Improving the Accuracy of the Benford Algorithm via Monte Carlo Simulation for Sample Size Determination

by

Franco Arda

to

A dissertation
submitted to the department of business administration
in partial fulfillment of the requirements
for the degree of
Doctor of Business Administration.

WSB University, Poland

December 2020

Abstract

Advisor: Prof. Dr. George Iatridis
Author: Franco Arda
Title of Dissertation: Improving the accuracy of the Benford algorithm via Monte Carlo simulation for sample size determination

Research Aim: Bedford's Law can be used to detect fraud. Previous research has highlighted a weakness of the algorithm, though: a high number of false positives. For auditors, false positives have become a strong deterrent to using the algorithm. This thesis aimed to decrease the false positive rate and, ultimately, increase the overall accuracy of Bedford's algorithm.

Research Hypotheses: To the best of our knowledge, no study has focused on estimating the sample size required for Benford's Law. The first research hypothesis was that we could determine the required sample size for a Benford sample, at a given confidence level. The second research hypothesis was, given the required sample, whether we can improve the Benford algorithm's accuracy at a statistically significant level.

Research Methodology: We do not know how fraudsters invent numbers. Our best assumption is that fraudsters invent numbers by randomizing data. This approach's advantage is that we can test our hypotheses objectively against other research papers.

In order to test both hypotheses, we created a synthetic dataset of 100 vendors with a total of almost one million invoices and five datasets were randomized (i.e., fraud).

To evaluate the first research hypothesis, we randomly sampled from the synthetic dataset to determine the sample size required, at a 95% confidence interval. Additionally, we iterated over the sample size until the samples' distribution was normally distributed, based on the Shapiro-Wilk normality test at a p-value of 0.05. The key breakthrough was to solve for both statistical measures iteratively.

To evaluate the second research hypothesis, we created two benchmarks based on current research. The alternative hypothesis was accepted if our algorithm's accuracy was higher than the benchmark, at a p-value of 0.05. To test the hypothesis, we used a simulation-based permutation hypothesis test. We compared the classification results with a confusion matrix.

We used the statistical programming language R and added short code snippets, where they supported the text.

Research Conclusion: The study concluded that researchers and practitioners require a sample of 1,532 invoices to be 95% confident that the sample conforms to Bedford's distribution. Our findings provide strong evidence, that using the sample size required improves the overall accuracy by 4.50% compared to other benchmarks. The improvement was solely due to the reduction of false positives (a.k.a. type I errors).

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor George Iatridis, for pushing me to be scientifically rigorous.

In terms of research on Bedford’s law, without the extensive work by Professor Mark Nigrini, few advanced would be possible, and indeed not my thesis.

Finally, the ability to simulate millions of data on my laptop feels like a superpower. Essentially, my laptop became my laboratory. Special thanks to Professor Rafael Irizarry for teaching us probability and random sampling related to Monte Carlo simulations in the statistical programming language R.

"Bedford's Law differs from other fields of study in that it has a serious mathematical component and also an applied component where real-world data is tested against the expectations of the law."

Prof. Mark J. Nigrini

For my father, Sami Arda-Keller.

1939 - 2019

Table of content

Table of Figures

Introduction

The first chapter of the research sets out the scene for the investigation by highlighting the rationale to research the Benford's law and development of an alternative model to determine accurate sample size. The chapter provides a background overview of the chosen law, its application till date, algorithms and statements. Based on the contextual analysis of the research, the chapter presents the research problem related to the integration of individual datasets. The chapter also justifies the research methodology selected for the analysis in this investigation and the practical and theoretical contribution to the empirical findings of the current research. The last section of the research provides the layout or structure of the research.

Background Overview

The Benford algorithm, or Bedford's law, is used extensively by researchers and practitioners to detect fraud. The law states that the sequence of numbers is likely to be distributed in a specific, non-uniform way. Until now, the practical implementation of the Benford's law is visible in different investigations related with random pairs of growth rates and initial values, tax payment-based frauds and accounting data frauds. The law also allows significant support in assessing the operational effectiveness of employees' data and to deal with the survey data problems highlighted in large datasets (Gonzalez-Garcia & Pastor, 2009).

The theorem is based on the conceptualization and operationalization of the digits appearing in the number. According to the law, leading digits are the digits appearing in the first place in a number, which can be used for understanding the uniformity of the distribution and for detecting the fraudulent use of certain digits the business, finance and accounting records. Additionally, the application of the law appears to be mainly significant for the large dataset comprised of several individual datasets. In this context, there is a possibility that individual datasets in a large dataset may not be complying with the criteria of Benford's law but such compliance is visible in the integrated dataset exhibiting new series of behaviors. Hence, the application of the law is extremely critical for real-world problems, mainly in the field of data and number management and uses.

Based on current research, there are two popular algorithms: the first-digit and the first two-digit algorithms. In this research paper, we focus on the first two-digit algorithm. In datasets that obey Benford's law for the first two digits, the number 10 appears as significant leading digits about 4.14% of the time, while the digits 99 appear as the significant leading digits 0.44% of the time.

Hence, the current research has attempted to test the statements of Benford's law by exhibiting that the law of leading digit is applicable upon the leading two-digits too rather than leading a single digit. Hence, in pursuance of the same statement of the law, the current research can focus on the finite observations of the selected database of the invoices for algorithmic assessment based on accurate sample size. On the other side, Benford's law in the second statement does highlight the importance of good visual fit for the observations. The use of Monte Carlo Simulation in the current research has allowed the researchers in evaluating the goodness of fitness test simultaneously.

Figure X visualizes the expected Benford distribution (blue line) for the digits 10 - 99. We can see that the distribution is highly skewed to the right — the bars in a grey display the observed first two digits in percentages. For example, we observed the number 49, almost 2.5% of the time. Based on the number of standard deviations, the number 49 represents the largest deviation from the expected Benford distribution and is therefore marked in red.

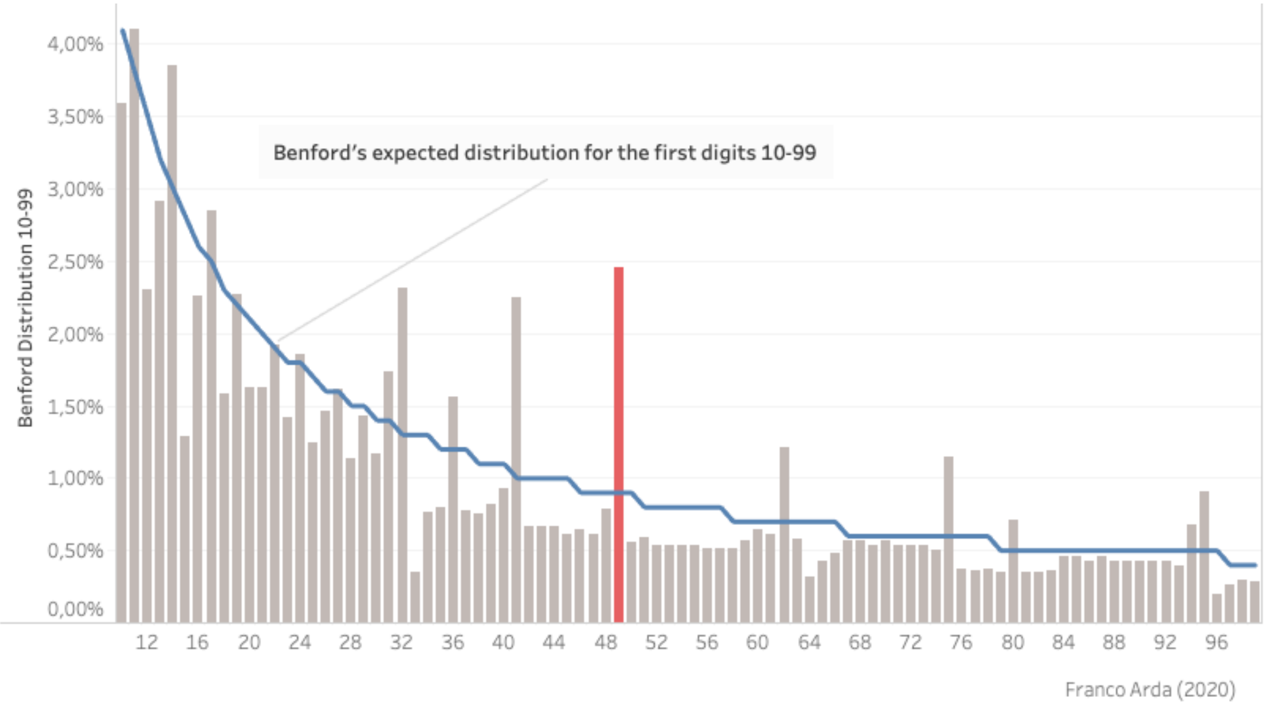


Figure 1: Expected distribution of the digits 10-99 based on Bedford's Law (blue line) vs. observed distribution (gray bars). The red bar indicates the largest deviation from the expected distribution based on standard deviations. Source: Franco Arda (2020).

One popular method to classify if a dataset obeys the law is calculating the average deviation from each digit, mathematically known as mean average deviation. If the mean average deviation is above a certain threshold, the dataset is classified as non-conforming to Benford's Law. Some researchers work on fraud detection using Bedford's law (Tota, Aliaj, and Lamcja, 2016), aggregate vendors (or companies) to calculate conformity to Bedford's distribution. The reason for aggregating is that, to date, we do not know the required sample size necessary. In other words, if we have one hundred vendors, we aggregate all the invoices and test for conformity. The aggregation is also supported by the Central Limit Theorem that justifies the point that one can get a normal distribution of a large dataset by combining the individual datasets having nonnormal or skewed distributions.

Research Problem

However, aggregating is mathematically and intuitively wrong. Intuitively, the explanation is straightforward. If we have one hundred vendors, all vendors' invoices aggregated might conform to Bedford's Law, while analyzing vendors individually might not. There might be one single vendor who creates invoices at random and commits fraud. In aggregate, we most likely will not spot this vendor while analyzed individually; we might. This is one of the significant limitations associated with the traditional statistics or traditional Benford's law as they undermine the importance of individuality of each dataset as well as individual transactions and invoices in such datasets. The differences in the behaviors of the vendor's maintaining individual datasets cannot be overlooked when assessing the fraudulent invoices through random sampling. Therefore, in the presence of appropriate sample size, the researcher may draw the relevant sample size from the individual databases and compare the activities of the different endeavors with each other rather than integrating their databases to perform collective analysis to identify normal distribution.

In traditional statistics, the determination of the sample size is an essential concept of any empirical study in which the goal is to make inferences about a population from a sample.

We deliberately use the phrasing "traditional statistics" because the methods we use in this research paper can be classified as "modern statistics." The distinguishing is non-judgmental.

In traditional statistics, we often deal with a normally distributed population. However, with Benford's Law, we are dealing with statistically speaking unknown distribution.

The Benford's Law distribution is highly skewed to the right (i.e., low digits occur more often than high numbers), the numbers are not continuous (i.e., numbers are discrete in bins), and last but not least, the population is not defined by the median, but by a mean average deviation.

In other words, calculating the required sample size for Benford's Law is challenging because we most likely cannot use traditional statistics. In our research, we use techniques from modern statistics that incorporate simulations. Technically, we use simulations because we do not have closed formulas to determine the sample size. The primary approach we use is a random sampling method called Monte Carlo simulation.

With Monte Carlo simulations, we can randomly sample from the population (i.e., the expected Benford distribution) several thousand times to approximate the required sample. In other words, rather than using a closed formula to determine the sample size, we empirically simulate the sample size needed. This approach works because the Monte Carlo simulation does not require a predefined distribution. Statistically speaking, the Monte Carlo simulation is non-parametric and should work on any distribution.

Research Aim and Objectives

Similar to traditional statistics, in fraud detection, we aim for the smallest possible sample size. While in traditional statistics, a sample size, which is too large wastes scarce resources, in fraud, we might waste another scarce resource; money. In other words, with fraud, we want to detect fraud as early as possible to keep the losses to fraud as low as possible. In a business context, knowing the required sample size allows us to monitor for fraud in real-time. However, with Benford's algorithm, monitoring fraud in real-time requires us to understand the necessary sample size.

Rationale behind Conducting Current Research

In the aftermath of the recent US presidential election 2020, several researchers applied Bedford's Law to investigate potential election fraud. One prominent publication was titled "Inappropriate applications of Benford's Law regularities to some data from the 2020 presidential election in the United States" by Professor Walter R. Mebane, Department of Political Science and Department of Statistics (Mebane, 2020). Unfortunately, we could not find any reference to the sample size in the research paper.

For example, in the small US county Fulton, some sample vote counts for Trump and Biden, with less than 1,000 samples, were probably too little for a Benford analysis. In other words, such small samples were most likely insufficient to classify whether the distribution conformed to Bedford's Law or not.

For this introduction, we simulated the US election example from a Benford perfect dataset. We randomly sampled 1,000 samples and repeated this process 10,000 times. For each simulation, we calculated whether the sample obeyed the Benford distribution (mean average deviation). An example with a mean average deviation below 0.0022 is considered to conform to Benford's Law (grey bars). Only about 18% of the simulations resulted in conformity.

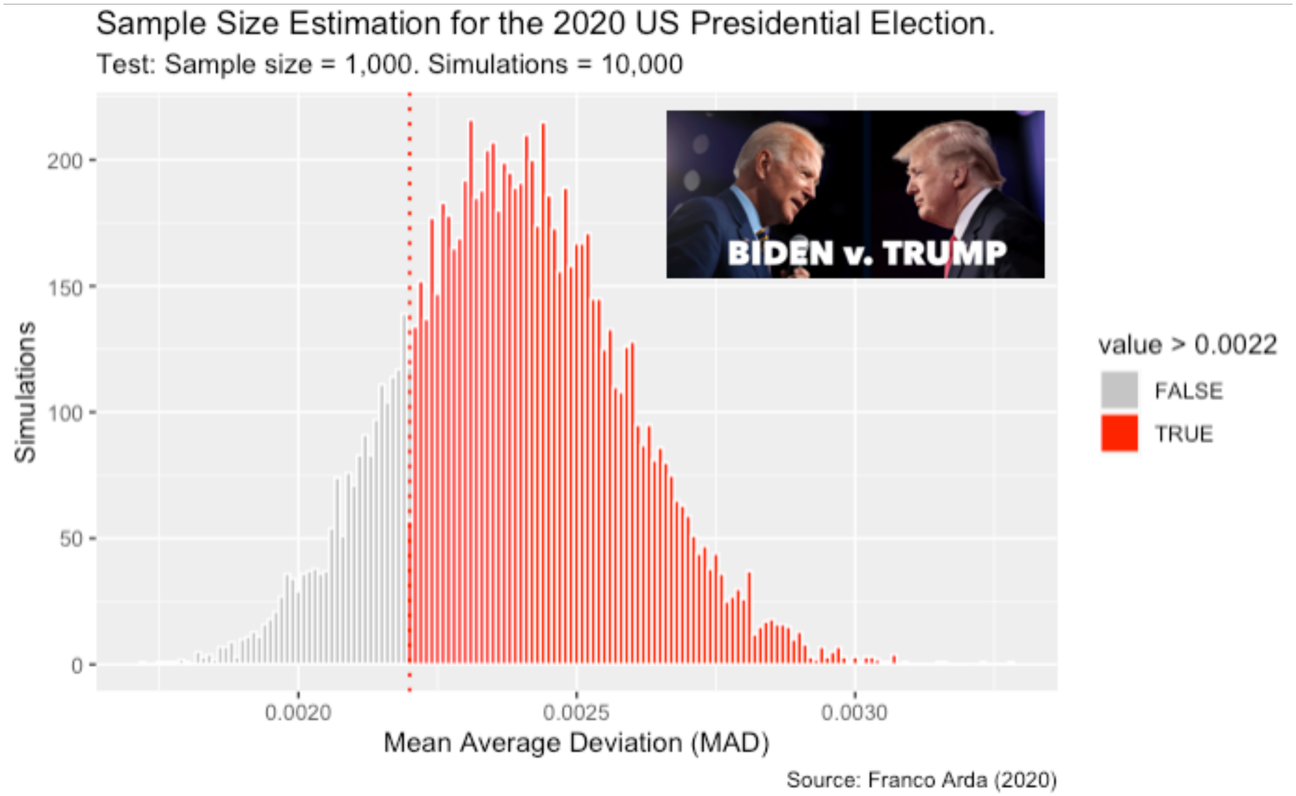


Figure 2: A sample size of 1,000 from a perfected Benford dataset, simulated 10,000 times, only conforms about 18% of the time (grey bars) to Bedford's Law. Source: Franco Arda (2020).

In other words, if we have a sample size of 1,000 from a dataset that perfectly obeys the Benford distribution, we would only get 18% of the time conformity to Bedford's Law. By most measures, being correct 18% of the time is too low. With confidence intervals, the most common levels are 90%, 95%, and 99%. With only 18%, we are far off even from the lowest confidence level at 90%. In conclusion, a sample size of 1,000 is with a high degree of certainty not enough.

Two additional comments to our approach:

1. We define the probability in terms of the relative frequency with which an event will occur if we repeated our "experiment many times. This is a classic "frequentist" interpretation of probability.

2. A Monte Carlo simulation is fundamentally different from a traditional resampling technique because the researcher constructs the true population. In other words, we construct a dataset that conforms to the Benford distribution, which acts as the population. On the other hand, with a resampling technique, the researcher does not know the true population (Carsey and Harden, 2013).

Based on the most commonly used conformity calculation, mean absolute deviation (MAD) set at 0.0022 (Nigrini, 2020), we will empirically test the hypothesis based on a synthetic dataset with one hundred vendors and close to one million invoices. We do not know how fraudsters invent numbers, and our best estimation is that fraudsters invent numbers by randomizing them. Therefore, we created a synthetic dataset as follows:

- 85 vendors with every 10,000 invoices conforming to Bedford's distribution.
- 10 vendors with a tiny number of invoices (< 89 samples).
- 5 vendors with randomized invoices (fraudsters).

The choice of the number of invoices for the "tiny number of invoices" was difficult. The number of invoices goal was that the number of samples had to be too small for the Benford algorithm. However, what the number of samples could be considered too small by the majority of researchers? Our assumption for this study is that any sample below 100 must be too small. The reasoning behind the choice of 89 is that the first-two-digit Benford algorithm consists of 89 bins (numbers 10 - 99). Furthermore, it should be evident that we need a few samples for each bin, at least more than one. Based on this logic, most research should agree that a sample size of 89 is too small.

With fraud in general and invoice fraud in particular, we need to detect fraud as early as possible. The reason for speed is that if we pay fake invoices, we want to stop them as early as possible.

Research into Benford, such as (Durtschi, Hillison, and Pacini, 2004), states, "the more data, the better." One of the few attempts to quantify the problem of sample size was by Heilig and Lusk (Heilig and Lusk, 2019), Barney and Schulzke (Barney and Schulzke, 2017), and Iorliam and Tirungari (Iorliam and Tirungari, 2016). Another approach through entropy was researched by (Goldman and Perez-Mercader, 2016).

In a recent research article by the Royal Statistical Society (Goodman, 2016) regarding Bedford's law's promise and pitfalls, a sufficient sample size of 20 numbers is not enough. As we will demonstrate in our research, a sample size of 20 is indeed far from enough.

Even with accounting experts, there is little explicit guidance for companies concerning the required sample. The larger, the better (Collins, 2017) is no help. Of course, that is not the fault of accountants. Even technical books, such as mathematical statistics (Wackerly, Mendenhall, and Schaeffer, 2008), only guide taking samples from a normally distributed underlying.

For early fraud detection, this answer might be in practice, for many companies, too vague. Ideally, based on a synthetic population created by us of 10,000 data points that conform precisely to the Benford distribution, can we identify the required sample needed at a specific confidence interval? To get a real benefit for companies, we set a statistically significant level at 5,000 data points (i.e., 50% earlier). Thus, the null hypothesis is that we cannot set the required sample at a lower level than 5,000 data points with a confidence interval of 95%. The alternative hypothesis is that we can define the required sample < 5,000 datapoints at a 95% confidence interval.

Few studies, if any, have investigated the impact of aggregating MAD levels. To date, scant attention has been paid to the MAD levels in aggregate. We argue that previous research has mostly overlooked the effect of aggregation MAD levels and its significance.

Research Hypothesis

Research on Bedford's Law has relied primarily on analyzing, in the case of invoice fraud, a single company. In the case of analyzing invoices from a single company, we do not argue about the correctness of Bedford's Law's application and the corresponding MAD levels. However, we hypothesize that by combining several companies, the MAD level's accuracy diminishes. In other words, our alternative hypothesis states that by keeping the companies on a granular level, we can get a better accuracy based on MAD.

Overview of Selected Methodology

We plan to offer an alternative hypothesis at a p-value of 0.05. Remarkably few studies have been designed to validate the correctness of MAD levels when analyzing companies in aggregate. Our research is concerned with offering an alternative hypothesis.

This research study examines the relationship between calculating the MAD level on a company level and an aggregate level (i.e., aggregating MAD levels of several companies together).

We will run each dataset against the aggregation, compare the MAD levels, and use Nigrini's MAD level of 0.0022 to test the algorithm. If our alternative hypothesis can be accepted at a 95% confidence level, we can reject the null hypothesis.

In conclusion, several researchers, in particular Prof. Nigrini, have laid out the mathematical framework for evaluating the Benford algorithm. We have the scientific knowledge of how to measure a given dataset and apply a threshold for classifying the dataset for conformity. We plan to extend our scientific and practical knowledge of Benford in regards to sample size determination.

Research Theoretical and Practical Contributions

To our knowledge, there has been no study till date in which researchers have focused on the estimation of accurate sample size for the application of Benford algorithm to a large dataset. In this regard, the empirical findings of the current research would be of value addition to the existing literature database that will provide representable and accurate sample size based on the controlled, reliable and valid research. On the other side, practically the findings of the current research would have significant implications for the researchers investigating the fraudulent invoices in a large dataset through rapid and fast detection using the small and accurate sample size relatively.

This will also reduce the costs associated with the investigation as well as the resulting loss of fraudulent activities that are often left undetected due to the lengthy and time taking estimation procedure within the traditional mathematical processes. These findings will have significant implications for the future data management field related to artificial intelligence, data mining, and remote databases administration, which are comprised of a multitude of invoices and transactions. The use of appropriate sample size can allow the fraud examiner in detecting fraud efficiently limited timeframe and with adequate outcomes.

Research Layout

Besides this introduction chapter, the research is divided into three main chapters. The second chapter is comprised of a critical review of past literature about the Bedford Theorem and the implications of these literature findings for the current research methodology framework. The third chapter presents empirical findings gathered from the Monte Carlo simulation of a dataset of 10,000 invoices. This chapter of the report presents histograms developed through the examination of a random sample via Benford's law and Central Limit Theorem.

The chapter also provides a critical analysis of the results of histograms concerning the traditional plan for distribution. The last chapter of the research provides a critical discussion on the empirical findings and their implications for future investigations. It also offers conclusions and path towards future research based on major findings of accuracy, recall and representability of the

Literature Review

To the best of our knowledge, no study has focused on the sample size required for Benford's Law. A recent study (Gomez-Camonovo et al., 2016) has listed its samples, ranging from 99 to slightly over 500. Again, we hypothesize that the study's accuracy could have been higher if the researchers had used a pre-defined sample size. One possible problem that occurs with small sample sizes is that we get a high false-positive rate. As the algorithm's classification accuracy is reflected in the overall accuracy, we will test this hypothesis extensively.

Last year, one of the first research papers (Cerioli et al., 2019) we have discovered used a Monte Carlo simulation-based simulation approach. What the researchers simulated was the expected distribution of the first nine digits. They used a pre-defined number of transactions (or samples) ranging from 50 to 500. We believe, though, that they have overlooked the sample size required defined at a specific confidence interval. Additionally, they did not justify their choice of 10,000 Monte Carlo simulations. In general, many simulations such as 10,000 are an excellent choice in a scientific research paper. However, we would argue that we do not always have the luxury of a large dataset with fraud. In other words, we need a statistically significant sample size with the least number of samples. In our research hypothesis, we offer a possible solution. We use the Shapiro-Wilk normalization test in combination with a Monte Carlo simulation. As soon as our sample size is normally distributed (based on the Shapiro-Wilk test at a p-value of 0.05), we stop increasing the simulation number. In conclusion, we hypothesize that if the researchers had used a sample size determination, they would have achieved a higher accuracy.

To date, little attention has been paid to determining the sample size for Benford's Law. For example, in a recent research paper (Whyman et al., 2016), the authors paid most of their focus on proofing the distribution (or probability in their words) on the frequency of the first digits based on Benford's Law.

Few attempts have been made to investigate the importance of sample size with Benford's Law. In even the most recent research papers (Noorullah et al., 2020), the focus has been on applying Fraud and the expected digit distribution.

Few studies have investigated the impact of small sample size, or sufficient sample size, in combination with Benford's Law. One recent study (Moreno-Montoya, 2020) offered a promising title referring to small sample size. Unfortunately, the study came short, in our opinion, in providing a statistical sound method to determine the sample size required for Benford's Law.

There is limited research investigating the sample size required for Benford's Law. One recent study explored a dataset with a population ranging from 10,640 to 1,439,323,776(Koesters et al., 2020). Based on our research, we know that those numbers are sufficient for a sample size. Those population samples are probably large by any standard. Unfortunately, we do not have the luxury of waiting for fraud detection until we have enormous datasets. For fraud detection, we require the smallest possible sample size given a confidence interval.

Previous research in Benford's Law has been mostly restricted to the application of the Law. For example, researchers (Sugiarto et al., 2017) have tested different statistical methods for determining Benford's Law, such as the 5% limit. We believe this is a gap in the research.

Several studies have shown that Benford's Law works exceptionally well with financial data, such as financial statements. However, important questions regarding the required sample size remain unanswered. In a recent study regarding financial statements and Benford's Law (Subago, 2017), the authors applied the first-digit algorithm's distribution. Based on our research, the algorithm requires a substantial amount of samples. Analyzing financial statements at the aggregate level (e.g., monthly basis) bears the risk that the sample size is potentially insufficient.

There remain many unanswered questions about the application of Benford's Law, such as the sample size and the problem with a high false-positive rate. Some researchers (Silva and Fijho, 2020) explicitly stated the number of samples based on Brazil's COVID-19 death rates. The sample size ranged from 6 to 203. We believe this is a gap in research. We hypothesize that we can increase the accuracy of the Benford algorithm with pre-defined sample size.

While there has been a great deal of research on Benford's Law, very few studies have paid attention to the sample size (Hidayat and Budiman, 2020).

The evidence points to a variety of applications for Benford's Law. However, the role of sample size is still poorly understood. Some researchers (Kaiser, 2019) focused almost solely on statistical parameters such as mean, median, and skewness. We believe this is a gap in research and that higher accuracies and lower false-positive rates could be achieved using a sample size threshold.

Existing research has focused on applying Benford's Law, often in very technical terms, but has failed to explore the sample size's impact. In a recent study (Gauvit et al., 2017) in advanced cognitive psychology, the focus has been laid on issues with Benford's Law regarding sensitivity and specificity. We would argue that calculating sensitivity (or correct positive classifications) is a good start. We see that the underlying problem of calculating high false positives is not easily detected. In other words, a high false-positive rate highlights the problem but does not lead us directly to the solution. Thus, we hypothesize that the sample size determination can decrease the false positive rate and increase the algorithm's overall accuracy.

Remarkably few studies have accounted for the sample size in Benford's Law. Researchers today (Tota et al., 2016) still quote the approximate sample size required from earlier research (Nigrini, 2012) where Nigrini stated, "the general rule is that the data set should have at least 1,000 records before we should expect a good conformity to Benford's Law." We strongly believe that analyzing the required sample size is a research gap worth exploring. In other words, our primary goal is to explore the relationship between sample size and accuracy. In order to understand the mechanisms underlying the sample size, we have to explore different simulation methods. In science, the most popular simulation technique is the Monte Carlo simulation. The approach in this thesis is challenging; on the one hand, we establish a hypothesis, and on the other hand, we have to explore methods to test the hypothesis. For those reasons, we split the literature review into two parts; Benford's Law and Monte Carlo simulations.

Even one of the grandsons of Benford (Benford, 2020) pays no attention to the required sample size. In general, existing research has focused mostly on Benford's mathematics but has failed to explore the sample size.

Within the field of Benford's Law, at least one crucial question remains unanswered. In a recent study (Luo and Li, 2018), the researchers provide a new technique for testing a dataset. Unfortunately, the researchers do not test the accuracy of the algorithm. Our main goal is to create a synthetic dataset to verify the accuracy of the algorithm. In other words, we believe not using a synthetic dataset that can be used to compare algorithms is a research gap. In our research, we propose a simple but transparent approach; a dataset that includes data that conforms to Benford's Law and data, which is randomized. We are aware of our assumption that nonconform Benford data might not be random or only partially random. The essential advantage of a synthetic dataset is that we have a transparent way of measuring algorithms.

Recent research offered an exciting approach (Marif et al., 2020) to creating a synthetic dataset. In testing the Benford algorithm's distribution (not the accuracy), the researchers created a synthetic dataset of 100000 invoices. We plan to approach our research study similarly. As Monte Carlo simulations are computationally exhaustive, we might use only 10000 invoices for each vendor. As a precursor to our study, we will test the synthetic dataset variations (a.k.a. the population) on whether a smaller dataset will suffice. In summary, it was fascinating to see that other researchers approach simulations with a synthetic dataset of invoices.

Challenges associated with Benford's conformity have been approached differently in a recent study (Nguyen et al., 2020). Rather than using sample size, the researchers have used a robustness test based on multiple linear regression. While we applaud this innovative approach, the effectiveness of this approach is hard to quantify. For example, we could not find any accuracy tests in the mentioned research paper. In our research paper, we aim to define the sample size required by defying the sample size at a given confidence interval without judgment.

Additionally, we want to evaluate the accuracy of our approach against a traditional algorithmic approach. Finally, our goal is to compare the accuracy of the different algorithms in a confusion matrix. Accuracy in an algorithm is often not enough; that is why we use a confusion matrix that compares the accuracy and the correct classification of Fraud (i.e., sensitivity or recall).

Despite decades of research on Benford's Law, the accuracy has often been less than satisfactory, particularly with the high false-positive rate. A very recent study of accounting data (Özevin et al., 2020) analyzed Benford's Law's application to accounting data. In this research paper, mostly large datasets have been analyzed. We hypothesize that the accuracy could have been improved if the researchers had used a minimum sample size. For example, some sample sizes were as small as 91. In our view, such a small sample size leads to lower accuracy in general and a high false-positive rate in particular.

A recent research paper touches (Matakovic, 2019) on the financial numbers game. Within accounting, this is a fascinating topic. The researcher worked with extensive data (e.g., 5,325 and 4,916) so that a potential minimum sample size did not distort his results. In our research, we aim for the smallest sample size to detect fraud as early as possible. Therefore, unlike this researcher, we focus on speed and need the smallest sample size possible.

Conformity to Benford's Law is an essential area of inquiry; however, relatively little is known about the required sample size. A recent research paper (Larsen, 2017) found anomalies in REITs. The study used 2248 observations. We hypothesize that this is a large enough sample size for Benford's Law. However, if the researcher had different datasets, the results might not have been that favorable. In other words, our research goal is to find a minimum threshold (i.e., the sample size required) to determine if a dataset follows the Benford distribution or not.

Research-based on Bedford's law is rich and with a long history dating back to 1881. The sheer amount of interest is probably best illustrated by a recent research paper (Beebe, 2020), stating all publications about Bedford's law on 279 pages. These publications provide significant insights related with the importance of Bedford law's savviness for the chief financial experts, asymptomatic properties of the digit sequences of random numbers, distribution of leading digits in statical tables, the applicability of the Benford's law in geoscience, natural science, and payment system auditing.

In general, the most significant contributions in recent years have come from accounting experts, in particular, identifying problems in financial fraud and finding ways of applying Bedford's Law to them. Those research findings have leapfrogged our understanding of which numbers represent sizes of facts or events.

Among all researchers, one stood out the most: Mark J. Nigrini. In his research (Nigrini, 2020), he contributed to the two different algorithmic approaches (first-digit and first-two digits), the algorithmics' scale invariance, and, most importantly, to researching and quantifying when a dataset does and does not conform to Benford's Law are the highest contributions. Nigrini's extensive research into Benford's Law led to one of the most important statistical insights; the mean average deviation applied to Benford's distribution.

Unfortunately, one of the foremost experts (Nigrini, 2012 and 2020) in Benford has no clear answer to the required sample size: *" … it is not clear, how large our numbers (sample size) have to be …"*

According to recent research (Miller, 2015), a major weakness of the Benford algorithm is the high rate of false positives: *" …without a doubt, for auditors, the false positives have become a strong deterrent to the use of Benford's Law in recent years..."*

While our research paper tries to generalize, we focus on a single fraud example: invoice fraud. On a global level, the mean average deviation determines the conformity to Benford's distribution. On a local level, (Nigrini 2020) argues that we can apply standard deviation (a.k.a. z-statistic or z-score). His theory is that the standard deviation increases when the difference between observed values is compared to expected values. In our case, the highest standard deviation within a flagged dataset refers to the invoice with the most substantial deviation from Benford's distribution.

Some researchers (Vaughan, 2018) have proposed the statistical conformity measure for categorical data, the chi-square test. In general, high chi-square values could be associated with nonconformity to Benford's distribution. This approach's weakness is that there is no scientific threshold for when a distribution can be considered to conform to Benford's Law. We prefer the mean average deviation approached proposed for the lack of a clear threshold cut (Nigrini, 2013 and Nigrini, 2020).

Our simulation and hypothesis tests rely on a correct setup of the synthetic dataset for Benford's distribution. Numerous research (Nigrini, 2020) have shown evidence that Benford's distribution is scaled invariant. In other words, it is not relevant to the algorithm whether we create a dataset in USD or AUD or if we start with the synthetic dataset at the number 10 or 1000.

It is no surprise that Nigrini was one of the most prominent characters in the Netflix documentary digits, which covered Benford's Law in Connected: The Hidden Science of Everything, in episode 4, August 2020.

Research in the complex area of nonparametric statistics has emerged in recent years. Scientific knowledge is required, as Benford's distribution is nonparametric. In other words, the distribution parameters (e.g., mean or standard deviation) are not known with the nonparametric distribution. Recent research (Linebach, Tesch, Kovacsiss, 2014) confirms that randomly sampling data from independent samples work with nonparametric distributions. Such distributions are comprised of data, which are not assumed to come from prescribed models of a normal distribution or linear regression model. In our case, this research confirms that we can apply a Monte Carlo simulation to Benford's distribution. Monte Carlo method is recognized as a useful approach for estimating the uncertainty associated with an estimated expected value proportionally based on the number of histories or samples of f (x).

Several authors have shown evidence of the misuse of p-values (Ionannidis, 2019) in scientific papers. One common theme is p-value hacking, where the researchers try out different statistical simulations until they find a statistically significant result. One weakness of the study is not highlighting that a researcher might get a statistically significant result just by random chance. In other words, the author focuses on researchers with unethical behavior while neglecting those who are merely unlucky by chance. We agree with the researcher that simulation several thousand times an outcome based on confidence intervals should limit the risk of uncovering a statistically significant outcome by pure chance.

Recent research has confirmed that (Sander, Senn, Rothman, Carlin, Poole, Goodman, and Altman, 2016) we can apply the confidence level of 100% - alpha to get to the desired p-value. In other words, a confidence interval of 95% can also be viewed as a p-value of 0.05. One key finding in their research is that they do not differentiate between a simulated and a static p-value. We would argue that a simulation-based p-value is more stable and accurate.

On the mathematical side, recent research (Berger and Hill, 2020) has uncovered some mathematically erroneous approaches regarding Bedford's law. These set of common errors offers significant insights for the researchers in the current investigation to help them understand how they can avoid such errors in future research and applications. According to the researchers, the first error relates with the order of magnitude, the second error with exponential sequence, the third error with the large spread and regularity of the distribution or dataset for increasing closeness to Benford, and the last and fourth error relate with the relatively intuitive arguments. Those findings are crucial in pushing scientific progress.

In our case, those research findings were crucial in creating the synthetic datasets. For example, a dataset does not need to cover several orders of magnitude to comply with Benford's distribution. Unfortunately, the belief that a random variable needs to cover at least several magnitude orders is widely propagated.

Those research insights motivated us to investigate the alternative Benford algorithm's accuracy and analyze false positives and false negatives. Interestingly, we would have never thought of applying a confusion matrix to evaluate the performance of the Benford algorithm's classification accuracy. In general, we used confusion matrixes in the past for predictive models, such as a machine learning algorithm.

Kwak and Kim (Kwak and Kim, 2017) argue that the central limit theorem requires a larger sample size for skewed distributions in the real world. Their leading theory is focused on the binomial distribution, whereas the Benford distribution is multinomial. With Benford's highly skewed distribution, we must simulate extensively to see if we can get random samples that adhere to the central limit theorem. In other words, the mathematical distribution of the Benford bins (10 - 99) will probably not "comply" with the central limit theorem. For determining the sample size for Benford's distribution, the central limit theorem is of little importance. However, if the random samples did comply with the central limit theorem, we could expect much more stable accuracy results with fewer samples.

Rizzo (Rizzo, 2019) wrote extensively about the computational tools in applied statistics. One of the main theories was the application of Monte Carlo simulation for statistical inference. While she did not provide theories on simulating a sample size, she gave great coverage on the probability of confidence intervals. One potential gap is sample size determination. However, we understand the mathematics and code to approach Monte Carlo simulations for sample size determination with the different literary works.

However, the confusion matrix proved extremely useful in comparing the traditional Benford algorithm with our modified version. With the benefit of hindsight, applying a confusion matrix to the Benford algorithm makes sense. For example, a high number of false positives (or false alarms) can deter an accountant from using the Benford algorithm. On the other hand, if we find methods to reduce the false positive rate, we might give accountants and fraud examiners more confidence in applying the Benford algorithm (Miller, 2015).

From a researcher's perspective, it is also interesting to see how a programmer would try to "hack" (Vaughan, 2018) Bedford's law—in other words, trying to create programs that go undetected in terms of conformity. Contrary to our initial assumptions, creating a dataset that complies with Benford's distribution was relatively straight-forward. In other words, creating a large dataset of 10,000 invoices that comply with Benford's distribution is simple.

Therefore, "hacking" Benford's Law with one dataset can be achieved without advanced mathematical knowledge. Where it gets complicated is "hacking" Benford's Law with continuous data entries. In our example with invoices, a vendor hands in one invoice, and we run the Benford algorithm on the whole dataset. Then, the vendor hands in another invoice, and we rerun the Benford algorithm. This process repeats until we reach several thousand invoices.

"Hacking" the Benford algorithm in this sequential process could be extremely difficult. In a sequential process, the fraudster must continuously update his invoices' digit distribution to stay undetected. The positive result is an unintended benefit of our approach; as we aim to determine the required sample size, we can continuously monitor the dataset's conformity. "Hacking" this improved Benford algorithm is most likely very complicated.

Without correctly importing and transforming data (Wickham and Grolemund, 2017), a research job would be significantly harder. We feel that programming and statistics are inherently challenging. The statistical programming language R makes it relatively easy to load datasets and prepare them to run simulations and algorithms. R allows us to focus cognitive focus on the statistical challenges and less on programming.

With the blog post, "There is only one test" (Downey, 2016), Downey shuck up the statistics world by illustrating how a hypothesis test should be simulated. The essential idea is that a hypothesis test is not based on a static dataset but that the null hypothesis should be stochastic and based on a simulation, which reduces many problems with the p-value—a deeper understanding of the pitfalls of p-values followed by several researchers (Reinhart, 2015).

Shortly afterward, the American Statistical Association released a statistical significance statement and p-values (American Statistical Association, 2016). In the last chapter of our research paper, we test the traditional Benford algorithm against our proposed alternative. We run hypothesis tests with 1000, 5000, and 10000 simulations.

Only a handful of researchers have investigated the explanations and understanding of why a particular dataset confirms Benford's distribution (Cole, Maddison, Zhang 2019). We were surprised to discover that the researcher explained why stock markets conform to Benford's Law. We believe that the researchers wrongly argue in favor of multiple orders of magnitude for a Benford dataset. Study results compared the first-digit with the first-two digit algorithm as other researchers found evidence supporting more stable accuracy of the first-two digit algorithm.

Some researchers have applied Benford's Law to the epidemic growth model of COVID-19 (Lee, Han, and Jeong 2020). We are not convinced that Benford's Law can be applied to growth rates in general. For creating a synthetic dataset, it was interesting to observe that the researchers used a similar approach to simulating data.

