

FERRET: REVIEWING TABULAR DATASETS FOR MANIPULATION

PREPRINT

 **Devin Lange**
University of Utah

 **Shaurya Sahai**
University of Utah

 **Jeff M. Phillips**
University of Utah

 **Alexander Lex**
University of Utah

ABSTRACT

How do we ensure the veracity of science? The act of manipulating or fabricating scientific data has led to many high-profile fraud cases and retractions. Detecting manipulated data, however, is a challenging and time-consuming endeavor. Automated detection methods are limited due to the diversity of data types and manipulation techniques. Furthermore, patterns automatically flagged as suspicious can have reasonable explanations. Instead, we propose a nuanced approach where experts analyze tabular datasets, e.g., as part of the peer-review process, using a guided, interactive visualization approach. In this paper, we present an analysis of how manipulated datasets are created and the artifacts these techniques generate. Based on these findings, we propose a suite of visualization methods to surface potential irregularities. We have implemented these methods in Ferret, a visualization tool for data forensics work. Ferret makes potential data issues salient and provides guidance on spotting signs of tampering and differentiating them from truthful data.

1 Introduction

Data manipulation is an unfortunate reality of the scientific publication process. Like plagiarism, it is an unethical attempt to game the system, usually to further academic careers. The effects of falsified data in research vary. Manipulated data and the resulting incorrect claims can mislead scientists who want to build on the incorrect knowledge or lead to actions not based on evidence. Manipulated data can even lay a faulty foundation for a whole area of research, leading to years of wasted effort by researchers. At worst, incorrect and dishonest findings can result in the inappropriate application of knowledge in society, with potentially severe consequences, such as the harmful treatment of patients. It is suspected that in a far-ranging Alzheimer’s scandal [Pil22], for example, image and numerical data was manipulated in what was considered one of the most important publications on the topic. Based on this — now considered false — knowledge, drugs were developed and even FDA-approved, exposing patients to potentially useless medication while foregoing alternative treatments and causing side effects. Pharmaceuticals have also invested “millions of dollars, or even billions” [Pil22] based on the manipulated findings.

Unlike plagiarism — which can be discovered by checking articles against other published sources — falsified data is difficult to detect. Plagiarism checks are now part of the editorial process of many conferences and journals. However, in several high-profile data manipulation cases, scientists have had seemingly productive careers, and only after a single case of misconduct surfaced did the community critically scrutinize their whole academic record to find many instances of wrongdoing [Vig20].

Besides urging individuals to refrain from such activity, how can we prevent or at least mitigate this issue? To address this, we look to peer review, a cornerstone of the scientific process. While peer review has known flaws, the premise of peer review is that experts can verify the soundness of the research and increase the quality of published works. So why is fabricated data not caught in this step of the publishing pipeline? There are many factors: reviewers may assume a good-faith effort by their peers and are not looking for falsified data. In addition, combing through data to find signs of malpractice is difficult and time-consuming, especially when reviewers are not educated on what to look for and have no tools that can help at their disposal. Also, checking data requires that the data is made available to the reviewers and, subsequently, the readers, a practice gaining momentum with the open science movement but still far from universally adopted [Har18].

Existing tools that help find cases of data fabrication tend to focus on finding duplicated regions in images. The goal of our work is to equip editors, reviewers, and scientists

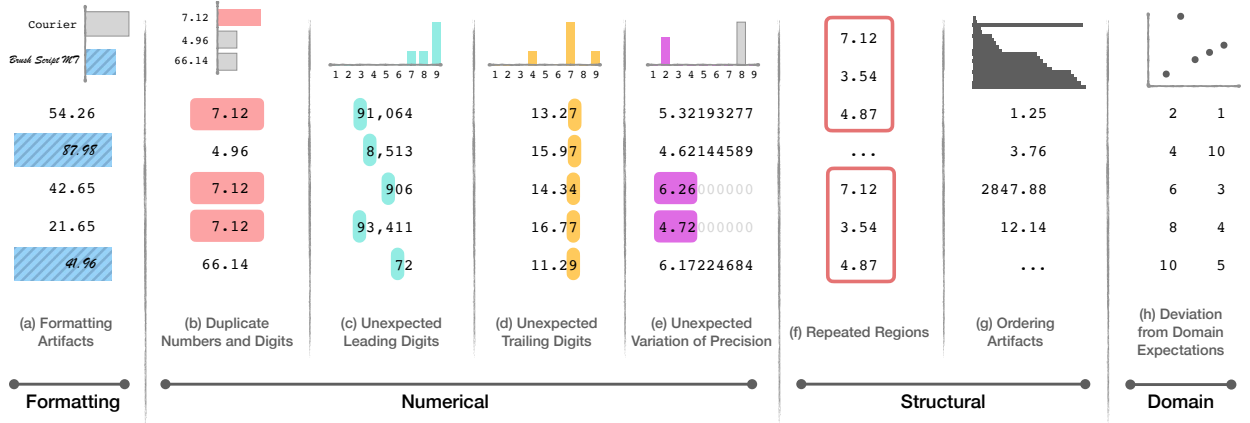


Figure 1: Artifacts of manipulation we discovered when analyzing manipulated datasets. The artifacts range from (a) unexpected formatting; to numerical issues such as (b) duplicates, unusual distributions of (c) leading or (d) trailing digits, (e) variations in precision; to structural issues such as (f) repeated regions and (g) artifacts associated with sorting and ordering items; to (h) the unrealistic relationships in the data.

with knowledge and tools to make the investigation of fabricated tabular data feasible. In particular, the tools we provide are designed to aid and enhance human judgment, as scientific data can be noisy and extremely varied, and alternatives leveraging automated statistical analysis can potentially encourage false accusations.

Our work has two primary contributions: first, we identify common artifacts of data manipulation using a combination of analyzing datasets known to be manipulated and interviews with researchers investigating fraudulent datasets. Second, we propose an array of design principles and visualization methods to saliently surface these artifacts enabling experts to easily and confidently identify fraudulent datasets.

As an additional contribution, we have developed Ferret, a prototype in which we have implemented these visualizations. In addition to these different visualization designs, we also include guidance on interpreting the results directly in the tool. Since artifacts can arise both from falsified and truthful data, it is important for users of the tool to have guidance on interpreting the results without being prescriptive in how they use the tool.

We evaluate our methods and our tool using case studies from a series of known fraudulent datasets, demonstrating that these patterns become evident by leveraging Ferret. Finally, we discuss the ethics and the potential for abuse of our approaches.

2 Related Work

While we are unaware of research on using interactive tools to detect manipulated tabular data, there are published approaches on detecting duplications in more general cases, detecting errors in spreadsheets, and detecting manipulations with numerical methods, which we discuss in this section.

2.1 Detecting Duplicated Data

Data duplication and data manipulation share some commonalities, as copying and pasting parts of a dataset is a common approach. The duplication of data, broadly speaking, is of interest in many domains and for many types of data. The detection of text plagiarism is an active research field [FMG19]. Even though it is not a solved problem, progress has been made, as is evident by the use of plagiarism-checking tools in many journals’ review processes.