Random sampling, including Monte Carlo simulations and bootstrap, has been propelling scientific progress for decades. Years ago, complicated simulation techniques required a solid background in mathematical statistics.

Today, with the introduction of coding libraries, simplified code, and the spread of knowledge, those methods have become useful for a wider audience. With a little bit of code, as we will see in our research paper, we can run simulations on datasets with millions of variables. Without those advances in statistical computation, this research paper would have been tough to implement. Just the simulations alone, which amount to several hundred, would have had taken several weeks to run on a laptop. At the core of many simulations are Monte Carlo methods. Those techniques might look simple on the surface, but the applications (Irizarry, 2020) require extensive knowledge.

Although Frefix (Frefix, 2018) argues that precision and recall are sufficient binary classification measures for fraud detection, we disagree. He bases his theory that with unbalanced datasets, accuracy can be useless, which is true. This observation is correct, and we will cover it with our confusion matrix as well. We believe that his argument is weak because, in a confusion matrix, we can reflect several measures of a binary classification algorithm. By combining several metrics in a confusion matrix, we would argue that we give a better overview.

In discussing the reduction of false positives in fraud prediction (Wedge, Kanter, and Veeramachaneni, 2017), they essentially proposed a feature engineering. In testing their hypothesis, they achieved a lower rate of false positives. One potential weakness of this approach is that it required human intervention. The goal of our research paper is to generalize our findings. If we allowed human intervention in the dataset, we fear that our research findings would not be reproducible.

Grabowski poses the question, "how many more?" (Grabowski, 2016) In his study, he explored the effects of sample size. While not explicitly mentioning Monte Carlo simulations, the main topic covered the usefulness of simulations. The central theme was focused on estimating the sample size for more complex distributions. A clear novelness was the quantification of errors by using a formula combining imprecision and bias. We would argue that with simulations in general, and Monte Carlo simulations in particular, the study of confidence intervals is well researched. For this reason, we will evaluate simulation errors based on confidence intervals, such as 90%, 95%, and 99%.

The researcher's Pan, Liu, and Miao (Pan, Liu, and Miao, 2018) tackled sample size determination for longitudinal data. The author's expertise is in the medical field, where they applied statistical concepts for grant applications. The central theme in their field was the challenges related to sample size determination for longitudinal design. Their central hypothesis was to simulate the required sample data via bootstrap. The bootstrap approach is the first method we want to test our hypothesis. Their study's potential weakness is that they combine an empirical approach, such as bootstrap, with statistical power. In our opinion, and based on statistical textbooks, statistical power requires a parametric approach (i.e., the distribution should be normal) while the bootstrap, in general, is nonparametric. In other words, we are not sure about the correctness of their approach, but we are sure that the Benford distribution is nonparametric and works with the Benford distribution.

As the famous statistician John Tukey said, "The simple graph has brought more information to the data analyst's mind than any other device." Visualizations are meaningful in general for statistical analysis but for simulation-based modeling in particular. It is incredibly beneficial to see how often specific values have been in or out of a range with simulations.

Additionally, it is helpful to see immediately if a particular value's distribution is relatively normal or not. Prebuild libraries such as infer (Kim and Ismay, 2018) help us create bootstrap distributions and hypothesis tests with only a few code lines. For example, the library allows us to visualize a simulated based hypothesis test with a few code lines. As researchers, with such libraries, we can focus on the problem rather than on coding.

From the beginning of this research paper, we knew that determining Benford's algorithm's required sample size needed simulations. But we did not know how to do it. The building block of our research paper lies in simulations, which is part of mathematical statistics. We can solve challenging, real-world problems that are close to impossible to solve with traditional or closed formulas with simulations.

Therefore, we consulted some research that was focused on the mathematical aspects of simulations. One requirement was that the authors used the statistical programming language R. Code in R to bridge the gap between mathematical formulas and apply the knowledge to our problem. Harvard's most popular probability source (Blitzstein and Hwang, 2019) helped us at the beginning. While it fulfilled the mathematical rigor and code snippets in R, it lacked our depth in simulations for our purposes.

Monte Carlo simulations is applied in sampling to gather information about a random object by observing many realizations of it (Kroese et al., 2014). With repeated simulations, it is very important to have fast implementations of the estimator in the corresponding programming language (Templ, 2016). Ideally, everything in R is vectorized. In other words, any function call applied to a vector operates directly on all of its values. The statistical programming language R also enables us to simulate random numbers which are independent and identically distributed (i.i.d.).

We never found a blueprint on how to simulate the sample size required for Bedford's algorithm. Still, we saw incredible value in the highly technical and practical book "Mathematical Statistics and Resampling with R" where the authors (Hesterberg and Chihara, 2019) share experiences with simulation problems used with large datasets, particularly at Google.

Researchers such as Kak (Kak, 2018) provide a methodological framework for inferencing a synthetic dataset that conforms to the Benford distribution. The author's focus is on the first-digit approach. However, find no evidence that we can not apply the same methodology to the first-two digit algorithm version.

Numerous researchers have pointed out (Martino, Luengo, Miguez, 2018) that random sampling methods with non-uniform random numbers are the building block for Monte Carlo simulations. Those researchers have worked on a range of applications of the Monte Carlo algorithm, such as statistical physics problems. From our perspective, it is often hard to "translate" those mathematical notations into R code. Previous studies (Templ, 2016) with corresponding R code examples on GitHub (public and private code repository) have empowered us to translate Monte

Carlo simulation into applicable code. Empirical evidence appears to confirm the notion that a statistical simulation is carried out in drawing a random outcome and the subsequent observation of multiple simulations. In other words, we take several random samples from a population and repeat this process several times.

Many scholars have researched decimal representation (Berger and Hill, 2015). The leading theory is that Benford's confirm digits should match positive real numbers with a minimum of 10 and no maximum. This is also the main characterization associated with the Benford’s property of the uniform distribution of a sequence of real numbers. Such sequence is obtained when the decimal algorithm of the absolute values of the real numbers is uniformly distributed. For our research, we will incorporate those findings in creating a synthetic dataset.

In this 2011 study, Peng (Peng 2011) touches on a sensitive point in quantitative research such as this paper: reproducibility. Reproducibility means that the same researcher, or a different researcher, gets the same results with the same data. While difficult to prove, some argue that more than half of the doctoral thesis is not reproducible. There are essentially two parts to reproducibility: (1) code and (2) datasets. In our research paper, we try to comply as much as possible to further push for reproducibility. In the following research paper, we add short R code snippets, where it supports the text.

We acknowledge, though, that we could do better. For example, we could use R Markdown, which allows other researchers to run our research paper in R and replicate our results. Unfortunately, this is not possible in our research paper as it is written in Microsoft Word. What we did, though, is adding in the appendix several of our datasets. The first dataset was used to test the first research hypothesis: determining the required sample size for Benford's distribution. The second and third datasets were used to test the second hypothesis: the statistically significant difference in accuracy. Measurement of the data accuracy was necessary for the representation of the true behavior of the economic variables. The significance of Ci-square test in confirming the accuracy of the data set, which ultimately assist in understanding the implications of the specific dataset on the application of Benford’s law.

In this study by Hegazy et al. (Hegazy, Median, and Regaie, 2016), the authors points out the required sample size for false positives. The potentially small sample is a challenge we have never considered. The author's leading theory is that we should consider the number of false positives in evaluating an algorithm's accuracy. In general, the authors argue, we should be cautious with samples below five percent. Unfortunately, the authors never tested the theory, or rather hypothesis. However, for our algorithmic evaluation, we will consider their findings.

A note on mathematical notations; our research paper is light on mathematical notations. The reason for this is not that we do not like mathematics, quite the contrary, but that we are trying to solve challenging problems directly in code via simulations. Simulation allows testing the true probability of the dataset through the analysis of distribution and skewness of the dataset relatively. The effectiveness of the simulation cannot be undermined in reducing the variation and high probability of the same number to appear again in the process.

For example, our synthetic datasets consist of almost one million variables. Suppose we simulate the behavior of nearly one million variables with a Monte Carlo simulation several thousand times. In that case, we can create empirical studies that are almost impossible with a closed mathematical formula.

In other words, the combination of today's computational power, even on our laptop, large datasets, and a statistical programming language such as R, empowers us today to test ideas and hypothesis which are almost impossible with mathematics alone.

The critical review of the past studies and its implications for the current research can be used for understanding how the Benford’s law facilitates experimental inquiries in the financial investigations including large datasets in the banking sector, auditing, forensic accounting, natural science and others. However, using the law, the literature has informed that future researchers should be cautious about the erroneous approaches of the law to lead towards desired outcomes. However, the literature review has substantiated the use of Bedford theorem for the large datasets for the significant assessment of its operative reputation.

A theoretical framework

The last decades have seen an increased interest in applying Bedford's law in connection with accounting (Nguyen, Nguyen, and Duong, 2018) and taxes (Demir, 2020).

Especially in the financial world, ample evidence seems to support that if humans make up numbers, they should follow a particular order that conforms to Benford's distribution.

In the present study, we focus on invoice fraud, but the same framework can apply to any accounting activity (invoices, reimbursements, or financial statement manipulation). The framework should work for any made-up numbers, no matter the background.

As we will see in our empirical findings, if a fraudster uses randomized numbers, the algorithmic framework works exceptionally well in terms of accuracy. How well? In short, if a fraudster uses a random number generator, the algorithm is going to catch him with 100% accuracy.

A short introduction to Bedford's Law

Contrary to common intuition that all digits should occur randomly with equal change in real data, empirical examinations consistently show that not all digits are created equal, but rather that low digits occur much more frequently than high digits in almost all data types, such as those relating to geology, chemistry, astronomy, physics, and engineering, as well as in accounting, financial, econometrics, and demographic data sets.

The finding is named after Frank Benford, a physicist, who published the seminal article on the topic. Benford started his article by noting that the first few pages of a book of common logarithms show more wear than the last few pages.

Of interest here is the fact that the first few pages of logarithm books give us the logs of numbers with low first digits (e.g. 1,2, and 3). He concluded that the worn first pages were there because most of the numbers in the world had low first digits. The first digit is the leftmost digit in a number and, for example, the first digit of 153,451 is 1.

Zero is inadmissible as a first digit and so there are nine possible first digits (1,2 …9). The signs of negative numbers are ignored and so the first-two digits of -32.12 are 32.

In his research, Benford tried to collect data from as many sources as possible to include a variety of different types of data sets. His data varied from random numbers having no relationship to each other, such as the numbers from the front pages of newspapers and all the numbers in an issue of Reader's Digest, to formal mathematical tabulations such as mathematical tables and scientific constants.

Other data sets included the drainage areas of rivers, population numbers, American league statistics, and street numbers from an issue of American Men of Science. He analyzed either the entire data set at hand, or in the case of large data sets, he worked to the point that he was assured that he had an fair average. All his work and calculations were done by hand and the work as probably quite time consuming.

His research showed that on average 30.6% of the numbers had a first digit 1, and 18.5% of the numbers had a first digit 2. This means that 49.1% of his records had a first digit that was either a 1 or a 2. In contrast, only 4.7% of his record had a first digit 9.

Benford then saw a pattern of his results. The actual proportion for the first digit 1 was almost equal to the common logarithm of 2 (or 2/1), and the actual proportion for the first digit 2 was almost equal to the 9 with the proportion for the first digit 9 approximately the common logarithm of 10/9.

He then calculated the expected frequencies of the digits in lists of numbers, and these frequencies have now become known as Bedford's law. Bedford's law is usually associated with the expected first digit (1-9) proportions, but we use a slightly more advanced version, the first two-digit (10-99) version, proposed by Nigrini (Nigrini, 2020).

Mathematically, a dataset satisfies Bedford's Law for the first-two-digits if the probability that the first-two digits, D1 D2 equal d1 d2 is approximate:

$$P(D_1\,D_2\ = d_1\,d_2\ ) = log\left(1\ +\ \frac{1}{d_1\,d_2}\right)\quad d_1\,d_2\ \in [10\,,11,\ldots,98\,,99]$$

Figure 3 : Bedford's Law for the first-two digits. Source: Mark Nigrini (2020).

Converting the mathematical formula in R code, gives us for the first-two digits 12, an expected distribution of 3.476%:

```
ben_law <- function(digit)
        log10(1 + 1 / digit)
ben_law(12)
# 0.03476
```

Testing the hypotheses based on Mean Average Deviation (MAD)

Most of the research on Bedford's distribution suggests that real-data will never perfectly reflect Benford's expected proportion. Without an error term it is too imprecise to say that data set "does not conform to Bedford's distribution."

In order to run hypotheses scientifically, we need to have an "error measure" to evaluate an invoice dataset on whether it is conforming or not. By how much does it have to differ from expected values to "non conform"?

In statistics, measure called chi-square compares two categorical distributions. This is not an error, as Benford's proportions are binned (i.e., 10, 11 …99), the data is categorical, and not numerical. But given how it is calculated (Goodman, 2016), it is misleadingly sensitive to the size of the data sets being tested.

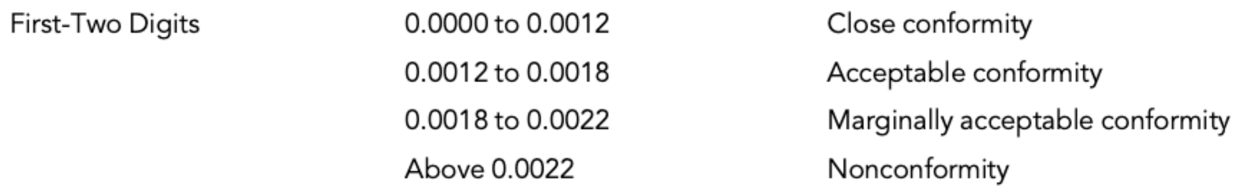
Again, Nigrini (Nigrini, 2020) came up with an ingenious research approach: first, he proposed a measure which is not sensitive to the number of records, and second, he meticulously researched natural and unnatural data in order to determine conformity.

The mean absolute deviation (MAD) is such a test which ignores the number of records because the denominator is the number of records, but the number of bins (i.e., 10-99), which always stays constant in the MAD formula:

$$MAD = \left(\frac{\Sigma \mid OP_i \; - \; EP_{\;i} \mid}{K}\right)$$

OP: observed proportion.
EP: expected proportion based on Benford’s distribution.
K: number of bins, which is 89 (first-two digits 10-99).

A considerable amount of research went into analyzing natural datasets for the MAD levels, mainly by Nigrini (Nigrini, 1997). For example, earth science data gave near-perfect conformity to Benford's distribution and a MAD level of 0.0001. The results based on Nigrini's research (Nigrini, 2020):



First-Two Digits	0.0000 to 0.0012	Close conformity
	0.0012 to 0.0018	Acceptable conformity
	0.0018 to 0.0022	Marginally acceptable conformity
	Above 0.0022	Nonconformity

Figure 4: Critical Mean Average Deviation (MAD) values. Source: Mark Nigrini (2020).

In summary, figure 4 shows the individual levels for conformity based on Mean Average Deviation. For this research thesis, we will ignore the different granular levels and only focus on nonconformity at 0.0022.

In terms of conformity, there are two levels: a global and a local level. The global maxima refer to the high level for conformity, i.e., the MAD level.

The local maxima refer to the individual invoices. In general, we can use any statistical measure such as the z-score or t-test for statistical significance tests. In the context of invoice fraud, the local maxima label individual invoices as a fraud.

Our research will only focus on global maxima for two reasons: first, focusing only on MAD as a conformity level makes the hypothesis testing objective. Second, with invoice fraud, a vendor is most likely to conform or nonconform. In other words, we assume that a vendor commits fraud or does not.

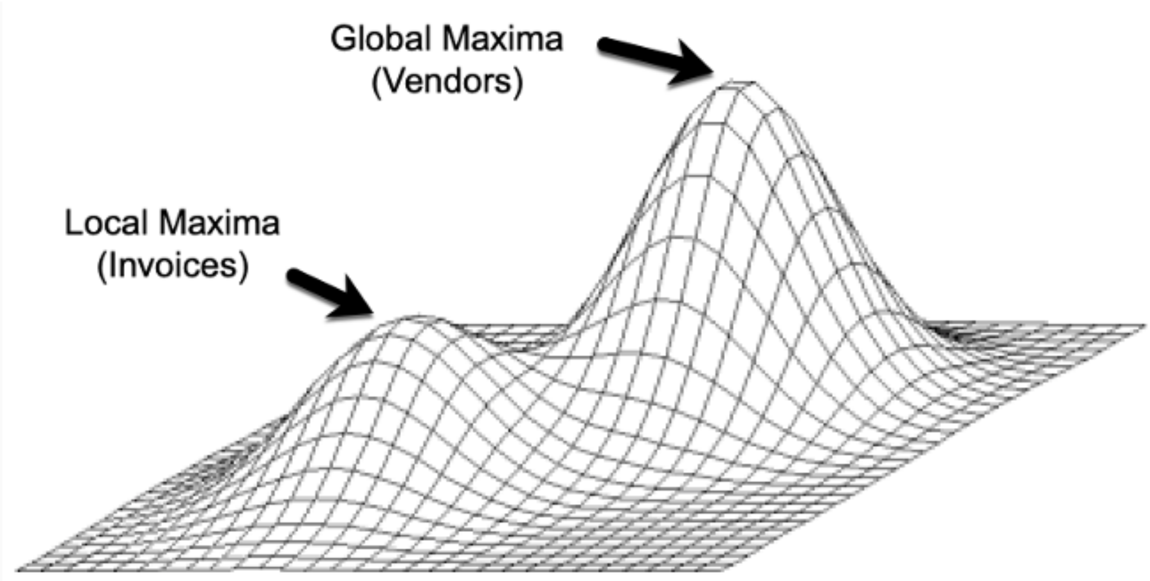


Figure 5: Global maxima and local maxima with invoices. Source: Franco Arda (2020).

Creating a dataset of 100 vendors.

According to some of the most prominent Benford researchers (Berger and Till, 2020), there is no simple, intuitive argument to explain the appearance of Benford's law in the vast array of contexts in which it has been observed, including statistics, number theory, dynamical systems, and real-world data.

In our view, Benford's distribution is similar to pi (π), which is approximately equal to 3.14159, for finding the area and circumference of a circle. Both phenomena occur naturally, but we have no scientific explanation of why.

As mentioned in the abstract, we do not know how fraudsters invent numbers. Our best assumption is that fraudsters invent numbers by randomizing data. To our knowledge, there is no public dataset of invoice fraudsters. The risk of this assumption is that we might be wrong.

However, if we were wrong, there is most likely no discernible pattern in creating fake data. The advantage of this assumption is that we can test our hypotheses objectively against other research papers.

To create a synthetic dataset, we pretend that the data reflects invoice fraud. Invoice fraud occurs when a vendor sends fictive invoices. Nevertheless, the data entries could be anything, not only invoices.

The synthetic dataset with 100 vendors includes invoices for goods and services received from vendors. For example, a company might have maintenance done for servers for $10,500, and the vendor will send an invoice for this amount to the accounts payable departments.

For the synthetic dataset, we selected five vendors to be nonconform. Based on other research (Insurance Information Institute, 2020), fraud often ranges between 1% to 10%, depending on the industry.

For our synthetic dataset, we chose the "middle" or 5% as a fraud. To select the invoice amounts for those fraudsters, we randomized the invoice amounts. We chose an invoice range from $10 to $99,999. Based on Berger and Till (Berger and Till, 2020), it is an error that the random variable or dataset needs to cover several orders of magnitude. In other words, it is a standard research assumption that the Benford distribution requires data samples from several magnitudes.

The choice of 10,000 invoices for each dataset is somewhat arbitrary and based on reverse elimination. We did some primary research and found that one thousand invoices are too small. In the same primary research, we found no difference between testing for 10,000 invoices or 100,000 invoices. Based on those experiments, we chose a dataset of 10,000 invoices.

Benford's law is a counterintuitive statistical regularity that is part of many tests in forensic accounting tests. Benford-based tests are effective at detecting abnormal duplications in our invoice data. Actual frequencies that deviate significantly based on MAD (mean average deviation) from the norm point us towards a set of abnormal duplications.

Under certain conditions, nonconformity to Benford indicates an increased risk of invoice fraud, an error, or just a regular invoice structure (e.g., numerous small invoices from certain vendors). Therefore, nonconformity to Benford does not signal fraud or error with certainty. This part becomes particularly apparent when we simulate data. By simulating datasets, we become much more alert to what the Benford algorithm considers to conform and nonconform.

Benford's logarithmic pattern's closest conformities come from data in which the numbers had no relationship to each other, such as the invoice numbers based on a particular user. We believe that visualizations of the logarithmic graph, e.g., ranging from 10 to 10,000, are misleading. In other words, in a sorted histogram, we can sort the variables. This sorting can lead to wrong conclusions.

In our synthetic dataset, with five nonconform vendors, we need another 95 vendors conforming to Benford's distribution.

There are a few rules for creating a synthetic Benford distribution (Nigrini, 2020). The invoice data forms a sequence, which is a finite or infinite ordered progress of invoices. An example of such a sequence is $\mathbb{N}$ {1,2,3} or $\mathbb{R}$ {$11.25, $23.55, $48.59 …}. Whether the sequence has a chance of conforming to Benford's distribution depends on whether it converges to Benford's distribution (i.e., whether it reaches some limit). Two issues stand out: (1) since I start with the "population" of invoice data of 10,000, We need to mold the dataset based on Benford's distribution. Issue (2) is more of practical benefit: if a fraudster gets his hands on this research paper, they will have no idea how to "beat" the model. We do not even know how to build an invoice data set, which grows from 1 to n without being detected.

For an invoice fraudster, this is terrible news. For us, this is excellent news. We do not want fraudsters to find a template to beat the algorithm. Additionally, this should give users of the algorithm confidence that beating the algorithm is, if not impossible, at least extremely hard.

One researcher (Meitner, 2019) argued that we should not trust Benford's findings as many (cheating) CFO's knows about the law. In defense of the researcher, he refers to Benford's law's simple version, the single-digit version. We do not refer to the single-digit algorithm Benford version in this research paper for space reasons but agree with the researcher.

The simple algorithmic version is too simple for real-world purposes, and too many people know about it. However, the complexity of the first-two digits algorithm, hopefully, its accuracy, is tough to beat, hopefully, shown by this chapter's length and complexity.

To detect fake datasets, we need to be able to compare them to a perfect Benford dataset. Perfect in the sense of Benford distribution in order to measure the performance against a fake dataset. First, we need to set the numbers range. The numbers range, for example, 10 – 10,000 give us a lower and upper bound.

The following formula gives us the difference between the logs of the upper and lower bounds.

d = log(upper_bound) – log(lower_bound)
  = log(10,000) – log(10)
  = 3

With d we can calculate the expected and perfect Benford distribution:

= lower_bound*(lower_bound^(d/n))^(n-1)

The result is a perfect geometric sequence. The MAD for the first-two digits is an extremely low 0.00016. Some small deviations occur even with 400 records. The general rule is that as n increases, we will get an ever-closer level of perfection. As we are concerned with detecting fake data as soon as possible, meaning with the lowest samples statistically possible, we are not concerned with large samples.

Applying this log transformation, we get the following expected Benford distribution for the first-two digits:

Digit	Log transformation	Distribution	Digit	Log transformation	Distribution
10	$\text{Log}_{10}(1+(1/10))$	4.14%	55	$\text{Log}_{10}(1+(1/55))$	0.78%
11	$\text{Log}_{10}(1+(1/11))$	3.78%	56	$\text{Log}_{10}(1+(1/56))$	0.77%

12	Log$_{10}$(1+(1/12))	3.48%	57	Log$_{10}$(1+(1/57))	0.76%
13	Log$_{10}$(1+(1/13))	3.22%	58	Log$_{10}$(1+(1/58))	0.74%
14	Log$_{10}$(1+(1/14))	3.00%	59	Log$_{10}$(1+(1/59))	0.73%
15	Log$_{10}$(1+(1/15))	2.80%	60	Log$_{10}$(1+(1/60))	0.72%
16	Log$_{10}$(1+(1/16))	2.63%	61	Log$_{10}$(1+(1/61))	0.71%
17	Log$_{10}$(1+(1/17))	2.48%	62	Log$_{10}$(1+(1/62))	0.69%
18	Log$_{10}$(1+(1/18))	2.35%	63	Log$_{10}$(1+(1/63))	0.68%
19	Log$_{10}$(1+(1/19))	2.23%	64	Log$_{10}$(1+(1/64))	0.67%
20	Log$_{10}$(1+(1/20))	2.12%	65	Log$_{10}$(1+(1/65))	0.66%
21	Log$_{10}$(1+(1/21))	2.02%	66	Log$_{10}$(1+(1/66))	0.65%
22	Log$_{10}$(1+(1/22))	1.93%	67	Log$_{10}$(1+(1/67))	0.64%
23	Log$_{10}$(1+(1/23))	1.85%	68	Log$_{10}$(1+(1/68))	0.63%
24	Log$_{10}$(1+(1/24))	1.77%	69	Log$_{10}$(1+(1/69))	0.62%
25	Log$_{10}$(1+(1/25))	1.70%	70	Log$_{10}$(1+(1/70))	0.62%
26	Log$_{10}$(1+(1/26))	1.64%	71	Log$_{10}$(1+(1/71))	0.61%
27	Log$_{10}$(1+(1/27))	1.58%	72	Log$_{10}$(1+(1/72))	0.60%
28	Log$_{10}$(1+(1/28))	1.52%	73	Log$_{10}$(1+(1/73))	0.59%
29	Log$_{10}$(1+(1/29))	1.47%	74	Log$_{10}$(1+(1/74))	0.58%
30	Log$_{10}$(1+(1/30))	1.42%	75	Log$_{10}$(1+(1/75))	0.58%
31	Log$_{10}$(1+(1/31))	1.38%	76	Log$_{10}$(1+(1/76))	0.57%
32	Log$_{10}$(1+(1/32))	1.34%	77	Log$_{10}$(1+(1/77))	0.56%
33	Log$_{10}$(1+(1/33))	1.30%	78	Log$_{10}$(1+(1/78))	0.55%
34	Log$_{10}$(1+(1/34))	1.26%	79	Log$_{10}$(1+(1/79))	0.55%
35	Log$_{10}$(1+(1/35))	1.22%	80	Log$_{10}$(1+(1/80))	0.54%
36	Log$_{10}$(1+(1/36))	1.19%	81	Log$_{10}$(1+(1/81))	0.53%
37	Log$_{10}$(1+(1/37))	1.16%	82	Log$_{10}$(1+(1/82))	0.53%
38	Log$_{10}$(1+(1/38))	1.13%	83	Log$_{10}$(1+(1/83))	0.52%
39	Log$_{10}$(1+(1/39))	1.10%	84	Log$_{10}$(1+(1/84))	0.51%
40	Log$_{10}$(1+(1/40))	1.07%	85	Log$_{10}$(1+(1/85))	0.51%
41	Log$_{10}$(1+(1/41))	1.05%	86	Log$_{10}$(1+(1/86))	0.50%
42	Log$_{10}$(1+(1/42))	1.02%	87	Log$_{10}$(1+(1/87))	0.50%
43	Log$_{10}$(1+(1/43))	1.00%	88	Log$_{10}$(1+(1/88))	0.49%
44	Log$_{10}$(1+(1/44))	0.98%	89	Log$_{10}$(1+(1/89))	0.49%
45	Log$_{10}$(1+(1/45))	0.95%	90	Log$_{10}$(1+(1/90))	0.48%
46	Log$_{10}$(1+(1/46))	0.93%	91	Log$_{10}$(1+(1/91))	0.47%
47	Log$_{10}$(1+(1/47))	0.91%	92	Log$_{10}$(1+(1/92))	0.47%
48	Log$_{10}$(1+(1/48))	0.90%	93	Log$_{10}$(1+(1/93))	0.46%
49	Log$_{10}$(1+(1/49))	0.88%	94	Log$_{10}$(1+(1/94))	0.46%
50	Log$_{10}$(1+(1/50))	0.86%	95	Log$_{10}$(1+(1/95))	0.45%
51	Log$_{10}$(1+(1/51))	0.84%	96	Log$_{10}$(1+(1/96))	0.45%
52	Log$_{10}$(1+(1/52))	0.83%	97	Log$_{10}$(1+(1/97))	0.45%
53	Log$_{10}$(1+(1/53))	0.81%	98	Log$_{10}$(1+(1/98))	0.44%
54	Log$_{10}$(1+(1/54))	0.80%	99	Log$_{10}$(1+(1/99))	0.44%

Figure 6: Benford distribution for the first-two digits (10-99). Source: Franco Arda (2020).

To summarize those findings more formally:

The sequences ℕ {1,2,3} or ℝ {$11.25, $23.55, $48.59 …} denotes to the positive integers, or natural numbers. For simplicity, we only consider numbers >10 or higher. Lower numbers only work for the first-digit, but not the first-two digit algorithm.

In other words, an invoice amount can be any amount, $\mathbb{R} = (10, \infty)$, ranging from 10 to infinity.

The Benford limiting properties for sequences include three basic facts (Berger and Till, 2020) that:

(i) powers of every continuous random variable converge to Benford's law;

(ii) products of random samples from every continuous distribution converge to Benford's law; and

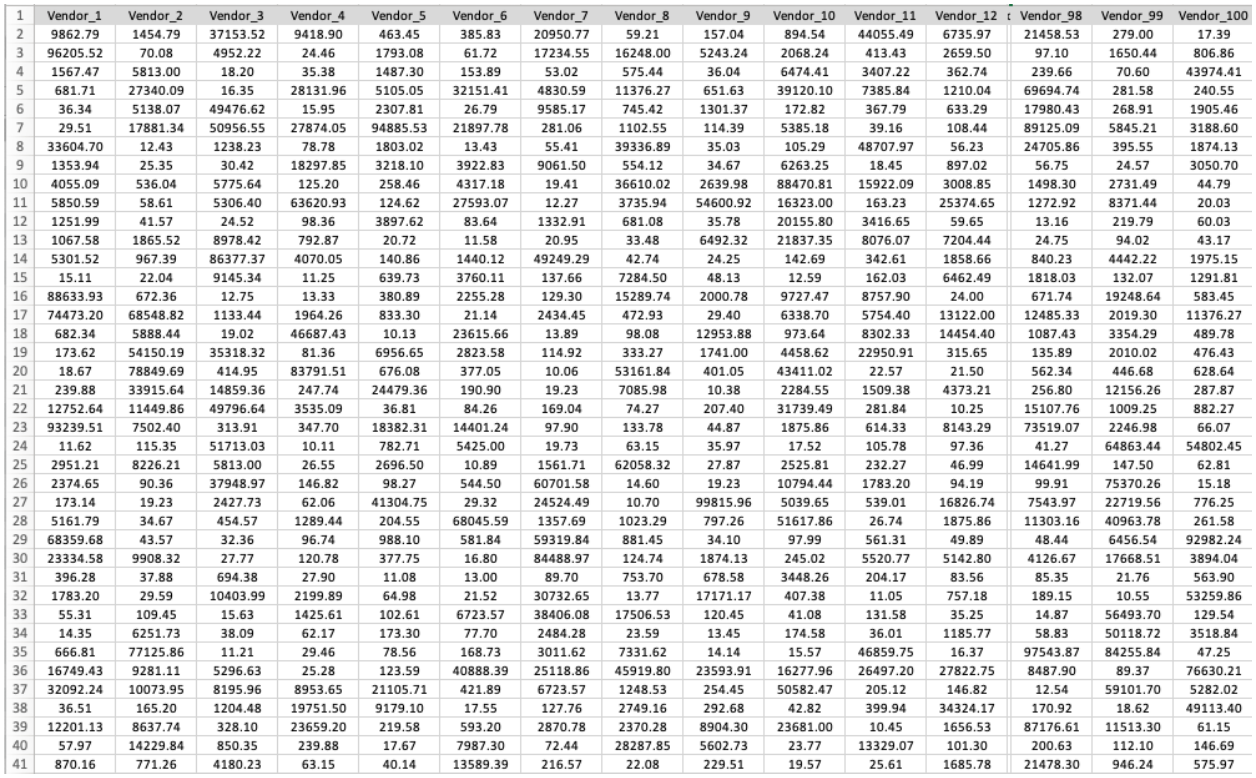
(iii) if random samples are taken from random distributions chosen in an unbiased way, then the combined samples converge to Benford's law.

The following follows in mathematical terms: an infinite sequence of random invoices (X_1, X_2, X_3, …) converges in distribution to Benford's law if:

$$\lim_{n\to\infty} P(S(X_n) \leq t) = \log_t \qquad \text{for all } t \in [10, \infty)$$

In Figure 6, we can see the first 40 rows from a number of vendors (in total 100 columns / vendors, but shortened for visual purposes).

In figure 7 below, we see a snapshot of the synthetic dataset created for the thesis. The columns represent the vendors (Vendor 1 to Vendor 100). With almost 10,000 invoices for each vendor, we have nearly one million invoices for the dataset.



1	Vendor_1	Vendor_2	Vendor_3	Vendor_4	Vendor_5	Vendor_6	Vendor_7	Vendor_8	Vendor_9	Vendor_10	Vendor_11	Vendor_12	Vendor_98	Vendor_99	Vendor_100
2	9862.79	1454.79	37153.52	9418.90	463.45	385.83	20950.77	59.21	157.04	894.54	44055.49	6735.97	21458.53	279.00	17.39
3	96205.52	70.08	4952.22	24.46	1793.08	61.72	17234.55	16248.00	5243.24	2068.24	413.43	2659.50	97.10	1650.44	806.86
4	1567.47	5813.00	18.20	35.38	1487.30	153.89	53.02	575.44	36.04	6474.41	3407.22	362.74	239.66	70.60	43974.41
5	681.71	27340.09	16.35	28131.96	5105.05	32151.41	4830.59	11376.27	651.63	39120.10	7385.84	1210.04	69694.74	281.58	240.55
6	36.34	5138.07	49476.62	15.95	2307.81	26.79	9585.17	745.42	1301.37	172.82	367.79	633.29	17980.43	268.91	1905.46
7	29.51	17881.34	50956.55	27874.05	94885.53	21897.78	281.06	1102.55	114.39	5385.18	39.16	108.44	89125.09	5845.21	3188.60
8	33604.70	12.43	1238.23	78.78	1803.02	13.43	55.41	39336.89	35.03	105.29	48707.97	56.23	24705.86	395.55	1874.13
9	1353.94	25.35	30.42	18297.85	3218.10	3922.83	9061.50	554.12	34.67	6263.25	18.45	897.02	56.75	24.57	3050.70
10	4055.09	536.04	5775.64	125.20	258.46	4317.18	19.41	36610.02	2639.98	88470.81	15922.09	3008.85	1498.30	2731.49	44.79
11	5850.59	58.61	5306.40	63620.93	124.62	27593.07	12.27	3735.94	54600.92	16323.00	163.23	25374.65	1272.92	8371.44	20.03
12	1251.99	41.57	24.52	98.36	3897.62	83.64	1332.91	681.08	35.78	20155.80	3416.65	59.65	13.16	219.79	60.03
13	1067.58	1865.52	8978.42	792.87	20.72	11.58	20.95	33.48	6492.32	21837.35	8076.07	7204.44	24.75	94.02	43.17
14	5301.52	967.39	86377.37	4070.05	140.86	1440.12	49249.29	42.74	24.25	142.69	342.61	1858.66	840.23	4442.22	1975.15
15	15.11	22.04	9145.34	11.25	639.73	3760.11	137.66	7284.50	48.13	12.59	162.03	6462.49	1818.03	132.07	1291.81
16	88633.93	672.36	12.75	13.33	380.89	2255.28	129.30	15289.74	2000.78	9727.47	8757.90	24.00	671.74	19248.64	583.45
17	74473.20	68548.82	1133.44	1964.26	833.30	21.14	2434.45	472.93	29.40	6338.70	5754.40	13122.00	12485.33	2019.30	11376.27
18	682.34	5888.44	19.02	46687.43	10.13	23615.66	13.89	98.08	12953.88	973.64	8302.33	14454.40	1087.43	3354.29	489.78
19	173.62	54150.19	35318.32	81.36	6956.65	2823.58	114.92	333.27	1741.00	4458.62	22950.91	315.65	135.89	2010.02	476.43
20	18.67	78849.69	414.95	83791.51	676.08	377.05	10.06	53161.84	401.05	43411.02	22.57	21.50	562.34	446.68	628.64
21	239.88	33915.64	14859.36	247.74	24479.36	190.90	19.23	7085.98	10.38	2284.55	1509.38	4373.21	256.80	12156.26	287.87
22	12752.64	11449.86	49796.64	3535.09	36.81	84.26	169.04	74.27	207.40	31739.49	281.84	10.25	15107.76	1009.25	882.27
23	93239.51	7502.40	313.91	347.70	18382.31	14401.24	97.90	133.78	44.87	1875.86	614.33	8143.29	73519.07	2246.98	66.07
24	11.62	115.35	51713.03	10.11	782.71	5425.00	19.73	63.15	35.97	17.52	105.78	97.36	41.27	64863.44	54802.45
25	2951.21	8226.21	5813.00	26.55	2696.50	10.89	1561.71	62058.32	27.87	2525.81	232.27	46.99	14641.99	147.50	62.81
26	2374.65	90.36	37948.97	146.82	98.27	544.50	60701.58	14.60	19.23	10794.44	1783.20	94.19	99.91	75370.26	15.18
27	173.14	19.23	2427.73	62.06	41304.75	29.32	24524.49	10.70	99815.96	5039.65	539.01	16826.74	7543.97	22719.56	776.25
28	5161.79	34.67	454.57	1289.44	204.55	68045.59	1357.69	1023.29	797.26	51617.86	26.74	1875.86	11303.16	40963.78	261.58
29	68359.68	43.57	32.36	96.74	988.10	581.84	59319.84	881.45	34.10	97.99	561.31	49.89	48.44	6456.54	92982.24
30	23334.58	9908.32	27.77	120.78	377.75	16.80	84488.97	124.74	1874.13	245.02	5520.77	5142.80	4126.67	17668.51	3894.04
31	396.28	37.88	694.38	27.90	11.08	13.00	89.70	753.70	678.58	3448.26	204.17	83.56	85.35	21.76	563.90
32	1783.20	29.59	10403.99	2199.89	64.98	21.52	30732.65	13.77	17171.17	407.38	11.05	757.18	189.15	10.55	53259.86
33	55.31	109.45	15.63	1425.61	102.61	6723.57	38406.08	17506.53	120.45	41.08	131.58	35.25	14.87	56493.70	129.54
34	14.35	6251.73	38.09	62.17	173.30	77.70	2484.28	23.59	13.45	174.58	36.01	1185.77	58.83	50118.72	3518.84
35	666.81	77125.86	11.21	29.46	78.56	168.73	3011.62	7331.62	14.14	15.57	46859.75	16.37	97543.87	84255.84	47.25
36	16749.43	9281.11	5296.63	25.28	123.59	40888.39	25118.86	45919.80	23593.91	16277.96	26497.20	27822.75	8487.90	89.37	76630.21
37	32092.24	10073.95	8195.96	8953.65	21105.71	421.89	6723.57	1248.53	254.45	50582.47	205.12	146.82	12.54	59101.70	5282.02
38	36.51	165.20	1204.48	19751.50	9179.10	17.55	127.76	2749.16	292.68	42.82	399.94	34324.17	170.92	18.62	49113.40
39	12201.13	8637.74	328.10	23659.20	219.58	593.20	2870.78	2370.28	8904.30	23681.00	10.45	1656.53	87176.61	11513.30	61.15
40	57.97	14229.84	850.35	239.88	17.67	7987.30	72.44	28287.85	5602.73	23.77	13329.07	101.30	200.63	112.10	146.69
41	870.16	771.26	4180.23	63.15	40.14	13589.39	216.57	22.08	229.51	19.57	25.61	1685.78	21478.30	946.24	575.97

Figure 7: The synthetic dataset of 100 vendors with each 10,000 invoices. Source: Franco Arda (2020).

To apply statistical methods, we needed to ensure that the samples are truly random, and the actions have not influenced that invoice datasets during the creation. Technically, we can not create truly random numbers, but only pseudo-random numbers.

We also had to ensure that we later sample from the "right" population. Here, we used information from the later experiments and argue that it is not cheating. As stated before, a synthetic population size of 50,000 or 100,000 would have made no sense. Such a large population would have made fraud detection most likely futile. The more delicate decision was if a population size of 1,000 would have been sufficient.