Detecting software plagiarism is a similar problem. The most widely used tool for determining software similarity is MOSS [SWA03]. The authors of MOSS argue that it should not be used as an automated tool but rather as a way to surface potentially questionable data to reviewers, which is consistent with our motivations. Duplication of software can also occur when programmers copy/paste regions of code and then modify them for convenience. Detecting such copies is of interest to software engineers [BKA*07]. Similarly, in spreadsheet programs, an analyst may copy and paste a table, and just like in code, when one is updated, copies possibly should be as well. Hence methods to detect such copies exist [HSPv13, ZDZ*20]. Although detecting copies in tables shares some similarities with detecting manipulation in datasets, the structure of intentional clones compared to duplicated regions due to data manipulation can not be expected to be identical. Also, duplication is only one of the many artifacts, as we discuss in Section 5.

Some forms of image manipulation create duplicated regions, such as the use of Adobe Photoshop’s Clone Stamp tool. Image manipulation through duplication are a common problem in science [Bik22]. While much of the work on identifying manipulations remains manual, recent work relies on machine-learning techniques such as CNNs [WWO*19, LH19, BNTZ19, YLL*20, BCM*20, CDJ*21, KNY*21]. This approach is effective for images,

where large datasets can be acquired or generated. However, tabular data is more varied in its structure. More importantly, the context associated with the data is also critical for interpretation. A tabular dataset could be completely plausible given one context and obviously manipulated in another. Such contextual understanding is difficult to encode in machine learning models today, especially with limited data, as is the case for manipulated tabular datasets. Therefore, we believe that a human-in-the-loop approach is needed when detecting manipulation in tabular data.

2.2 Detecting Errors in Spreadsheets

Detecting unintentional errors in spreadsheets is a well-researched problem [PBL08, Boc16]. The detection of formula errors [BBZ18] is useful when working with spreadsheets but is unlikely to apply to manipulated data. Detecting structures that can lead to errors in tables ([CCLC16]) shares commonalities with our work since table clones are one of these structures. Beyond these structures, other methods for detecting errors in spreadsheets exist [JSHW14, KSJ*21, LWX*19b, LWX*19a, HXJ*20]. However, unintentional errors do not always produce the same artifacts as intentional manipulations, so such techniques will not identify all manipulated datasets.

2.3 Detecting Manipulation With Numerical Methods

An alternative to our interactive visual system is to inspect data for statistical anomalies. Rules like Benford’s law [Nig12, Mil15] — which state that the leading digit is more likely to be 1 and then 2 than the large digits 7, 8, 9 — has been used in domains such as accounting [DHP04]. In these settings, financial fraud has been spotted by observing that Bedford’s Law was violated over a series of transactions; not as many had leading digits of 1 or 2 as suspected. Statistical hypothesis tests can be leveraged to assess the statistical significance of deviation from this expected distribution of leading digits [Mil15, NM09]. However, to apply Benford’s law assumption must be made on the background (null) distribution. In particular, the distribution must at least span multiple orders of magnitude, which applies to some data, e.g., in astronomy or finance, but not in many others. For instance, the time in seconds to run a mile in a professional competition will almost surely start with a 2 (the current world record is 223 seconds).

Similar concerns exist in applying any statistical hypothesis testing method to look for anomalous patterns in data. All of these methods start with an assumed background (null) distribution and look for a fixed type of pattern which may deviate from it. However, the *choice* of the background distribution requires domain knowledge and human judgment. Hence, one should not automatically or generically apply tests such as those for Benford’s Law. In the breadth of tabular datasets we analyzed, we found it very rare to be able to feasibly apply such tests. As a result, we decided not to include statistical tests as we

believe that they would lead to numerous false-positives (claimed detection when a wrong background distribution was assumed).

Beyond statistical tests, there are numerical tests that do not check for statistically unlikely data but rather data that is numerically impossible. Notably, StatCheck [RNE16] checks for internal consistency of statistical measures. StatCheck is used in some peer review processes. However, it has received criticism due to concerns for its accuracy and its automated testing of papers in bulk [Cha17]. The convenience of these automatic systems carries the risk that they will be used without consideration for authors or provide the opportunity for them to respond to claims. Furthermore, if the raw dataset has been manipulated before a correct statistical analysis is run, StatCheck will not be able to identify any errors. Another algorithm for detecting manipulation is Park et al.’s work [PSL21] on detecting (and recovering) integer data when it has been multiplied by a nonintegral real number and has been rounded. While such techniques can be useful in the right situation, they are generally limited to only identifying a narrow set of problems.

2.4 Visualizing Systems

Many visualization systems visualize tabular data, yet not with a focus on detecting manipulations. The Table Lens [RC94] and Taggle [FGS*20] inspired our tabular layouts, yet our focus is on custom visualizations and descriptions specifically designed to expose artifacts. Most closely related is maybe Taco [NSH*18], a system for comparing similar datasets, but Taco could not be used to find patterns of similarity within a single table.

3 Methods

To further understand how datasets can be manipulated, we collected datasets with known issues predominantly associated with retracted publications. To identify fraudulent datasets, we leveraged a database collected by the Retraction Watch Project, a website that tracks retractions in their database and disseminates them through blog-style articles [Ora10], through community feedback on social media, and through interviews with two researchers who have investigated and reported evidence of data falsification. In total, we identified 10 datasets, with strong evidence that some manipulation occurred on the datasets, summarized in Table 1.

We obtained a complete version of the Retraction Watch Database [The18] through a special request to the database curator. Since this included a large set of papers that were retracted for various reasons, we filtered to papers that included “Falsification/Fabrication of Data” as one of the reasons for retraction, resulting in 1161 candidate papers. Next, we began manually examining the papers to find fraudulent datasets by reading the official reason for retraction and checking the retracted publication for any

references to public data. After checking 103 papers, we only found a single tabular dataset with signs of manipulation. We hypothesize that this low success rate is due to a focus on manipulated images in the database and because authors that manipulate data are incentivized to not publish it.

We then elicited help through social media. In this way, we identified four datasets associated with retracted papers. All four of these datasets also have an associated blog post where the evidence for manipulation and process of investigation has been posted. Search for the paper titles in the Retraction Watch Database revealed that these papers are in the dataset but were not flagged with “Falsification/Fabrication of Data”. Three of the four include a flag related to data, such as “Error in Data,” “Unreliable Data,” and “Concerns/Issues About Data.” The fourth paper, which is a preprint, only has a flag of “Notice - Limited or No Information.” We suspect that these less serious classes may have been used due to an abundance of caution by editors.

We also interviewed two researchers who have investigated and reported evidence of data falsification. These interviews provided us with two additional datasets. One of these was in the Retraction Watch Database, again without the “Falsification/Fabrication of Data” flag but with other flags related to data. The other paper was not in the Retraction Watch Database as of Oct 28, 2022. The interviews also introduced us to patterns of manipulations and approaches for data fabrication that these experts had encountered. For example, we had not considered checking the plausibility of the data in a larger, domain-specific context. The interviews also provided additional context for how analysts search for anomalies.

To find common patterns of artifacts across datasets, we performed a primary analysis of the data in Excel and Ferret. We also reviewed existing discussions of anomalies in the data in published works, blog posts, and online forums such as PubPeer.

4 Datasets Overview

As described in the previous section, we collected datasets that contain data manipulations associated with retracted papers. All datasets are listed in Table 1. Here we briefly introduce a subset of these datasets, and how they were likely manipulated so that it is easier to understand the artifacts present in the datasets.

DS-Driving This dataset comes from a retracted study on honesty in the field of psychology. One experiment asked participants to report the odometer mileage of their car both before and after some period of time. It appears that the “after” column was generated by adding a random number between 0 and 50,000 to the “before” number. In addition, half of the rows also appear to be generated by adding a small amount of noise to the original values.

Name	Status	Statement	Domain	Blog
DS-Priming	R	[Edi16]	Mrkt.	[Cha21] [PRA*16]
DS-Driving	R	[Edi21]	Psy.	[SSN21]
DS-Covid	W	[Law21]	Med.	[Bro21]
DS-Gaming	R	[SKV*20]	Med.	[Bro20]
DS-Spider-P	R	[LMD*20]	Bio.	
DS-Spider-E	R	[LP20]	Bio.	[Las20]
DS-Spider-I	R	[LMP20]	Bio.	
DS-Glioma	R	[Wan19]	Med.	
DS-Fly	C	[EB21]	Bio.	[Aut20]
DS-Fish	R	[Tho22]	Bio.	[Ens21]

Table 1: Table of datasets associated with retracted or withdrawn papers. Clicking on the dataset name will open Ferret with the dataset loaded. The *Status* column indicates whether a paper was retracted (R), withdrawn (W), or has earned an expression of concern (C). References in the *Statement* column link to the retraction statement. References in the *Blog* column link to blog posts that discuss how the data was manipulated.

DS-Gaming In this study, a survey was sent over email asking about video gaming habits, demographic information, and sleeping habits. The paper contains a table with summary statistics that include duplicate regions.

DS-Spider-E This study measured the “boldness” of spiders by recording how long it will take spiders to reemerge from their enclosure after a simulated predator attack. The dataset includes a large number of duplicates, as well as repeated regions.

DS-Fly In this study, the sizes of flies were measured, as well as the distance that they traveled. Both of these measurements include values that have a high degree of precision, with roughly 16 digits after the decimal point, as well as values with a precision of two. One possible explanation for this is that most values were generated with a function in excel, and a few were modified by hand, or that the low-precision numbers are actual measurements, while the high-precision numbers were generated.

5 Artifacts of Manipulation

The act of manipulating or completely fabricating a dataset can leave behind signs: We call these signs artifacts of manipulation. Using the process outlined in Section 3. As shown in Fig. 1, we have organized these artifacts into four common categories, **formatting** — relating to how the data appears in the data files, **numerical** — relating to patterns of numbers and digits in and across columns, **structural** — related to patterns that appear when analyzing multiple rows or columns together, and **relational**, relating to patterns that show impossible or implausible effects in the data given the meaning of the data. The types of artifacts we found in each of our ten datasets

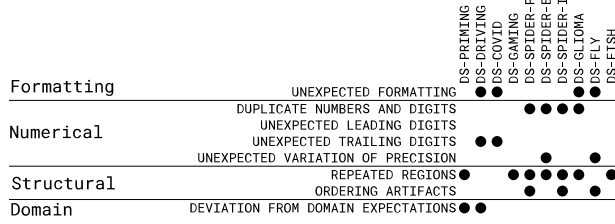


Figure 2: An overview of which datasets exhibit which artifacts.

are summarized in Figure 2. It is important to note that while we believe that these artifacts cover a wide array of signs that manipulation leaves behind, this list may not be complete. Also, the presence of artifacts is not always an indication of wrongdoing — they can be produced by a valid data processing step or be an artifact of the data collection methodology. Finally, it is sometimes difficult to distinguish intentional wrongdoing from honest mistakes while working with data. We discuss the implications of our visualization design in Section 6 and the necessary care in our section on broader impacts (Section 10).

5.1 AR-Formatting: Formatting Artifacts

Spreadsheet tools like Excel or Google Sheets allow users to format the appearance of the data. This includes choosing a *font*, *font size*, methods of text emphasis such as *bold*, *italics*, and *underlining*, and font as well as background *colors*. In addition, users can select a *data format*. For example, changing a cell to a date format will alter how the cell is displayed without changing the underlying information. These formats can be flexibly chosen for cells, columns, or rows, and combinations thereof. Formatting is typically consistent and logical in inconspicuous data. However, if odd patterns of formats occur, this can hint at manipulation, as illustrated in Fig. 1a. For instance, in the study on honesty (**DS-Driving**), it appears that data was copied to a temporary file, where the values were modified, and later copied back into the master spreadsheet. Notably, these two files seem to have used different fonts, so in the final dataset, exactly half of the rows contained text using the font *Calibri* and the other half using *Cambria*. These rows were interspersed, likely due to shuffling or sorting the table after augmenting it. In total, we found evidence of strange formatting in three of the datasets we collected (**DS-Driving**, **DS-Covid**, **DS-Fly**).

This type of artifact, however, could also appear in authentic data, for example, when assembling a dataset from multiple data sources. Whether or not such a pattern is a sign of manipulation will depend on details, such as whether a whole column has a different format (likely not suspicious), or whether individual cells are formatted differently (possibly suspicious).

5.2 Numerical

The variety of possible numerical artifacts left behind by bad actors is considerable. Here we describe common types that we have observed. All of these artifacts occur both in individual columns and across columns.

5.2.1 AR-Duplicate: Duplicate Numbers and Digits

This artifact describes cases when (whole) numbers or sequences of digits (parts of numbers) are repeated more frequently than expected (Fig. 1b). Encountering duplicate numbers or digits can suggest that data was copied and pasted or manually entered. When measuring a natural phenomenon there is typically variation in the data, either from differences in the signal being measured or from noise introduced by the tools used to measure the signal. For a specific number of values sampled from a distribution at a specified precision, a certain number of duplicate values can be expected. When there are more duplicates than expected there are a few possible explanations. First, the underlying distribution could be different than expected. For instance, a narrow Gaussian distribution would result in more duplicates than a wider one. Next, low precision generally would make duplicates more likely.

A common cause of duplicate numbers and sequences of digits that may seem suspicious at first, but is typically innocent are high-precision duplicates caused by converting measurements. For example, converting fractions to decimals could introduce duplicates with seemingly high precision. If an experiment recorded the length of an animal in inches as integers, but in a subsequent step, the data would be converted to feet using decimals, we would expect that the resulting decimals have values with high precision, such as 0.33333333 and 0.41666667. In this case, the number 0.33333333 may appear more often than naively expected and an n-gram of digits, such as 3s, or 6s may appear frequently.

Another common cause of duplicates that is likely innocent is thresholding or reaching a maximum/minimum value. In many scientific experiments, there is a terminating condition, such as a maximum time of the experiment or a score corresponding to a maximum or minimum achievable value. For example, while the spider datasets (**DS-Spider-E**) are definitely manipulated, they also only measured a time period of 10 minutes, recorded as 600 seconds, and that maximum threshold was reached often. Hence, the frequent occurrence of 600 in such a dataset is likely inconspicuous.

Duplicate numbers can also appear when a dataset is manipulated by copying items or measurements or by manually inventing numbers. Humans are bad at generating random numbers [TLB14, SSBW12, FSK08] and random sequences of digits. When humans simulate the process of sampling from a distribution by repeatedly typing numbers, they tend to produce patterns (duplications) that often can be distinguished from collected data. In addition, sequences of digits appear more frequently in fabricated sets

of numbers. For example, 54.23 and 23.54 are not duplicate numbers, but they do contain duplicate digit sequences, 54 and 23. We observed a suspicious amount of duplicated numbers and digits in four of our datasets (**DS-Spider-E**, **DS-Spider-P**, **DS-Spider-I**, and **DS-Glioma**).