With hindsight from later experiments, we know now that the sample size is around 1,600, depending on the given confidence level. We argue that this knowledge of hindsight is not cheating, as we could have displayed all the steps in this part. Had we done this and showed all tests ranging from 1,000 to 100,000, the page count of this thesis would have increased unnecessarily.

In other words, we formulated both hypotheses at the beginning, with a confidence level at 95% and a p-value of 0.05. Using knowledge from the experiment regarding the synthetic population size does, therefore, not affect our hypotheses.

In the formulation of the hypothesis, we also defined what we are going to measure. For the first research hypothesis, it was the sample size required, given a confidence level. The second research hypothesis was the p-value for the difference in accuracy between benchmarks and our alternative hypothesis.

The importance of sample size in fraud detection.

For many researchers, detecting fake data as early as possible is not a concern, which is understandable. Suppose a researcher is concerned about the effectiveness of an algorithmic approach, ex-post, or "after the fact" tests are sufficient.

In other words, a researcher might take thousands of invoices from several years of collection and test different approaches. However, it is of ultimate interest for a company to detect invoice fraud as fast as possible in a real-world business application. For example, let us say a company receives fake invoices. The following fictive visualization shows the problem clearly. The total amount of fake invoices over several years amount to over $4 million. Now, if the company had successfully detected the fake invoice scam, it could have saved millions.

Figure 8 visualizes the importance of detecting fraud as early as possible. With invoice fraud, the earlier we can detect fraud, the smaller the number of losses in fraud we incur.

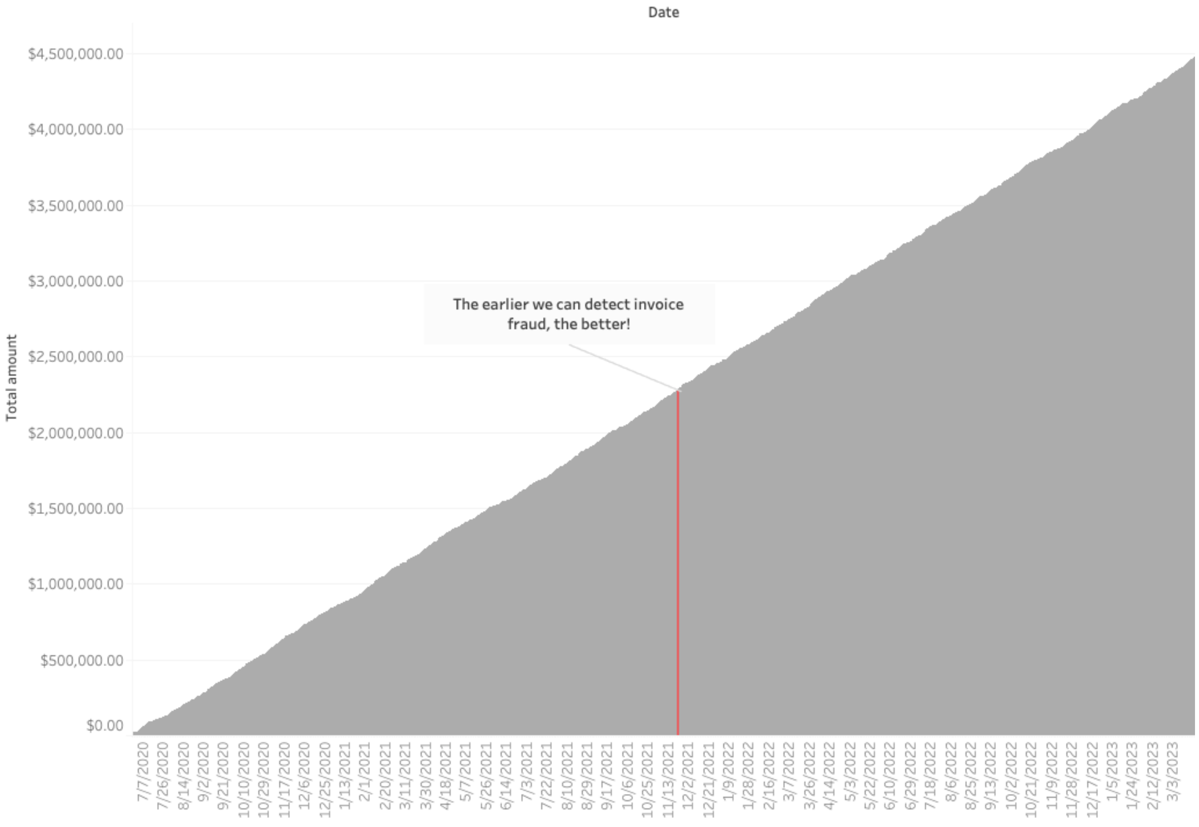


Figure 8: With fraud in general, but invoice fraud in particular, a real-time monitoring of invoice fraud, the credo is "detect as early as possible." Source: Franco Arda (2020).

Economically the ideal amount lost to fake invoices would be zero, but that is virtually impossible to achieve. Mathematically, the best alternative to no fake invoices is the smallest possible amount of fake invoices. In order to determine the smallest possible amount of fake invoices, we need a statistical tool. In other words, after how many numbers of invoices can our algorithm decide, with a certain confidence level (e.g., 95%), when we have enough samples?

As part of our research hypothesis two, this study examines the relationship between a minimum sample required and its corresponding confidence interval. Although a considerable body of research has focused on the application of Benford's Law, less attention has been paid to the sample required. As we will see shortly, the lack of attention to the required sample size might stem from calculating the required sample size is quite challenging.

However, for any company monitoring fraud in general, and invoice fraud, the required sample at a given statistical confidence level is crucial. The required sample not only helps in reducing type 1 errors (false alarm) but also in implementing real-time monitoring. With the required sample given a confidence interval, real-time monitoring can set in place, and a company can potentially catch invoice fraud much earlier.

Companies often use audits at the end of the fiscal year, and invoice fraud might light. With real-time monitoring, an invoice fraud might become apparent – any time of the year. Real-time monitoring can be financially extremely beneficial. For example, the company Facebook paid out $100 million in fake invoices (CNBC, 2019). A detection only after half the invoices could have said the company millions.

In summary, with fraud detection, the challenge is two-fold: on the one hand, we need a low false-positive rate, and on the other hand, we need a low sample size. If our sample size is statistically speaking too small, we will incur too many false positives. However, if our sample size is too large, we will not detect fraud early enough.

A statistical method to calculate the required sample given a confidence level.

The Benford distribution with its bins ranging from 10 to 99 is a Multinomial distribution. The Multinomial distribution is a generalization of the Binomial. Whereas the Binomial distribution counts the successes in a fixed number of trials that can only be categorized as success or failure, the Multinomial distribution keeps track of trials whose outcomes fall into multiple categories (in this case, bins ranging from 10 – 99).

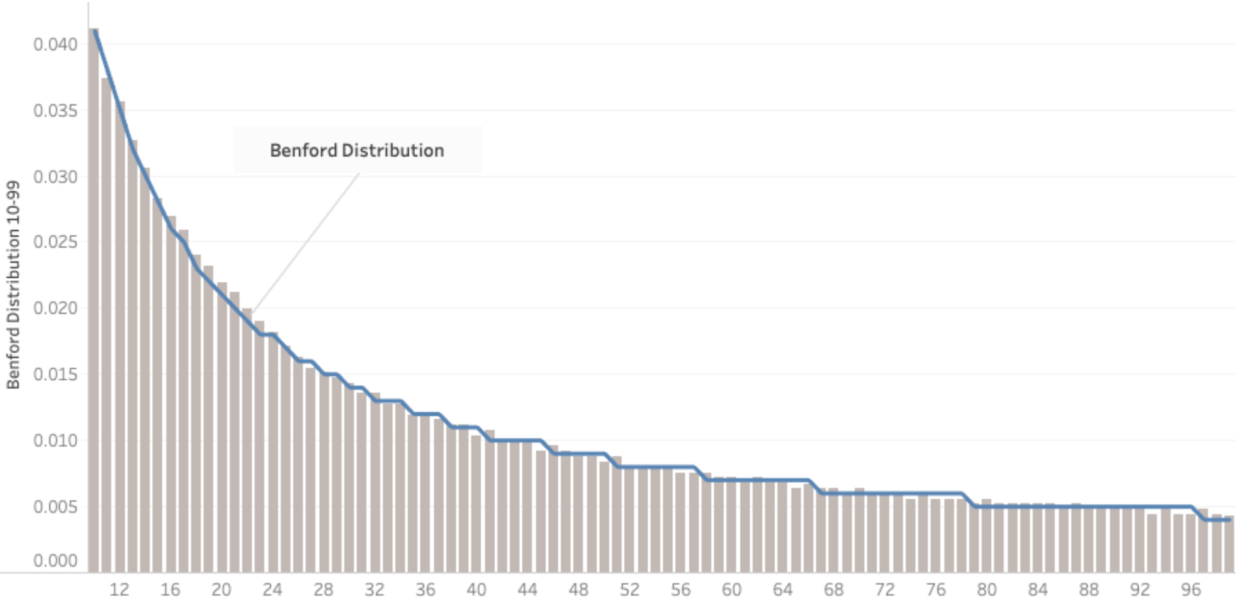


Figure 9: A synthetic data that conform almost perfectly to the Benford distribution (blue line). Source: Franco Arda (2020).

Visually, the expected Benford distribution is non-normal and skewed to the right. In order to support this observation statistically, we will perform a distribution test. For example, if we take nonparametric data as data that does not look Gaussian, we can use statistical methods that quantify how Gaussian a sample of data (or rather the population in this case) is and use a nonparametric method if the data fails those tests.

The Shapiro-Wilk test evaluates a data sample and quantifies how likely it is that the data was drawn, or not drawn, from a Gaussian distribution named Samuel Shapiro and Martin Wilk. To quantify this observation, we will run the Shapiro-Wilk test on a perfect Benford dataset. Unfortunately, the test limits the data intake to 5,000 data points. As the original Benford dataset consists of 10,000 data points, we randomly sample from the original dataset to curb it to 5,000 rows.

```
set.seed(12)
benford_test <- dplyr::sample_n(benford, 5000)
shapiro.test(Benford_test$Benford)
# p-value < 2.2e-16
```

The null hypothesis of this test is that the sample distribution is normal (Gaussian). If the test is significant, we need to reject the null hypothesis and accept that the given distribution is non-normal. The R function for shapiro_test() can be used to perform the Shapiro-Wilk test of normality for one variable (univariate).

The resulting p-value is 2.2e-16, which is close to zero. From the output, the p-value is < 0.05, implying that the distribution of the data is significantly different from the normal distribution. In other words, we have mathematically confirmed that our visual interpretation was correct that we can assume that the data is not normally distributed.

This step was crucial as we know now that we cannot use a statistical power test to calculate the required sample size. In other words, a statistical power test assumes a normal distribution, which is here not the case.

We feel that we need to make this point even more explicit: if the Benford distribution had been normal or close to normal, we could have applied a statistical power test. Running a statistical power test would have been extremely simple, but that is not the case. Therefore, we need a different approach to test the required sample.

Monte Carlo simulation is a technique used to approximate an event's probability by running the same simulations multiple times and averaging the results (Guttag, 2016). For the first research hypothesis, we will use different stochastic sampling techniques with and without replacement. Those techniques are required to determine the sample size. Which exact technique, not sure yet. One of the possible solutions is bootstrapping with or without replacement.

For the second research hypothesis, the method another Monte Carlo simulation, but this time the exact technique is given for simulation-based hypothesis testing. In this hypothesis test, we have two groups of accuracy.

We want to compare them if one is different based on a p-value of 0.05. The underlying technique is a permutation, where we shuffle the results, assuming the null hypothesis is valid.

Comparing the sample results with the actual results allows us to determine the p-value. This approach is fundamentally different in hypothesis testing than with classic statistics, where the p-value reflects a static number.

For both research hypothesis tests, we use the family of Monte Carlo simulations. Unlike deterministic simulations that do not have a different result with resampling, Monte Carlo simulations do. They are stochastic.
Stochastic simulations incorporate randomness in our hypothesis tests. Multiple runs, in general, over one thousand of the sample model most likely generate different values. These random elements generate many outcomes to see the range of possibilities in confidence levels and p-values.

One potential alternative is bootstrapping. Bootstrapping generates an empirical distribution of a test statistic by repeated random sampling with replacement from the original sample. It allows us to generate confidence intervals without assuming a specific underlying theoretical distribution (i.e., a nonparametric approach).

A nonparametric bootstrapping is another Monte Carlo simulation method for approximating the variation in the test statistic distribution. Nonparametric refers to the idea that the sample data are not generated from a particular parameterized probability model.

Our nonparametric bootstrapping makes no assumptions about the probability model underlying the population. This method uses the distribution of the observed sample data, the empirical distribution, as a proxy for the population distribution. Replicate data sets are then randomly generated, with replacement from the empirical data.

Without replacement, we would always the same results. Since, in theory, the observed sample data should represent the population in which they are drawn, replicates drawn from these data should represent what one would get if many samples from Benford's expected population were drawn.

Bootstrapping tests are feasible only with software to automate the heavy computation that the resampling methods require. Historically, bootstrapping was only available to expert statisticians. However, statistics are changing rapidly.

Modern programming languages such as R make it possible to look at data graphically and numerically in ways previously inconceivable. The bootstrap is part of this revolution. Bootstrapping involves randomly resampling with replacing the observed data many times to generate an empirical distribution for interest statistics.

The resulting distribution allows us to create statistical estimates of variability and confidence limits. There are no distributional assumptions (nonparametric), but most bootstrap estimates run under the premise that the observed data comes from an independent and identically distributed population.

The steps for the first research hypothesis, in regards to the bootstrap, are:

1. Randomly select n invoices from a vendor dataset.

2. Calculate and record sample means or medians.

3. Repeat the first two steps n times (starting at one and going to n where n equals the required sample.)

4. Order the n sample means or medians from the smallest to the largest

5. For a 95% confidence interval, we trim [(1 – [5%/100]) / 2]% of the sample results.

Regarding point (1), in a standard bootstrap, the sample and the size are equal. Of course, in this case, we must use replacement = TRUE. If we did not do so, we would always resample the same data. In other words, if we resampled the data 1,000 times, we would always take the median of the same data 1,000 times.

Nevertheless, here, in point (1), our bootstrap approach differs from the traditional approach. Our sample is the population of Benford's expected distribution of 10,000 data points. Based on the population, we want to know the required sample we need.

The percentage associated with the confidence interval is the confidence level. The higher the level of confidence, the wider the interval. Also, the small the sample, the wider the interval (i.e., the more uncertainty). Both make sense: the more confident we want to be, and the less data we have, the wider we must make the confidence interval to be sufficiently assured of capturing the correct range.

The methods for calculating confidence intervals for a population mean and differences between two populations is only valid for normal distributions. This approach uses the sample means, standard errors, and the t distribution should strictly be used only for continuous data from normal distributions, although small deviations from normality are not substantial. This approach is a so-called parametric approach as it essentially estimates the two parameters of the normal distribution: the mean and standard deviation.

Calculating the 90% confidence and 99% confidence limits is then a simple modification.

As we use sample data based on our bootstrap to infer the population, we get four types of errors. In our case, once we have identified a vendor as nonconform, our null hypothesis (not the research hypothesis) is that all individual invoices are conforming. The four possible outcomes from our decision are:

- If the null hypothesis is false and the statistical test leads to rejecting it, we have made the correct decision.

- If the null hypothesis is true and we accept it (extremely unlikely as we have already defined the dataset as nonconform), again, we have made a correct decision.

- If the null hypothesis is false and we fail to reject it, we have committed a Type I error.

- If the null hypothesis is false and we fail to reject it, we have committed a Type II error. Again, this is extremely unlikely as we have already defined the dataset as a whole as nonconform.

Fortunately, running a hypothesis test on a local level is much straightforward with fake data. The reason for this is that we have already determined whether a vendor's dataset is conforming or not.

We specify a hypothesis about a population parameter (our null hypothesis, or H0), get amounts or samples from this population, and calculate a statistic used to make inferences about the population parameter. Assuming that the null hypothesis is true, we calculate the probability of obtaining the observed sample statistic or one more extreme.

If the probability is sufficiently small, we reject the null hypothesis in favor of its opposite (referred to as the alternative or test hypothesis, H1). To make it crystal clear, this is not the research hypothesis, but the hypothesis to test individual amounts from a vendor once he has been flagged as nonconform (i.e., MAD level > 0.0022).

Monte Carlo simulation is a powerful technique, and there are many problems where there is the only reasonable approach currently available. One problem with a simulation result is that it is easy to criticize (Blitzstein and Hwang, 2019). How do we know that we ran it long enough?

How do we know that our result is close to the truth? Moreover, how close is "close"? We wanted to raise this point that a statistician or mathematician might want provable guarantees. We cannot deliver them, but we can deliver empirical proof that our calculations should be very close to the truth.

The law of large numbers tells us that the approximations should be useful if *n* is large. Now, what is large? According to the Central Limit Theorem (CLT), a sample of 30 is often enough, though sometimes we might need 100.

In general, hypothesis tests can be one-sided or two-sided. We are interested in knowing that there is a difference of digits, regardless of whether the difference is positive or negative for a two-sided test. In other words, the distribution of the first two digits can be too high or low compared to the expected Benford distribution.

With confidence intervals, we calculate a range of similar values between the population and the sample. In other words, our goal is to uncover the ranges for samples iteratively. With this approach, using bootstrap, we should be able to determine the sample size required.

The width of a confidence interval associated with a sample statistic depends partly on its standard error, and hence on both the standard deviation and the sample size. It also depends on the degree of "confidence" that we want to associate with the resulting interval, though this level is much more in our control.

More precisely, in a statistical sense, the confidence interval means if a series of identical samples simulations were carried out repeatedly on different samples from the same population, and a 95% confidence interval for the difference between the sample means calculated in each test, then, in the long run, 95% of these confidence intervals would include the population difference between the medians.

The sample size affects the standard error size, which in turn affects the confidence interval's width. If more significant or less confidence is required, different intervals can be constructed: 99%, 95%, and 99% confidence intervals for the data. As would be expected, greater confidence that the sample reflects the population is obtained with wider intervals.

In practice, intervals other than 99%, 95%, or 99% are rarely quoted. There is a close link between the use of a confidence interval and the two-sided hypothesis test. If the confidence interval is calculated, then the hypothesis test result can be inferred at an associated level of statistical significance.

If the difference observed is outside the 95% confidence interval, this indicates that a statistically significant difference between the two observed values at the 5% level would result from applying a hypothesis test.

The 95% confidence interval covers a wide range of possible mean or median differences, even though the sample difference between the means or medians is different from zero at a 5% level of statistical significance.

The 95% confidence interval shows that the test results are compatible with a small difference. Thus, the confidence interval provides a range of possibilities for the population value, rather than an arbitrary dichotomy based solely on statistical significance. It conveys more useful information at the expense of the precision of the p-value test.

The two extremes of the confidence interval are known as confidence limits. However, the word "limits" suggests that there is no going beyond and maybe misunderstood because, of course, the population value will not always lie within the confidence interval.

The 1-1 relationship between confidence intervals and hypothesis tests make it possible to construct a bootstrap test using any bootstrap method to obtain a confidence interval.

Suppose we have a $100(1 - \alpha)\%$ bootstrap confidence interval; then, if the null hypothesis H_0 is $\theta = \theta_0$, you obtain a bootstrap test by rejecting H_0 if and only if θ_0 lies outside the confidence interval. Use two-sided confidence intervals for two-tailed tests and one-sided confidence intervals for one-tailed tests.

On the other hand, suppose we have an α-level test of the null hypothesis H_0: $\theta = \theta_0$ versus the alternative H_1: $\theta \neq \theta_0$, then a $100(1 - \alpha)\%$ bootstrap confidence interval is given by the set of estimates (or their corresponding test statistics) that fall between the upper and lower critical values for the test.

We see from the bootstrap that we do not involve any assumptions about the data or the customarily distributed sample statistics. If the data were normally distributed (parametric), it would be reasonably straightforward to find the required samples. Conceptually, we can imagine the bootstrap as replicating the original sample thousands of times to have a hypothetical population that embodies all the knowledge from the original sample.

We then draw samples from this hypothetical population to estimate a sampling distribution. Here, though, our research differs from the traditional bootstrap. While the standard bootstrap uses samples to infer to the population, we use the samples to see if they are similar to the population (i.e., the Benford distribution).

In practice, it is not necessary to replicate the sample a vast number of times. We replace each observation after each draw; that is, we sample with replacement. In this way, we effectively create an infinite population in which the probability of an element being drawn remains unchanged from draw to draw.

The algorithm for a bootstrap resampling median for a sample of size n draws a sample value, repeats it n times, and records the n resampled values' median. Now we repeat this process several times and record the results.

The result from this procedure is a bootstrap set of sample statistics or estimated model parameters, which we can then examine to see how variable they are. However, we have to be careful as the bootstrap does not compensate for a small sample size as it does not create new data, nor does it fill the holes in an existing data set. It merely informs us about how many additional samples would behave when drawn from a population like the original population. What too small data is, is very hard to estimate. If we had parametric data, then a sample in the range of 30 – 100 should conform to the theorem's central limit.

Nevertheless, our original population data in Benford's expected distribution has two significant challenges; it is highly skewed and has 90 bins. Going forward, we know that we need at least 100 samples.

The bootstrapping tests are statistically and computationally demanding, but the code snippets combined with the visualizations should give us a tremendous theoretical framework to test our hypothesis. In other words, we can focus our cognitive energy on the hypotheses rather than on coding and statistical labor.

Five or ten years ago, our research paper would only be possible for researchers with a solid statistics background. Today, in particular, with R programming advances for statistical problems, we Data Scientists have tools at our hands to solve challenging business problems.

While not being part of our research work, we can take our research paper, based on those findings, and create a real-time invoice fraud detection system. A system that analyzes in near real-time millions of invoices evaluates each vendor for conformity and visually displays potentially fraudulent invoices.

In conclusion, we now have a theoretical framework in order to proceed to test our first research hypothesis. At this stage, we understand the mechanics of the algorithm and created a synthetic dataset against which we can test our hypothesis. To simulate the first hypothesis, we have chosen a bootstrap method. Bootstrap is part of the Monte Carlo simulation family.

To the best of our knowledge, no other research paper has attempted to determine the sample size required for the Benford algorithm. Therefore, we are currently guessing if the bootstrap method is the correct approach. Intellectually, we know that we need to apply a sample method. The Monte Carlo simulation is the most well-known simulation method.

Methodology for the first research hypothesis

In recent years, researchers have become increasingly interested in determining the sample size required to determine a specific dataset's conformity regarding Benford's distribution. To this date, we have not seen any scientifically based clear answer to this question. Recently, in a statistical research article (Goodman, 2016), they claimed that a sample size of 20 is not enough.

Researchers in accounting struggle with the same question (Collings, 2017). Their result was "the larger, the better," which is not helpful.

While we argue for the importance of determining a statistically significant sample size, we understand that at the beginning of researching Benford's law and its applications, the sample size should not be at the forefront of any researcher. Why? We believe we need to start with a simplified assumption before digging deeper into the sample size required.

However, nowadays, as research in Benford's law has matured to a certain degree, we need to tackle challenging problems of Benford's law. The hardest problem we see, at least in terms of real-world applications, is the required sample size for Benford's law.

To demonstrate the importance and complexity of the sample size required, we need to start when it is not essential. If we have large datasets (e.g., >10,000) and or analyzing datasets that are can be analyzed far from real-time, we do not need to be concerned with the sample size.

With fraud in general and invoice fraud in particular, the required sample size becomes incredibly important. Knowing the required sample size empowers us two essential aspects:

1 – We can detect invoice fraud in real-time.
2 – We can detect invoice fraud as faster.

The two points are incredibly similar and only are distinguishable on a very granular level. We know that an invoice dataset is potentially fraudulent at a 0.0022 mean average deviation level based on years of research. If we know the required sample size required, we have all the parameters required to scan for invoice fraud in an automated manner at a given confidence interval.

In other words, based on those two parameters, we can run our algorithm continuously and scan invoices for potential fraud.

Those two parameters allow us to scan for invoice fraud in real-time.

Second, we can run the algorithm in real-time once we have solved the required sample data. Often, in accounting and auditing, invoice fraud is analyzed once a year.

However, by solving the sample size required, a company can run the algorithm all year and not wait until the end of the year.

For the computational simulations in the statistical programming language R, we will use the following packages. For space efficiency, we will only list them once:

```
library(tidyverse) # R package for data manipulation.
library(infer) # R package for bootstrapping.
library(benford.analysis) R package for Benford analysis.
set.seed(12) # Pseudo random generator starts at 12 (useful for reproducibility of my research).
```

For all hypotheses tests, we will always use the same dataset. The dataset consists of 100 vendors with every 10,000 invoices. We name it benford_100:

```
benford_100 <- read_csv("DBA_dataset_100.csv")
```

In order to simulate the required sample size, we need one perfect Benford dataset. Perfect in the sense that the distribution of the dataset conforms to Benford. To run our simulations, we only need a single dataset as all datasets are identical in distributions. From creating the datasets, we know that Vendor_1 conforms to the Benford distribution.

We name this simulation dataset "Benford_perfect”:

```
Benford_perfect <- benford_100 %>%
  select(Vendor_1)
```

Because this is months of work and the whole research hypothesis, we rely on the correctness of the data; we manually verify the conformity of Vendor_1 to Benford's distribution.

```
library(benford.analysis)
test <- select(Benford_perfect, Vendor_1)
bfd <- benford(test$Vendor_1)
bfd
# Mean Absolute Deviation (MAD): 0.0001214141
```

With the dataset Vendor_1, we get a MAD level of 0.0001214141, close to a perfect Benford distribution. As real data probably will never conform perfectly (perfect MAD = 0), we are close. Alternatively, in other words, Benford's nonconformity is defined as MAD level 0.002 from which we are far off.

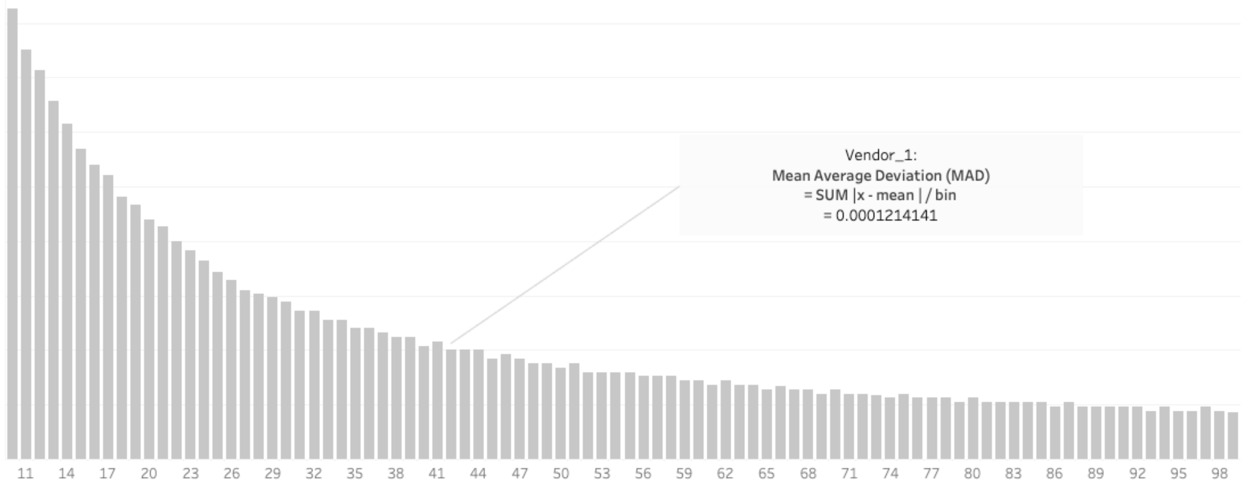


Figure 10: Histogram of the first-two digit frequency for a Benford conformity dataset of Vendor_1. Source: Franco Arda (2020).

The challenge with the conform Benford distribution is that it does not follow any know statistical distribution such as a normal distribution, Bernoulli distribution, or uniform distribution. It is a multinomial distribution with a positive skew. While the multinomial distribution is a generalization of the binomial distribution and refers to categorical data ranging from 10 to 99 (the bins for the first two digits), the positive skew refers to the long-tail the right.

Therefore, we are dealing with a nonparametric distribution (unknown mean and standard deviation) and need to simulate the required sample. Unfortunately, research on the subject is light, as the multinomial and nonparametric distribution is probably rare. Hints to our approach are mentioned (Bruce, 2019): "The bootstrap can be used for sample size determination; experiment with different values for n to see how the sampling distribution is affected."

A similar method (Blitzstein and Hwang, 2019) has been used to simulate another hard problem: estimating pi via computational simulation.

Technically, what we use is known under the umbrella term, a Monte Carlo simulation where we sample with or without replacement at random from a given dataset. The bootstrap is part of the Monte Carlo family, where we estimate the confidence interval. It can solve challenging problems, such as this research attempts, without extensive study of mathematical approximations to sampling distributions.

We are keeping the mathematical part to a minimum, not because we do not like mathematics, quite the contrary, but because we are trying to solve statistical problems through code.

In general, this trend has been seen in the last ten years in statistics, where statistics is being taught mainly through code. More prominently, in our industry, Data Science, the focus on code, has been fundamental.

Many of today's scientific problems have reached a complexity level, such as determining the sample size in the first research hypothesis, that there are almost impossible to solve mathematically. The shortness of the following code might suggest simplicity, but the computational complexity is relatively high. High enough that they are hard to solve mathematically.

As discussed before, because we evaluate a dataset's conformity based on Mean Average Deviation (MAD), we can conclude that we should determine the sample based on the mean as well. We admit this is statistically unusual. With a highly skewed dataset, the median tends to give us more stable results. However, as we do not have acceptable errors based on Benford's expected distribution, this is essential MAD's usefulness, we do our simulations based on the mean.

Let us go step-by-step from the known population to the sample size required. What is the mean of all invoices from Vendor_1?

```
mean(benford_100$Vendor_1)
# 10843.01
```

In the bootstrap, we want to generalize from a sample. While we have never done such an extensive bootstrap, we will start with a population simulation (Vendor_1). We have never seen this in a research paper or a textbook, but it makes intuitive sense. As the bootstrap is a procedure that uses the given sample to create a new distribution, called the bootstrap distribution that approximates the sampling distribution for the sample mean, we start first with the population.

We start first with the population. If the simulation approach works on the population, we can deduce that the same approach works for smaller sizes (i.e., our goal is to go as small as possible.)

In this step, we take a sample size from the Vendor_1 (size = 1000), simulate it 1000 times (reps = 1000), and use replacement (replace = TRUE).

```
resample_population <- benford_100 %>%
  rep_sample_n(size = 1000, reps = 1000, replace = TRUE)

resample_population_median <- resample_population %>%
  group_by(replicate) %>%
  summarize(mean_population = mean(Vendor_1))
# 10840.99
```

The goal is to see if the 1,000 times resampled mean with only 1,000 samples gives us a similar mean.

```
resample_population_median %>%
  select(mean_population) %>%
  summarize(mean(mean_population))
```

Indeed, the mean is extremely stable:
- Vendor_1 mean = $10,843.01
- Resampled mean = $10,840.99

We believe it is beneficial to summarize what we just did. To find the bootstrap distribution of the mean, we draw samples of size n (i.e., 1000), with replacement, from the original or population (Vendor_1), and then compute the mean of each resample. In other words, we now treat the original sample (i.e., 1000) as the population.

A certain amount of difference (or error) is expected as we are essentially using a random method to solve a deterministic problem.

Visually, in a histogram, this is our result (I omit the code to avoid repetition):

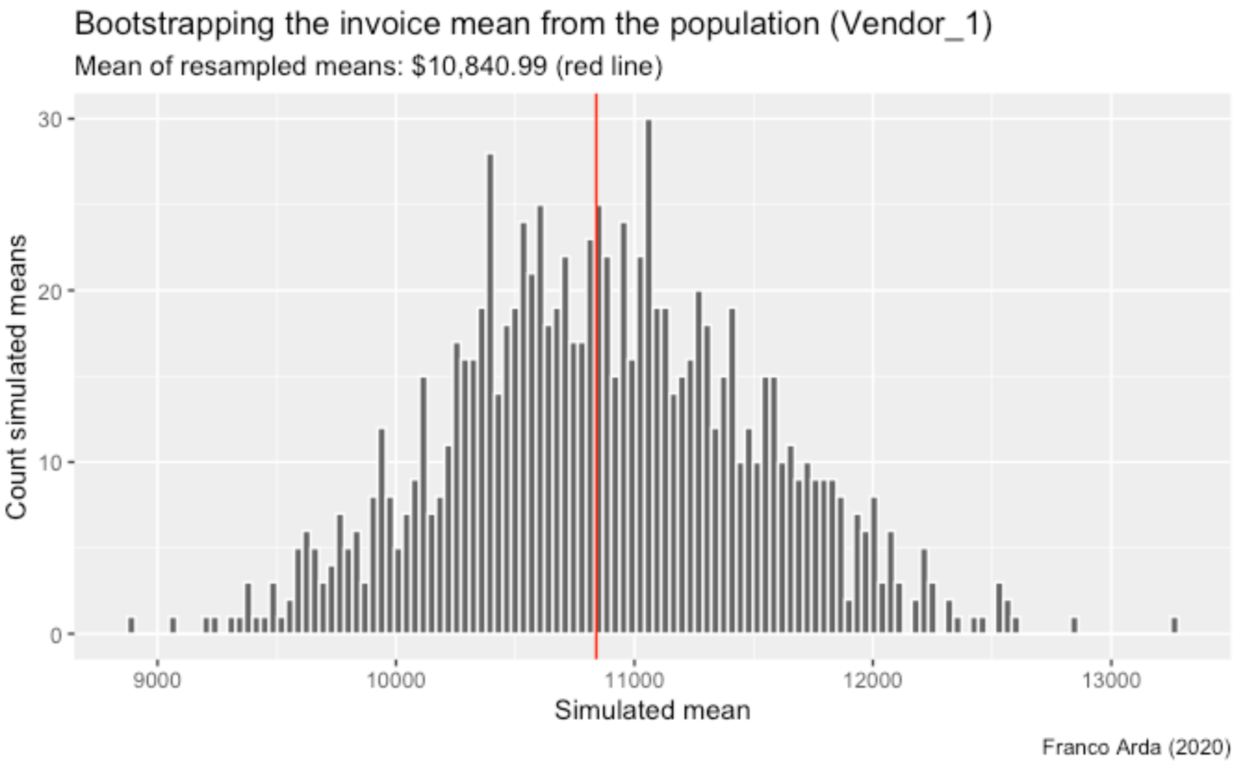


Figure 11: Bootstrapping the invoice mean from the population (Vendor_1). Source: Franco Arda (2020).

The bootstrap's idea is that if the original sample represents the population, then the mean's bootstrap distribution will look approximately like the sampling distribution of the mean. In other words, the sampling distribution has roughly the same spread and shape.

However, the bootstrap distribution's mean will be the same as the original sample's mean, not necessarily that of the original distribution (Hesterberg and Chihara, 2019). As our research experiment will show, we come to the same conclusion where Benford's distribution is nonnormal, while our sample distribution is normal (i.e., conforms to the central limit theorem).

Brad Efron, the inventor of the bootstrap (Efron and Tibshirani, 1993), suggested in 1993 that n = 200, or even as few as n = 25, suffices for estimating the standard errors and that n = 1000 is enough for confidence intervals.

Again, our research experiment shows that we need around 1,000 simulations for the bootstrap.

Using confidence intervals to determine the sample size

The work of Murphy (Murphy, 2012) demonstrates that the bootstrap, a simple Monte Carlo technique, can be used to approximate the sampling distribution. This approach is particularly useful in cases where the estimator (i.e., sample size) is a complex function of the parameters.

We add the sampled mean of $10,840 and simulate at the 95% confidence interval in the first simulation.

We use the same number of samples (i.e., size = 1000) and repetitions (i.e., reps = 1000):

```
sample_size_test_1 <- benford_100 %>%
  specify(response = Vendor_1) %>%
  generate(size = 1000, reps = 1000, type = "bootstrap", replace = TRUE) %>%
  calculate(stat = "mean")

percentile_ci <- sample_size_test_1 %>%
  get_confidence_interval(level = 0.95, type = "percentile")
percentile_ci
# 2.50% at 10428 and 97.50% at 11241
```

The 2.50% confidence level is at 10428 and 97.50% at 11241. A suitable confidence interval for the mean need not necessarily be symmetric: an endpoint from the sample mean in the direction of any outliers. A confidence interval is an insurance policy: rather than relying on a single statistic, the sample means, as an estimate of the mean, we give a range of plausible values for the mean (Hesterberg and Chihara, 2019).

Visualizing the bootstrap sample reveals statistically exciting behavior, that the invoice sample man is normally distributed:

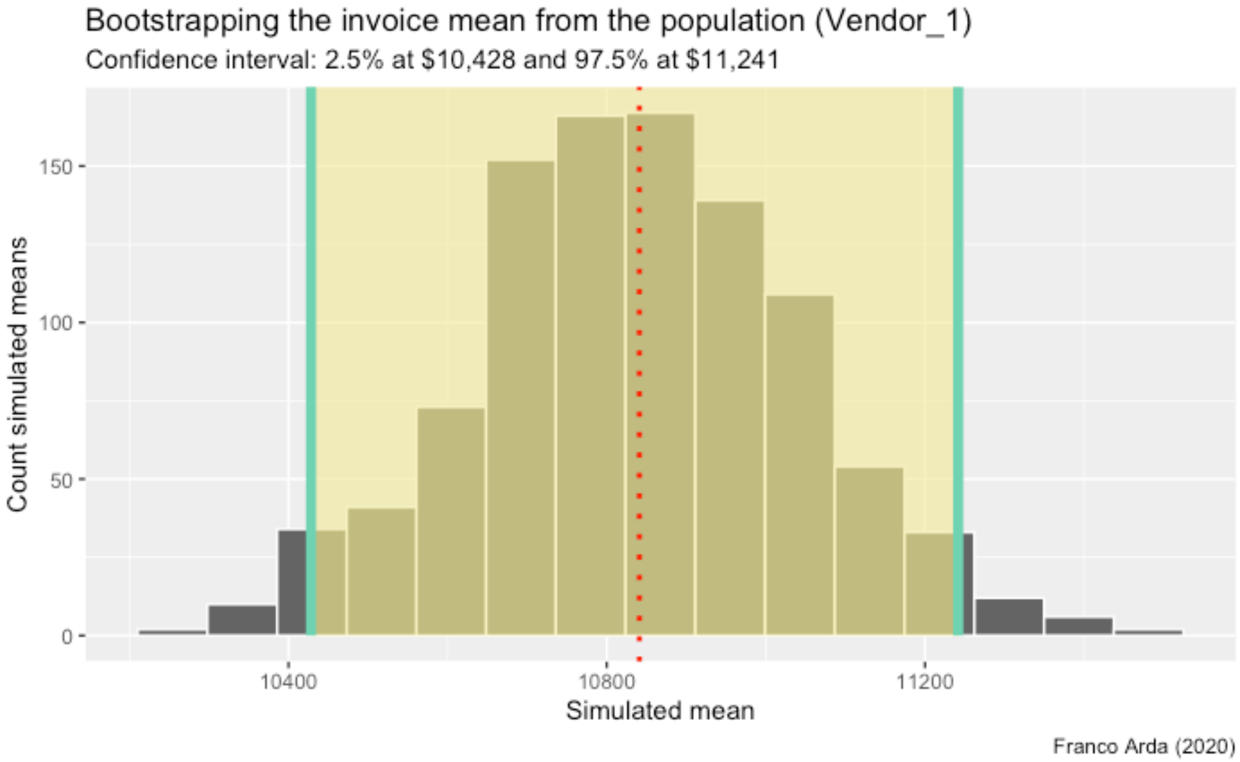


Figure 12: Bootstrapping the invoice mean from the population (Vendor_1). Source: Franco Arda (2020).

Interestingly, one of the most important mathematical concepts holds, even for the Benford distribution: The Central Limit Theorem (CLT). While the Benford distribution is extremely skewed to the right, the samples taken from the population (Vendor_1) are fairly symmetrically distributed.

Researchers and practitioners do not use the sampled population mean (dotted red line) but only focus on the confidence intervals; in this case, 2.5% and 97.5%. We have never seen anyone else using it. However, we like having the dotted line as an addition to the confidence interval.