5.2.2 AR-Leading: Unexpected Leading Digits

Benford’s Law [Nig12, Mil15] (also discussed in Section 2.3) is an expected pattern of the first digits of numbers in a dataset (Figure 1c). In short, it states that in datasets that span multiple orders of magnitude, the most frequent first digit should be a one, followed by a two, then a three, and so on. For example, in a dataset of the number of people living in cities and villages, we would expect that there are more cities with 100,000–199,999 inhabitants (leading digit 1) than there are cities with 900,000–999,999 inhabitants (leading digit 9). We have included this artifact in our collection since checking for violations of Benford’s law is a known technique for unearthing fabricated data. However, none of the scientific datasets in our collection spreads densely over such multiple orders of magnitude; hence we did not identify this pattern.

5.2.3 AR-Trailing: Unexpected Trailing Digits

We have also found it useful to examine the last digit of numbers (Figure 1d). In some situations, the last digit of a collection of measurements might represent a randomly sampled uniform distribution. In other situations, different patterns would be expected. For example, in a list of prices for grocery-store products, an increased frequency in the digit nine would be expected since prices ending with 99 are strategically selected to make a product appear cheaper. On the other hand, if people are asked to provide an estimate for a value, we expect a final digit of zero to be more frequent than other digits. For example, if participants at a large event were asked how many people attended, we would expect an answer of 15,000 to be much more common than 14,872. We consider a trailing digits artifact to be a mismatch between the expected pattern of the last digit and the pattern observed in the data or an unexplained inconsistency of trailing digits between parts of a dataset.

In **DS-Driving** two different columns represent values where drivers are asked to give the mileage of their car. In one column, this rounding effect — showing a large amount of numbers ending with zeros — is present. In the other column, the trailing digits follow a uniform distribution. The manuscript does not describe any difference in data collection between these two columns that might explain the difference in pattern between the final digits.

Although this example could be identified by shifting the decimal place and performing an analysis on precision, this is not always the case. The frequency of numbers ending in nine would not be noticeable in a precision analysis. Furthermore, a precision analysis of **DS-Covid** would not catch a strange pattern where even trailing digits occur

more frequently than odd digits. Vice versa, not all precision artifacts are noticeable through a trailing digit analysis. For instance, the variance of precision of the stopwatch example is independent of the frequency of different trailing digits.

5.2.4 AR-Precision: Unexpected Variation of Precision

The formatting of data in spreadsheet programs can also obfuscate data, leading to numerical artifacts that may not be evident in the source spreadsheet. In particular, this can occur with the precision of numbers. Numerical data may record a varying number of digits after the decimal places. However, if the data is formatted as a *number*, the default in excel is to show two digits after the decimal place. We assume inconspicuous data has similar precision for similar observations. Time measured with a stopwatch, for example, would typically have a precision of up to 1/100 of a second. Most numbers should have two digits after the decimal points, a few with one digit (e.g., exactly 3.1 seconds), and even fewer numbers with no digits (3 seconds). Manipulated data may have extremely varied precision Fig. 1e. This could happen if data is recorded (or generated) with a high degree of precision, then manually manipulated to change some values. Such a difference may not be apparent in a spreadsheet program when two digits are displayed. Alternatively, some authentic data could be collected with limited precision, and a function with high precision could be used to generate the rest. However, such a phenomenon could also arise in innocent ways, like in converting between fractions and decimals, as explained earlier. We have observed unexplained varied precision in **DS-Fly**.

5.3 Structural

Beyond the frequency of data or attributes of data, the structure of data can also play a role in detecting manipulation. Structural patterns are concerned with both the value of measurements and the order of the observations in the data file.

5.3.1 AR-Regions: Repeated Regions

While six duplicate numbers may be considered a weak signal of manipulation, two identical sequences of six numbers are a much stronger one. We consider a region to consist of multiple cell values that have a spatial relationship in a spreadsheet, as illustrated in Figure 1f. While the simplest example is a sequence of numbers in a column, regions include adjacent patterns, vertically or horizontally, and may include gaps. Repeated regions can be artifacts of manipulation. While some repeated regions could be caused by how the data is collected, such an innocuous structure is likely obvious. In the case of manipulated data, regions were likely copied and pasted multiple times, either accidentally or as a convenient way to augment a dataset. In addition to simply copying and pasting regions, parts

of the region are sometimes modified manually, resulting in similar regions with gaps. We saw this type of artifact in seven of our ten datasets, making it the most common artifact we discovered (**DS-Priming**, **DS-Gaming**, **DS-Spider-E**, **DS-Spider-I**, **DS-Spider-P**, **DS-Fish**).

5.3.2 AR-Ordering: Ordering Artifacts

It is natural for ordering artifacts to exist in authentic datasets. For instance, if multiple observations are recorded over time it would be expected that time increases throughout the dataset. Our interviews revealed that some experts consider it a good practice to avoid changing the order of a dataset. However, it is not uncommon and not automatically suspicious that data is re-sorted.

The ordering of the data can still indicate manipulation, as illustrated in Figure 1g. For example, if a bad actor wants to show that an experimental condition has an effect on the weight of animals, they might sort the data based on weight. Then, they might modify values at the distribution’s tails — altering the data to match their hypothesis. This is an economical approach since changing the extreme values will have the largest effect on aggregate measurements. However, this approach can leave behind ordering artifacts. This kind of dataset where a column is nearly sorted is one example of an ordering artifact. If the order is reset after modifications, such a pattern might be difficult to detect. However, if the data is reset by sorting on a column with duplicates (e.g., by a categorical value), then the effects of sorting on weight before the reset will still be seen within the groups. This kind of ghost sorting is another variation of an ordering artifact. A different order artifact exists in **DS-Fly**. Here one column has a mixture of high and low precision (**AR-Precision**). Additionally, the cells with low precision do not appear to be randomly interspersed throughout the rows, but rather appear in a repeated structured way.

5.4 AR-Domain: Deviation from Domain Expectations

So far, we have assumed that artifacts are visible in the formatting, structure, or values of the data. However, authors may use more sophisticated techniques for generating fabricated data that cannot be detected with the aforementioned methods. In these situations, more sophisticated techniques are required to find the artifacts.

Single-Dimensional. In the case of a single dimension of data, there is often prior knowledge about how that data should look, at least in the aggregate. For instance, many natural measurements, such as the weight of an animal, will exhibit a normal distribution. We consider drastic variations from these expectations, such as a uniform distribution occurring when a normal distribution is expected, or a normal distribution with an obviously clipped tail, to be a single-dimensional domain artifact.

Relational. If authors are careful, a single column of fabricated data may be indistinguishable from authentic

data. However, ensuring that all columns have a reasonable (based on domain expectations) relationship with all other columns is a more difficult task. For example, if an experiment measured the length and weight of an animal, there likely should be a correlation between the two values (longer animals of the same species are likely to be heavier, on average). If such data is generated or manipulated using, for example, spreadsheet functions for individual columns, the data may look innocent when only looking at one column, but the relationships between columns may not be meaningful, as illustrated in Figure 1h. Hence, a dataset exhibits a relational artifact when the relationships between columns differ from expectations.

Relational artifacts can be more nuanced than a missing correlation. In **DS-Driving**, for example, comparing the relationship of cars’ mileages before and after a period of time, the miles driven in this period is always less than 50,000 miles, with many drivers very close to driving 50,000 miles, violating an assumption of a smooth distribution. **DS-Priming** shows large groupings at the extreme ends of two experimental conditions.