The primary reason is that we get a better intuition of the behavior for the underlying data. Moreover, on the off chance of making a horrible calculation mistake, we might spot it. In such a case, the dotted red line might be outside of the two confidence intervals. If this happened, then we have produced an extreme error or used a way too small sample. We very much like having this insurance that, if we produce a gross mistake, it might be visually visible.

While we try to keep the potential error as small as possible, determining the required sample size through simulation is less structured through a mathematical formula. An error can occur quickly. However, determining the sample size at a confidence interval of 95% is extremely important for the algorithm.

We use resampling to mimic the sampling variation. It approximates the sample mean sampling distribution, in the sense that both distributions (sample and population) will have a similar shape and similar spread. In particular, we study the typical "error" of our estimates, which can be quantified as a standard error of the difference in confidence levels. The later is an idea I developed during our research, and we are not sure how reliable it is.

When we use simulation with bootstrap to determine the sample required, we approach bootstrapping slightly differently than in the traditional application. For example, in general, bootstrap distributions are constructed by taking multiple resamples from a single sample. The emphasis is here on a "single sample." In fact, by determining the sample required, we go the other way. In a standard bootstrap, we take multiple resamples from a sample to infer to the population.

In contrast, here we go the "other way." We sample deliberately from the population and want to find the smallest possible sample that resembles the population at a given confidence interval. We are genuinely surprised by the lack of research and literature on the subject. We have read every book on bootstrap techniques and have never found a detailed explanation of how to do bootstrap in "reverse." Based on research forums, finding the sample required for a nonparametric distribution is a common problem.

However, we have not found clear instructions. We assume that the more general problem is to infer from a small sample to the population – and not the reverse as we research here. Luckily, we still feel confident tackling such a challenging problem because, in the end, once we determined a sample size based on confidence intervals, we can cross-reference the results with multiple samples based on Mean Average Deviation (MAD) to determine conformity. In other words, even if it takes several steps to simulate to determine the sample required, in the end, we can verify the results by testing the mean average deviation levels for conformity.

The worst that can happen is that we do not get the lowest possible sample size. However, what should not happen is that we get the given confidence level wrong. The reason for this is again that we can verify our simulation via mean average deviation.

The confidence intervals 2.50% and 97.50%, with the corresponding sample size, based on 1,000 simulations (reps = 1000):

Sample size	2.5%	97.50%
size = 10,000	10,442	11,259
size = 5000	10,448	11,248
size = 4000	10,438	11,242
size = 3000	10,443	11,278

Sample size	2.5%	97.50%
size = 2000	10,418	11,229
size = 1000	10,470	11,239
size = 100	10,430	11,234
size = 10	10,437	11,285

Figure 13: The confidence intervals 2.50% and 97.50%, with the corresponding sample size based, on 1,000 simulations (reps = 1000). Source: Franco Arda (2020).

Of course, the bootstrap does not compensate for a small sample size. It does not create new data, nor does it fill in holes in an existing data set. It merely informs us about how plots of additional samples would behave when drawn from a population (i.e., Vendor_1).

However, the power of bootstrap works now against us. With only a sample size of 10 and 1000 simulations (reps), our lower confidence interval of 2.50% is already significantly close to the population level.

Unfortunately, that feels wrong, and the problem is the high number of simulations set in 1000. Before we test a smaller number of simulations, let us verify that the results are incorrect in the MAD level.

The simulation-based confidence intervals were unsuccessful since the confidence intervals via a bootstrap do not correctly capture the sample size. Based on the bootstrap simulation, n = 1000 should be close to the correct sample size of MAD 0.002.

Sample size	Confidence interval	Mean Average Deviation (MAD)
n = 10	95%	0.1744
n = 100	95%	0.0083
n = 1,000	95%	0.0028

Figure 14: MAD results based on confidence intervals. Source: Franco Arda (2020).

Using Mean Average Deviation (MAD) to determine the sample size.

As seen above, determining the required sample size based on the distribution was unsuccessful due to the error rate. We calculated the percentile method computing the 2.5% and 97.5% percentiles.

Unfortunately, by cross-referencing the percentiles' MAD levels, the results were unsatisfactory (see Figure 12 above). In other words, while according to the percentiles, we were close to the correct sample size at n = 1,000, based on MAD levels, we were not.

The goal was to generalize the data of a sample to the population, or inductively infer. Intellectually, this approach via a bootstrap made sense to us, but the results do not confirm our intuition. We now test the traditional Monte Carlo simulation based on "trial and error," where we let the computer draw many samples from the population to test the hypothesis. Nevertheless, rather than simulating for the percentiles, we simulate for the MAD levels. With hindsight and the experience in sample size determination (1), this step makes more sense in determining the sample size required for the Benford distribution.

Essentially, this is a classic Monte Carlo simulation where we have the following formula in R code:

```
replicate(k, sample(invoices from Vendor_1, replace = FALSE, size = n))
```

replicate(): replicate applies a function over a vector.
k: refers to the number of simulations (I use 1,000).
sample(): this function takes n random samples from our population (i.e., Vendor_1).
n: refers to the number of samples (this is our determined sample at the end).
replace: refers to whether we draw from the invoice samples with or without replacement.

As we take samples from the population, we use them without replacement.

```
# Number of samples:
n <- 1000
# Number of simulations:
k <- 1000
```

The following R code calculates the number of simulations with a MAD level of > 0.002 (i.e., non conformity:

```
cutoff <- 0.0022
number_nonconform <- sum(MAD_events_A >= cutoff)
number_nonconform
# 829
```

Our very first simulations, with an invoice sample of 1,000, give us 829 non-conform datasets.

```
# Confidence interval level
1 - (number_nonconform / n)
```

0.171

Expressed as a confidence interval, this gives us 17.10%. The confidence interval is way too low, and we are looking at a range of 90% - 99%. However, this approach via sampling the MAD levels looks promising as we get more stable results with this approach.

An additional benefit that sees visually at least, how stable the samples are. The distribution is slightly normally distributed, and the Central Limit Theorem (CLT) is starting to take shape. In order to quantify the sample distribution, we will use another statistical test.

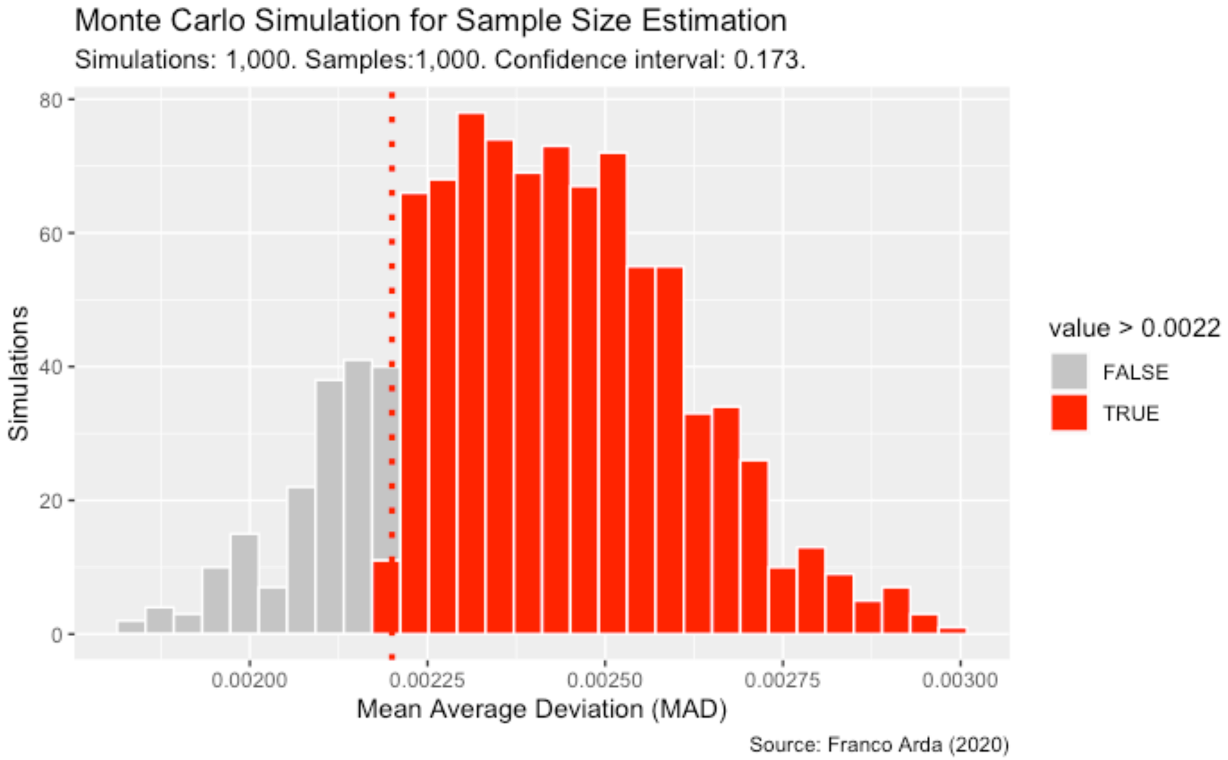


Figure 15: Monte Carlo Simulation for Sample Size Estimation with 1,000 simulations and 1,000 samples. Source: Franco Arda (2020).

Statistical attributes of Benford conform and nonconform datasets.

Four thousand randomly selected samples were taken from a conform dataset (Vendor_1) and a non-conform (Vendor_12) dataset to extract the statistical summaries. The main requirement to use the Monte Carlo method for simulation is that it must be possible to describe the underlying in terms of the probability density function. In this study, a bootstrap research methodology exploring the difference was used to take samples at random.

For the study, the samples were taken with replacement. At the core of a bootstrap, simulation is a random number generation. As in all our studies, the seed was set at 12. The seed refers to pseudo-random numbers corresponding to a deterministic sequence that passes randomness (Martino, Luengo, Miguez, 2018). The code snippet for the statistical summary:

```
# Number of samples:
n <- 4000
# Number of simulations:
k <- 1000

MAD_MC_simulation <- replicate(k, sample(benford_vendor_1$Vendor_1, replace = TRUE,
size = n))
MAD_MC_results_1 = apply(MAD_MC_simulation, 2, benford)
MAD_MC_results_2 = mapply(MAD, MAD_MC_results_1)
```

Replicate runs the same chunk of code 1,000 times, each time getting a sample MAD. The sample takes 4,000 random invoice samples from the population (10,000 invoices). We store the 1,000 MAD samples as a vector, and ultimately visually it in a histogram.

The analysis results consist of some of the main statistical measures, such as minimum, maximum, mean, median, and standard deviation. It can be inferred from the figure X below that the Mean Average Deviation (MAD) variation between the Benford conform dataset (Vendor_1) and the non-conform dataset (Vendor_12) deviate strongly. There was a significant difference in min and max values across both datasets.

Statistical summary	Benford Conform Vendor_1	Benford Non-conform Vendor_12
Min	0.00097	0.002349
Max	0.00161	0.003279
Mean	0.00126	0.002855
Median	0.00126	0.002854
Standard Deviation	0.00010	0.001450

Figure 16: Statistical summary for Mean Average Deviation (MAD) for a conform (Vendor_1) vs. a non-conform (Vendor_12) dataset. Bootstrap resampling with replacement. Source: Franco Arda (2020).

From a Benford distribution perspective, the most important number is the maximum (max) with a value of 0.003279 for the non-conform dataset. As we have established earlier, the threshold for nonconformity is 0.0022. In other words, statistical significance is accepted at or above 0.0022 MAD level for nonconformity.

From a statistical perspective, the difference's significance is best seen in the standard deviation, which is over ten times higher for the non-conform dataset.

In other words, there seems to be a strong correlation between the standard deviation and nonconformity. In fact, the strong correlation between all statistical calculations and nonconformity is easily visible. The figures suggest that regardless of the statistical summary, a higher level tends to lead to nonconformity. These findings suggest that we could use any statistical summary to determine conformity.

However, as we have already set a threshold for conformity via the MAD level, the data only provide strong evidence in favor of sticking to the MAD level for determining conformity. In other words, the results yield no alternative to the MAD level.

Therefore, we can conclude with high confidence that our findings reveal a high success rate for the MAD level without any other alternative. Analyzing the summary statistic provides conclusive evidence that we continue using MAD for the conformity test.

A statistical approach for defining the required number of simulations.

Some researchers (Hesterberg and Chihara, 2019) argue that at least 10,000 simulations are required if accuracy matters.

In general, this statement might be sufficient, but in our case, it is not. With fraud, we need the lowest possible sample. However, if we increase the number of simulations, our confidence intervals are being reduced. Therefore, we need to quantify statistically what number of simulations is sufficient. Again, the lower, the better.

Form the figure above, and we can see that for our simulated MAD levels, the Central Limit Theorem (CLT) seems to hold. That is fortunate in our case with invented numbers in general and invoice fraud, as we can use a traditional normality test.

Before we continue, let us illustrate the effect of simulations on the confidence interval. The following figure shows the inverse relationship between the number of simulations and the confidence interval. The number of invoices samples is fixed at 1,500, and the MAD level is fixed at 0.0022. By fixing those parameters, we can see the inverse relationship between the number of simulations and confidence intervals:

Number of simulations	Confidence Interval
k = 1,000	98.66%
k = 2,000	96.46%
k = 3,000	93.66%
k = 4,000	91.40%
k = 5,000	89.46%

Figure 17: The inverse relationship between the number of simulations and the confidence interval. The number of invoice samples is fixed at 1,500. Source: Franco Arda (2020).

For each given confidence interval, we can use the Shapiro-Wilk test for normality. The null hypothesis is always that the sample came from a normally distributed population. In our research, normally distributed population refers to the MAD levels and not the Benford distribution, which is not normally distributed. We set the p-value for the Shapiro-Wilk test at 0.01.

The reason for this is that we want to very confident that the number of simulations is statistically significant. For a business application of accountants or fraud examiners, the main parameter should be the confidence interval, while all other parameters are set to a high confidence at a p-value of 0.01.

Similar to our last test, we keep only a single variable (number of simulations) and fix the number of samples (i.e., 1,500 invoice samples) and the MAD level (i.e., 0.0022). With those parameters fixed, we can see that the Shapiro-Wilk p-value reduces our targeted p-value with 3,000 – 4,000 simulations.

Shapiro-Wilk test	Number of simulations
p-value = 0.1004	2,000
p-value = 0.02534	3,000
p-value = 0.00054	4,000

Figure 18: Shapiro-Wilk test p-value in relationship to the number of simulations. Source: Franco Arda (2020).

Based on our target confidence interval of 95%, we must recalculate the Shapiro-Wilk test based on a p-value of 0.01 for our optimal number of simulations. With this approach now, we can quantify the number of simulations required to determine the sample, which is extremely exciting.

We are not sure if we are just fortunate that the CLT holds in our example. If it did not, we would need another test, such as the Kolmogorov-Smirnov test. This test is nonparametric, while the Shapiro-Wilk test is parametric. We are unsure if we would get those incredible results if Benford's MAD would not hold under CLT.

However, it is sure that Benford's MAD distribution holds to CLT, and we, therefore, can generalize the application to any Benford test. In other words, the very same Benford algorithm can be used with invoice fraud, reimbursement fraud, Ponzi schemes, or tax fraud, to name a few.

When we set out for this research paper, we did not expect to generalize our research hypotheses to any Benford test.

From a statistical standpoint, the Monte Carlo simulation methodology's goal is to determine the characteristics of the probability distribution associated with the sample invoice.

Benford’s distribution and the Central Limit Theorem (CLT)

According to most statistic textbooks, the usual rule of thumb is that the CLT is reasonably accurate if n > 30 unless the data are quite skewed. This rule is probably wishful thinking, dating back to the pre-computer age (Chihara and Hesterberg 2019). In general, we can obtain better approximations and visualize them using simulation-based methods such as simulating the MAD levels a thousand times or more.

In our view, the best technical definition (Tilde, 2016) of the central limit theorem is: “When the distribution of any population has a finite variance, then the distribution of the arithmetic mean of random sampis is approximately normal, if the sample size is sufficiently large.”

Given the central limit theorem, the distribution of the invoice samples should form a normal distribution, the mean of which can be taken as the approximated quantity and the variance used to provide a confidence interval for the Mean Average Deviation (MAD). Goodfellow, Bengio, and Courville (Goodfellow, Bengio, and Courville, 2016) have advanced the hypothesis that the central limit theorem allows us to estimate the estimate's confidence intervals, using the cumulative distribution of the normal density.

This chapter investigated the central limit theorems with Benford's Distribution for experimental testing of up to 10,000 invoices. The purpose was to investigate the Central Limit Theorem's application for a large dataset created followed Bedford's distribution. Then, we obtained a statement for Bedford's Distribution, CLT, and LLN simultaneously. Subsequently, methodological testing to assess the accuracy and goodness of fit test were described.

Later findings are presented in histograms showing n-1, n=10, and n=100 for k = 4000. The results confirmed that we do not get a normal distribution with only one sample (n=1), but a slightly skewed distribution due to high mean deviation. However, if we take ten or even 100 random samples, we see that the distribution becomes normal. The is the Central Limit Theorem (CLT), wherewith larger and larger samples (law of large numbers), we get a normal distribution even though our population is not normally distributed.

The theorem helps in understanding the relationship between the large dataset and a normal distribution of samples.

Drawing inferences from Benford's sample data, we can see that where individual datasets may fail to meet Bedford's law criteria, integrating the different data sets may result in a new behavior series, which is characteristically nearer to the Bedford's law.

On the other side, the Central Limit Theorem (CLT) importance cannot be undermined. It contributes vitally to gaining insights about the ubiquity of Benford's law. Datasets design in line with the Benford’s Law, it is easier for the researcher to focus upon the situations exhibiting values not concentrated in a small interval (Peng, 2019). Therefore, CLT allows the researcher to disregard the sample's size chosen for the dataset not distributed normally. CLT is based upon probability theory, emphasizing the appropriately normalized sum of independent random variables even if the individual variables are not normally distributed.

In this regard, CLT describes that the values of the variables can differ in a population in terms of different distributions such as normal, left-skewed, right-skewed, and uniform, among others. Hence, the theorem explains that despite any type of population distribution, the mean's sampling distribution will always be approximate to the normal distribution.

CLT also highlighted that with increasing the sample size of the distribution, there is always a decrease in the sampling error. Hence, the theorem works on the three components, including the population's successes, increased sample size, and population distribution. Under this approach, the sample size is crucial for determining the sampling distribution (Peng, 2019).

The application's scope is apparent in almost all disciplines, including physics, chemistry, astronomic, economics, engineering, and other disciplines requiring mathematical and statistical expressions. The result should manifest the quantitative skewness associated with the forward band phenomena in the natural sciences and real-life data. CLT allows us to understand that there is a high order of magnitude or enough variability in almost all datasets; therefore, it is impossible to obtain the data set, which is an exception to the given rule.

Hence, following the mathematical formula of multiplicative CLT, one can get two actual results. The first one is related to a dramatic increase in skewness, which is also in line with the Bedford behavior, and the second one is related to the increasing order of magnitude, which is also an essential criterion for Bedford behavior.

The law of large numbers (LLN) is since the probability of an event E depends on the number of occurrences of that event. The probability is defined as the longer and frequency of the event in many trials, m denotes the frequentists definition of probability of an event.

In the following figure, we randomly select n invoices from the population of 10,000 (invoices not normally distributed). In the first figure, we take 1 sample (n = 1), in the second 10 samples (n = 10), and in the last 100 samples (n = 100).

With only one sample (n=1), we do not get a normal distribution, but a slightly skewed distribution. However, if we take ten or even 100 random samples, we see that the distribution becomes normal. The is the Central Limit Theorem (CLT), wherewith larger and larger samples (law of large numbers), we get a normal distribution even though our population is not normally distributed.

In Figure 1 below, a single sample was withdrawn from the population of 4,000 invoices. Based on the results gathered from more than 200 simulations of n=1, the Mean Average Deviation (MAD) showed a slightly right-skewed distribution and not normal distribution. It showed that by using the single-digit number in a large dataset of the invoice, the probability of normal distribution decreases due to large MAD values with the increase in some simulations ranging between 0.0205 to 0.0220.

It can further be depicted that the data points in the simulation reflect slightly positively skewed histogram due to the deviation of the data point values from their MAD values. It shows that with the increase in the number of simulations, the distance between the data points and the MAD also increases, due to high variation and low probability of the same number to appear again in the simulations.

The same discussion can be justified from the distance between the two data points: better-given diagrams such as 0.0205, 0.0210, and 0.015, and 0.0220. The CLT further justifies skewness in the dataset by explaining the reason behind the symmetric normal distributions in the data side as the eventual distribution develops after several editions of random variables. It shows that small data samples are far from eventuality as the histogram does not show an excellent cover around the center, falling off almost unevenly on one side.

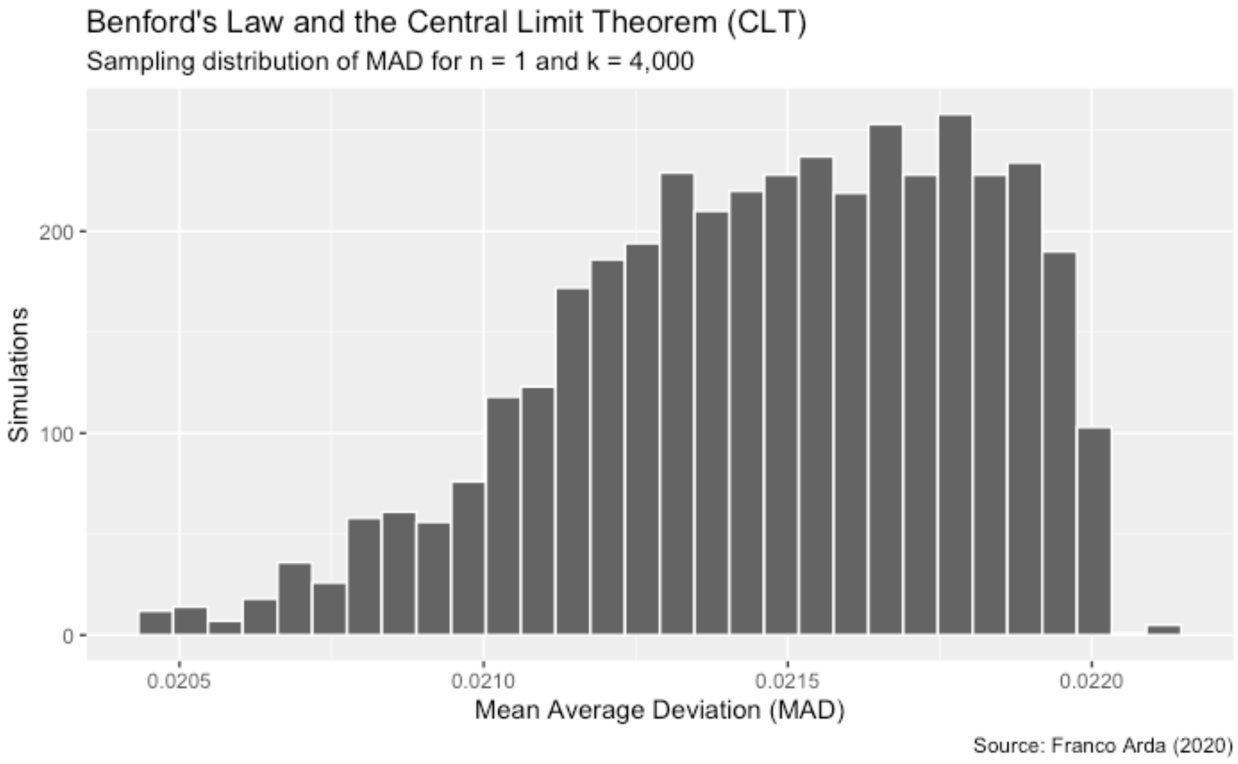


Figure 19: Benford's law and the Central Limit Theorem with n = 1. Source: Franco Arda (2020).

Moreover, in Figure 2 below, a sample of 10 numbers was withdrawn from the population of 4,000 invoices. Based on the results gathered from more than 200 simulations of n=10, the Mean Average Deviation (MAD) depicted normal distribution. The distribution confirms that by using the sample of 10 number in a large dataset of invoices, the probability of normal distribution increases due to large MAD values with the increase in some simulations, which later returned to the low deviation values after reaching a peak number of simulations ranging between 0.017 to 0.020.

It can be analyzed that the data points in the simulation reflect slightly distribution returning towards normal due to the reduced deviation of the data point values from their MAD values. It shows that with the increase in the number of simulations, the distance between the data points and the MAD decreases when the datasets are comprised of a large set of numbers.

This behavior can be explained via the fact that due to reduced variation and high probability of the same number to appear again in the simulations. The same discussion can be justified from the distance between the two data points: better-given diagrams such as 0.017, 0.018, 0.019, 0.020, and 0.021. This figure is a strong hint that the data has started even distribution on both edges to resemble the normal distribution of MAD.

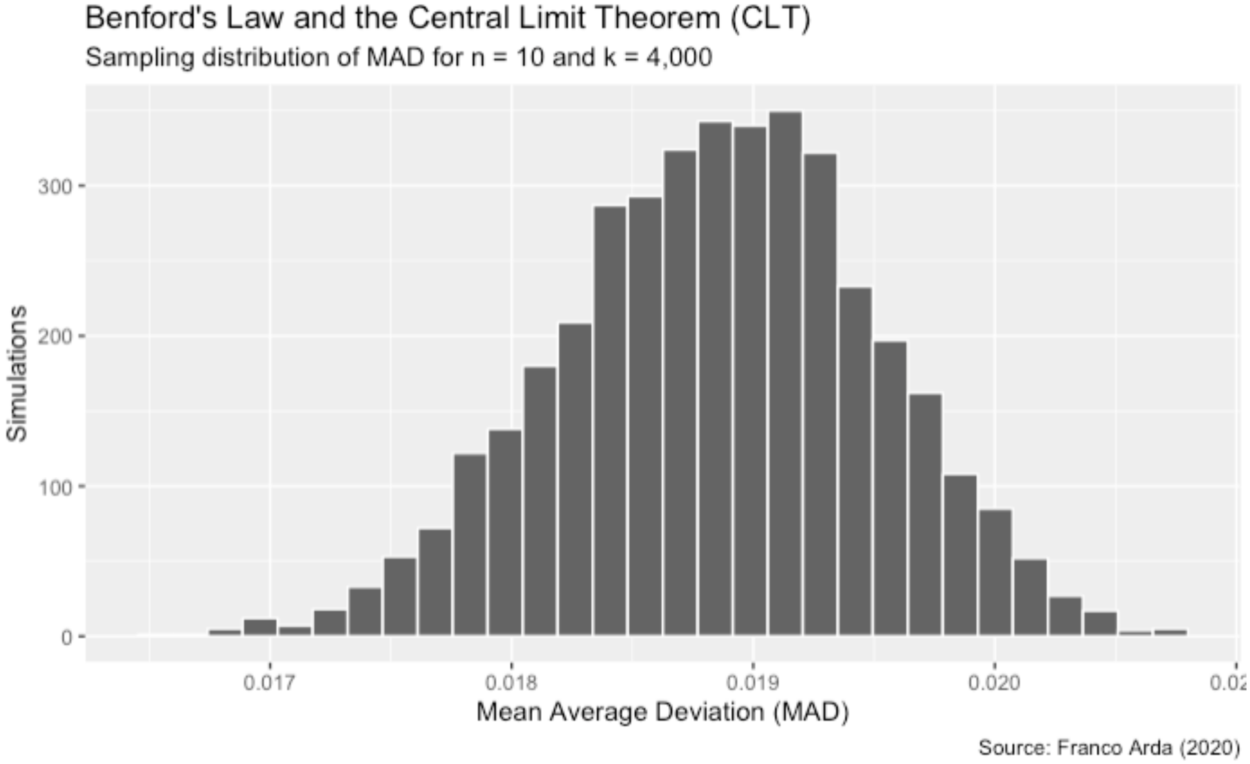


Figure 20: Benford's law and the Central Limit Theorem with n = 10. Source: Franco Arda (2020).

Like the results gathered from the sample (n = 10), in Figure 3 below, a sample of 100 numbers was withdrawn from the population of 4,000 invoices. Based on the results gathered from more than 300 simulations of n=100, the Mean Average Deviation (MAD) depicted normal distribution. The distribution confirms that by using the sample of 100 numbers in a large dataset of invoices, the probability of normal distribution increases due to large MAD values with the increase in several simulations, which later returned to the low deviation values after reaching a peak number of simulations.

This behavior can be explained by the CLT assumption that with large numbers in a dataset, probability of deviation among the sample invoice values decreases, ranging from 0.006 to 0.010. Similarly, the distribution explains that the simulation's data points reflect a perfectly normal histogram due to the very minimal deviation of the data point values from their MAD values.

It shows that with the increase in the number of simulations, the distance between the data points and the MAD decreases and comes to a negligible point. This is because of the reduced variation and increased probability of the same number to appear again in the simulations. The same discussion can be justified from a distance between the two data points: better-given diagrams such as 0.006, 0.007, 0.008, 0.009, and 0.010.

Through the application of CLT in these figures, we can understand how with the increase in the sample size, the histogram inclined towards showing the eventuality of data explained from the nice curves around the center that further falls off almost evenly in both the edges.

It is clear from the quantitative configuration of the data that CLT had understood the criteria for Bedford behavior as with the large sample size, there was a lack of an increase in skewness and added focus on increased concentration around the center. Likewise, there was lacking in increasing order of magnitude beyond the existing maximum order of magnitude (Kossovsky, 2019).

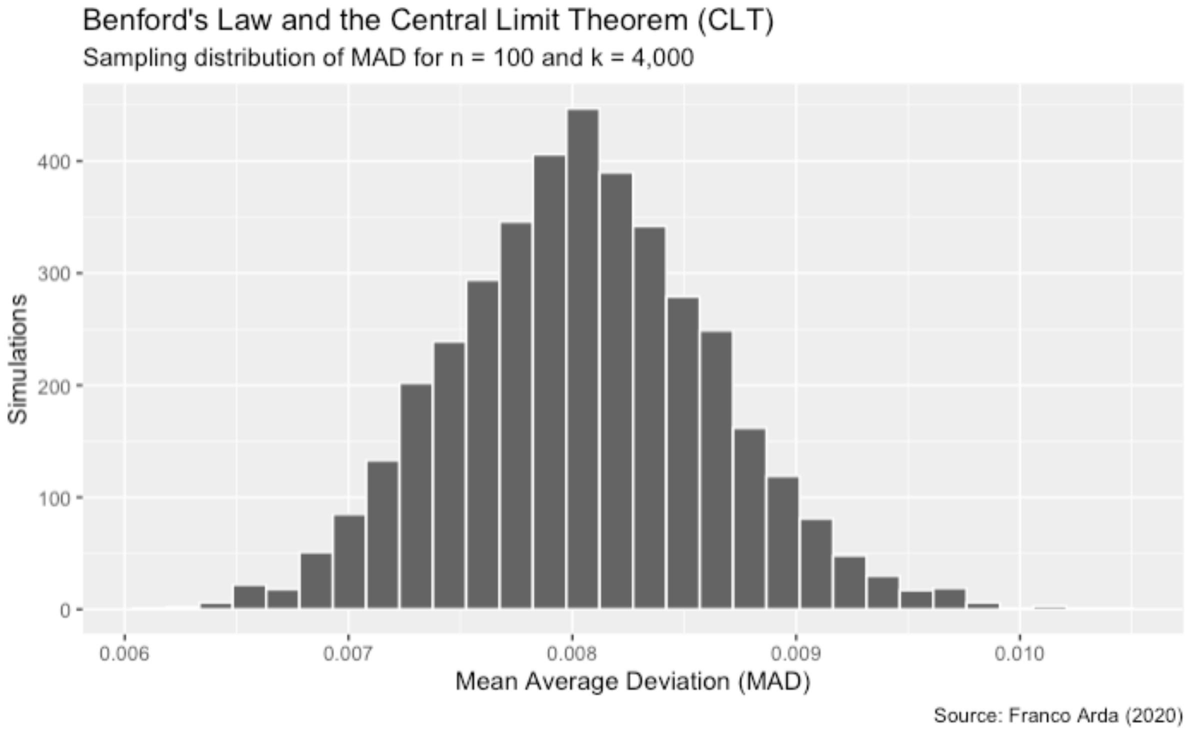


Figure 21: Benford's law and the Central Limit Theorem with n = 100. Source: Franco Arda (2020).

These figures must successfully help in explaining the application of Benford's Law and CLT to a large dataset of invoices in this research.

However, explaining the spread can be examined in detail by applying the theoretical 's accusations and mathematical statements of the two approaches in the next section of the report.

In this section, the analysis of the spread of the distribution of invoices dataset is explained to answer several questions using the Benford's Law and Central Limit Theorem together. The as gathered in all the three figures have substantially confirmed the effectiveness of using CLT on the dataset satisfying the Benford's Laws conditions.

With an increase in sample size, there is always a chance to move towards normal distribution despite the individual datasets and not exhibiting the normal distribution. This can be further explained by answering numerous questions.

Firstly, the researcher dealt with the question to identify why the datasets behave similarly and to identify why the datasets behave differently. The answer to the similarity hypothesis can be well explained with the help of Benford's law as similarities in the leading digit or first digit of the numbers in dataset shapes there are similarities between them. In looking upon the histograms' detailed analysis above, figures 3 and 4 show similar behaviors as these datasets with n=10 and n=100 were spread out on a logarithmic plot over several orders of magnitude.

The normal distribution is one of the significant components of the probability theory, known as Gaussian distribution, which helps understand the real-valued random variables. The significance of normal distribution in statistics cannot be undermined as it helps in the representation of real value trend variables within the natural and social sciences. The importance of normal distribution is associated with the central limit theorem.

Normal distribution states that physical quantities that are expected to be the sum of many dependent processes represent weirdly normal distributions. The Gaussian distributions' unique properties, such as its reflection from a distance combination of a fixed collection of normal deviates. The spread analysis of the invoices in this research is also visible from the mathematical support it has provided in understanding that under some conditions, the average of many samples of a random variable with finite mean and variance is itself a random variable.

Another property of such distribution is based upon the results and methods like propagation of uncertainty and least square parameter fittings that have allowed detailed analysis of the relevant variables normally distributed. Hence, a normal distribution with bell-curve is always helpful in understanding the situation.

The investigation results have confirmed that with only one sample (n=1), we do not get a normal distribution, but a slightly skewed distribution due to high mean deviation. However, if we take ten or even 100 random samples, we see that the distribution becomes normal. This is the Central Limit Theorem (CLT), wherewith larger and larger samples (law of large numbers), we get a normal distribution even though our population is not normally distributed.

The study has provided a detailed analysis of the theory behind normal distribution associated with many datasets, where each of the datasets can be positively or negatively skewed. It is essential to understand that the integration of CLT and Benford's law is the real essence of the current research. Mathematical statements of both the theorem and law explained significant randomization processes associated with the real-life data.

In the final note, the discussion in this chapter has explained CLT's great effectiveness for randomly sampling invoices from the population. Those findings can be used in the next chapter to quantify the normal distribution using a statistical method called the Shapiro-Wilk normalization test.

Monte Carlo sample size determination at a 90% confidence interval.

The first research hypothesis used quantitative techniques to determine the sample required from the population of invoices conforming to the Benford distribution.

Before we run the simulations and analyze the data, it would be wise to define the statistical parameters. Ott and Longnecker (Ott and Longnecker, 2016) state the confidence interval's general formula (1 – alpha) where alpha is between 0 – 1.

In other words, alpha is the significance level, which corresponds to the probability that we will make the mistake of rejecting the null hypothesis (i.e., a type II error). The p-value measures the probability of getting a more extreme value than the one from the simulation.

In this section, we determine the sample size, given a confidence interval of 90%. As in the Benford distribution, the Mean Average Deviation (MAD), the level for conformity, is calculated on an absolute basis; we are dealing with a two-tailed distribution. It can be inferred from the table that the two-tailed testing is represented by the lower bound and upper bound confidence interval.

Whether a statistical significance is achieved based on a one-tail or two-tail testing should not concern the practitioner. Therefore, the tables present the confidence interval as a total number, e.g., 90%, not 5%, and 95%, which might be confusing in practical terms.

The idea is based Value at Risk (Kenton, 2019), which is often used in finance, where only a single number represents the risk, even though it is a two-tailed risk. This approach's reasoning is that similarly to Value at Risk, a single number should reflect the risk.

Unlike with Value at Risk, where the number reflects a potential loss at a given confidence interval, the Benford algorithm's confidence interval reflects the classification risk. A lower confidence interval at 90% increases the risk of a type I error (false alarm) but reduces the risk of committing a type II error (missing a potentially fraudulent transaction).

We recommend that practitioners take internal risk policies into account. Some companies might be more comfortable with type I errors, while others are not.

Judging which confidence interval works best for a given company is an extremely individual choice. For example, if a company has the human resources to deal with a higher number of type I errors (false positives), the company might opt for a lower confidence interval such as 90%.

On the other hand, if a company deals with a vast number of invoices, they might want to keep the false positive rate (type I error) as low as possible. Many large companies deal with tens of thousands of invoices per year, where a higher confidence interval might be more appropriate.

Sample size	Number of simulations	Shapiro-Wilk normality test p-value	Mean Average Deviation	Lower bound confidence level	Upper bound confidence level	Confidence Interval
10	1,000	0.048	0.0022	-	-	0%
100	1,000	0.001	0.0022	-	-	0%
1,000	1,000	0.321	0.0022	10801	10900	17.1%
1,500	1,000	0.556	0.0022	10367	11360	98.6%
1,500	2,000	0.100	0.0022	10416	11285	96.4%
1,500	3,000	0.025	0.0022	10468	11241	93.6%
1,400	4,000	0.019	0.0022	10597	11101	75.8%
1,600	4,000	0.021	0.0022	10375	11344	97.6%
1,500	4,000	0.001	0.0022	10496	11210	91.4%
1,450	4,000	0.016	0.0022	10541	11146	84.4%
1,480	4,500	0.055	0.0022	10526	11178	88.1%
1,480	4,000	0.031	0.0022	10578	11207	90.0%

Figure 22: Solving for the required sample size, at a 90% confidence interval (size = 1000), a Mean Average Deviation at 0.002, and with a Shapiro-Wilk p-value at 0.05. Franco Arda (2020).

In figure 21 below, we can visually see that with only 800 samples, we are starting to get a bell-shaped curve, which corresponds to the Central Limit Theorem (CLT). The large part in red shows that the confidence level is low at around 24%.

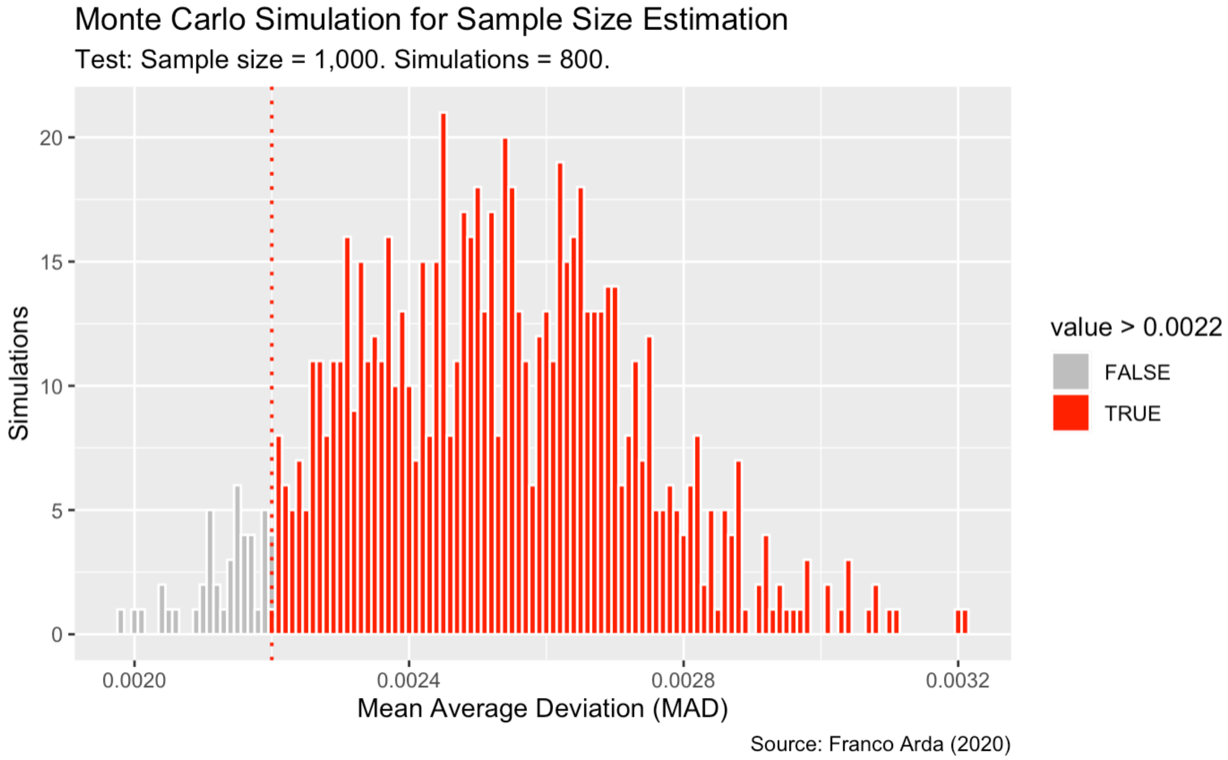


Figure 23: Test based on a Monte Carlo method, with 800 simulations and a sample size of 1,000. Source: Franco Arda (2020).

With 100 simulations more, as shown in the following image, the confidence level (histogram in gray) hardly improves. The lack of variable chance is an indication that we must increase the number of simulations considerably.

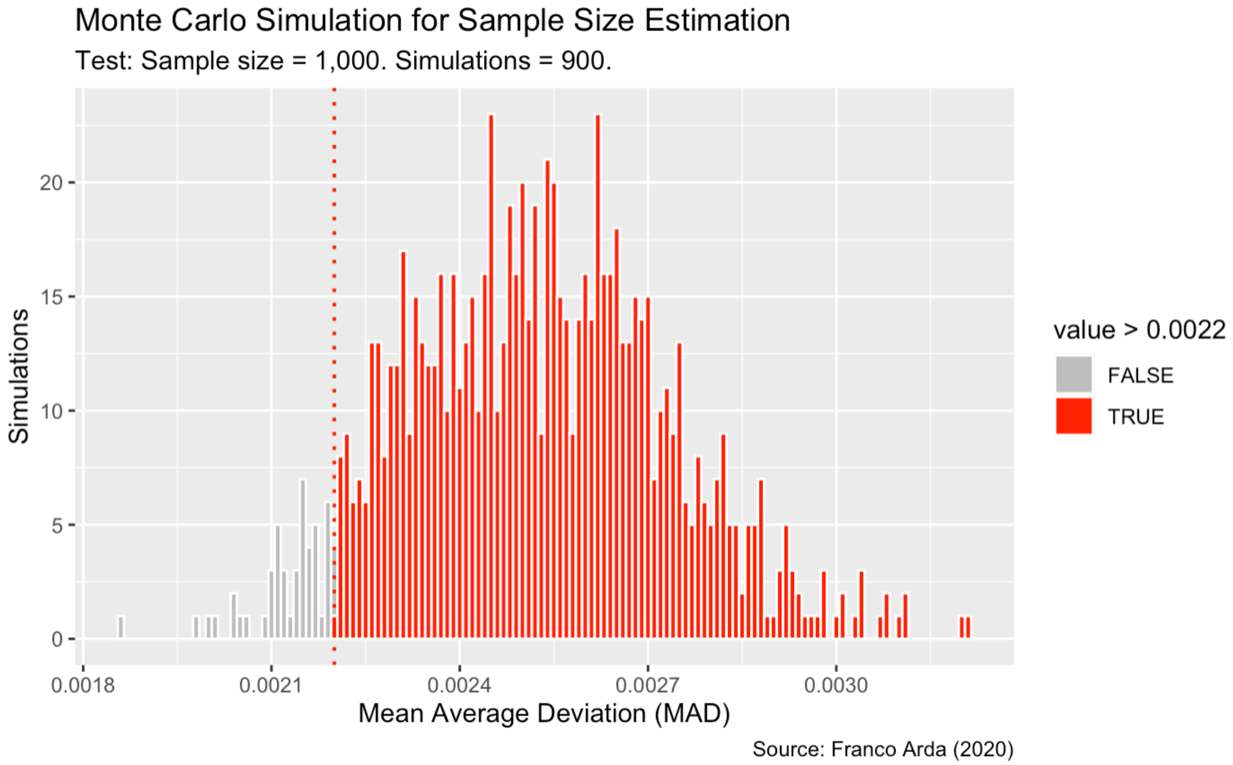


Figure 24: Test based on a Monte Carlo method, with 900 simulations and a sample size of 1,000. Source: Franco Arda (2020).

In figure 22, we increase the number of simulations from 800 to 900 and see a slight move to the right in the confidence interval. The move to the right is an indication that we are on the right track with the Monte Carlo simulation. With a higher number of simulations, we can already say that we are likely to observe a normal distribution based on the sample.

Given this tradeoff between a high number of false positives and many false negatives, it is no wonder that many researchers and practitioners try to strike a balance with a confidence interval of 95% or a p-value of 0.05. Additionally, this common choice has another advantage: it is intuitive. Given the confidence interval at 95% or a p-value at 0.05, we can expect one false positive out of twenty flagged transactions.

An accountant or fraud examiner might feel comfortable with this number, while with a 90% confidence interval, this would result in one out of ten (which might feel a lot), while with a 99% confidence interval with one in one hundred (might feel too few).

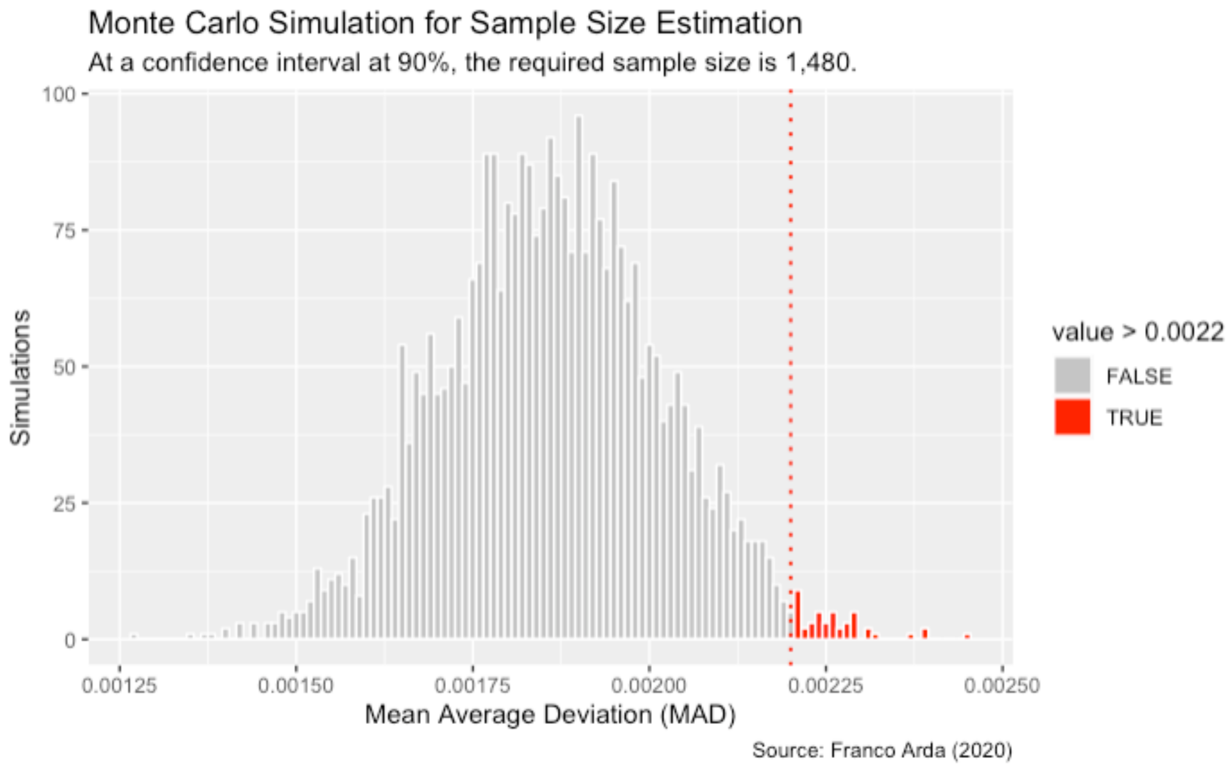


Figure 25: Based on a Monte Carlo method, with 4000 simulations, the required sample size assesses to 1480, at a 90% confidence interval. Source: Franco Arda (2020).

Mathematically, this makes sense—the lower the confidence interval, the lower the required sample. At a 90% confidence interval, we determined a sample size at 1,480. This result is considerably lower than the sample size required at a 99% confidence interval, assessed at 1,670 invoice samples.

As for each simulation, we started with a sample size of 10 and several simulations at 1,000. We soon realized that the statistically significant number of simulations is around 4,000. The statistical significance was determined by the Shapiro-Wilk normality test with a p-value of 0.05. While the Benford distribution is non-normal (highly skewed to the right), the samples based on Mean Average Deviation (MAD) are normally distributed with a large sample (e.g., over 1,000 samples).

This effect, of a normal distribution of the sample, can be attributed to the Central Limit Theorem (CLT). Initially, we did not expect the CLT to hold for the Benford distribution. With a normally distributed sample, we can expect a more robust algorithmic prediction.

In statistical learning, this effect has been highlighted by James, Witten, and Hastie (James, Witten, and Hastie, 2013), which means that we can expect tested accuracy to be similar to the accuracy for new data. In other words, the higher accuracy of the algorithm should also apply to new data in the real-world.

Figure X below shows the confidence interval at 90% based on bootstrapping with a sample size = 1,480, number of simulation = 4,000, and a Shapiro-Wilk normality test p-value = 0.031.

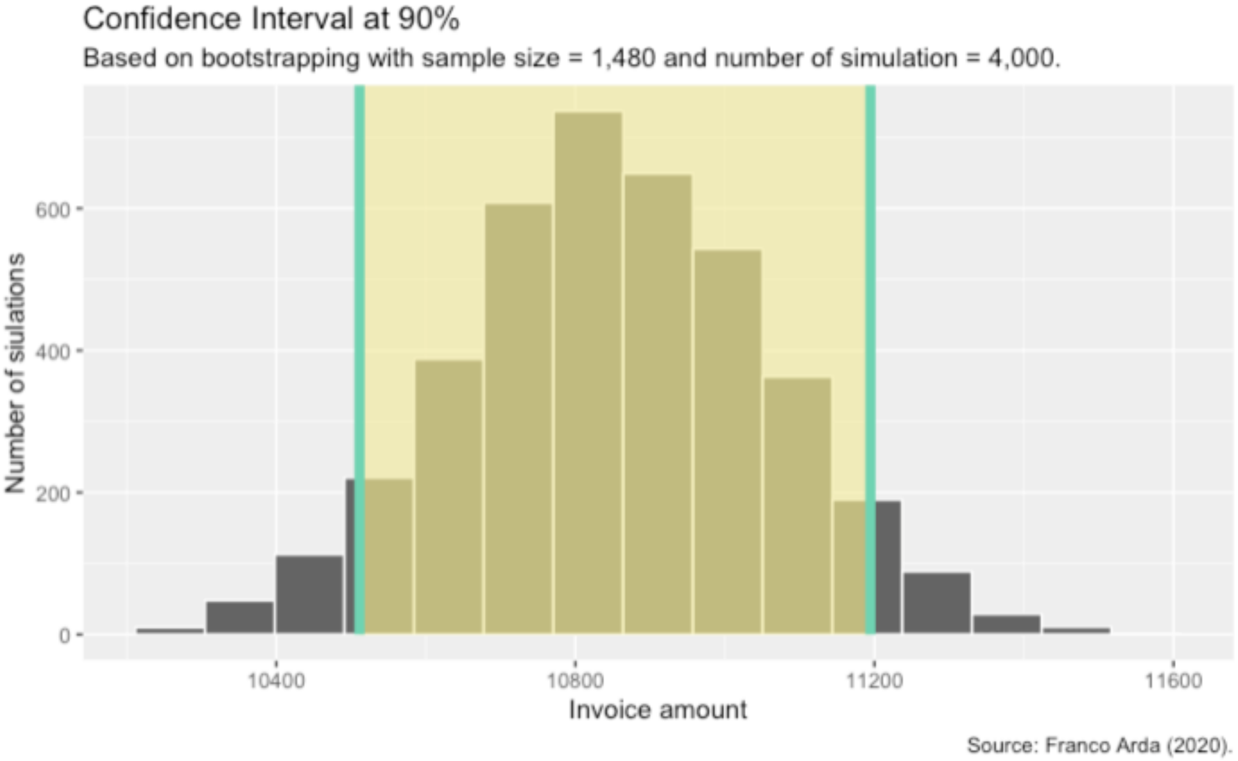


Figure 26: 90% confidence interval based on bootstrapping with a sample size = 1,480, number of simulations = 4,000, and a Shapiro-Wilk p-value = 0.031. Source: Franco Arda (2020).

We have realized why we could not assess the sample size directly from the confidence intervals during the research process. The most likely reason for this is that confidence intervals assume a normal distribution. As we have already established, the Benford distribution is non-normal and does not fall in any distribution category.

While the research design involved a non-parametric random sampling via bootstrapping, the confidence interval itself assumes a normal distribution. In hindsight, this makes sense. In general, the bootstrapping method is being used for inferring from a sample to the population.

Research designed to investigate with bootstrapping is, in general, used for large datasets. Statistical significance is probably determined better through inferring then deducting. Moreover, deducting is what we tried to attempt by determining the sample size via the confidence intervals.

In other words, with future research, we can recommend resampling for the sample size via a Monte Carlo simulation rather than via bootstrap and its confidence intervals.

Monte Carlo sample size determination at a 95% confidence interval.

It can be inferred from figure 24, that the determined sample size at a 95% confidence interval assesses to 1,532 invoice samples. Compared to the sample size required at a 90% confidence interval, we only need 52 additional invoice samples to get to a confidence interval of 95%. Intuitively, we would have expected a higher number to be statistically significant. The simulation results were considered significant at several simulations at 4,000. This result was no surprise and similar to statistical significance at 90%, ultimately at 99%.

The simulations resulted in some exciting findings. A cursory glance at figure X reveals that the Shapiro-Wilk normality test p-value was sometimes quite unstable, particularly with the sample size at 1,550. Based on similar simulations with p-values below 0.05, we suddenly got a p-value of 0.388 with 1,550 simulations.

The results yielded no statistically significant relationship between 1,525 – 1,535 simulations. We suspect the instability in the Shapiro-Wilk p-value stems from the fact that our tests are stochastic. In other words, small differences in the number of simulations, e.g., 1,525 or 1,530, resulted in close to zero p-values while one example deviated massively (e.g., 1,550 simulations).

We will get into the instability of the Shapiro-Wilk test shortly, but for calculating the confidence intervals, we see no problem. In general, the p-values were consistent and only returned once inconsistent results.

These figures suggest that regardless of one extreme p-value, we can be confident that the p-value outliers were, in fact, an outlier. Results obtained for 90% and 99% confidence intervals are consistent with our findings that we have a single outlier due to stochastic instability caused by the Monte Carlo simulation. In other words, the data from the 90% and 99% confidence intervals provide strong support that we saw here a single outlier in p-values.

Taken altogether, the simulation results for the 90%, 95%, and 99% confidence interval provide evidence that the simulated results are correct and stable.

Future studies might provide deeper insights into the behavior of the Shapiro-Wilk normality test p-values based on Monte Carlo simulation.

Sample size	Number of simulations	Shapiro-Wilk normality test p-value	Mean Average Deviation	Lower bound confidence level	Upper bound confidence level	Confidence Interval
10	1,000	0.048	0.0022	-	-	0%
100	1,000	0.001	0.0022	-	-	0%
1,000	1,000	0.321	0.0022	10801	10900	17.1%
1,700	3,000	0.044	0.0022	10328	11405	99.4%
1650	4000	0.065	0.0022	10330	11402	99.3%
1600	4,000	0.021	0.0022	10396	11333	97.6%
1,550	4,000	0.388	0.0022	10417	11284	96.3%
1,540	4,000	0.000	0.0022	10435	11277	95.4%
1,525	4,000	0.000	0.0022	10484	11230	92.4%
1,530	4,000	0.000	0.0022	10462	11253	94.2%
1,535	4,000	0.000	0.0022	10435	11277	95.4%
1,532	4,000	0.000	0.0022	10439	11269	95%

Figure 27: Solving for the required sample size, at a 95% confidence interval (size = 1000), a Mean Average Deviation at 0.002, and with a Shapiro-Wilk p-value at 0.05. Franco Arda (2020).

Figure 26 below shows us visually the 95% confidence intervals in a histogram. We undertook a large number of simulations in order to get confident and stable confidence intervals. All simulations were conducted using a Monte Carlo simulation with the code and random seed generator detailed earlier. All the random simulations via a Monte Carlo simulation were weighted against the target population (Vendor_1 conforming to the Benford distribution).

All simulations are weighted to represent the determined sample given a confidence interval of 90%, 95%, or 99%. As visually highlighted in red, the non-conformity level based on Benford's distribution was always set at the Mean Average Deviation (MAD) of 0.0022.

There is limited research investigating the sample size required for Benford's law. Moreover, no study has explicitly looked at a sample-sized combined with a confidence interval to date.

Figure 26 below shows a test with a sample size of 1,000 and a simulation of 1,000. We can visually see that the confidence level is low. The histogram's red part should reflect 10% of the distribution to get to a confidence level of 90%. We had to increase the sample size considerably from 1,000 to 1,700.

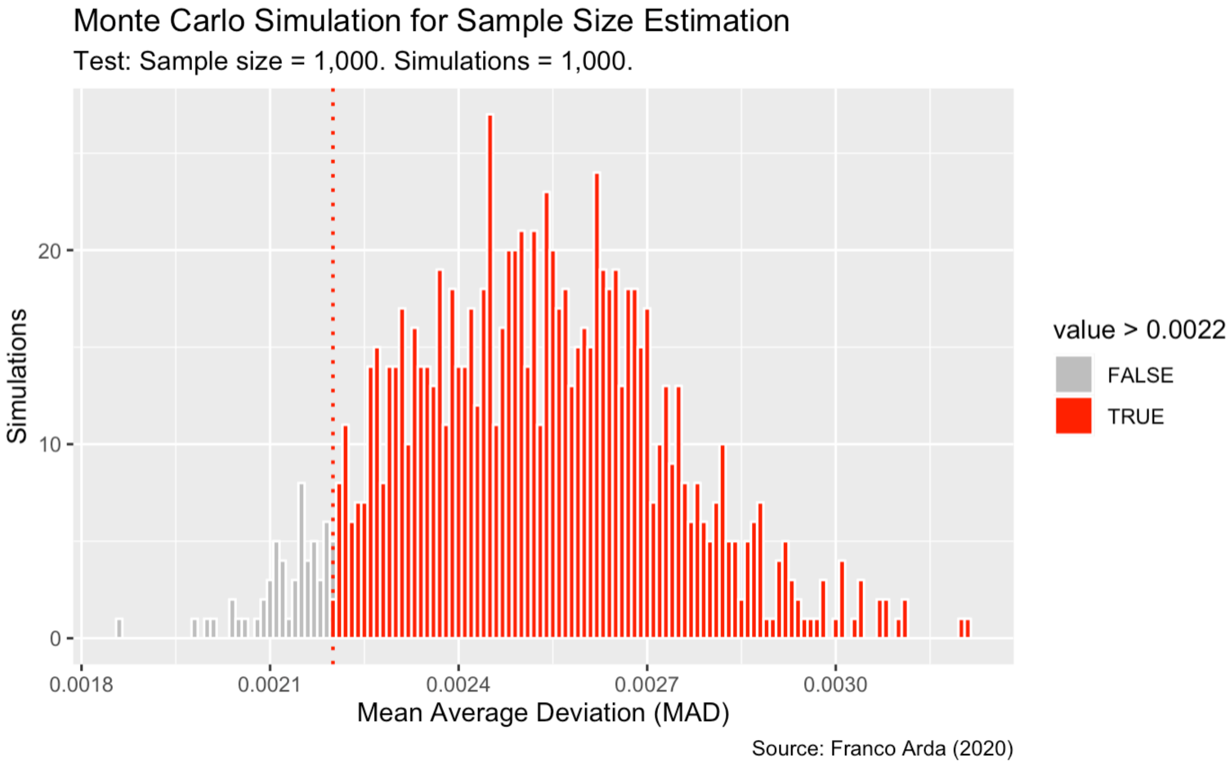


Figure 28: Test based on a Monte Carlo method, with 1,000 simulations and a sample size of 1,000. Source: Franco Arda (2020).

Figure 27 below shows a test with a sample size of 1,700 and a simulation of 3,000. With a confidence level at 99.4%, we could reduce the sample size and increase the simulation number.

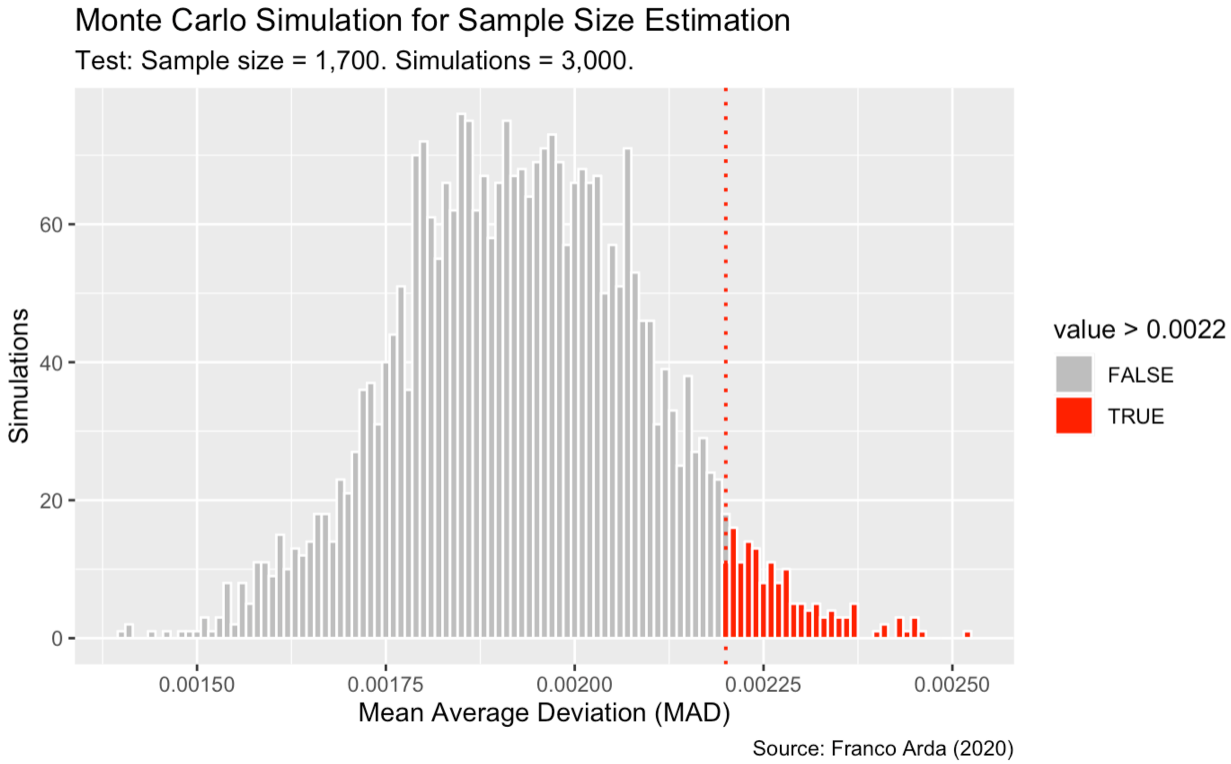


Figure 29: Test based on a Monte Carlo method, with 3000 simulations and a sample size of 1,700. Source: Franco Arda (2020).

This research allows researchers and practitioners to make a powerful statement: "At a 95% confidence interval, we require a sample size of 1,532 invoices."

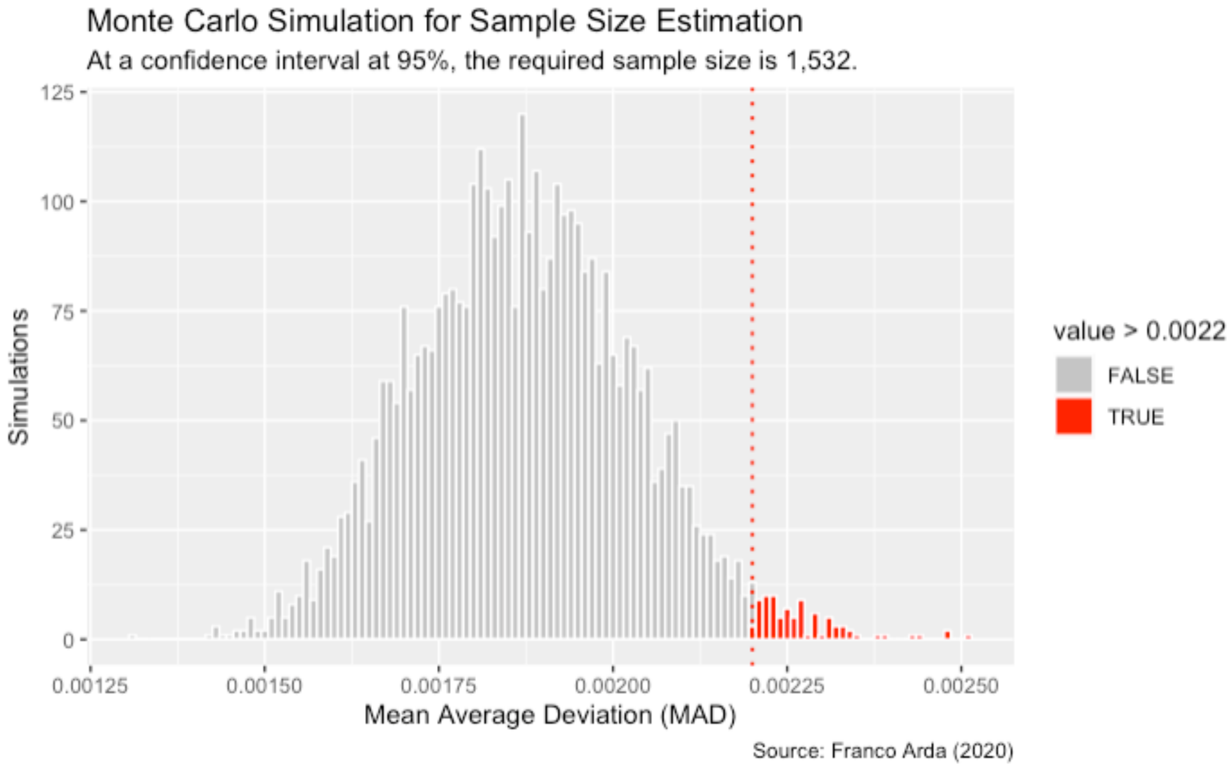


Figure 30: Based on a Monte Carlo method, with 4000 simulations, the required sample size assesses to 1532, at a 95% confidence interval. Source: Franco Arda (2020).

In the histogram, each bar on the x-axis refers to the number of observations of Mean Average Deviation (MAD) levels. Each outcome is the number of invoices from a particular vendor. Each outcome is mutually exclusive. In other words, an outcome cannot be both conform and nonconform.

The histogram of Benford's Mean Average Deviations (MAD) is called a multinomial trial.

Figure 29 below visualizes the bootstrapping results at a confidence interval at 95% with a sample size = 1,532, a number of simulations = 4,000, and a Shapiro-Wilk normality test p-value = 0.000.

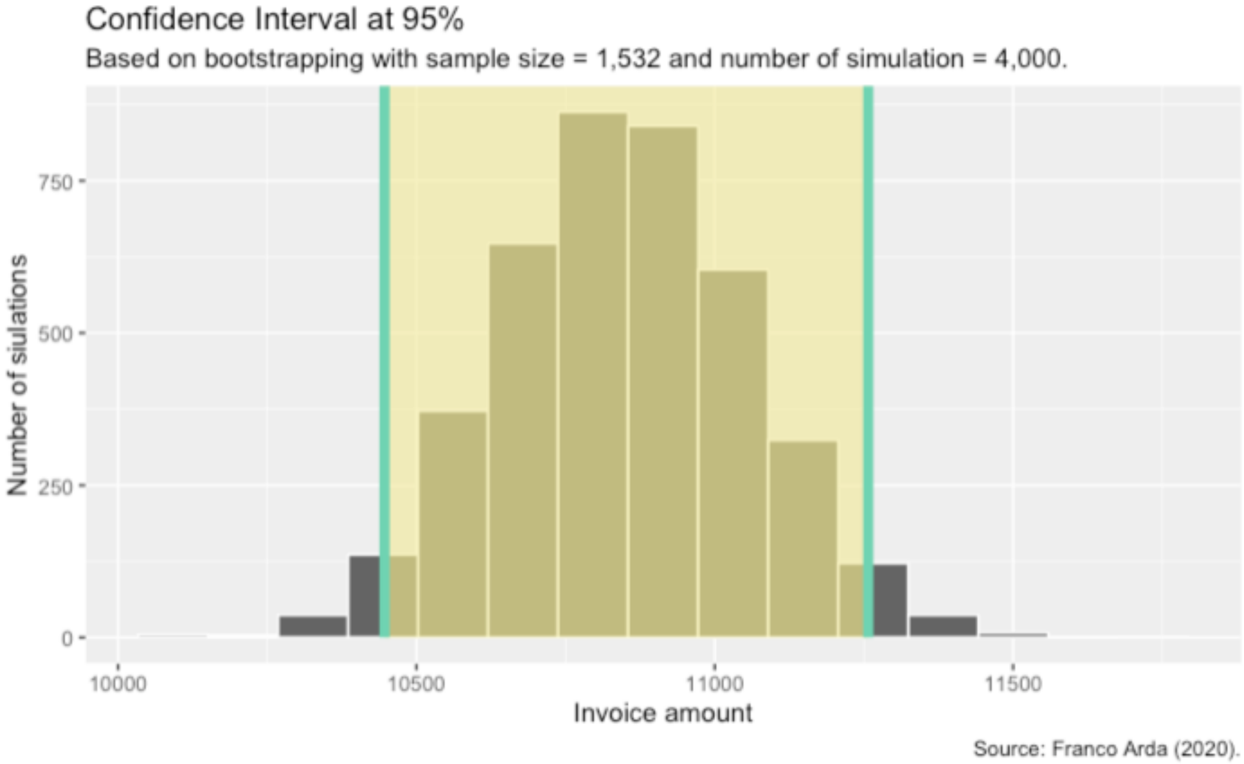


Figure 31: 95% confidence interval based on bootstrapping with a sample size = 1,532, number of simulations = 4,000, and a Shapiro-Wilk p-value = 0.000. Source: Franco Arda (2020).

As discussed in the previous chapter, the confidence interval is limited in determining the sample size required. No significant correlation was obtained between the confidence interval and the sample size required, which is disappointing. On the other hand, a Monte Carlo simulation based on Mean Average Deviation (MAD) was successful.

Monte Carlo sample size determination at a 99% confidence interval.

Figure 30 below shows the findings for the Monte Carlo simulations at a 99% confidence interval. Because of the tighter confidence interval, the assessed sample size required is larger.

For example, with a much lower confidence interval at 90%, we only required a sample invoice size of 1,480. The smaller sample size illustrates the inverse relationship between confidence interval and sample size nicely.

At a confidence interval of 99%, the required sample size was assessed to 1,670, considerably more than at 90% confidence interval.

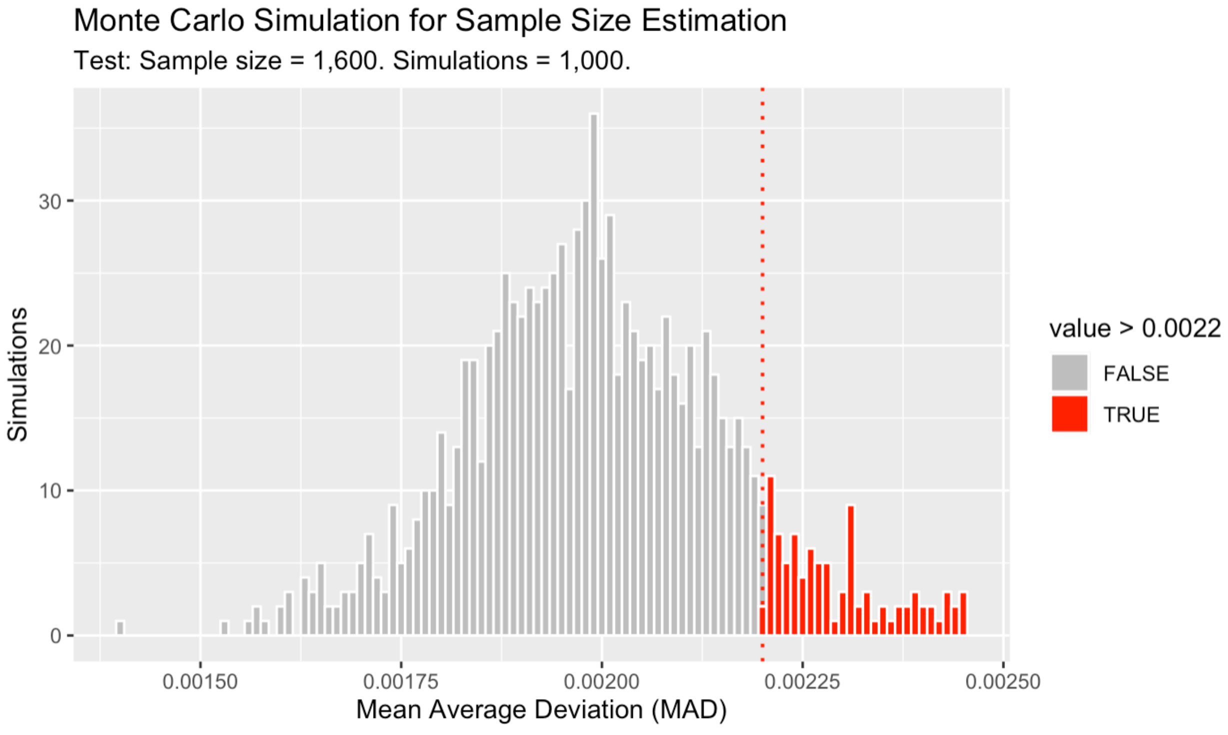
Sample size	Number of simulations	Shapiro-Wilk normality test p-value	Mean Average Deviation	Lower bound confidence level	Upper bound confidence level	Confidence Interval
10	1,000	0.048	0.0022	-	-	0%
100	1,000	0.001	0.0022	-	-	0%
1,000	1,000	0.321	0.0022	10801	10901	17.1%
1,700	4,000	0.000	0.0022	10331	11399	99.2%
1,675	4,000	0.001	0.0022	10331	11399	99.2%
1,650	4,000	0.065	0.0022	10330	11402	99.3%
1,600	4,000	0.021	0.0022	10397	11329	97.6%
1,625	4,000	0.003	0.0022	10368	11355	98.5%
1,635	4,000	0.000	0.0022	10371	11348	98.3%
1,645	4,000	0.160	0.0022	10367	11368	98.7%
1,660	4,200	0.000	0.0022	10367	11377	98.8%
1,670	4,200	0.000	0.0022	10358	11383	99%

Figure 32: Solving for the required sample size, at a 99% confidence interval (size = 1000), a Mean Average Deviation at 0.002, and with a Shapiro-Wilk p-value at 0.05. Franco Arda (2020).

Researchers and practitioners who would like to be very confident in assessing potential invoice fraud might opt for a tighter confidence interval of 99%.

In general, with such a tight confidence interval, we should expect the lowest type I error (false alarm), compared to the 90% and 95% confidence interval. However, this high confidence comes with a price: a lower type I error leads to a higher type II error. In other words, with a 99% confidence interval, while we can be quite confident of getting low type I errors, we potentially miss some fraudulent cases.

Figure X below shows a sample size of 1,600 with only 1,000 simulations. The simulation resulted in a fascinating finding: the distribution needs considerably more simulation to conform closer to the Central Limit Theorem (CLT), measured by the Shapiro-Wilk p-value



Source: Franco Arda (2020)

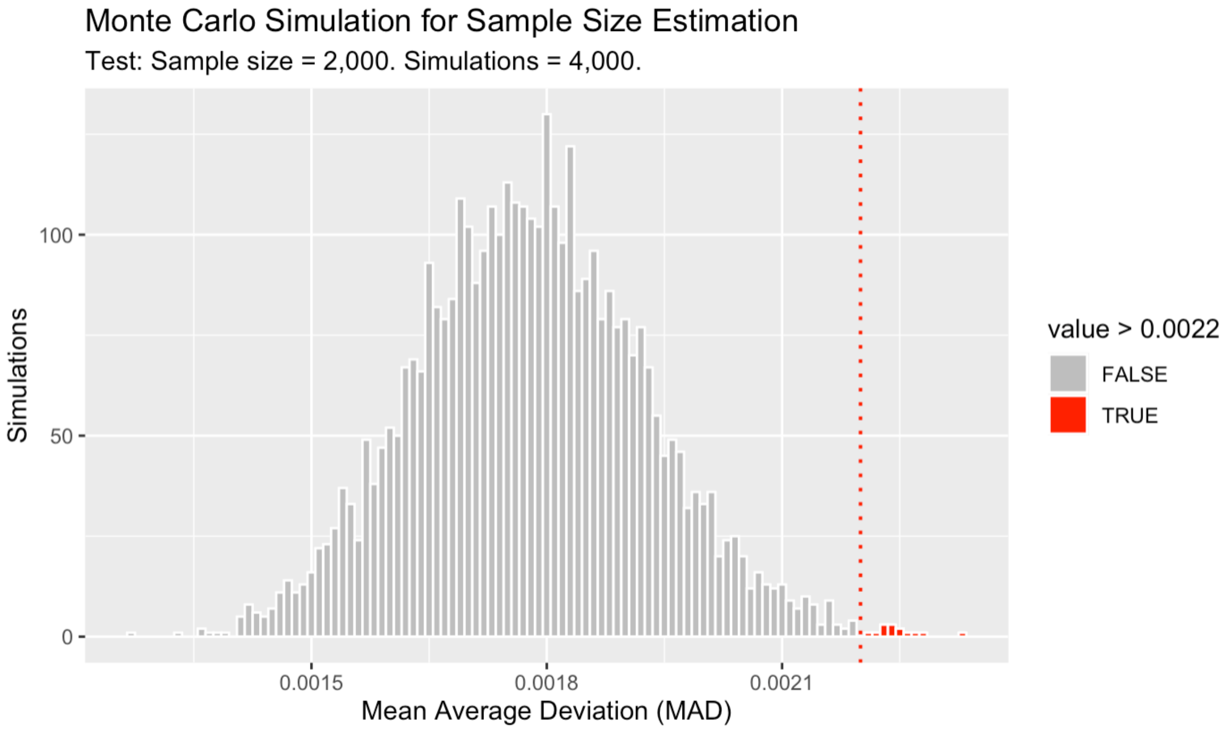
Figure 33: Test based on a Monte Carlo method, with 1,000 simulations and a sample size of 1,600. Source: Franco Arda (2020).

Random sampling forces us to speak in probabilistic terms, rather than absolutes.

Fortunately, fraud events are mutually exclusive. In other words, a vendor can only be conforming or nonconform, but not both. Just because there are two possibilities does not make them equally likely.

As we saw with the synthetic dataset, the probability of fraud is already apparent: out of 100 vendors, only five vendors are nonconform. This "unbalance" makes detecting fraud incredibly tricky. Of course, we do not imply that there should be more fraud to make the balance more equal.

In figure 32 below, we can see from the red part in the histogram that a sample size of 2,000 is potentially too large, as the confidence level is minimal (i.e., 99.3%):



Source: Franco Arda (2020)

Figure 34: Test based on a Monte Carlo method, with 4,000 simulations and a sample size of 2,000. Source: Franco Arda (2020).

A key element in virtually all statistical procedures is independence. By default, we assume independence, but do not test for independence. If two or more vendors are dependent, the dependence should reflect the Mean Average Deviation (MAD).

There are potential exceptions to this case, where one vendor might send invoices from different departments. We assume that invoices are sent from a single vendor. If, within one vendor, there are fraudulent and non-fraudulent departments, we would not detect them. Incorporating this potential case would be very complicated. We assume, though, that we can expect a single vendor to be either conform or nonconform.

With fraudulent transactions occurring rarely, we expect fraud to occur similarly to a Poisson distribution: rare events in time and space.