6 Visualization Design Principles

Analyzing datasets for manipulation is a difficult and potentially fraught endeavor. A claim of manipulation, even during the review process, is a serious accusation and should be levied with caution. Hence, we believe it is essential that the agency should be firmly in the hand of the domain experts analyzing a dataset. Analysis tools should support experts by providing guidance without being prescriptive. To realize this sentiment, we developed design principles that guided our development of Ferret, a visualization tool for reviewing tabular datasets for manipulation. In this section, we introduce these design principles, while we describe the particulars of Ferret in the next section. Some of our guidelines are related to general visualization guidelines, such as Shneiderman’s Mantra (overview first, zoom and filter, details on demand) [Shn96], yet we provide more specific guidance for the use case of detecting manipulations in datasets.

6.1 Show, Don’t Tell

One early observation we made is that the breadth of artifacts of manipulation is significant and that domain knowledge about the data is often necessary to make accurate judgments. Hence, we argue that a human, ideally with domain expertise, is needed to discern whether an artifact is the result of manipulation. As a result, our first design principle is to provide guidance through the artifacts of manipulation and to provide salient visualizations of potential issues (*show*) but to not be prescriptive, for example, by describing why a data set is manipulated or recommending a particular analysis or statistical test (*don’t tell*).

This principle is manifested in Ferret in several ways: First, Ferret lists and explains the different types of artifacts

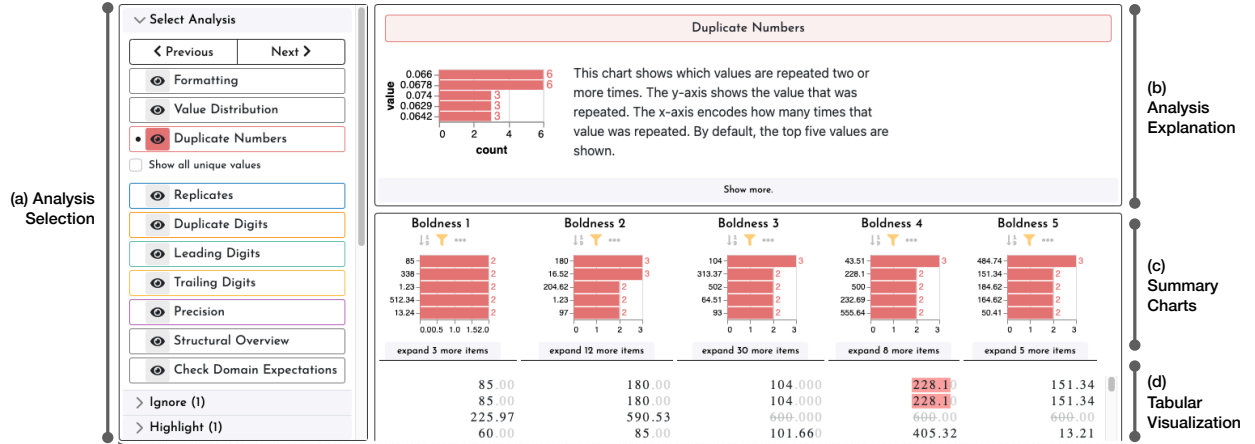


Figure 3: Overview of the Ferret visualization tool. (a) The *Analysis Selection* panel gives quick access to the available analysis modes. (b) An explanation is provided for each analysis mode to help users understand artifacts of manipulation and guard against false positives. (c) The *Summary Charts* display aggregate information for each column in the dataset. (d) The *Tabular Visualization* gives access to the raw tabular data with relevant data surfaced through highlighting and rearranging. Values can be highlighted (such as 228.1) and ignored (600).

(Fig. 3 a and b). The introductory text gives advice on how to spot an artifact but also lays out common benign causes of those artifacts. Second, Ferret provides visualizations to identify and confirm suspicious patterns (Fig. 3 c and d). And finally, Ferret refrains from using statistical tests to identify issues. Our argument for not using tests is that most tests would only be valid under narrow circumstances (such as a specific type of distribution) and that the danger of inappropriately using a test outweighs the benefits.

6.2 Make Artifacts Salient

Our next design principle is to make artifacts of manipulation salient. Since our first design guideline necessitates an expert human to investigate the data, our goal is to make that investigation more efficient by quickly exposing artifacts of manipulation. This principle is best illustrated with an example: It can be difficult to notice things like the difference between Calibri and Cambria fonts or between 11.9 and 12-point font sizes in traditional spreadsheet software. In Ferret, cells with deviating formatting are highlighted with a distinct background color and pattern (Fig. 8a) so that differences are salient. Similarly, spreadsheet tools will often round decimals in their display to two digits. Ferret will display all of the digits recorded and aligned at the decimal point (Fig. 5a).

6.3 Use Overview and Details

A well-designed visual overview handles large datasets and helps analysts quickly spot suspicious patterns. At the same time, making the raw tabular data a first-class citizen within the visualization is essential. Only access to the raw data enables an analyst to confirm their suspicion or identify a benign explanation. In other words, any overview visualizations should be tightly integrated with a visualization of the details. If an interesting feature is

noticed in an overview, it should be possible to query for details and easily see the rows generating that feature, as shown in Fig. 3d. Conversely, if an interesting pattern is found by inspecting the raw tabular data, it should be easy to switch to the overview visualization and observe that pattern from a higher vantage point (Fig. 5b and Fig. 8b).

6.4 Leverage Interactivity

While investigating artifacts of manipulation, interactive sorting and filtering is essential. **Sorting** by different columns provides many ways to view the data, and combining this ability with different visual encodings, can reveal interesting patterns, such as alternating fonts Fig. 8b. **Filtering** is useful for focusing/excluding specific items. However, unlike most systems, ignoring only the values of specific cells (in contrast to filters that remove a row from a dataset), is more useful for detecting manipulations. For instance, in the case where values are clamped to an upper bound (**DS-Spider-E**), there may be many duplicates. Such duplicates will affect the analysis of several artifacts. Excluding those frequent values from the analysis is a convenient way of running the visualizations on the remaining data without excluding entire rows (Fig. 3d).

7 Visualizations in Ferret

Ferret is based on the design principles described to surface artifacts of manipulation. Ferret provides various visualizations for different aspects, yet some visual encodings can be used for multiple patterns. At the heart of Ferret is a tabular visualization technique [RC94, FGS*20] combining spreadsheet-like raw values with graphical marks, with a series of domain-specific custom visual encodings, enriched by a set of supplementary views.

Formatting Ferret uses dedicated visual encodings for formatting artifacts within the tabular visualization and considers font styling and emphasis as well as the *data format*. Ferret does not use the styling of the source, since the exact formatting is usually immaterial for detecting manipulations. Instead, our encoding emphasizes the differences in formatting: The most frequent combination is assigned the default white background, while all other unique combinations of formats are assigned a background color and texture/pattern (see Figure 8a). We chose to use both patterns and colors as the number of unique combinations can easily exceed the number of reasonably distinguishable colors. When a cell is selected, the exact formatting parameters and a count for the number of cells that share the same formatting are listed.