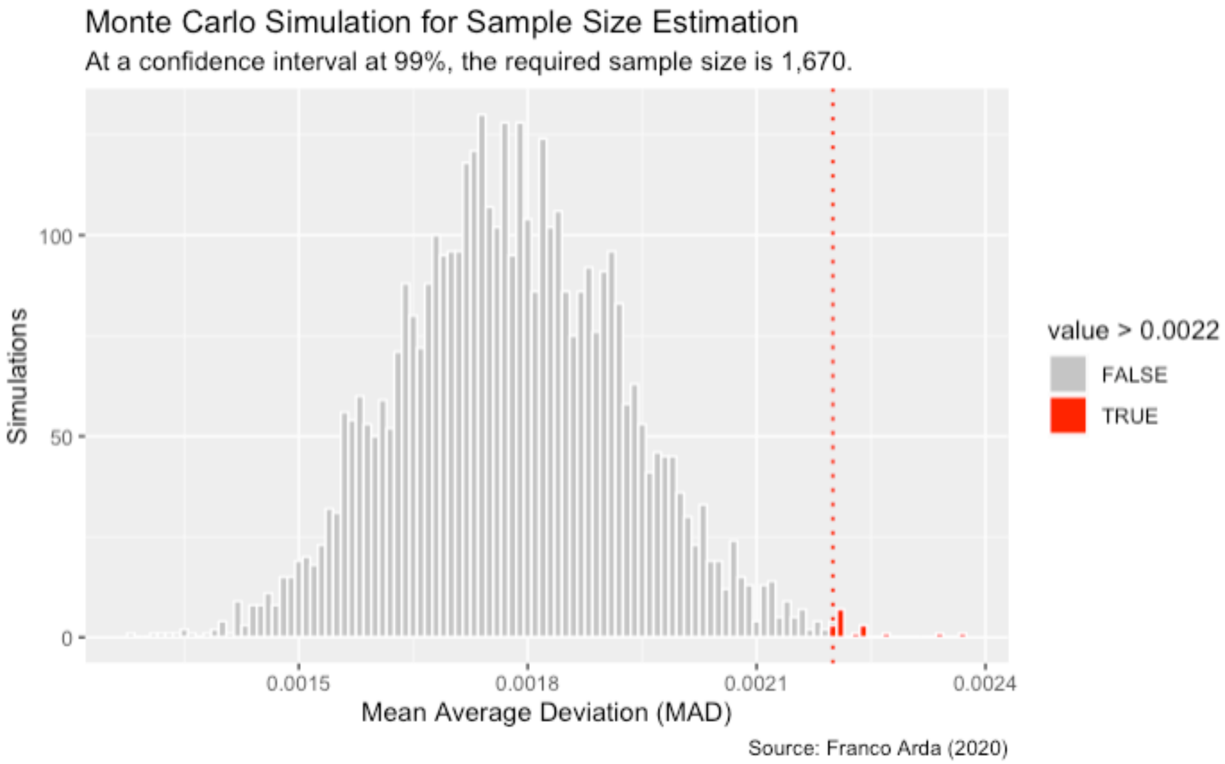


Figure 35: Based on a Monte Carlo method, with 4200 simulations, the required sample size assesses to 1670, at a 99% confidence interval. Source: Franco Arda (2020).

Our second research hypothesis will compare the empirical results of type I and type II errors.

Determining the exact sample size is so complicated that we can never expect to obtain a precise answer. Because of this inherent randomness, we are using a Monte Carlo simulation. The simulation returns an answer for the given confidence interval with a random amount of error. The amount of error can typically be reduced by increasing the number of simulations.

At a confidence interval of 99%, we used 4,200 simulations. Given the law of large numbers from statistics, the more random samples (or resamples) we perform, the more accurate the approximated quantity will become. Goodfellow, Bengio, and Courville (Goodfellow, Bengio, and Courville, 2016) argue that by generating enough samples, then the average almost surely converges to the expected value.

Figure 34 below shows the 99% confidence interval's visualization with a bootstrapping sample size = 1,670, a number of simulations = 4,200, and a Shapiro-Wilk test at a p-value of 0.000.

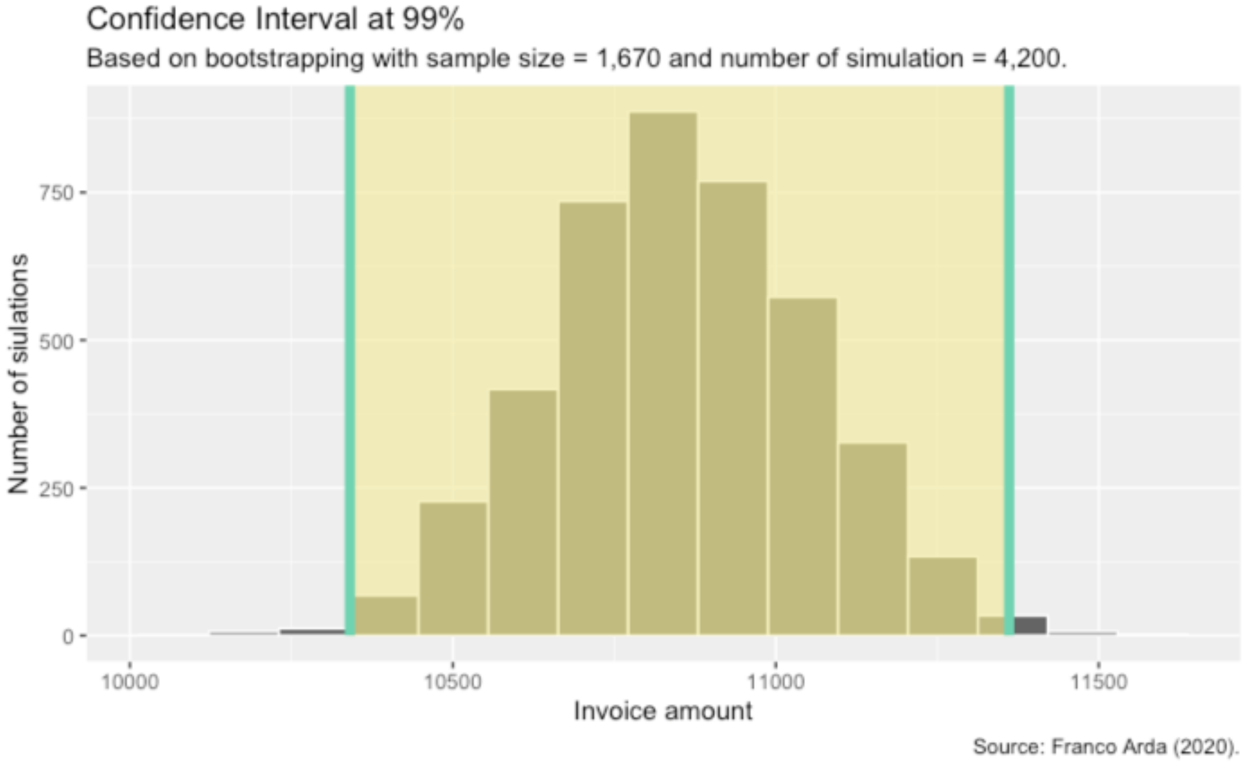


Figure 36: 99% confidence interval based on bootstrapping with a sample size = 1,670, number of simulations = 4,200, and a Shapiro-Wilk p-value = 0.000. Source: Franco Arda (2020).

In particular, for practitioners, communicating the confidence level and correctly can be quite challenging. In the presented research, we try to refer from communicating confidence intervals (e.g., 5% and 90%) but focus on communicating a potentially simpler number, a single value.

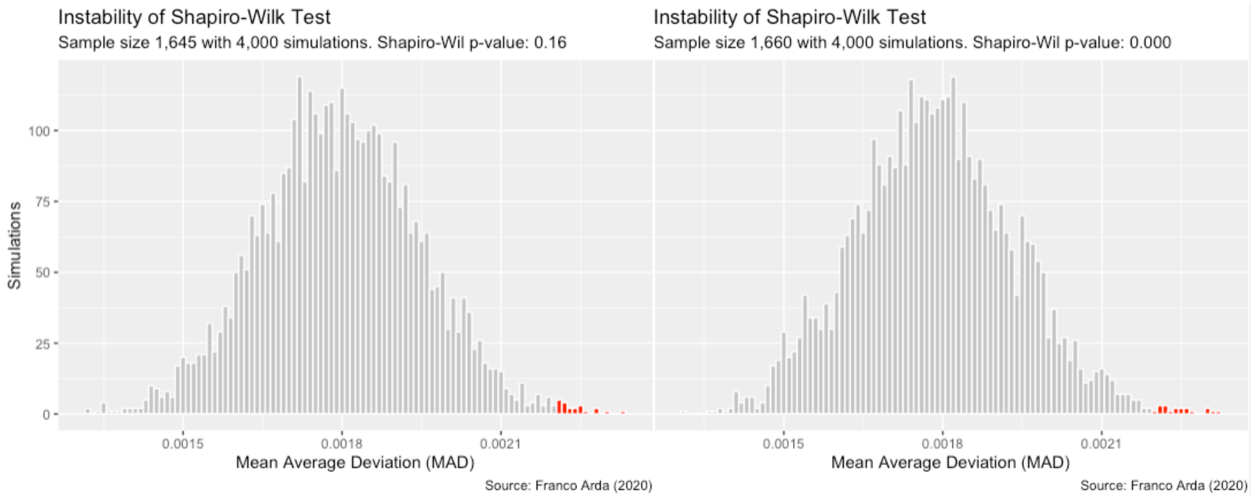
This single value can be a 90%, 95%, or 99% confidence level. Following the Shapiro-Wilk problem with confidence levels, we will dedicate a short chapter on correctly communicating the confidence level.

Instability in the Shapiro-Wilk test with stochastic invoice resampling.

To look at the sample distribution, from a comparable normal distribution, we used the Shapiro-Wilk test. It compares the sample scores to a normally distributed set of scores with the same mean and standard deviation. If the test is not significant (p-value > 0.05), it tells us that the sample's distribution is not significantly different from a normal distribution.

If, however, the is significant (p-value < 0.05), then the distribution of the invoice sample is significantly different from a normal distribution (i.e., it is non-normal). However, it seems to have limitations because, with large samples sizes, it sometimes became precarious, and it was easy to get significant and non-significant results from small deviations from normality to bias the statistical analysis of the sample data.

Below are two identical simulations, but with sample sizes of 1,645 and 1,660. For the first simulation, we got a Shapiro-Wilk p-value of 0.160, while for the later, a p-value of 0.000.


Instability of Shapiro-Wilk Test

Sample size 1,645 with 4,000 simulations. Shapiro-Wil p-value: 0.16

Instability of Shapiro-Wilk Test

Sample size 1,660 with 4,000 simulations. Shapiro-Wil p-value: 0.000

Source: Franco Arda (2020)

Source: Franco Arda (2020)

Figure 37: The instability of the Shapiro-Wilk test visualized. Sample size 1,645 with a p-value of 0.16 vs. a sample size of 1,660 with a p-value of 0.000. Source: Franco Arda (2020).

The p-value results are impossible and are most likely caused due to the sensitivity of the Shapiro-Wilk test. The variations were less of a problem with multiple tests while indicating the number of simulations' general correctness. We state from the research general because the variations seemed to normalize with +/- 100 simulations.

The critical consideration for the first research hypothesis is to determine the sample size given a specific confidence interval with the ultimate goal to reduce type I and type II errors in order to increase the accuracy of the Benford algorithm.

Confirming whether the data simulation with a Monte Carlo simulation within our first research hypothesis sufficiently meets a normal distribution assumption should be seen as an informed judgment based on a series of simulations, rather than a definitive black and white decision.

A parametric test, such as the Shapiro-Wilk test assumes that the variance within different sample data parts is equal.

With a Monte Carlo simulation taking only around 1,500 samples from the invoice population of 10,000, we ultimately can expect random variations that make it ultimately hard for the Shapiro-Wilk test to deliver locally stable results.

Initially, during our research, this threw us off. However, with extensive simulations, the pattern emerged: while we saw some substantial variations locally from the Shapiro-Wilk test, globally, we got a strong indication of where the required number of simulations should be. Mostly, this global level of simulations was around 4,000.

In other words, with the Monte Carlo simulations, we ultimately get variations in the sampling. For the sampling distribution, this can is seen in figure 35. We can think of the sampling distribution only as of the frequency of sample MAD means from the population (i.e., 10,000 invoices).

The sampling distribution tells us about samples' behavior from the population, which is due to the Central Limit Theorem, centered at a similar value as the population's mean. In other words, based on the Central Limit Theorem, a sample taken from the sample should have a similar mean as the population.

Our findings fall into two categories: first, the Shapiro-Wilk normality test provides us with a statistical measure to define the number of simulations required. Second, we should not use a lower p-value than 0.05 due to the instability with stochastic simulations.

Communicating confidence levels in the context of fraud.

The language in which we communicate those confidence intervals to accounting, auditing, and business management can be tricky. To be clear, it is not about what confidence intervals mean, but how we communicate and interpret those results.

In the context of sample size determination for invoice fraud, we do not communicate the confidence intervals as we would typically do in a research paper or communicating to a regulator. In other words, we do not communicate confidence intervals but confidence levels.

We were researching confidence levels (i.e., 2.5% and 97.5%) as any researcher would do, but once we communicate to the business, I believe we should only refer to the confidence level.

Confidence intervals expressed as a confidence (or coverage) level communicate how high a confidence level is. In general, we would consider a 95% confidence level as high and 99% as too high. Technically, one way to think of a 95% confidence level is as follows: it is the interval that encloses the central 95% of the bootstrap sampling distribution of the sample statistic. More generally, a 95% confidence level should, on average, contain similar sample estimates at 95% of the time.

A common but incorrect interpretation is: "There is a 95% probability that the confidence interval contains p." (Kim and Ismay, 2018).

In other words, a 95% confidence interval indicates that 95 out of 100 times, the sample reflects the Benford distribution. With the 95% confidence interval, we are statistically referring to the proportion constructed confidence interval that would likely resemble Benford's distribution. If we ran the experiment 1,000 times, in about 950 of those replications, the invoice samples would lead to calculating a confidence interval that overlaps if the invoices adhere to Benford's distribution.

About 50 of those 1,000 replications would give us a confidence interval that was either too high or too low – both ends of the confidence interval would either be above or below the expected distribution of Benford's law.

Of course, if the invoice samples do not comply to Benford's law, we will never see a confidence level of 90% or higher. Conclusions that were drawn inductively from invoice samples are never fixed or firm. We may characterize our uncertainty, but we can never be 100% sure of anything when using confidence intervals based on statistical inference.

Technically, even with 10,000 invoices samples, we cannot be 100% sure that the vendor conforms to Benford's distribution. We can say that so far, the invoice samples have conformed to Benford's distribution, but it might change in the future, and therefore we cannot be sure.

The second part of our research deals with aggregation. The results from the first research hypothesis provide a statistically significant sample sized.

In conclusion, we succeeded in finding a suitable Monte Carlo simulation approach to determine the required sample size. Unfortunately, our first attempt via a bootstrap did not succeed. The extracted confidence intervals in a bootstrap proved to be unstable for the Benford algorithm. Fortunately, we succeeded when we switched to a traditional Monte Carlo simulation approach based on the mean average deviation.

We were surprised to see that a sample from the Benford population (our synthetic dataset) adhered to the CLT (Central Limit Theorem). In other words, given that the Benford distribution is highly skewed, we did not expect samples to be normally distributed. The fact that samples did adhere to CLT allowed us to add another statistical measure: the Shapiro-Wilk normality test. With this test, we could fine-tune the simulations required for the Monte Carlo simulation, leading to more stable classification results.

Methodology for the second research hypothesis

This study aims to address the issue of aggregation in determining conformity based on Benford's distribution. Many researchers researching Benford's law (Tota, Aliaj, and Lamcja, 2016) calculate conformity to Benford's law based on aggregated data, which is understandable. Without determining the sample size required, those researchers aggregate data, most likely, to avoid too few samples.

This study examines the relationship between conformity and non-conformity to Benford's distribution based on aggregation. For example, in our dataset created for this research, we simulated one hundred vendors. This study's central hypothesis is that we can get better accuracy if we test each vendor on a granular level for conformity.

The notion of accuracy underlying this study is a broad one. Accuracy is one metric for evaluating a classification model. We have a binary classification model with the Benford algorithm that classifies a dataset as conforming or non-conform.

Some researchers (Nigrini, 2020) even go on a more granular level and use multiclass classification: close conformity, acceptable conformity, marginally acceptable conformity, marginally acceptable conformity, and non-conformity.

To understand the nature of aggregation, we focus only on a binary classification: conformity and non-conformity. Informally, accuracy is the fraction of predictions our model got right.

Formally, the simple accuracy formula has the following definition:

Accuracy = Number of correct predictions / Total number of predictions.

In order to test the research hypothesis two, we examine variability associated with aggregation. As the link between aggregation and granularity is the focus of this research hypothesis, we calculate accuracy in positives and negatives.

Formally, for binary classification (Luque, Carasso, Martin, and de la Herras, 2020), accuracy can be calculated in terms of positives and negatives as follows:

Accuracy = TP + TN / TP + TN + FP + FN

Where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

The more detailed binary classification model's choice is to detect better why we might see different accuracy results. For example, we want to know why we get different accuracy by comparing the research results in aggregation vs. granularity. Are we getting a better accuracy because of a reduced FP (false positives) or a reduced FN (false negatives)?

Our main goal is to understand the mechanism underlying aggregation and corresponding accuracy. Investigating the effect of the binary classification should help us understand the reasons for better accuracy.

With fraud, we often have a considerable class imbalance. Our dataset has a class imbalance of 1 / 100, where 1 is a nonconform dataset out of one hundred datasets. Several researchers have addressed (Hossin and Sulaiman, 2015) the class imbalance on classification performance metrics.

The class imbalance has no impact on Benford's algorithm because it is hardcoded and does not "learn" like a machine learning algorithm.

In other words, the Benford algorithm is fundamentally different from statistical learning algorithms or statistical Machine Learning that it does not learn.

Research assumptions and defining the ground truth.

The purpose of this research hypothesis is defined as follows: compare the binary classification accuracy of this modified Benford algorithm against a traditional approach. The research design so that we compare the accuracy directly. One challenge arises, though: what is the ground truth?

As defined by Wikipedia, ground truth is a term in statistics and machine learning, which refers to the accuracy of a classification algorithm. Ground truth is used to prove or disprove a research hypothesis.

Ground truth can be wrong, however. As ground truth is a measure, there can be errors in it. One assumption of this research is that the ground truth equals to a randomized dataset. As for the second research hypothesis, we need to compare the accuracy of two different approaches; we need to make another assumption:

The ground truth for classifying a dataset is based on a granular, and not aggregated, level.

Justifications for the two assumptions:

(1) As stated earlier, the justification for assuming that a fraudster uses a similar approach to randomized data is our best guess how a fraudster works. In general, fraud is committed through inventing numbers. The potential impact of this assumption is not without risk. A fraudster might be lucky and randomize data that conform to the Benford distribution.

Another risk is that a fraudster uses techniques we are not aware of, such as a fraudster algorithm. We believe though, the assumption of random data is close to how a fraudster creates invoices. Additionally, this assumption has a substantial benefit that we can objectively compare different approaches to the Benford algorithm.

(2) The potential impact of this assumption is limited. In general, when researchers test the Benford algorithm on datasets, they start on a granular level. We take a vendor's invoices and

test the conformity based on the Mean Average Deviation (MAD) or another similar statistical technique.

The potential algorithmic problems with data aggregation

We will adopt the determined sample size combined with data on a granular level for the second research hypothesis. To our knowledge, no research has investigated the potential algorithmic problems when inferring conformity from a granular level to an aggregate level. The arithmetic problem that might arise from aggregating is easily formulized:

Aggregation: Conformity to Benford's distribution, based on Mean Average Deviation (MAD), is calculated by taking all vendors' sum (aggregation).

Granularity:
Conformity to Benford's distribution, based on Mean Average Deviation (MAD), is calculated by each vendor individually, on a granular basis, no matter the number of vendors.

Previous research has mostly overlooked the effect of aggregation, even very recent research (Blondeau da Silva, 2019). To test the research hypotheses empirically, at a p-value of 0.05, we first compare our approach to that of another researcher (Cinelli, 2019) based on a publicly available dataset of invoice fraud.

The presented research study in figure X below represents the difference in Mean Average Deviation (MAD) based on aggregation (middle column) vs. granularity (right column).

The presented research study in figure X below represents the difference in Mean Average Deviation (MAD) based on aggregation (middle column) vs. granularity (right column).

Vendor number:	MAD based on aggregation:	MAD based on granularity:
2001	0.00234	0.00334
2004	-	0.00338
2006	-	0.00738

Figure 38: Differences in Mean Average Deviation (MAD) based on aggregation vs. granularity. Source: Franco Arda (2020).

As mentioned before, the threshold for conformity is a MAD level < 0.0022. In the table, all vendors displayed are non-conform based on the MAD level. For a scientific hypothesis test, the given dataset is too small.

Table 38 above highlights a significant difference in MAD levels. MAD levels were determined for the vendor numbers 2001, 2004, and 2006. The computation shows that vendor number 2001 was nonconform with aggregation and a granular approach. The resulting MAD levels were very similar (i.e., 0.00234 and 0.0034), suggesting non-conformity for vendor number 2001.

In this somewhat primary research, our findings already illustrate vast differences. For example, on a granular level, vendor number 2006, with a MAD level of 0.00738, displayed the strongest non-conformity.

The results yielded some interesting findings. The test provides some preliminary evidence that computing the conformity level on a granular level might yield higher accuracy. In other words, vendor number 2006 was not flagged on an aggregation level, while on a granular level, this vendor should provide the highest non-conformity level. These results provide first convincing evidence in favor of testing conformity on a granular level.

Admittedly, the dataset was small, and these findings might be less surprising if we consider the size. A possible interpretation of these findings is still that the granular level yields a higher accuracy. The test provides some first convincing evidence in favor of granularity.

Theoretical support for this hypothesis is that we need to evaluate conformity on a granular level, no matter the sample size. If the sample size falls below the defined sample size in the first research hypothesis, we might ignore the conformity results. In other words, if the sample size falls below the determined sample size, we cannot evaluate the conformity.

Evaluating accuracy for benchmark I.

In the preceding chapter, the concept of aggregation and granularity was explored. The first part of this chapter deals with the results based on all hundred vendors.

The simulation in figure 39 below shows the result. For vendors, where there was not enough data, no ground truth was determined. Not enough data is defined at 1,532 at a 95% confidence level. The assumption was that if we do not have enough data, we can not define ground truth. The risk of this assumption is that MAD aggregate might label the corresponding vendor correctly. The advantage is that we get a more objective and replicable result.

Vendor:	MAD aggregate:	Ground truth:	MAD granular:	Ground truth:
Vendor 1	0.000156	✓	0.000121	✓
Vendor 2	0.000156	✓	0.000121	✓

Vendor:	MAD aggregate:	Ground truth:	MAD granular:	Ground truth:
Vendor 3	0.000156	✓	0.000121	✓
Vendor 4	0.000156	✓	0.000121	✓
Vendor 5	0.000156	✓	0.000121	✓
Vendor 6	0.000156	✓	0.000121	✓
Vendor 7	0.000156	✓	0.000121	✓
Vendor 8	0.000156	✓	0.000121	✓
Vendor 9	0.000156	✓	0.000121	✓
Vendor 10	-	✓	Not enough samples	✓
Vendor 11	0.000156	-	0.000121	✓
Vendor 12	0.000156	✗	0.000239	✓
Vendor 13	0.000156	✓	0.000121	✓
Vendor 14	0.000156	✓	0.000121	✓
Vendor 15	0.000156	✓	0.000121	✓
Vendor 16	0.000156	✓	0.000121	✓
Vendor 17	0.000156	✓	0.000121	✓
Vendor 18	0.000156	✓	0.000121	✓
Vendor 19	0.000156	✓	0.000121	✓
Vendor 20	0.000156	✓	0.000121	✓
Vendor 21	0.000156	✓	0.000121	✓
Vendor 22	0.000156	✓	0.000121	✓
Vendor 23	0.000156	✓	0.000121	✓
Vendor 24	0.000156	✗	0.000239	✓

Vendor:	**MAD aggregate:**	**Ground truth:**	**MAD granular:**	**Ground truth:**
Vendor 25	0.000156	✓	0.000121	✓
Vendor 26	0.000156	✓	0.000121	✓
Vendor 27	0.000156	✓	0.000121	✓
Vendor 28	0.000156	✓	0.000121	✓
Vendor 29	0.000156	✓	0.000121	✓
Vendor 30	0.000156	✓	0.000121	✓
Vendor 31	0.000156	✓	0.000121	✓
Vendor 32	0.000156	✓	0.000121	✓
Vendor 33	0.000156	✓	0.000121	✓
Vendor 34	0.000156	✓	0.000121	✓
Vendor 35	0.000156	✓	0.000121	✓
Vendor 36	0.000156	✗	0.000239	✓
Vendor 37	0.000156	✓	0.000121	✓
Vendor 38	0.000156	✓	0.000121	✓
Vendor 39	0.000156	✓	0.000121	✓
Vendor 40	0.000156	✓	0.000121	✓
Vendor 41	0.000156	✓	0.000121	✓
Vendor 42	0.000156	✓	0.000121	✓
Vendor 43	0.000156	✓	0.000121	✓
Vendor 44	0.000156	✓	0.000121	✓
Vendor 45	0.000156	✓	0.000121	✓
Vendor 46	0.000156	✓	0.000121	✓
Vendor 47	0.000156	✓	0.000121	✓

Vendor:	MAD aggregate:	Ground truth:	MAD granular:	Ground truth:
Vendor 48	0.000156	✗	0.000239	✓
Vendor 49	0.000156	✓	0.000121	✓
Vendor 50	0.000156	✓	0.000121	✓
Vendor 51	0.000156	✓	0.000121	✓
Vendor 52	0.000156	✓	0.000121	✓
Vendor 53	0.000156	✓	0.000121	✓
Vendor 54	0.000156	✓	0.000121	✓
Vendor 55	0.000156	✓	0.000121	✓
Vendor 56	0.000156	✓	0.000121	✓
Vendor 57	0.000156	✓	0.000121	✓
Vendor 58	0.000156	✓	0.000121	✓
Vendor 59	0.000156	✓	0.000121	✓
Vendor 60	0.000156	✗	0.000239	✓
Vendor 61	0.000156	✓	0.000121	✓
Vendor 62	0.000156	✓	0.000121	✓
Vendor 63	0.000156	✓	0.000121	✓
Vendor 64	0.000156	✓	0.000121	✓
Vendor 65	0.000156	✓	0.000121	✓
Vendor 66	0.000156	✓	0.000121	✓
Vendor 67	0.000156	✓	0.000121	✓
Vendor 68	0.000156	✓	0.000121	✓
Vendor 69	0.000156	✓	0.000121	✓

Vendor:	MAD aggregate:	Ground truth:	MAD granular:	Ground truth:
Vendor 70	-	✓	Not enough samples	✓
Vendor 71	0.000156	✓	0.000121	✓
Vendor 72	0.000156	✓	0.000121	✓
Vendor 73	0.000156	✓	0.000121	✓
Vendor 74	0.000156	✓	0.000121	✓
Vendor 75	0.000156	✓	0.000121	✓
Vendor 76	0.000156	✓	0.000121	✓
Vendor 77	0.000156	✓	0.000121	✓
Vendor 78	0.000156	✓	0.000121	✓
Vendor 79	0.000156	✓	0.000121	✓
Vendor 80	0.000156	✓	0.000121	✓
Vendor 81	0.000156	✓	0.000121	✓
Vendor 82	0.000156	✓	0.000121	✓
Vendor 83	0.000156	✓	0.000121	✓
Vendor 84	0.000156	✓	0.000121	✓
Vendor 85	0.000156	✓	0.000121	✓
Vendor 86	0.000156	✓	0.000121	✓
Vendor 87	0.000156	✓	0.000121	✓
Vendor 88	0.000156	✓	0.000121	✓
Vendor 89	0.000156	✓	0.000121	✓
Vendor 90	-	✓	Not enough samples	✓
Vendor 91	0.000156	✓	0.000121	✓

Vendor:	MAD aggregate:	Ground truth:	MAD granular:	Ground truth:
Vendor 92	0.000156	✓	0.000121	✓
Vendor 93	0.000156	✓	0.000121	✓
Vendor 94	0.000156	✓	0.000121	✓
Vendor 95	0.000156	✓	0.000121	✓
Vendor 96	0.000156	✓	0.000121	✓
Vendor 97	0.000156	✓	0.000121	✓
Vendor 98	0.000156	✓	0.000121	✓
Vendor 99	0.000156	✓	0.000121	✓
Vendor 100	0.000156	✓	0.000121	✓

Figure 39: Classification results for one hundred vendors for benchmark I. Source: Franco Arda (2020).

Classification accuracy: benchmark I vs. alternative

Many researchers have highlighted that with unbalanced datasets, accuracy is not enough (Sturm, 2013). Our dataset of 100 vendors is highly unbalanced: we have ninety-five conform and five non-conform vendors.

If an algorithm always predicted not fraud, the accuracy would look deceptively good. In this case, the accuracy would result in 95%. In other words, this algorithm would correctly classify 95 vendors. To reflect this accuracy problem, we need different measures of accuracy.

For the benchmark (H_0), we get the following results:

TP (true positive) = 0 / 5 = 0
TN (true negative) = 97 / 97 = 1
FN (false negative) = 5 / 97 = 0.05
FP (false positive) = 0 / 97 = 0

For the alternative (H_1), we get the following results:

TP (true positive) = 5 / 5 = 1.0
TN (true negative) = 97 / 97 = 1
FN (false negative) = 0 / 97 = 0

FP (false positive) = 0 / 97 = 0

We first calculate the simplest of all binary classification metrics: accuracy. Accuracy is simply defined as the proportion of invoices that were correctly classified. The way we calculated all formulas is by defining the ground truth for all vendors. Correctly classified was defined as 1 and wrongly classified as 0.

Calculating the accuracy for the benchmark, or alternative hypothesis, resulted in:

$$\text{Accuracy}_0 = \frac{TP + TN}{TP + TN + FP + FN} = \frac{0 + 1}{0 + 1 + 0.05 + 0} = 95.23\%$$

Figure 40: Accuracy of benchmark I. Source: Franco Arda (2020).

Calculating the accuracy for the alternative, or alternative hypothesis, resulted in:

$$\text{Accuracy}_1 = \frac{TP + TN}{TP + TN + FP + FN} = \frac{1 + 1}{1 + 1 + 0 + 0} = 100\%$$

Figure 41: Accuracy of alternative against benchmark I. Source: Franco Arda (2020).

Since all the positive invoices are either correctly or incorrectly predicted, the sum of the number of true positives and the number of false negatives equals the total number of positive invoice samples.

True and false positives and negative scans are conveniently summarized in a table called a confusion matrix. A confusion matrix for a binary classification problem is a 2 x 2 matrix where the true class is along the on-axis, and the predicted class is along with the other. The confusion matrix gives a quick summary of how many true and false positives and negatives there are:

Since we hope to make correct binary classifications, we hope that the diagonal (that is, the entries along a diagonal line from the top left to the bottom right: true negatives and true positives) of the confusion matrix are relatively large, while the off-diagonals are relatively small, as theses present incorrect classifications.

For the confusion matrix for the benchmark (H_0), we get the following results:

	Predicted	
	Positive	Negative
Actual True	0	0.05
Actual False	0	1

Figure 42: Confusion matrix for benchmark (H$_0$). Source: Franco Arda (2020).

For the confusion matrix for the benchmark (H$_1$), we get the following results:

	Predicted	
	Positive	Negative
Actual True	1	0
Actual False	0	1

Figure 43: Confusion matrix for alternative (H$_1$). Source: Franco Arda (2020).

Calculating the precision for the benchmark, or null hypothesis, resulted in:

$$\text{Precision}_0 = \frac{TP}{TP + FP} = \frac{0}{0 + 0} = 0$$

Figure 44: Precision of benchmark I. Source: Franco Arda (2020).

Calculating the precision for the alternative, or alternative hypothesis, resulted in:

$$\text{Precision}_1 = \frac{TP}{TP + FP} = \frac{1}{1 + 0} = 1$$

Figure 45: Precision of alternative against benchmark I. Source: Franco Arda (2020).

Calculating the recall for the benchmark, or null hypothesis, resulted in:

$$\text{Recall}_0 = \frac{TP}{TP + FN} = \frac{0}{0 + 0} = 0$$

Figure 46: Recall of benchmark I. Source Franco Arda (2020).

Calculating the recall for the alternative, or alternative hypothesis, resulted in:

$$\text{Recall}_1 = \frac{TP}{TP + FN} = \frac{1}{1 + 0} = 1$$

Figure 47: Recall for alternative against benchmark I. Source: Franco Arda (2020).

We have been using the terms precision and recall, but there are other terms used as synonyms. For example, in medical diagnosis, the terms sensitivity and specificity are used. Those are just different terms for the same calculation.

Getting a statistically significant result at a p-value of 0.05 does not necessarily mean a difference between the benchmark and alternative results. Because technically, with a p-value at a threshold of 0.05, we have a 5% chance of falsely concluding that the alternative results are indeed significant. Because the base rate of invoice fraud detection is low in the dataset (only 5%), we have many opportunities for false positives.

The p-value tells us the probability of obtaining a difference in accuracy between benchmark and alternative, given the null hypothesis is true. In other words, our statement about a more accurate model than the benchmark most be taken with caution, as statistically significant with a p-value of < 0.05 can fluctuate with changes in the base rate.

The dangers of false positives need to be weighted with the possible early detection of fraud based on different confidence levels, such as 90%, 95%, and 99%. With industry-specific experience in fraud, a company might even modify the confidence level to their specific needs. For example, a fraud examiner might select 97.50% confidence as more appropriate for their analytics.

Accuracy is not enough with fraud.

Decisions based on the selected or even modified confidence level are one key question to using recall: if the Benford algorithm flags a vendor, what is the probability that it is invoice fraud? If this probability is too low, most positive results will be false, and a great deal of time and effort will be wasted for no benefit.

The challenges with recall, or the probability of detecting fraud, can only partially be the bridge with experience. The reason for this is that we might never find out the true ground truth. That is, we might never detect all fraudulent transactions.

In other words, if we never have the exact ground truth, we can never know the probability of recall. In our formulas, this problem is quantified as false negatives. Likely, we never know all false negatives in the real-world. However, this should not make the model unusable. If we can detect a large part of the invoice fraudsters, in particular fraudsters with large invoice amounts, the investment from a company, calculated in time and money, can be worth it.

In other words, accuracy is not enough, because with an unbalanced dataset where only 5 out of 100 are invoice fraudsters in the synthetic dataset, predicting just always not fraud would already result in a 95% accuracy. In general, the challenge with unbalanced datasets is a concern for most fraud detection. For example, insurance fraud accounts for "only" about 15% (Insurance Information Institute, 2020).

Accuracy is not enough, because, with an unbalanced dataset, we could always predict not-fraud and get a 95% accuracy. Precision, in the context of fraud, delivers a better measure but does not take into account false negatives in the denominator.

We are concerned with detecting fraud for rare fraud data modeling and want to keep false negatives as low as possible. False negatives are in the denominator of recall. Therefore, we can deduce that recall is most likely much more important than any other accuracy measure with fraud detection.

We need to strongly differentiate between the probability of classifying fraud and the p-value. We cannot express the p-value at 0.05 that the result is with a 95% chance true, as this would be an incorrect interpretation of the p-value. A more appropriate interpretation would be that there is a 5% chance (p-value at 0.05), what we see such an extreme value of a better accuracy for the alternative approach.

Calculating the probability of false positives (Reinhart, 2015) is fairly straight forward. If we have *n* independent hypotheses to test, and none is true, then the probability of obtaining at least one false positive is at a p-value of 0.05:

$$\text{P(false positive)} = 1 - (1 - 0.05)^{n}$$

With this formula, if n equals 50, then the probability of a false positive increase to 92.31%. One technique for correcting multiple comparisons is the Bonferroni correction method.

The method takes the number of comparisons (i.e., n) in the trial and modifies the p-value accordingly; 0.05/n. By applying the Bonferroni correction, we reduce the risk of a false positive.

However, this method potentially reduces the statistical power as we demand much stronger correlations before we can conclude that they are statistically significant.

Therefore, we use a simulation-based hypothesis testing to see if the alternative statistically significantly outperforms the benchmark:

H_0 : $\text{accuracy}_{\text{benchmark}} - \text{accuracy}_{\text{alternative}} = 0$
H_1 : $\text{accuracy}_{\text{benchmark}} – \text{accuracy}_{\text{alternative}} > 0$

For the hypothesis with alternative H_1: $\text{accuracy}_{\text{benchmark}} – \text{accuracy}_{\text{alternative}} > 0$, we compute a p-value by finding the proportion of resamples equal to the observed test statistic. Two-sided p-values are the default in many statistics practice, and we should, in general, perform a two-sided test unless we have clear reasons to pick a one-sided alternative hypothesis.

Additionally, it is not scientific to look at the data before deciding to use a one-sided hypothesis. In our case, though, we have strong reasons for deciding on a one-sided alternative: we do not want to know if the accuracy between the benchmark and alternative is different, we want to know if the alternative produces a statistically significant higher accuracy. If not, the experiment fails. The benchmark might have a statistically significant higher accuracy, but this fails in the experiment as well.

Simulation-based hypothesis test for benchmark I

We randomly sample 1,000 times the accuracy of the benchmark vs. the alternative. The null hypothesis simply assumes that there is no difference. The p-value, set at 0.05, is the probability of obtaining a test statistic just as extreme or more extreme than the observed test statistic assuming the null hypothesis H_0 is real.

In the simulation, which follows, we shuffle (permute) the accuracy 1,000 times as there was no difference in accuracy between the methods. Then, we measure the real difference and evaluate how likely we would see such a difference if the null hypothesis were correct.

From our simulations, we define classifications as “correct”, for both, the benchmark and the alternative.

```
set.seed(12)
null_distribution <- accuracy %>%
specify(formula = Classification ~ Method, success = "correct") %>%
hypothesize(null = "independence") %>%
generate(reps = 1000, type = "permute") %>%
calculate(stat = "diff in props", order = c("alternative", "benchmark"))
null_distribution
# 1,000
```

Similarly, as in the research hypothesis one, we are randomly sampling. With a hypothesis test, we use permutations which are a kind of resampling, but unlike the bootstrap method from the first research hypothesis, we know resample without replacement.

From the code above, we see that we got 1,000 rows returned for the accuracy proportion for all of the 100 datasets, multiplied by 1,000.

We defined “independence" for our hypothesis test, which refers to independence between the benchmark and alternative. In our case, we are testing whether the "alternative" variable is independent of the explanatory variable "benchmark." Formally, we define the hypothesis:

H_0 : $\text{accuracy}_{\text{alternative}}$ - $\text{accuracy}_{\text{benchmark}}$ = 0
H_1 : $\text{accuracy}_{\text{alternative}}$ - $\text{accuracy}_{\text{benchmark}}$ > 0

In other words, the null hypothesis H_0 refers to the proportions of how often, out of 1,000 simulations, the accuracy was different from the alternative Hypothesis_1. What was the observed difference in proportion rates? In other words, what was the $\text{accuracy}_{\text{alternative}}$ proportion – $\text{accuracy}_{\text{benchmark}}$?

We get a difference of 0.05, which is extremely unlikely to see by chance, as we will see in a moment.

```
obs_diff_prop <- accuracy %>%
    specify(Classification ~ Method, success = "correct") %>%
    calculate(stat = "diff in props", order = c("alternative", "benchmark"))
obs_diff_prop
# 0.05
```

For each of our 1,000 permutations, we calculated the test statistic by setting stat = “diff in props” since we are interested in simulating the accuracy difference between the benchmark vs. alternative. Furthermore, since we are interested in the difference in proportions, we set alternative vs. the benchmark.

In a famous blog post, Allen Downey (Downey, 2016) called this hypothesis testing framework, "there is only one test." In our second research hypothesis, we take the accuracy of benchmark II and alternative (i.e., data).

We test the difference in accuracy between benchmark II and alternative (i.e., observed effect). Then we shuffle (permute) the accuracy of benchmark II and alternative, as there was no difference (i.e., simulate data). Finally, we compare the observed effect with the simulated data under the assumption that the null hypothesis is true.

In figure 48 below, we visualized the simulation-based hypothesis testing. The red line reflects how likely it is to see a proportion difference of 0.05. As we see below, the chance of seeing such an extreme value if there were no difference between the benchmark and the alternative is 0.027, the p-value.

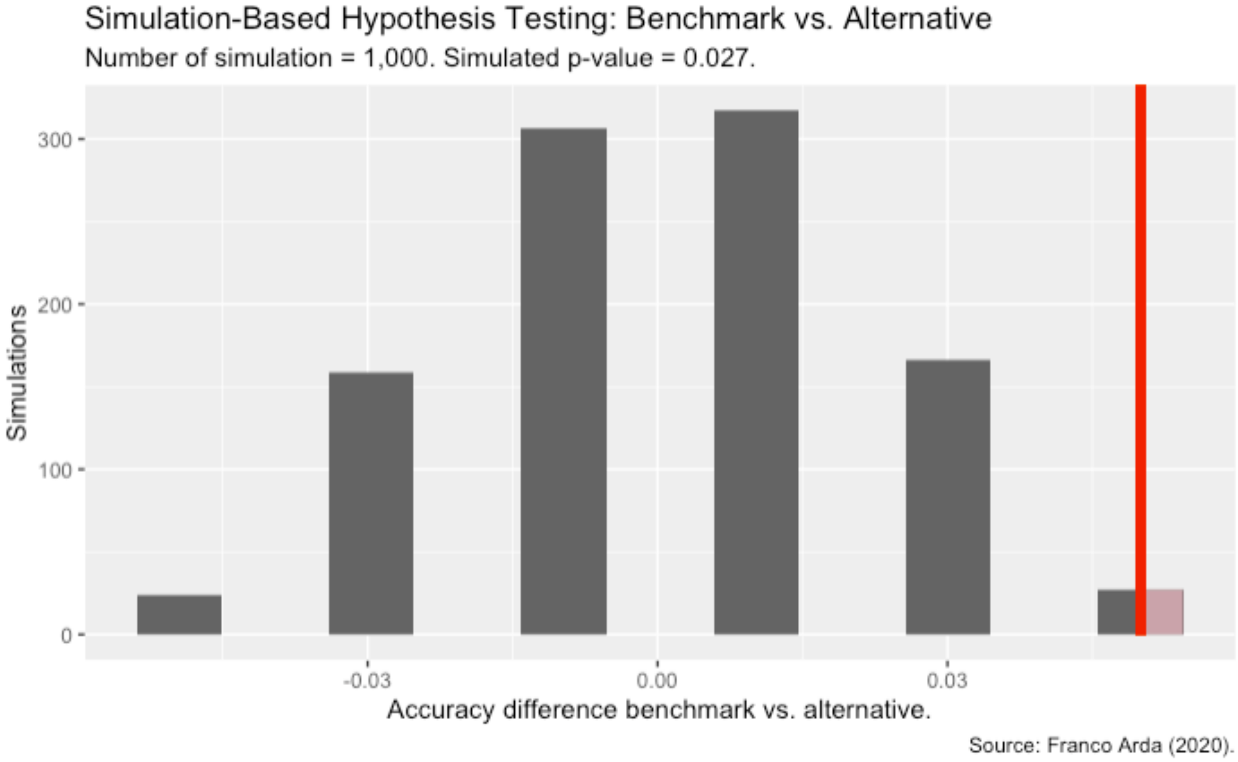


Figure 48: Simulation-Based Hypothesis Testing: Benchmark vs. Alternative. Number of simulations = 1,000. Simulated p-value = 0.027. Source: Franco Arda (2020).

Furthermore, we had set the direction = "right," reflecting our alternative hypothesis H_1: $\text{accuracy}_{\text{alternative}}$ – $\text{accuracy}_{\text{benchmark}}$, stating that there is a difference in the alternative rates in favor of the alternative.

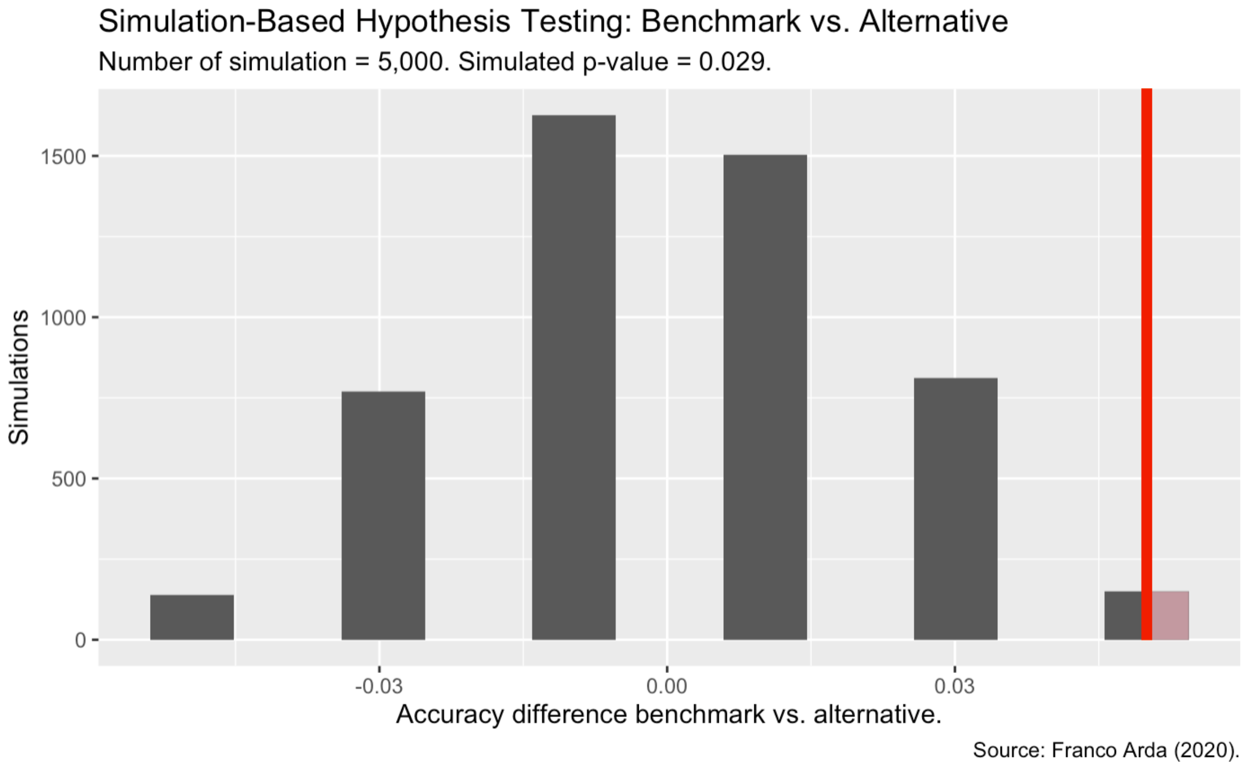
“More extreme” here corresponds to differences > 0.05, which would be even more unlikely by chance. Hence, we see a red shade in the visual for those even more extreme points.

Judging by the shaded region in red, it seems we would rarely observe differences in proportions of 0.05 = 0.05% or more if the accuracy of the benchmark and the alternative were the same.

The code snippets show that we are visualizing the difference to the "right." In other words, we are only interested whether the alternative performs better than the benchmark.

```
null_distribution %>%
        get_p_value(obs_stat = obs_diff_prop, direction = "right")
# p-value: 0.027
```

With a p-value of 0.027, we can reject the null hypothesis. In other words, we can reject H_0 since this p-value is smaller than our pre-specified significance level 0.05. We have enough evidence in favor of the alternative with improved accuracy.



Source: Franco Arda (2020).

Figure 49: Simulation-Based Hypothesis Testing: Benchmark vs. Alternative. Number of simulations = 5,000. Simulated p-value = 0.029. Source: Franco Arda (2020).

One reason to resample the accuracy results is that the cutoff value for the p-value at 0.05 is arbitrary. We might get different results with a p-value slightly lower (e.g., p-value 0.04) or a slightly higher value (e.g., p-value 0.06). In other words, we might mistakenly accept or reject a hypothesis simply because we picked the correct or wrong cutoff for the p-value.

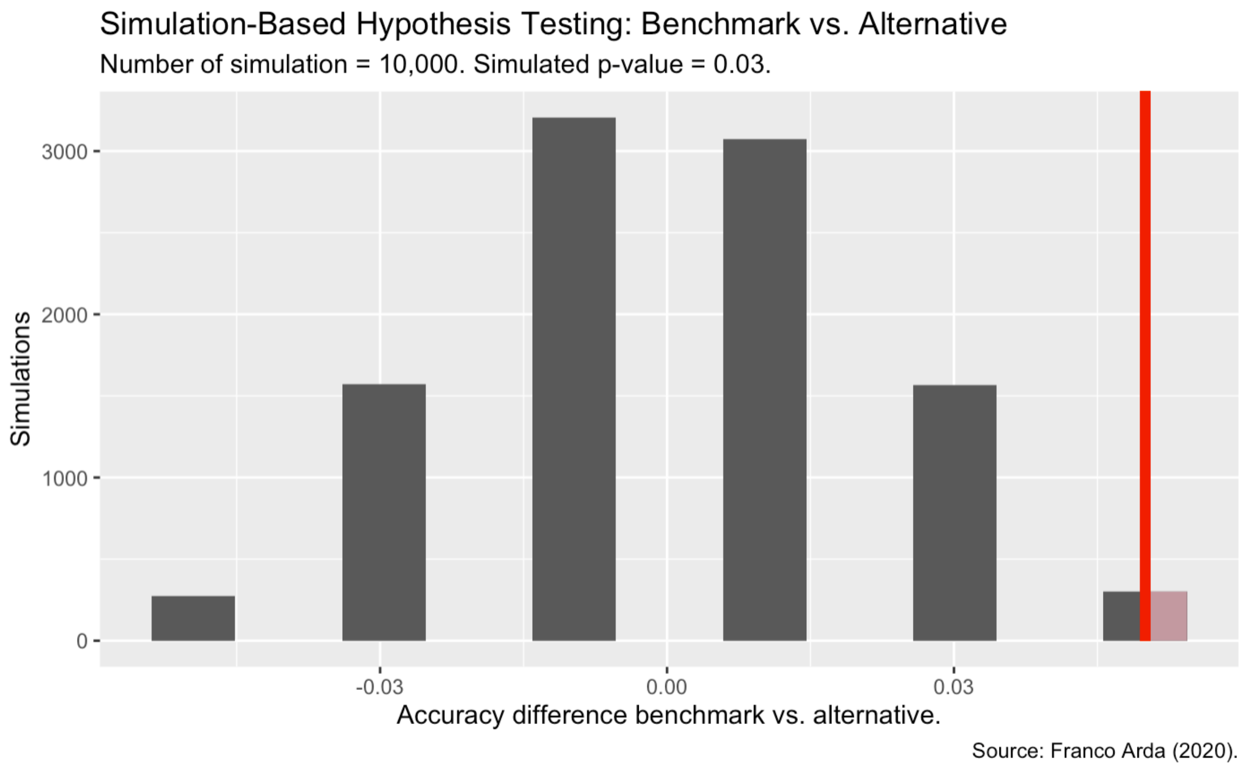


Figure 50: Simulation-Based Hypothesis Testing: Benchmark vs. Alternative. Number of simulations = 10,000. Simulated p-value = 0.03. Source: Franco Arda (2020).

We create a permutation resample with this permutation hypothesis test by drawing m = 100 observations without replacement from the pooled data (all binary classifications from benchmark and accuracy) to be one sample, leaving the remaining m = 100 observations to be the second sample.

We calculate the accuracy and repeat these 1,000 times. The p-value is then the fraction of times the random sampling exceeds the original statistic.

What assumptions are we making with the hypothesis test using permutation? The permutation test makes no assumption on the two-populations (benchmark vs. alternative) under consideration.

According to some research (Chihara and Hesterberg, 2019), the permutation test should still work, provided the sample sizes are equal. In our research, the later is the case, why we will not investigate the distribution of the benchmark vs. alternative accuracy.

Classification accuracy: benchmark II vs. alternative

Based on the preceding chapter, the benchmark's accuracy, we compared 95.23% accuracy to the alternative at 100%. The results are probably hard to believe in terms of recall: while benchmark I got a recall of 0, the alternative accomplished a recall of 1. In other words, the benchmark I algorithm could not detect a single invoice fraudster, while the alternative detected all invoice fraudsters (i.e., all five):

	Benchmark I	Alternative
Accuracy	95.23%	100%
Recall	0	1

Figure 51: Accuracy and recall of Benchmark I vs. Alternative. Source: Franco Arda (2020).

To the best of my knowledge, I'm not aware of any research paper that uncovered the aggregation problem associated with the Benford algorithm. Since the results are almost "too good to be true," I propose a Benchmark II in the last step of this thesis.

All assumptions remain the same, except Benchmark II, we determine conformity based on Benford's law by granularity. By modifying Benchmark II, the algorithm should improve accuracy, and more importantly, recall. Therefore, the only difference between Benchmark II and Alternative is based on the research hypothesis one: the required sample size.

To simulate realistically real-world fraud, we will randomize the number of invoices from 10 vendors. Provided my university and supervisor agree, we will publish this dataset for benchmark II. The goal is twofold: first, researchers can use the dataset to test their experiments. Second, companies can use the dataset to test their inhouse fraud detection models against the dataset and compare their results against ours.

Steps in the synthetic dataset "benchmark II":

1. Vendor 12, vendor 13, vendor 14, vendor 15, and vendor 16 invoices are randomized.

2. Vendor 91, vendor 92, vendor 93, vendor 94, vendor 95, vendor 96, vendor 97, vendor 98, vendor 99, and vendor 99 invoice samples are randomized to a level below the confidence level of 95%

We reduced the number of invoices for the vendors in (2) all below the given confidence level at 95% for simplification. The fast justify this simplification that the accuracy for the alternative would not change.

In order to calculate the accuracy for Benchmark II, we need to know the MAD levels for vendors 91 – vendor 100:

MAD levels:
Vendor 91: 0.00598 (non conform)
Vendor 92: 0.00719 (non conform)
Vendor 93: 0.00666 (non conform)
Vendor 94: 0.00654 (non conform)
Vendor 95: 0.00707 (non conform)
Vendor 96: 0.00749 (non conform)
Vendor 97: 0.00199 (conform)
Vendor 98: 0.00247 (non conform)
Vendor 99: 0.00247 (non conform)
Vendor 100: 0.00208 (conform)

The following list shows the classification results for one hundred vendors of the benchmark II against the alternative:

Vendor:	**Benchmark II:**	**Ground truth:**	**Alternative:**	**Ground truth:**
Vendor 1	0.000121	✓	0.000121	✓
Vendor 2	0.000121	✓	0.000121	✓
Vendor 3	0.000121	✓	0.000121	✓
Vendor 4	0.000121	✓	0.000121	✓
Vendor 5	0.000121	✓	0.000121	✓
Vendor 6	0.000121	✓	0.000121	✓
Vendor 7	0.000121	✓	0.000121	✓
Vendor 8	0.000121	✓	0.000121	✓
Vendor 9	0.000121	✓	0.000121	✓
Vendor 10	0.000121	✓	0.000121	✓
Vendor 11	0.000121	✓	0.000121	✓
Vendor 12	0.000239	✓	0.000239	✓
Vendor 13	0.000239	✓	0.000239	✓

Vendor:	Benchmark II:	Ground truth:	Alternative:	Ground truth:
Vendor 14	0.000239	✓	0.000239	✓
Vendor 15	0.000239	✓	0.000239	✓
Vendor 16	0.000239	✓	0.000239	✓
Vendor 17	0.000121	✓	0.000121	✓
Vendor 18	0.000121	✓	0.000121	✓
Vendor 19	0.000121	✓	0.000121	✓
Vendor 20	0.000121	✓	0.000121	✓
Vendor 21	0.000121	✓	0.000121	✓
Vendor 22	0.000121	✓	0.000121	✓
Vendor 23	0.000121	✓	0.000121	✓
Vendor 24	0.000121	✓	0.000121	✓
Vendor 25	0.000121	✓	0.000121	✓
Vendor 26	0.000121	✓	0.000121	✓
Vendor 27	0.000121	✓	0.000121	✓
Vendor 28	0.000121	✓	0.000121	✓
Vendor 29	0.000121	✓	0.000121	✓
Vendor 30	0.000121	✓	0.000121	✓
Vendor 31	0.000121	✓	0.000121	✓
Vendor 32	0.000121	✓	0.000121	✓
Vendor 33	0.000121	✓	0.000121	✓
Vendor 34	0.000121	✓	0.000121	✓
Vendor 35	0.000121	✓	0.000121	✓

Vendor:	Benchmark II:	Ground truth:	Alternative:	Ground truth:
Vendor 36	0.000121	✓	0.000121	✓
Vendor 37	0.000121	✓	0.000121	✓
Vendor 38	0.000121	✓	0.000121	✓
Vendor 39	0.000121	✓	0.000121	✓
Vendor 40	0.000121	✓	0.000121	✓
Vendor 41	0.000121	✓	0.000121	✓
Vendor 42	0.000121	✓	0.000121	✓
Vendor 43	0.000121	✓	0.000121	✓
Vendor 44	0.000121	✓	0.000121	✓
Vendor 45	0.000121	✓	0.000121	✓
Vendor 46	0.000121	✓	0.000121	✓
Vendor 47	0.000121	✓	0.000121	✓
Vendor 48	0.000121	✓	0.000121	✓
Vendor 49	0.000121	✓	0.000121	✓
Vendor 50	0.000121	✓	0.000121	✓
Vendor 51	0.000121	✓	0.000121	✓
Vendor 52	0.000121	✓	0.000121	✓
Vendor 53	0.000121	✓	0.000121	✓
Vendor 54	0.000121	✓	0.000121	✓
Vendor 55	0.000121	✓	0.000121	✓
Vendor 56	0.000121	✓	0.000121	✓
Vendor 57	0.000121	✓	0.000121	✓

Vendor:	Benchmark II:	Ground truth:	Alternative:	Ground truth:
Vendor 58	0.000121	✓	0.000121	✓
Vendor 59	0.000121	✓	0.000121	✓
Vendor 60	0.000121	✓	0.000121	✓
Vendor 61	0.000121	✓	0.000121	✓
Vendor 62	0.000121	✓	0.000121	✓
Vendor 63	0.000121	✓	0.000121	✓
Vendor 64	0.000121	✓	0.000121	✓
Vendor 65	0.000121	✓	0.000121	✓
Vendor 66	0.000121	✓	0.000121	✓
Vendor 67	0.000121	✓	0.000121	✓
Vendor 68	0.000121	✓	0.000121	✓
Vendor 69	0.000121	✓	0.000121	✓
Vendor 70	0.000121	✓	0.000121	✓
Vendor 71	0.000121	✓	0.000121	✓
Vendor 72	0.000121	✓	0.000121	✓
Vendor 73	0.000121	✓	0.000121	✓
Vendor 74	0.000121	✓	0.000121	✓
Vendor 75	0.000121	✓	0.000121	✓
Vendor 76	0.000121	✓	0.000121	✓
Vendor 77	0.000121	✓	0.000121	✓
Vendor 78	0.000121	✓	0.000121	✓
Vendor 79	0.000121	✓	0.000121	✓

Vendor:	Benchmark II:	Ground truth:	Alternative:	Ground truth:
Vendor 80	0.000121	✓	0.000121	✓
Vendor 81	0.000121	✓	0.000121	✓
Vendor 82	0.000121	✓	0.000121	✓
Vendor 83	0.000121	✓	0.000121	✓
Vendor 84	0.000121	✓	0.000121	✓
Vendor 85	0.000121	✓	0.000121	✓
Vendor 86	0.000121	✓	0.000121	✓
Vendor 87	0.000121	✓	0.000121	✓
Vendor 88	0.000121	✓	0.000121	✓
Vendor 89	0.000121	✓	0.000121	✓
Vendor 90	0.000121	✓	0.000121	✓
Vendor 91	0.00598	✗	Not enough data	✓
Vendor 92	0.00719	✗	Not enough data	✓
Vendor 93	0.00666	✗	Not enough data	✓
Vendor 94	0.00654	✗	Not enough data	✓
Vendor 95	0.00707	✗	Not enough data	✓
Vendor 96	0.00749	✗	Not enough data	✓
Vendor 97	0.00199	✓	Not enough data	✓
Vendor 98	0.00247	✗	Not enough data	✓
Vendor 99	0.00247	✗	Not enough data	✓
Vendor 100	0.00208	✓	Not enough data	✓

Figure 52: Classification results for one hundred vendors for benchmark II. Source: Franco Arda (2020).

For the benchmark II (H_0), we get the following results:

TP (true positive) = 5 / 5 = 1
TN (true negative) = 77/85 = 0.91
FN (false negative) = 0/85 = 0
FP (false positive) = 8/85 = 0.09

For the alternative (H_1), we get the following results:

TP (true positive) = 5 / 5 = 1.0
TN (true negative) = 85 / 85 = 1
FN (false negative) = 0 / 85 = 0
FP (false positive) = 0 / 85 = 0

Calculating the accuracy for the benchmark II resulted in:

$$\text{Accuracy}_0 = \frac{TP + TN}{TP + TN + FP + FN} = \frac{5 + 0.91}{5 + 0.91 + 0 + 0.09} = 95.50\%$$

Figure 53: Accuracy of benchmark II. Source: Franco Arda (2020).

Calculating the accuracy for the alternative, or alternative hypothesis, resulted in:

$$\text{Accuracy}_1 = \frac{TP + TN}{TP + TN + FP + FN} = \frac{5 + 1}{5 + 1 + 0 + 0} = 100\%$$

Figure 54: Accuracy of alternative against benchmark II. Source: Franco Arda (2020).

With the modified benchmark II, the accuracy is almost unaffected and increased only slightly from 95.23% (benchmark I) to 95.50% (benchmark II). What is extremely important with fraud, though, as the classes are incredibly unbalanced, is the increase of recall from 0 to 1. In other words, the benchmark II now detected all vendors committing to invoice fraud.

For the confusion matrix for the benchmark II (H_0), we get the following results:

	Predicted	
	Positive	Negative
Actual True	1	0
Actual False	0.09	1

Figure 55: Confusion matrix for benchmark II (H_0). Source: Franco Arda (2020).

For the confusion matrix for the benchmark (H_1), we get the following results:

	Predicted	
	Positive	Negative
Actual True	1	0
Actual False	0	1

Figure 56: Confusion matrix for alternative (H_1). Source: Franco Arda (2020).

The alternative resulted in the same accuracy, even though the numbers have changed slightly.

The alternative resulted in the same accuracy, even though the numbers have changed slightly. With benchmark II, the difference boils down to the false positives, or type I errors. With benchmark II, the false positives are now 0.09 vs. 0.00.

At first glance, this might not look like much, but a 0.09 or 9% reduction in false positives can amount to a large number. Let us say a company gets 100,000 invoices. With benchmark II, they get 9% or around 9,000 false alarms. As we have mentioned being problematic in other research papers, such a large number can deter from using the Benford algorithm.

Calculating the precision for the benchmark II, or null hypothesis, resulted in:

$$\text{Precision}_0 = \frac{TP}{TP + FP} = \frac{5}{5 + 0.09} = 0.9174$$

Figure 57: Precision for benchmark II. Source: Franco Arda (2020).

Calculating the precision for the alternative, or alternative hypothesis, resulted in:

$$\text{Precision}_1 = \frac{TP}{TP + FP} = \frac{5}{5 + 0} = 1$$

Figure 58: Precision for alternative against benchmark II. Source: Franco Arda (2020).

Calculating the recall for the benchmark II, or null hypothesis, resulted in:

$$\text{Recall}_0 = \frac{TP}{TP + FP} = \frac{5}{5 + 8} = 0.38$$

Figure 59: Recall for benchmark II. Source: Franco Arda (2020).

Calculating the recall for the alternative, or alternative hypothesis, resulted in:

$$\text{Recall}_1 = \frac{TP}{TP+FP} = \frac{5}{5+0} = 1$$

Figure 60: Recall for alternative against benchmark II. Source: Franco Arda (2020).

While recall has improved massively from benchmark I to benchmark II, we can see the true strength from our alternative, which can be credited to our work in the first research hypothesis: sample size.

The benchmark II now achieves a good precision of 0.9174 vs. the alternative with 1.0. Again, the true strength of the alternative is displayed in recall (number of invoice fraudsters detected): here, the benchmark II achieves a recall of 0.38 vs. the alternative of 1.00. In other words, while both algorithms catch invoice fraudsters, the alternative achieves a substantially lower false positive (type I error) rate.

	Benchmark II	Alternative
Accuracy	95.50%	100%
Recall	0.38	1.00

Figure 61: Accuracy and recall of Benchmark II vs. Alternative. Source: Franco Arda (2020).

To test our hypothesis on benchmark II, we test again for the hypothesis with alternative H_1: $\text{accuracy}_{\text{benchmark II}} – \text{accuracy}_{\text{alternative}} > 0$, we compute a p-value by finding the proportion of resamples is equal or greater to the observed test statistic.

H_0 : $\text{accuracy}_{\text{benchmark II}} - \text{accuracy}_{\text{alternative}} = 0$
H_1: $\text{accuracy}_{\text{benchmark II}} – \text{accuracy}_{\text{alternative}} > 0$
Based on the binary classification of benchmark II and alternative, we create an Excel file marking correct classification with "correct" and wrong classification with "wrong."

```
null_distribution <- accuracy %>%
  specify(formula = Classification ~ Method, success = "correct") %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "diff in props", order = c("alternative", "benchmark_II"))
```

The difference in accuracy, between benchmark II and alternative has now increased to 0.08:

```
obs_diff_prop <- accuracy %>%
  specify(Classification ~ Method, success = "correct") %>%
  calculate(stat = "diff in props", order = c("alternative", "benchmark_II"))
obs_diff_prop
# 0.08
```

The simulated results yielded an interesting finding: compared to benchmark I, the simulation-based p-value has slightly decreased to 0.02:

```
null_distribution %>%
  get_p_value(obs_stat = obs_diff_prop, direction = "right")
# 0.02
```

At first, we were surprised about the similar p-value for benchmark II as we have increased the number of vendors with invoice samples < 1,000 to 10, benchmark II misclassified several of those and thus reduce the corresponding p-value, even though it detected now all invoice fraudsters, which seems to have balanced off.

In other words, while benchmark I missed many false positives, benchmark II incurred several false positives. From a fraud analytics perspective, there is a fundamental difference between a false positive (false alarm) and a false negative (missed true fraudster), but it is not from a hypothesis perspective. For the hypothesis test, all it counts if the accuracy was correct, as defined as "correct" or not, as defined as "wrong."

We will verify those results with 1,000, 5,000, and 10,000 simulations.

A simulation-based hypothesis test based on 1,000 simulations returns the results visually:

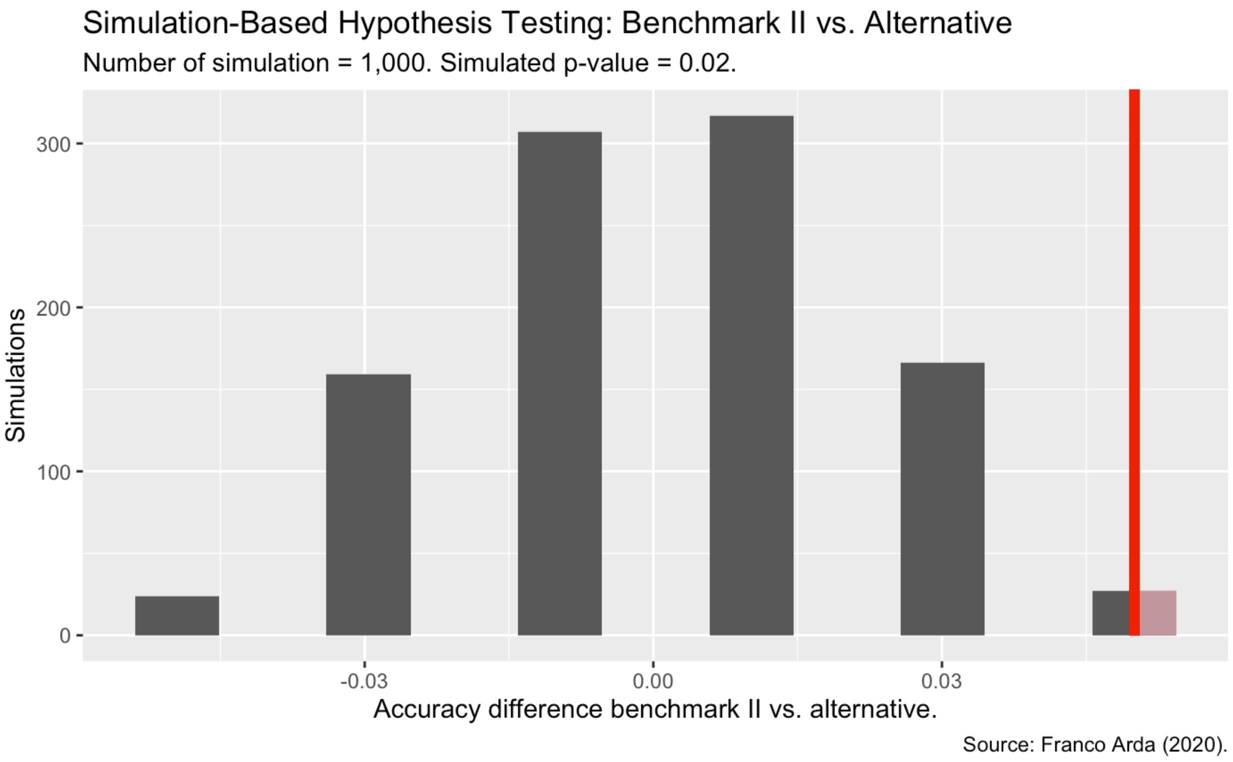
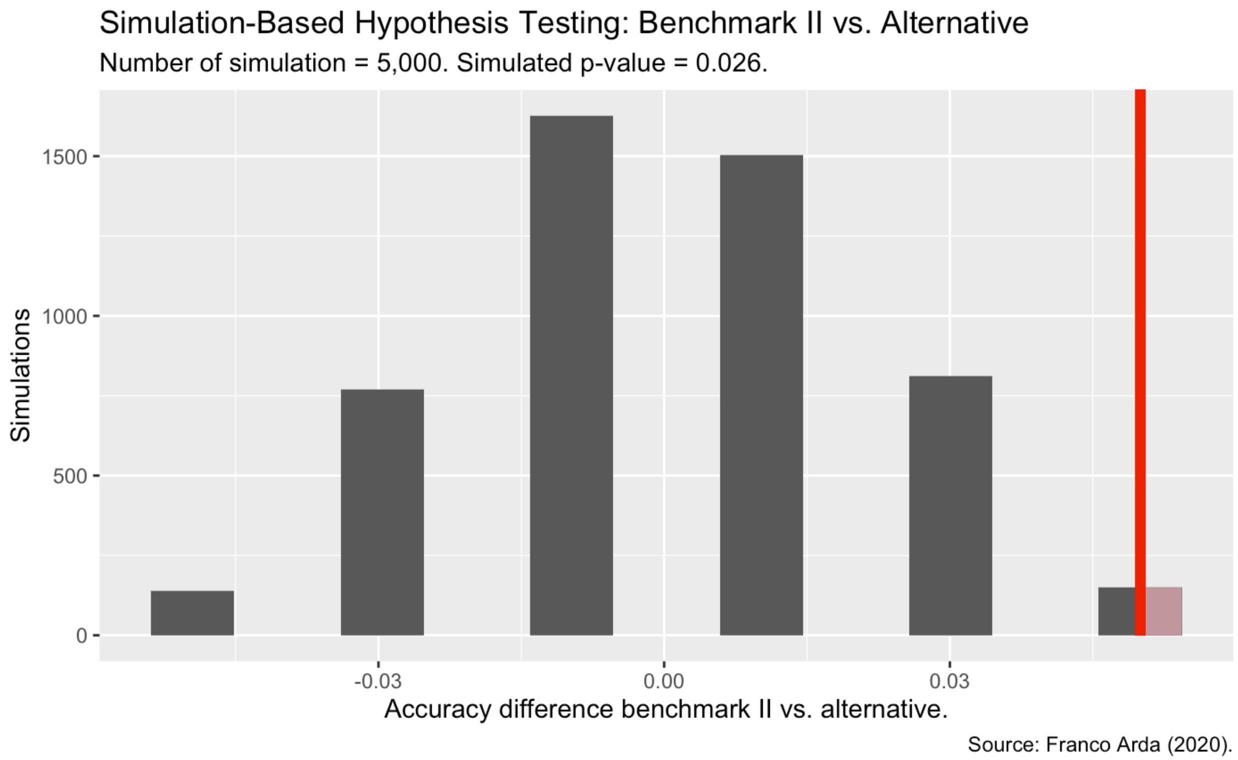


Figure 62: Simulation-Based Hypothesis Testing: Benchmark II vs. Alternative. Number of simulations = 1,000. Simulated p-value = 0.02. Source: Franco Arda (2020).

Visually, we can read bar referring to 0.08 and the chance of likelihood given that the null-hypothesis was correct.

H_0 : $accuracy_{benchmark\ II}$ - $accuracy_{alternative} = 0$

Our findings provide strong evidence that we can reject the null hypothesis. Not only do we see visually that a difference of 0.08 between benchmark II and the alternative is doubtful, but we see this also in the low p-value of 0.02.


Simulation-Based Hypothesis Testing: Benchmark II vs. Alternative

Number of simulation = 5,000. Simulated p-value = 0.026.

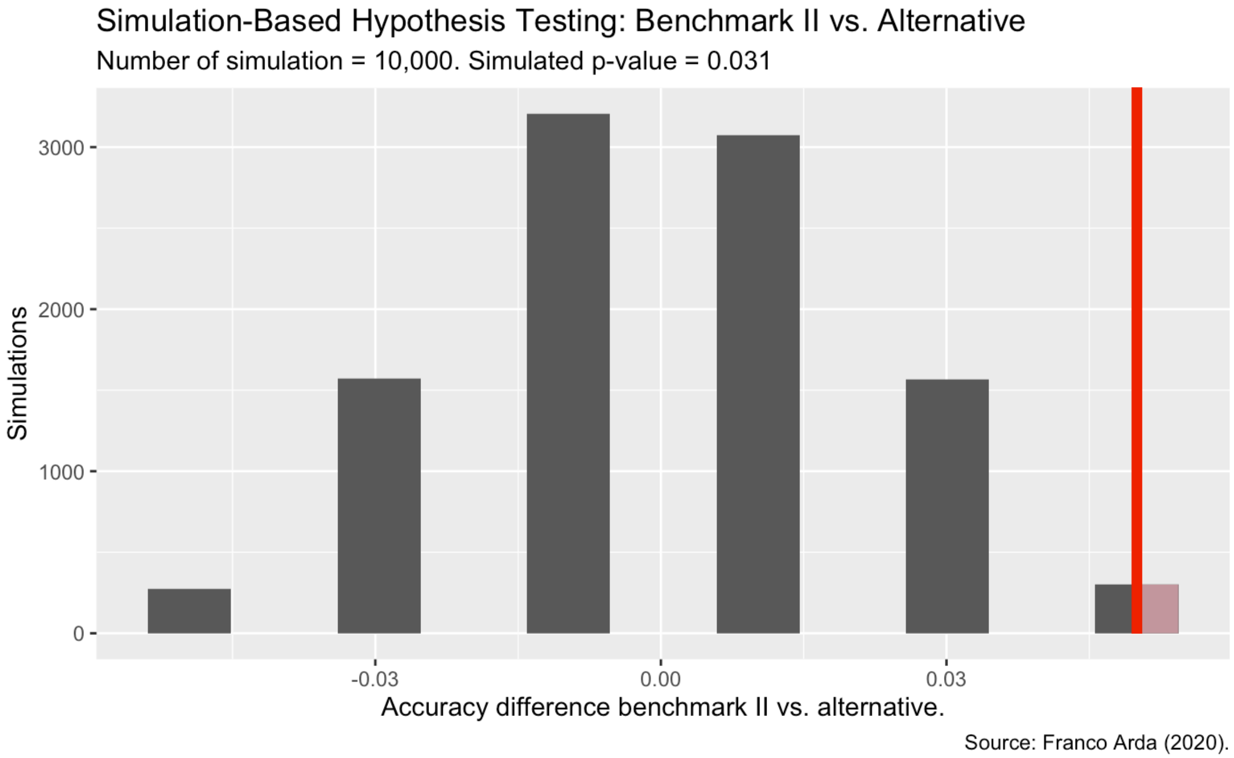
Source: Franco Arda (2020).

Figure 63: Simulation-Based Hypothesis Testing: Benchmark II vs. Alternative. Number of simulations = 5,000. Simulated p-value = 0.026. Source: Franco Arda (2020).

Our experimental findings can be compared to our previous study results, that the alternative's accuracy is statistically higher than that of benchmark II. The risk of bias or error was rated low as we tested the hypothesis with a different number of simulations and always got a p-value of less than 0.05.

The experiments provide convincing evidence demonstrating that the results are statistically significant. Contrary to our expectations, the p-value for benchmark II was almost identical to the p-values for benchmark I.

Given our statistically significant results, our findings should not be over-interpreted. As we will discuss in conclusion, all the experiments were done on a synthetic dataset with randomized data. Real-world fraudulent data in general and real-word invoice fraud data, in particular, might not offer such an exact classification. To our best abilities, the synthetic dataset presents the best possible simulation of fraudulent data.


Simulation-Based Hypothesis Testing: Benchmark II vs. Alternative

Number of simulation = 10,000. Simulated p-value = 0.031

Source: Franco Arda (2020).

Figure 64: Simulation-Based Hypothesis Testing: Benchmark II vs. Alternative. Number of simulations = 10,000. Simulated p-value = 0.031. Source: Franco Arda (2020).

The experiment provides convincing evidence, that the alternative hypothesis is statistical significantly more accurate than benchmark I and benchmark II. While we tested different number of simulations in order to verify our results, the p-values for always lower than the pre-defined p-value of 0.05:

Number of simulations:	P-value:
1,000	0.020
5,000	0.026
10,000	0.031

Figure 65: P-values for different number of simulations for the simulation-based hypothesis test of benchmark II vs. alternative. Source: Franco Arda (2020).

A vital component of the conclusion is that independently of the number of simulations, the p-values were always below 0.05.

In conclusion, we took the best performing Benford algorithm, called Benchmark II, and compared our alternative results. The alternative result is our Benford algorithm via sample size determination.

To test our hypothesis, we simulated the difference in proportion (R code "diff in props"). The accuracy of each algorithm measures the proportions. The number of simulations ranged from 1,000 to 10,000.

In other words, we combined both algorithms (permutation hypothesis test) and simulated the accuracy as there was no difference between the two algorithms. Finally, we compared the simulated accuracy against each algorithm's observed accuracy and measured how extreme the differences are (i.e., p-value).

Independent of the number of simulations, we always got p-values below 0.05. Therefore, we can say that our alternative algorithm's accuracy is statistically significant better with high confidence.

Empirical findings

The chapter of the research presents empirical findings for the current research gained after running a Monte Carlo Simulation method. According to research was done by Kabacoff et al. (Kabacoff and Girden, 2010), there are three concepts associated with empirical research: Controlled, Reliability, and Validity.

Controlled observation refers to the precision of conditions under which our data was collected. Our synthetic datasets of vendors are created by simulating invoices from vendors. Any potential risk of noise is minimized by creating some datasets strictly based on Benford's distribution. The randomized data was created in an isolated environment using a random number generator.

Hence, the controllability of data was ensured through the application of Benford’s algorithm for obtaining a random sample from the population, which was the main attribute of the current research. The dataset was tested for combining the traditional Benford laws properties with the new alternative model. The first factorization was rated with a uniform distribution of real numbers, the second was related to the sequence of real numbers and distribution, the third characterization indicated the significance of including distributions of the random variables and the fourth characterization relates with the probability distribution of the significands.

Technically, reproducing those random numbers fluctuates if replicated, but due to the large number (up to 10,000 invoices), the law of large numbers should apply, and a similar pattern should emerge, even with different random number generators seeds. These features should ensure that synthetic datasets will be objective, replicable, and reproducible.

Empirical research should be reliable, replicable, and reproducible. The notion that scientific findings can be confirmed through replication is fundamental to the centuries-old paradigm of science (Baumer, Kaplan, and Horton, 2020).

In our case, the datasets, simulations, and hypotheses should be reliable, replicable, and reproducible. More precisely, replicability means that different researchers and practitioners get the same results with different data. In this part, we believe we fully succeeded. A different person using our alternative should get the same results using their data. In other words, the alternative approach with the improved accuracy of the Benford algorithm is generalizable.