Summary Charts: Counts, Proportions, and Distributions We use histograms and bar charts to visualize distributions (how are values in a column distributed), counts (how often a number is duplicated), and proportions of values (what percentage of numbers has a precision of 2). To view the **counts of values**, we use horizontal bar graphs (Fig. 4a), which is useful for visualizing the count of duplicates and duplicate digits **AR-Duplicate**. These graphs can contain long labels, which is well suited for a horizontal layout. In Fig. 3c, for example, the duplicate numbers of one of the spider datasets (**DS-Spider-E**) are shown at the top of the five numerical columns. The duplicate digits chart works analogously; instead of visualizing duplicated whole numbers, it shows duplicated sequences of digits (2- or 3-grams). To view the **proportion of values** with certain properties, Ferret shows vertical bar charts, where each bar shows a percentage of the property on the overall column (Fig. 4b). We use proportion bar charts to show the frequency of trailing and leading digits (**AR-Leading** and **AR-Trailing**), as well as the frequency of a specific precision (**AR-Precision**). Finally, we use a histogram to show the **distribution of values** (Fig. 4c), which is useful for general sanity checks and alignment with domain expectations (**AR-Domain**).

Tabular Visualization These summary visualizations are tightly integrated with the tabular visualization. Using the summary charts, values can be highlighted or filtered. In Fig. 3 the number 600 has been filtered out, which removes it from the bar chart, and strikes it out in the tabular view. The value 228.1 is highlighted in red.

Figure 5a shows another example of tight integration between the summary visualization on top and the tabular visualization below. The bar chart shows the proportions of different levels of precision, while the tabular visualization

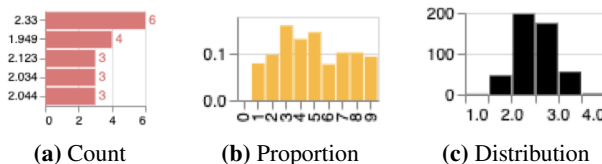


Figure 4: Different summary visualizations available in Ferret.

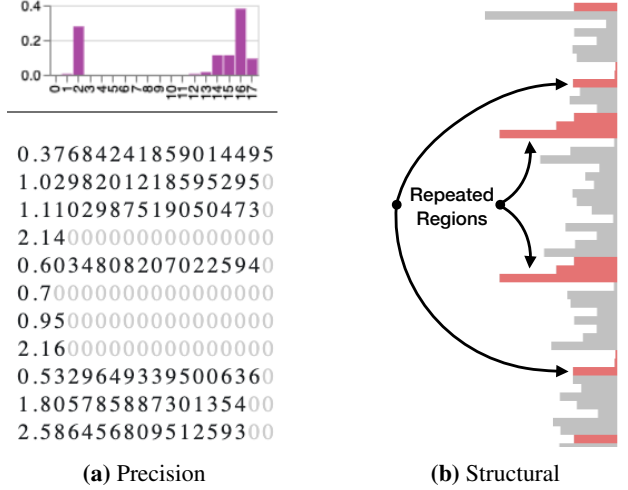


Figure 5: Visualizations for precision and structural artifacts. (a) A precision artifact (**AR-Precision**) is visible in **DS-Fly** through the proportion chart and the tabular visualization. (b) Repeated regions (**AR-Regions**) are visible for **DS-Gaming** using the overview.

tion below shows data, highlighting the precision through alignment.

Structural Visualization For large tables, it can be cumbersome to scroll through the full dataset, and raw numbers don't show structural effects well (**AR-Structural**). The table overview mode [FGS*20] in Ferret solves this problem by reducing the cell height to one pixel, maximizing the number of rows visible on the screen (see Figures 5b and 8b). In overview mode, exact values are elided, and graphical representations are shown.

Domain Visualizations Finally, Ferret includes a suite of domain visualizations to help reviewers test the data for deviations from their domain expectations. Ferret supports scatterplots (Fig. 7), faceted strip plots (Fig. 6), violin charts, bar charts, and parallel coordinate plots.

8 Implementation

Ferret is open source and implemented as a front-end web application. The code is available at <https://github.com/visdesignlab/ferret>, and a demo of the tool is available at <https://ferret.sci.utah.edu/>. The summary charts are built with Vega-Lite [SMWH17]. The table is built on top of LineUp [GLG*13] and Taggle [FGS*20] and uses custom code for the different kinds of cell rendering. The general visualizations are imple-



Figure 6: Strip-plot for miles driven in **DS-Driving** faceted by font. The data rendered in different fonts appears to be duplicated with minor noise added.

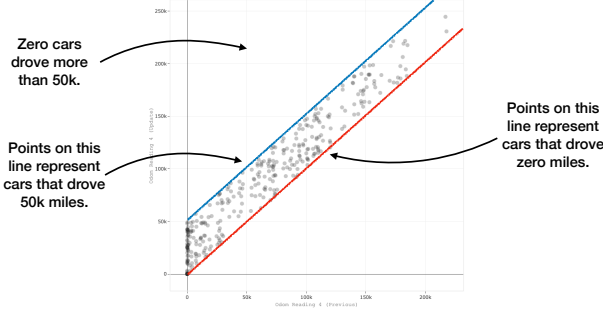


Figure 7: Scatterplot used to analyze a deviation from the domain expectation (**AR-Domain**). The x-axis corresponds to odometer readings taken at the beginning of the study, the y-axis corresponds to readings taken at a later time, as recorded in **DS-Driving**. Note that the distribution of miles driven seems uniform, up to a hard cut-off after 50,000 miles, indicated by the blue annotated line.

mented with React and Plot.ly [Inc15]. Ferret uses excel.js [ed22] to load and process Excel files.

9 Case Study

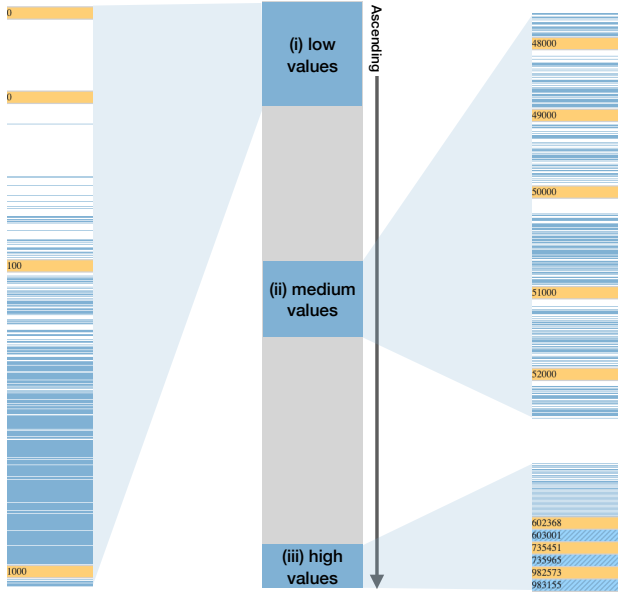
In the following, we demonstrate the utility of the classification of artifacts, our design guidelines, and the Ferret prototype in a case study. We include case studies for all 10 datasets in the supplementary material.