Strictly speaking, it is probably not easy to reproduce our research. For a thesis to be truly reproducible, all the code should be in one place. In other words, a researcher or practitioner using R should be able to re-run our research with only a few clicks, which is not possible.

However, what is possible is using the code snippets, loading our datasets, and reproducing our research. We made great efforts that the presented research is reliable, replicable, and reproducible. Nevertheless, if reproducible means that in a few clicks, the research is reproduced, that this is not possible. If a thesis includes all the steps conducted, including code, then our research paper is reproducible.

Validity is often referred to as the last key concept of empirical research. Regarding our research, we want to know whether the conclusions are valid: did we scientifically test the hypothesis, and is the difference meaningful?

As discussed, the expected distribution of Benford is nonparametric and does not conform to a normal distribution (i.e., is not Gaussian normal). However, from a synthetic dataset of 10,000 invoices conforming to Benford's distribution, a large enough random sampling does conform the CLT (Central Limit Theorem) based on the mean average deviation.

In other words, if we randomly sample 100 (n = 100) invoices from the population (i.e., the 10,000 invoices), calculate the mean average deviation for each sample, and simulate this process 4,000 times (k = 4,000), then we get a histogram that resamples a normal distribution (see figure below).

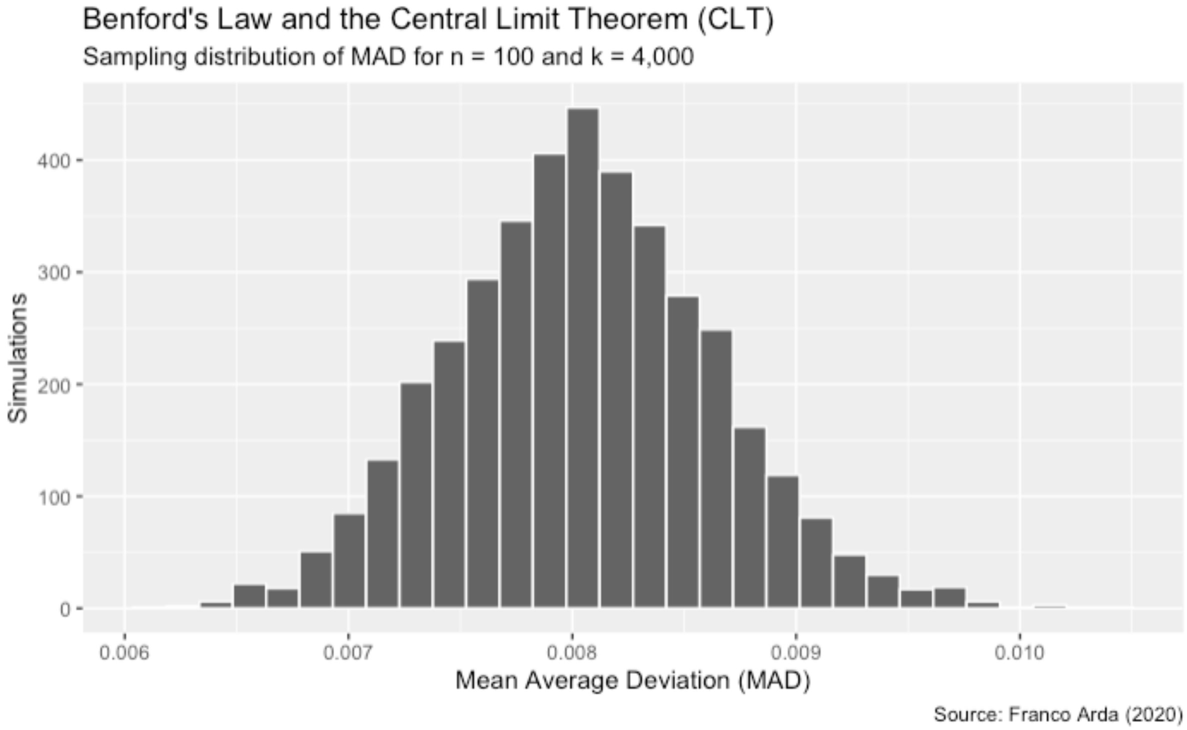


Figure 66: *Benford's law and the Central Limit Theorem with n = 100. Source: Franco Arda (2020).*

This empirical finding was unexpected as the Benford distribution is highly skewed to the right. We were extremely surprised that a Monte Carlo simulation produced a normal distribution of the mean average deviation from a Benford distribution. The unexpected normal distribution for the accurate sample size further challenges the Benford theorem approach and takes the attention of the researcher towards the erroneous approaches highlighted in the literature. It shows that small sample size does not always result in a skewed distribution of data set while the large sample size is found related with the normally distributed data set as presumed under the Central

Learning Theorem too. With adequate algorithmic modelling, the researchers can also reduce the order of magnitude, exponential sequence, and regularity of the distribution without increasing the sample size relatively. It is not always needed to have a random variable covering several magnitude orders for appropriate propagation of the data. In this context, the findings of the current research have attempted to bridge the gap related to the mathematically erroneous approaches related to Benford's law collectively.

The following empirical finding is based on our previous research: with a higher number of simulations (i.e., 1,480), we got a confidence interval of 90%. In other words, by randomly sampling from the population (i.e., a synthetic dataset of 10,000 invoices), we can be 90% confident, that the sample represents the population with a sample size of 1,480. These findings have great implication for understanding how the increased data (n =100) with an increased number of simulations (K = 4000) can assist the researcher in gaining the normally distributed datasets without any deviation of the data points from their center values. This confirms that the leading two-digits datasets also work similarly to the first digit dataset. As with the increasing the number of simulations and sample size, there is a significant decline in the skewness of the data along with the decreased order of magnitude leading to normal distribution.

In the figure below shows visually the Monte Carlo simulation, with 4,000 simulations:

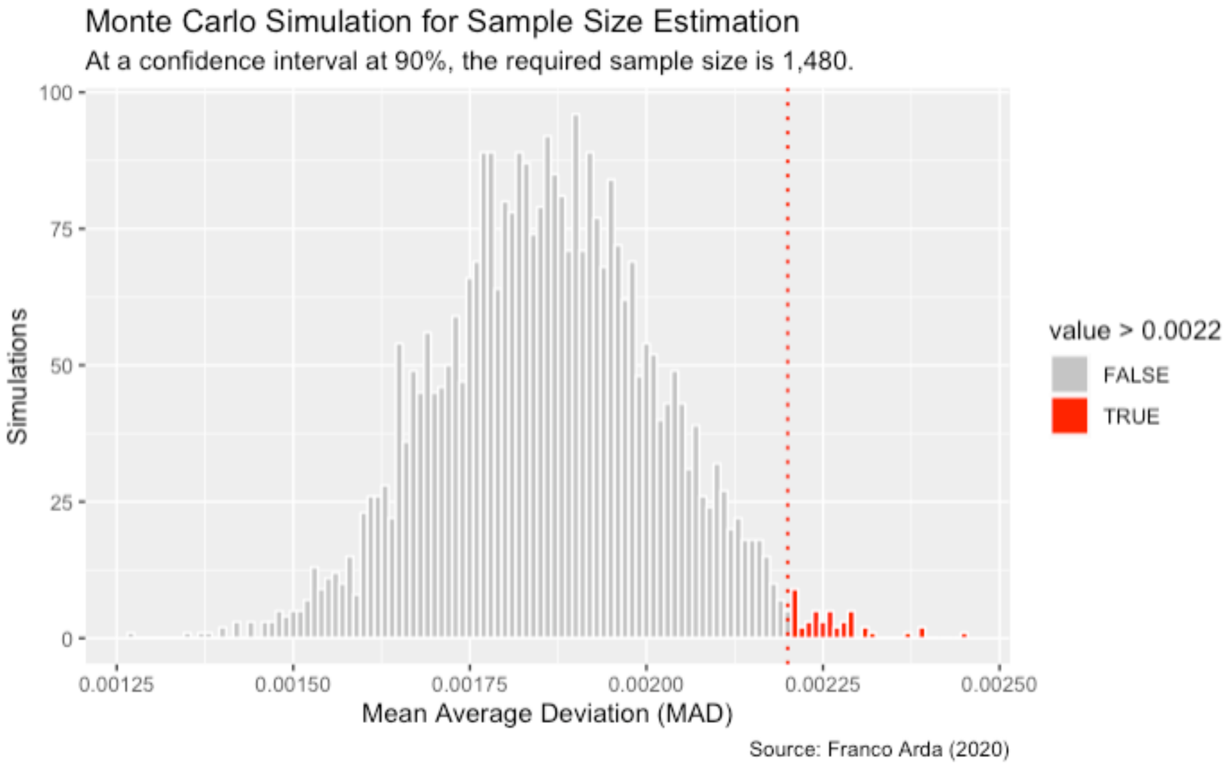


Figure 67: Based on a Monte Carlo method, with 4000 simulations, the required sample size assesses to 1480, at a 90% confidence interval. Source: Franco Arda (2020).

We started each empirical simulation at 1,000 random samples and increased it until the distribution was normally based on the Shapiro-Wilk normality test at a p-value of < 0.05. In other words, the Shapiro-Wilk normality test defined, when we would stop increasing the number of simulations. Increasing the number of simulations comes at a price: a higher number of samples. However, as our goal was to get the smallest number of samples possible, the Shapiro-Wilk normality test defined the upper limit of necessary simulations. Additionally, this statistical test also quantified our process which should make our research reproducible. In this regard, the dataset showed the value > 0.0022, which can be regarded as the fraudulent.

To compare the traditional Benford algorithm against our improved method, we tested the accuracy of the algorithms against a benchmark I and II. The accuracy of the binary classification algorithms was based on accuracy and recall.

Accuracy is a key measure for the assessment of precision in the data values in a large dataset, which is extremely important for carrying out statistics and probability-based analysis of the data set.

The recall is a key to measure the effectiveness of a model in memorizing or recollecting the trends of digits or numbers appear in a dataset. Such recall ability ultimately allows easy detection of the fraud, when any of the digits matches with the fraudulent criteria set in the specific algorithm. Hence, the three models were tested for their precision and recallability to justify the choice of the current algorithm over the previous benchmark models. In the table given below, it can be depicted that Alternative has shown relatively higher accuracy and recall percentages such as 100% and 1.

Contrarily, the earlier benchmarks have shown lower accuracy and recall like 95.23% and 95.50% with 0 and 0.38. It means that the alternative model is capable of reducing the margin of error is up to 0%, ultimately increasing the reliability and validity of the findings for the researchers. the main purpose behind the new proposed alternative model is to reduce the estimated time for detecting fraudulent invoices. By reducing the number of samples with accuracy for different types of intervals required to detect the fraud, the model has simultaneously successfully achieved the stated main goal.

To test our hypothesis, we used a permutation hypothesis test. As we can see in the following table, our Alternative algorithm outperformed the Benchmark I and Benchmark II:

	Benchmark I	Benchmark II	Alternative
Accuracy	95.23%	95.50%	100%
Recall	0	0.38	1

Figure 68: Comparing the accuracy for the benchmark I, benchmark II, and alternative. Source: Franco Arda (2020).

In both comparisons, the research hypothesis was supported at a p-value of 0.05. The accuracy for benchmark I and benchmark II were surpassed by the alternative at a statistically significant level. Based on previous research, to our knowledge, we have never seen comparisons with a benchmark. In this regard, the findings of the current research indicate a new value-added to the field of science, mathematics and statistics. Due to the exploratory nature of the current research, no comparable data to our study is present in the literature.
Therefore, it is impossible to relate our findings to previous studies. What is possible to conclude, though, is that the accuracy of the alternative is statistically significantly higher than the compared benchmarks.

However, it is important to identify the weaknesses associated with the alternative model for future researchers. In this regard, one challenge that might occur is if a statistical significance difference is meaningful? We got a difference of 4.77% (100% - 95.23%). In the second experiment, we got a difference of 4.50% (100% - 95.50%). To be conservative, we are taking a smaller difference of 4.50% as a research result. The difference might "feel" small, but this difference in accuracy can amount to a large difference.

For example, in the context of a fraud detection system at a company, processing 100,000 invoices yearly, the difference shows. A difference between the two algorithms inaccuracy of 4.50% requires the related company to deal with 450 (4.5% of 10,000) false positives. If it takes an accountant of fraud examiner half a day to deal with each detection, the 4.50% difference can result in one year of work. On the contrary, the same fraud detection system at the company processing the same invoices on animal pieces may not detect any false positive if the theme account into fraud examiner takes half a day to deal with each dataset in the detection.

Hence, the empirical findings in the current research have informed about the method that can be applied to reduce the sample size accurately yet applying a large number of simulations to the dataset respectively. These findings have confirmed that large datasets are not needed for detecting fraudulent transactions through simulations. Rather, by approximating the required sample size can help the researchers in eliminating the need for predefined distribution. Such sample size is also important to deal with the weaknesses of traditional statistics related to wastage of scarce resources like money as well as intangible resources like time.

Hence, Monte Carlo Simulation methods can be used for conducting several thousand simulations at a single point of time necessary for approximating the required sample. Thus, the alternative model can be applied to the Monte Carlo Simulation method to reduce the concerns and efforts of future researchers. These findings have further enhanced the effectiveness of random sampling in scientific mathematical research empty to identify fraudulent values. It can assist in simplifying the randomization process by highlighting the accurate sample size needed to allow each of the participants in the population equal chance of participation by running an approximate number of simulations. The traditional model does not provide accuracy of sample size rather it leaves at the discretion of the researcher to identify the appropriate sample size needed for estimation of

Discussion

To our knowledge, this was the first research that successfully quantified the sample size required for the Benford distribution.

We are incredibly pleased with this achievement. At the beginning of writing this thesis, the hypothesis was clear; can we determine the sample size required at a given confidence interval? The hypothesis was clear. What was not clear was how to calculate the sample size, and we struggled a lot.

For example, we can now state, "given a confidence level of 95%, we required 1,532 samples." Based on our experiments, the effects are substantial. Compared to other research approaches, we could improve the accuracy of our approach by 4.50%. The improvement in the accuracy was solely due to the reduction of false positives (false alarms).

As stated in the introduction of this thesis, one of the major complaints from practitioners such as accountants and fraud examiners is that the Benford algorithm often produces too many false positives.

For companies monitoring fraud in general and monitoring invoice fraud in particular, those research results might be significant. The ability to detect fake invoices as early as possible is of immense importance for companies. By knowing the required sample size, a company can monitor potential invoice fraud more frequently, instead of once a year when there are "enough" samples, and might detect fraud before they reach multimillion-dollar levels.

Technically, the breakthrough in determining the sample size required was combining simulation with a statistical normality test, as seen in figure 65 below.

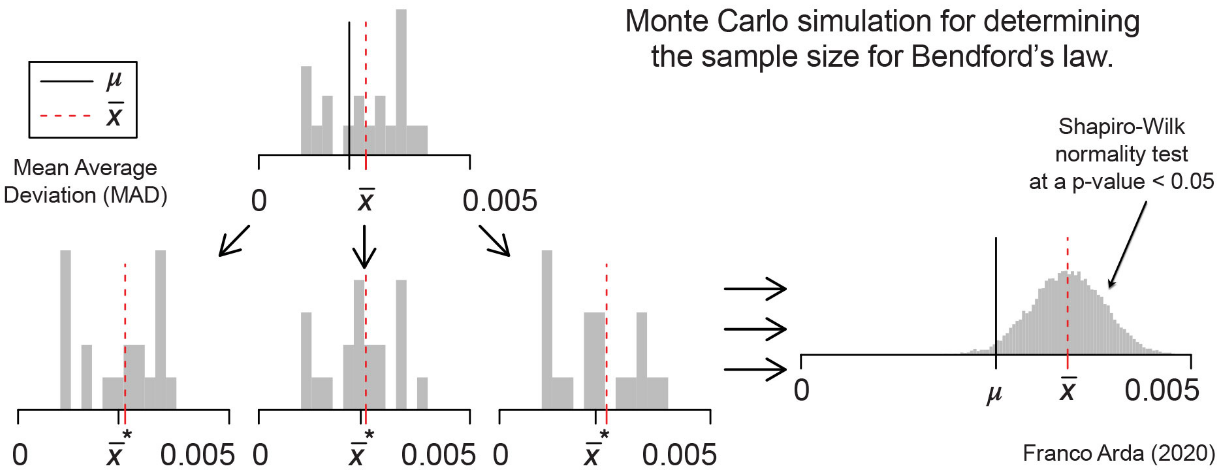

Monte Carlo simulation for determining the sample size for Bendford's law.
μ
$\bar{x}$
Mean Average Deviation (MAD)
0
$\bar{x}$
0.005
0
$\bar{x}^*$
0.005
0
$\bar{x}^*$
0.005
0
$\bar{x}^*$
0.005
Shapiro-Wilk normality test at a p-value < 0.05
0
μ
$\bar{x}$
0.005
Franco Arda (2020)

Figure 69: Breakthrough approach in determining the sample size required for Benford's law. Source: Franco Arda (2020).

The final approach was unexpected. From early on, we knew that the simulation required a kind of Monte Carlo simulation. Only a Monte Carlo simulation was capable of dealing with the nonconform distribution of Benford's Law. In other words, the Monte Carlo simulation does not require a specific distribution from the underlying, which is mathematically known as non-parametric.

The surprising part was that we expected the solution to be a bootstrap method. Bootstrap belongs to the Monte Carlo simulation. Intellectually, it made sense that we could narrow down the required confidence interval with a bootstrap. Unfortunately, this bootstrap approach did not work at all. Instead, what worked was using a plain-vanilla Monte Carlo simulation combined with a normality test.

We created a synthetic dataset with 10,000 variables and calculated the mean average deviation. This dataset acted as the population. Afterward, we randomly sampled from the population, starting at a low sample size number of 100, repeated the simulation several thousand times, and calculated the mean average deviation for the sample.

The histogram from the simulation allowed us to calculate the required confidence intervals of 90%, 95%, and 99%. The number of simulations was determined by a normality test called the Shapiro-Wilk normality test at a p-value of < 0.05. In other words, we increased the number of simulations (not samples) until the distribution was normal. In statistics, this effect is called CLT (Central Limit Theorem). With a higher number of simulations, the required sample size increased slightly. However, because each simulation was only valid if it was determined normally distributed, the results were stable in terms of confidence intervals

Our study's potential limitation might be our comparison of the traditional Benford algorithm with the improved version. We assumed that 5% of the dataset was fraudulent. Based on this assumption, we got an accuracy difference of 4.50% in favor of the Benford algorithm via sample size determination. However, with a different percentage of fraudulent cases in the dataset, we most likely would have gotten a different accuracy difference. We intentionally kept the potential variable change as low as possible, but the following scenarios might occur:

- A lower proportion of fraudulent cases in the dataset would have decreased the difference in accuracy between the traditional Benford algorithm and the improved algorithm. In other words, the alternative Benford algorithm would have still outperformed, but by a smaller amount.

- A lower proportion of "too tidy datasets" would have decreased the difference in accuracy between the traditional Benford algorithm and the improved algorithm. Again, the alternative algorithm would have still outperformed, but by a smaller amount.

- In relative terms, a higher proportion of fraudulent cases in the dataset would not have made any difference between the two algorithms.

- A higher proportion of "too tidy datasets" would have increased the difference between the two algorithms. With real-data, this scenario would most likely occur. In other words, the difference between the accuracy of the two algorithms would have been > 4.50%.

We believe that those different scenarios show the potential limitations of our study. Striking the "right" balance in creating a synthetic dataset is difficult. Our goal was to err on the side of caution. It would have been easily possible to show a more considerable difference in the accuracy of 4.50%. However, we wanted to be cautious. Additionally, in predictive modeling, an accuracy difference of 4.50% is enormous. In machine learning competitions such as on Kaggle, competitors often strive to improve an algorithm's accuracy by less than one percent

Another unexpected turn in this research paper was that we had to use a confusion matrix to emphasize the algorithmic approaches' differences. With unbalanced datasets in general and fraud datasets in particular, the accuracy of an algorithm is often not sufficient.

For example, if a dataset has one hundred variables, and one is fraudulent, then an algorithm that misses the fraudulent variable has the same accuracy as the one that does not miss it. In other words, both algorithms get an accuracy of 99%, while the second algorithm is superior to the first.

The results provided convincing evidence that we needed to use different measures of algorithmic success. In our case, one of the most critical measures was recall, in medical or bioresearch, often known as sensitivity. Recall measures how good an algorithm (or test) is at detecting true positives. In other words, an algorithm might have high accuracy, but if it does not flag fraudulent transactions, the algorithm is not useful.

As we have highlighted the potential limitations of our study, we would like to highlight potential suggestions for further research. For some companies, our determined sample size might be too large. In future studies, one possible approach is to reduce the sample size required. One possible approach might be to use our statistical techniques and apply it to only the Benford algorithm's first digit.

The first-digit Benford algorithm takes only the first digits 1 to 9 into account. The first-digits observe a much smaller range for the first-two digits algorithm, which observes the digits 10 - 99. As the first-digit algorithm observes a much smaller digits range, the required sample size is most likely smaller as well.

As discussed, smaller sample size can be beneficial for companies monitoring fraud. The only question is if the first-digit algorithm can deliver the same accuracy as the first-two digit algorithm. If we could get the same or a similar accuracy for the first-digit algorithm and sample size required below 1,000, this study could be very fruitful for further research.

Conclusions

This research aimed to identify whether we could improve the Benford algorithm's accuracy through sample size determination. Based on our empirical research, our findings show strong evidence that with defined sample size, we can increase the Benford algorithm's accuracy.

To identify the sample size required, we created a synthetic dataset, randomly selected samples via a Monte Carlo simulation, and increased the number of simulations until the sample size was normally distributed based on the Shapiro-Wilk normality test with a p-value of 0.05. The empirical results indicate that we can increase the accuracy of the Benford algorithm by 4.50%.

Answers to the research question

The first research hypothesis was whether we could determine the sample size required at a given confidence interval. From a technical point of view, this was the most challenging part. Because the expected Benford distribution does not conform to any known distribution, determining a sample size is challenging. In other words, if the distribution had been normal or Gaussian, determining the sample size would have been straight-forward.

However, because the Benford distribution is non-normal and skewed to the right due to the lower frequency of the expected higher digits, this was the biggest challenge. Few if any statistical frameworks exist to determine the sample size required for non-normal distribution. One approach we first were following is via a bootstrap. With a bootstrap, we resample invoice samples from a population. In our example, the population was a synthetic dataset of 10,000 invoices conforming to Benford's distribution. Based on the confidence interval, we should have been able to estimate the required sample size.

Unfortunately, this approach was not successful. The confidence interval based on the bootstrap was not stable enough to determine the sample size. In our second attempt, we randomly sampled from the population, calculated the mean average deviation for each sample, and increased the number of simulations until the sample was normally distributed. This approach was finally successful because sample distribution followed the CLT (Central Limit Theorem). As the Benford distribution is highly skewed to the right, we were quite surprised that the Benford distribution resampling followed the CLT. Our observation confirms (Tilde, 2016): “The most fundamental theorem in mathematical statistics, the central limit theorem, can also be shown by simulation.”

The second research hypothesis focused on whether we could improve the Benford algorithm's accuracy by using a sample size determination. Against our alternative, we used two other Benford approaches we named benchmark I and benchmark II. In other words, we measured the alternative approach, using sample sizes against two benchmarks.

With the alternative algorithm using sample size, we achieved a 4.50% higher accuracy than benchmark I and benchmark II. We could achieve substantially better accuracy by having a reduced false positive rate. In other words, or type I error rate (false positives) was substantially lower. The difference in accuracy amounted to 4.50% in favor of our alternative at a p-value of 0.05.

In a permutation hypothesis test, we simulated the different performances 1,000 times and evaluated the results against a p-value of 0.05. This simulation-based hypothesis test is the "modern way" (Downey, 2016), where we again use a Monte Carlo simulation to receive the p-value in a hypothesis test. Statistically speaking, our alternative algorithmic approach outperformed the benchmark in every simulation.

Reflections on the research

While reflecting on writing this thesis, we realized that we genuinely love the scientific process. At first, it was the observation that nobody considers the sample size when using the Benford algorithm. This observation did not come while reading about the Benford algorithm but by implementing it in code. As data scientists, we have often observed that implementing an idea in code is often fundamentally different. It is often the case that a seemingly complicated statistical method can be easily integrated into code.

On the other hand, the Benford algorithm looks relatively simple. However, when we try to implement the Benford algorithm in code, we often realize the difficulty in the nuances by analyzing millions of data. For example, we realized that no research paper mentions the sample size required for the Benford algorithm. Such "details" are often easy to miss until we have to code the instructions in a computer program explicitly. In other words, when we code the algorithmic details, we require each step.

When we coded the Benford algorithm, we realized that we needed the sample size for a successful program to run. For example, we had datasets of only a handful of variables. We cannot evaluate the conformity to the Benford distribution based on five or ten entries. This observation leads us to raise the following hypothesis: can we determine the sample size required at a given confidence interval? In a scientific process, what followed was the experiment. Based on current research, we could not find any method for determining the sample size for a complex distribution such as Benford. Benford's complexity lies in the fact that the distribution is highly skewed and that the data is categorical. The Benford data is categorical as there are bins: 10 – 99. What followed was several experiments to determine the sample size required for Benford's Law.

Because Benford's Law distribution is highly skewed, we cannot apply a traditional sample size determination. Traditional statistical tools such as power required the distribution to be normally distributed. However, the Benford distribution is not normally distributed and nonparametric. We iterated over several approaches simulating the sample size. Choosing the right simulation technique is quite challenging as there is no general guidance or method for simulation.

The choice of the right simulation technique depends on the underlying problem, data set, and study's aim. After several iterations, we found a way to determine the sample size required by combining a Monte Carlo simulation, sampling from the population via mean average deviation, and testing the normality sample. Finally, the scientific method was complete, and we had a scientific theory on how to determine the sample size required for Benford's Law.

Recommendations and future research

To better understand these results' implications, future studies might be able to reduce the required sample size for the Benford algorithm even further. For example, future researchers might test the hypothesis, whether the first-digit algorithm requires fewer samples while providing the same accuracy. In general, Benford's research (Nigrini 2013 and Nigrini 2020) found that the first-two-digit algorithm provides more stable results than the first-digit only algorithm.

The more stable results can be accredited to the higher number of digits observed. With the first-two digit algorithm, we compare the actual distribution of the first digits of 10 – 99 to the Benford distribution. By definition, the higher number of bins (e.g., 10, 11 …) requires a higher number of samples until the sample reflects the population. On the contrary, with the first-digit algorithm, a smaller number of digits are compared to the expected distribution. Technically, the smaller number of required digits should result in a smaller sample size requirement. Provided the first-digit algorithm could achieve the same or similar accuracy as the first-two-digit algorithm, practitioners might detect fraudulent transactions even earlier. Such a research paper could add even more scientific and practical knowledge.

Thesis contribution to the current scientific knowledge

Our research is probably the very first attempt to quantify the sample size required for Benford's distribution.

Researchers and practitioners can state eloquently and statistically: "given the sample size of X, we are Y% confident that the sample should reflect Benford's distribution."

From a technical perspective, we added scientific knowledge by showing how to determine the sample size for nonparametric distribution, such as the Benford's distribution. We created a synthetic dataset that conformed to Benford's distribution; the population. We randomly sample with a Monte Carlo simulation based on the conformity measure mean average deviation from the population. We started with a low number of simulations of one hindered and increased the simulations (and therefore the sample size) until the sample size was normally distributed. We quantified normally distributed with the Shapiro-Wilk normality test at a p-value of 0.05. This Monte Carlo simulation returned the proportion of simulation below the mean average deviation. This proportion served as the confidence interval. We increased the number of samples until we got the desired confidence intervals of 90%, 95%, and 99%.

The implications are not minor: with the sample size defined, companies have the last piece of the Benford puzzle to implement the algorithm in real-time. Additionally, as we have shown, the Benford algorithm's accuracy can be increased substantially, given that the required sample size at a given confidence level is known.

The highlight of our study's overall significance is probably a more accurate Benford algorithm via sample size determination. The higher accuracy of our model of 4.50% compared to the benchmark was purely due to the reduction in false positives. Based on other researchers' research, one problem with the Benford algorithm is the sometimes high number of false positives (or false alarms). Our study might help increase confidence in Benford's algorithm for practitioners, such as accountants and fraud examiners.

Our empirical study revealed for the first time the exact sample size required for the Benford algorithm:

- At a 90% confidence interval, we need a sample size of 1,480.
- At a 95% confidence interval, we need a sample size of 1,532.
- At a 99% confidence interval, we require a sample size of 1,670.

These findings of the empirical study have informed about the importance of using the exact sample size to assess the accuracy of the data set. Generally, it is believed that different datasets behave differently based on the differences in the leading digit.

The same differences also impact their logarithmic plot over several orders of magnitude. In the current research, by identifying the accurate sample size for the different confidence intervals to be tested in scientific investigations, the researchers had freed future researchers from complexities to deal with such dataset differences respectively.

Besides accuracy assessment, the findings of the empirical study will also increase the confidence of the researchers when using the sample size for assuring representability of their datasets e.g. 1480 invoices at 90% confidence interval can represent a population of a synthetic dataset of 10,000 invoices. This will have great implications for the researchers in different fields such as natural science, finance and data management when they are using the large synthetic datasets of invoices or measurable transactions. They would not need to waste their time in assessing the accuracy of the sample before the application of the Bedford theorem.

Consequently, to be able to detect the fraud as early as possible and keep the losses to fraud as low as possible in real-time monitoring. Hence, using the finding the current research, the future researchers would be able to select the sample size based on the specific choice of confidence interval and margin of error in their investigations. It can also be analyzed that there is very minimal variation in the sample sizes obtained from the set of three confidence intervals. Low variation ultimately predicts the accuracy of the method used for estimating the accurate sample size via Monte Carlo Simulation.

The two major discoveries of the current research would therefore be stated as follows:

- Without the sample size determined, the Benford algorithm produces many false positives based on current research.

- With the sample size determined, the algorithm predicts fraudulent data (e.g., randomized invoice amounts) with 100% accuracy.

With these conclusions, the current research has attempted to make significant value addition in the existing literature studies about the Bedford Law. This will also allow future researchers to compare the accuracy of their alternative model against the benchmarks that were not done in any earlier studies before current research. The differences in accuracy, recall and representation of the dataset will allow the future investigators to reap the benefits of evaluation by reducing the sampling error necessary for detecting fraudulent activities are the datasets items with false positive outcomes.

References

American Statistical Association, 2016.
https://www.amstat.org/asa/files/pdfs/P-ValueStatement.pdf

Barney, B. J., and Schulzke, K.S. (2017). Moderating "Cry Wolf" Events with Excess MAD in Benford's Law Research and Practice.

Baumer, B.S., Kaplan, D.T., and Horton, N.J. (2017). Modern Data Science with R.

Beebe, N.H.F. (2020) A Bibliography of Publications about Benford's Law, Heap's Law, and Zipf's Law.

Benford, F. (2020). Fourier analysis and Benford random variables

Benton, D. J. (2018). Monte Carlo Simulation: The Art of Random Process Characterization.

Berger, A. and Hill, T.P. (2015). An Introduction to Benford's Law. Princeton University

Berger, A. and Hill, T.P. (2020). The Mathematics of Benford's Law – A Primer.

Blitzstein, J., and Hwang, J. (2019). Introduction to Probability (2nd edition).

Blondeau de Silva, S. (2019). BeyondBenford: An R Package to Determine Which of Benford's or BDS's Distributions is the Most Relevant.

Bruce, P., Bruce A., and Gedeck, P. (2019). Practical Statistics for Data Science (2nd edition).

Carsey, T. and Harden J. (2013). Monte Carlo Simulation and Resampling Methods for Social Science.

Cerioli, A., Barabesi, L., Cerase, A., Menegatti, M., and Perotta, D. (2019). Newcomb-Benford law and the detection of frauds in international trade (page 110).

Cinelli, C. (2019). Benford Analysis R Package on GitHub.

CNBC, (2019). Facebook got tricked out of $100 million.
https://www.cnbc.com/2019/03/28/how-to-avoid-invoice-theft-scam-that-cost-google-facebook-123m.html

Cole, M. A., Maddison, D. J., and Zhang, L. (2019). Testing the emission reduction claims of CDM projects using Benford's Law.

Collins, J.C. (2017). Using Excel and Benford's Law to detect fraud. Journal of Accountancy.

Demir, B. (2020). Trade policy changes, tax evasion, and Benford's law.

Downey, A. (2016). There is only one test.
http://allendowney.blogspot.com/2016/06/there-is-still-only-one-test.html

Durtschi, C., Hillison, W., and Pacini, C. (2004). The Effective Use of Benford's Law to Assist in Detecting Fraud in Accounting Data.

Efron, B., and Tibshirani, R.J. (1993). An introduction to the bootstrap.

Few, S. (2019). The Perceptual and Cognitive Limits of Multivariate Data.

Frefix, B. (2018). Detection of credit card fraud.

Gauvrit, N., Houillon, J. C., and Delahaye, J. P. (2017). Generalized Benford's Law as a Lie Detector

Goldman, J.L.F., and Perez-Mercader, J. (2016). Ubiquity of Benford's law and emergence of the reciprocal distribution.

Gomez-Camponovo, M., Moreno, J., Idrovo, A. J., Paez, M., and Achkar, M. (2016). Monitoring the Paraguayan epidemiological dengue surveillance system (2009-2011) using Benford's Law

Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep Learning (Adaptive Computation and Machine Learning). Chapter 17: Monte Carlo Methods. Page 592.

Goodman, W. (2016). The promise and pitfalls of Benford's law. Royal Statistical Society.

Grabowski, M. (2016). How many more? Sample size determination in studies of morphological integration and evolvability.

Guttag, J.V. (2016). Introduction to Computing and Programming using Python (second edition). The MIT Press (Massachusetts Institute of Technology).

Hegazy, M, Median, A., and Ragaie, M. (2016). Enhanced fraud miner: credit card fraud detection using clustering data mining techniques.

Heilig, F., and Lusk, E. J. (2019). Testing the Small Size Effects for Benford Screening: The False Negative Signaling Error.

Hesterberg, T.C., and Chihara, L.M. (2019). Mathematical statistics with resampling and R (second edition).

Hidayat, T. and Budiman, A. I. (2020). Analysis benford law for company evidance from Indonesia

Holmes, S., and Huber, W. (2019). Modern Statistics for Modern Biology

Ioannidis, J. P. A. (2019). What have we (not) learnt from millions of scientific papers with p values?

Iorliam, A. and Tirungari, S (2016). "Flow Size Difference" Can Make a Difference: Detecting Malicious TCP Network Flows Based on Benford's Law.

Insurance Information Institute (2020). Background on: Insurance Fraud. https://www.iii.org/article/background-on-insurance-fraud

Irizarry, R.A. (2019). Data Science: Data Analysis and Prediction Algorithms with R.

James, G., Witten, D., and Hastie, T. (2013). An Introduction to Statistical Learning: with Applications in R (Springer Texts in Statistics).

Kabacoff, R., and Girden, E. (2010). Evaluating Research Articles from Start to Finish.

Kaiser, M. (2019). Benford's Law as an indicator of survey reliability - can we trust our data?

Kak, S. (2018). Variations on the Newcomb-Benford Law.

Kenton, W. (2019). Value at Risk (VaR).

Kim, A., and Ismay, C. (2018). Statistical Inference via Data Science.

Koesters, N., McMenemy, A., and Belanger, Y. (2020). Simulating epidemics with SIRD model and testing with Benford's Law

Kossovsky, A. E., 2019. Arithmetical Tugs of War and Benford’s Law, s.l.: arxiv.org.

Kroese, D.P., Brereton, T. T., and Botev, Z.I. (2014). “Why the Monte Carlo Method is so important today.”

Kroese, D. P., Taimre, T., and Botev Z.I. (2011). Handbook of Monte Carlo Simulations. John Wiley & Sons, Inc.

Kwak, S. G., and Kim, J. H. (2017). Central limit theorem: the cornerstone of modern statistics.
Larsen, J. E. (2017). Benford's Law and earnings management detection: the case of REITs (page 784)

Lee, K.B., Han, S., and Jeong, Y. (2020). COVID-19, flattening the curve, and Benford's Law.

Linebach, J. A., Tesch, B. P., and Kovacsiss, L. M. (2014). Nonparametric statistics for applied research.

Luo, P., and Li, Y. (2018). A new quantity for statistical analysis: "Scaling invariable Benford distance."

Luque, A., Carasso, A., Martin, A., de la Herras, A. (2020). The impact of class imbalance in classification performance metrics based on the binary confusion matrix.

Marif, M. Y., Jalal, A. A. A., Satar, N. S. M., and Samah, M. A. A. (2020). Detecting ERP data fraud using the first digits formula of Benford's Law (page 442)

Martino, L., Luengo, D., Miguez, J. (2018). Independent Random Sampling Methods (Statistics and Computing). Springer.

Matakovic, I. C. (2019). The empirical analysis of financial reports of companies in Croatia: Benford distribution curve as a benchmark for first digits (page 94)

Mebane, Walter R. (2020). Inappropriate applications of Benford's Law regularities to some data from the 2020 presidential election in the United States.

Meitner, M. (2019). Judging Steinhoff by Benford's Law (and its big limits.)

Miller, S. (2015). Benford’s Law: Theory and Applications.

Nigrini, M. (1997). The use of Benford’s Law as an aid in analytical procedures. https://search.proquest.com/docview/216734639?pq-origsite=gscholar&fromopenview=true

Murphy, K.P. (2012). Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning). Page 192.

Nguyen, T. T., Duong, C., and Nguyen, N. (2020). Benford's Law, earnings management, and accounting conservatism: The UK evidence (page 25)

Nigrini, M. (2020). Forensic Analytics: Methods and Techniques for Forensic Accounting Investigations.

Noorullah, A. S., Jari, A. S., Hasan, A. M., and Flayyih, A. M. H. (2020). Benford Law: A Fraud Detection Tool Under Financial Numbers Game

Moreno-Montoya, J. (2020). Benford's Law with small sample size. A new exact test useful in health sciences during epidemics
Özevin, Ö. Ü., Yücel, R., and Öncü, M. A. (2020). Fraud detecting with Benford's Law: an alternative approach with BDS and critic values (page 113)

Ott, R.L., and Longnecker, M. (2016). Statistical Methods & Data Analysis (seventh edition).

Pan, H., Liu, S., and Yuan, Y. (2018). Sample size determination for mediation analysis of longitudinal data.

Peng, R. D. (2011). Reproducible research in computational science.

Peng, S., 2019. Law of large numbers and central limit theorem under nonlinear expectations. Probability, Uncertainty and Quantitative Risk, 4(4).

Reinhart, A. (2015). Statistics Done Wrong: The Woefully Complete Guide.

Rizzo, M. L. (2019). Statistical Computing with R.

Sander, G., Senn, S. J., Rothman, K.J., Carlin, J.B., Poole, C., Goodman, S. N., and Altman, D. G. (2016). Statistical tests, p values, confidence interval, and power: a guide to misinterpretation).

Silva, L., and Fijho, D. F. (2020). Using Benford's Law to assess the quality of COVID-19 register data in Brazil

Sturm, B.L. (2013). Classification accuracy is not enough.

Subago, A. (2017). Application all digits number Benford Law in global financial statements.

Sugiarto, T., Budiman, A. I., and Rosini, I. (2017). The first digits analysis until the fifth Benford Law in financial statements

Templ, M. (2016). Simulation for Data Science with R.

Tota, I., Aliaj, A., and Lamcja. J. (2016). The Use of Benford's Law as a Tool for Detecting Fraud in Accounting Data.

Vaughan, L. (2018). Impractical Python Projects: Playful Programming Activities to Make You Smarter.

Wackerly, D., Mendenhall, W., and Schaeffer, R. L. (2008). Mathematical Statistics with Applications (7th edition).

Wedge, R., Kanter, J.M., Veeramachaneni, K. (2017). Solving the "false positive" problem in fraud prediction.

Wickham, H., and Grolemund, G. (2017). R for Data Science: Import, Tidy, Transform, Visualize, and Model Data.

Whyman, G., Shulzinger, E., and Brmashenko, Ed. (2016). Intuitive considerations clarifying the origin and applicability of the Benford law

Appendix

Link to the synthetic dataset:
https://www.dropbox.com/s/kqy3011ut2xjrl3/DBA_dataset_100.csv?dl=0

Link to the dataset for testing the hypothesis research II vs. alternative:
https://www.dropbox.com/s/a244e690fdf2j7r/hypothesis_2_benchmark_II.csv?dl=0