As our primary case study, we will analyze the driving dataset (**DS-Driving**) and recreate and expand upon the analysis in a blog post that leads to the paper’s retraction [Edi21]. The post discusses four anomalies and provides two hypotheses to explain them. A similar analysis with Ferret unearths the same and some additional anomalies, sometimes using different kinds of visualizations, that support the claims of the blog post. Upon loading a dataset, Ferret displays the Formatting visualization (**AR-Formatting**), also showing instructions on when to use it and how to read it. For the driving dataset (**DS-Driving**), it is immediately obvious that there is suspicious formatting in the second (*Odom Reading 1 (Previous)*) and the third (*Odom Reading 2 (Update)*) columns. Mixed formatting within a single column, as seen in Fig. 8a, is unusual and suspicious. In this case, there is a mixture of **Calibri** and **Cambria** fonts. If this was the only artifact found in a dataset, an editor could reach out to the authors and ask for an explanation. However, further investigation reveals additional irregularities. Switching to the overview mode allows a faster review of the table’s 13,488 rows. The pattern of seemingly random mixes of fonts continues throughout the column. Yet, sorting the data from low to high reveals several patterns, shown in Fig 8b.

First, most values less than 100 in this column are in **Calibri** font, and all rows with a value of zero are in **Calibri**. Conversely, values between 100 and 1000 are predominately **Cambria**. For the rest of the data, the two fonts are more interspersed, except for certain regions, where

Policy #...	Odom Reading 1 (Previous)	Odom Reading 1 (Update)
1	896	39198
2	21396	39198
3	21340	39198
4	23912	39198
5	16862	39198
6	147738	39198
7	18780	39198
8	41930	39198
9	28993	39198
10	78382	39198
11	58500	39198
12	99417	39198
13	93231	39198

(a) Detailed



(b) Overview

Figure 8: Visualizing formatting artifacts with color and patterns for **DS-Driving**. (a) The most frequent type of formatting is not highlighted (white background). All other formatting combinations are assigned a unique pattern/color combination. A tool-tip shows the formatting details on demand. (b) Showing structural patterns related to formatting. The pull-outs i-iii are taken from a large column, illustrated schematically in the center. Low values (i) are formatted in **Cambria** (white), while (ii) medium values alternate between **Calibri** and **Cambria** (blue), with **Cambria** clusters of round numbers. High values (iii) alternate between the fonts.

Calibri dominates. Inspecting the values in the region reveals they are duplicate round numbers, such as 75,000. Since these values represent self-reported car mileage, the data makes sense if people estimate the mileage of their car. Suspiciously, these rounding effects are not visible for the values in **Cambria** font, suggesting that the data collection method for the two fonts diverges.

Finally, the high values (Fig. 8b) alternate perfectly between **Calibri** and **Cambria**. Closer inspection reveals that every value styled in **Calibri** font has a corresponding

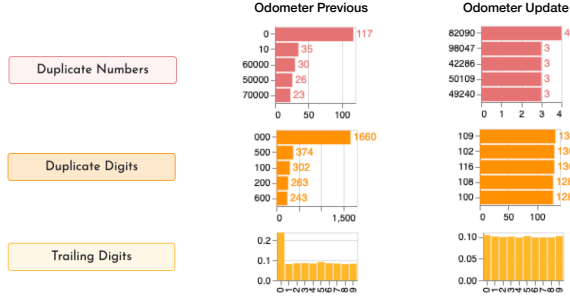


Figure 9: Rounding effects are clearly present in the duplicate numbers, duplicate digits, and leading digit frequency charts for the initial odometer reading (left column) but suspiciously absent for the follow-up reading (right column) in **DS-Driving**.

Cambria value that is within 1000 miles. This pattern suggests data was copied and a random number between 1 and 1000 was added. Visualizing this column (Fig. 6), reveals that the two datasets are extremely similar.

A different approach to analyzing this dataset is to look at rounding effects. Fig 9 reveals tell-tale signs of rounding using the duplicate numbers, duplicate digits, and leading digit frequency charts for the first column (the initial odometer reading). However, the second column (the follow-up reading after some time has passed) does not show any rounding effects.

To further explore this difference, we visualize the relationship between these two variables with a scatterplot. Fig. 7 shows that the miles driven never exceed 50,000, and the distribution of miles driven is uniform between zero and 50,000, an unlikely distribution for this dataset, supporting a hypothesis made in the blog that the odometer readings in the updated column were generated by adding a random number between 0 and 50,000.

10 Discussion and Broader Impacts

With the goal of increasing the trustworthiness of scientific research, our work collects manipulated datasets, categorizes artifacts of manipulation, designs visualization methods to explore them, and prototypes a tool to make those artifacts salient. We observe that manipulated datasets tend to present multiple artifacts simultaneously, which can be spotted with different techniques offered by Ferret. Hence, we believe that our approach of providing multiple visualizations that are easy to step through and interpret is a robust method for spotting artifacts and minimizing risks. That said, there are several potential unintended consequences from our work that could affect researchers and society in general.

False Positives. One concern is the possibility that our methods indicate that data has been manipulated when in reality, it has not. To remedy this, we suggest that when artifacts are identified, they should be used as a means of discussion with the authors, not as indisputable

evidence of wrongdoing. This concern is also one of the reasons we believe statistical tests or summary reports would be treacherous, as they might reduce the nuance and complexity of the topic to simplistic answers. Still, a tool such as Ferret can make it easier to levy accusations against authors. An overly zealous individual could cause harm if they place too much confidence in individual artifacts of manipulation and don’t give authors opportunities to respond. In the worst case, bad actors could use a tool like Ferret to maliciously target individuals. To remedy this, we suggest that Ferret should predominantly be deployed for general checks as part of the review process or when there are reasons to suspect wrongdoing with a paper.

Shaming. Our hope in collecting references to the manipulated datasets in this paper will be a resource for others interested in investigating data manipulation in tabular datasets. However, it is possible that our work leads to additional unwanted attention for the authors of these datasets. To minimize the potential impact of our actions, we have only published datasets that come with an official retraction or an expression of concern from the publishing journal.

Security Theatre. Reviewers and editors are often volunteers, hence limiting their workload is an important consideration, especially if the additional work would be akin to useless “security theater”. Similar concerns can be raised about plagiarism checkers, yet they have detected numerous cases of plagiarism. We also attempt to make Ferret easy to use to avoid unnecessary burdens. However, conducting a cost-benefit analysis in a trial run with a selected journal is a logical next step.

Abuse. Knowledge about artifacts of manipulation and the existence of tools to identify them may help bad actors avoid detection of their misconduct. Experience from plagiarism detection tools shows that they continue to catch manipulation. While we cannot ensure that abuse doesn’t happen, we hope that the burden of “engineering” a dataset that doesn’t raise suspicion is so high that bad actors may conclude that manipulation is not worth the risk.

Data Sharing. Using tools like Ferret may disincentivize authors to submit data with their manuscripts for fear of being unjustly accused of manipulation. While many journals and conferences already require the publication of data, researchers may choose to publish with journals that don’t. We hope that the scientific community can meet this challenge by (a) carefully using tools like Ferret and (b) more broadly endorsing open science practices.

11 Conclusion and Future Work

In conclusion, we believe our work will help future reviewers “ferret out” manipulations in tabular datasets. Knowing what artifacts of manipulation to look for will help analysts focus their search. Our design guidelines will aid in the development of tools for performing data forensics. Finally, Ferret is a first step towards instantiating this knowledge in

a tool. Due to the adversarial nature of catching instances of data manipulation, it is likely impossible to design a single static tool for catching all cases of data manipulation. However, we believe our approach, which emphasizes the importance of the human-in-the-loop, is robust to changes in future manipulation techniques.

12 Acknowledgments

The authors wish to thank Holger Stitz, Michael Pühringer, and the LineUp authors for their support using the library, the Retraction Watch Project for access to their database, Zach Cutler and Jack Wilburn for technical help, the interview participants for their time and expertise, and the Visualization Design Lab for feedback. This work was supported by NSF IIS 1751238 and CCF-2115677.

References

- [Aut20] AUTHOR A.: PubPeer discussion of "Host-parasitoid evolution in a metacommunity", Aug. 2020.
- [BBZ18] BAROWY D. W., BERGER E. D., ZORN B.: ExceLint: Automatically Finding Spreadsheet Formula Errors. *Proceedings of the ACM on Programming Languages* 2, OOPSLA (Oct. 2018), 1–26. arXiv:1901.11100, doi:10.1145/3276518.
- [BCM*20] BONETTINI N., CANNAS E. D., MANDRELLI S., BONDI L., BESTAGINI P., TUBARO S.: Video Face Manipulation Detection Through Ensemble of CNNs. *arXiv:2004.07676 [cs, eess]* (Apr. 2020). arXiv:2004.07676.
- [Bik22] BIK E.: Science Has a Nasty Photoshopping Problem. *The New York Times* (Oct. 2022).
- [BKA*07] BELLON S., KOSCHKE R., ANTONIOL G., KRINKE J., MERLO E.: Comparison and Evaluation of Clone Detection Tools. *IEEE Transactions on Software Engineering* 33, 9 (Sept. 2007), 577–591. doi:10.1109/TSE.2007.70725.
- [BNTZ19] BARNI M., NOWROOZI E., TONDI B., ZHANG B.: Effectiveness of random deep feature selection for securing image manipulation detectors against adversarial examples. *arXiv:1910.12392 [cs, eess]* (Dec. 2019). arXiv:1910.12392.
- [Boc16] BOCK A. A.: A Literature Review of Spreadsheet Technology. *IT University of Copenhagen Technical Report Series* (2016), 35.
- [Bro20] BROWN N.: Some issues in a recent gaming research article: Etindele Sosso et al. (2020), Apr. 2020.
- [Bro21] BROWN N.: Some problems in the dataset of a large study of Ivermectin for the treatment of Covid-19, July 2021.
- [CCLC16] CHEUNG S.-C., CHEN W., LIU Y., CHANGXU: CUSTODES: Automatic Spreadsheet Cell Clustering and Smell Detection Using Strong and Weak Features. In *2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)* (May 2016). doi:10.1145/2884781.2884796.
- [CDJ*21] CHEN X., DONG C., JI J., CAO J., LI X.: Image Manipulation Detection by Multi-View Multi-Scale Supervision. *arXiv:2104.06832 [cs]* (July 2021). arXiv:2104.06832.
- [Cha17] CHAWLA D.: Controversial software is proving surprisingly accurate at spotting errors in psychology papers. <https://www.science.org/content/article/controversial-software-proving-surprisingly-accurate-spotting-errors-psychology-papers>, Nov. 2017.
- [Cha21] CHARLTON A.: RETRACTED ARTICLE: Why money meanings matter in decisions to donate time and money. <https://openmkt.org/blog/2021/retracted-article-why-money-meanings-matter-in-decisions-to-donate-time-and-money/>, July 2021.
- [DHP04] DURTSCHI C., HILLISON W., PACINI C.: The effective use of benford’s law to assist in detecting fraud in accounting data. *Journal of forensic accounting* 5, 1 (2004), 17–34.
- [EB21] EDITORIAL BOARD P. B.: Editor’s Note on: Host-parasitoid evolution in a metacommunity. *Proceedings of the Royal Society B: Biological Sciences* 288, 1947 (Mar. 2021), 20210505. doi:10.1098/rspb.2021.0505.
- [ed22] EXCELJS DEVELOPERS T.: ExcelJS. exceljs, Dec. 2022.
- [Edi16] EDITORS J.: Retraction Note to: Why money meanings matter in decisions to donate time and money. *Marketing Letters* 27, 2 (June 2016), 409–409. doi:10.1007/s11002-016-9401-6.
- [Edi21] EDITORS J.: Retraction for Shu et al., Signing at the beginning makes ethics salient and decreases dishonest self-reports in comparison to signing at the end. *Proceedings of the National Academy of Sciences* 118, 38 (Sept. 2021), e2115397118. doi:10.1073/pnas.2115397118.
- [Ens21] ENSERINK M.: Sea of doubts. *Science* 372, 6542 (May 2021), 560–565. doi:10.1126/science.372.6542.560.

- [FGS*20] FURMANOVA K., GRATZL S., STITZ H., ZICHNER T., JARESOVA M., LEX A., STREIT M.: Taggle: Combining overview and details in tabular data visualizations. *Information Visualization* 19, 2 (2020), 114–136. doi:10.1177/1473871619878085.
- [FMG19] FOLTÝNEK T., MEUSCHKE N., GIPP B.: Academic Plagiarism Detection: A Systematic Literature Review. *ACM Computing Surveys* 52, 6 (Oct. 2019), 112:1–112:42. doi:10.1145/3345317.
- [FSK08] FIGURSKA M., STAŃCZYK M., KULESZA K.: Humans cannot consciously generate random numbers sequences: Polemic study. *Medical Hypotheses* 70, 1 (Jan. 2008), 182–185. doi:10.1016/j.mehy.2007.06.038.
- [GLG*13] GRATZL S., LEX A., GEHLENBORG N., PFISTER H., STREIT M.: LineUp: Visual Analysis of Multi-Attribute Rankings. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '13)* 19, 12 (2013), 2277–2286. doi:10.1109/TVCG.2013.173.
- [Har18] HAROZ S.: Open Practices in Visualization Research : Opinion Paper. In *2018 IEEE Evaluation and Beyond - Methodological Approaches for Visualization (BELIV)* (Oct. 2018), pp. 46–52. doi:10.1109/BELIV.2018.8634427.
- [HSPv13] HERMANS F., SEDEE B., PINZGER M., VAN DEURSEN A.: Data clone detection and visualization in spreadsheets. In *2013 35th International Conference on Software Engineering (ICSE)* (San Francisco, CA, USA, May 2013), IEEE, pp. 292–301. doi:10.1109/ICSE.2013.6606575.
- [HXJ*20] HUANG Y., XU C., JIANG Y., WANG H., LI D.: WARDER: Towards effective spreadsheet defect detection by validity-based cell cluster refinements. *Journal of Systems and Software* 167 (Sept. 2020), 110615. doi:10.1016/j.jss.2020.110615.
- [Inc15] INC. P. T.: Collaborative data science. Plotly Technologies Inc., 2015.
- [JSHW14] JANNACH D., SCHMITZ T., HOFER B., WOTAWA F.: Avoiding, finding and fixing spreadsheet errors – A survey of automated approaches for spreadsheet QA. *Journal of Systems and Software* 94 (Aug. 2014), 129–150. doi:10.1016/j.jss.2014.03.058.
- [KNY*21] KWON M.-J., NAM S.-H., YU I.-J., LEE H.-K., KIM C.: Learning JPEG Compression Artifacts for Image Manipulation Detection and Localization. *arXiv:2108.12947 [cs, eess]* (Aug. 2021). arXiv:2108.12947.
- [KSJ*21] KOCH P., SCHEKOTIHN K., JANNACH D., HOFER B., WOTAWA F.: Metric-Based Fault Prediction for Spreadsheets. *IEEE Transactions on Software Engineering* 47, 10 (Oct. 2021), 2195–2207. doi:10.1109/TSE.2019.2944604.
- [Las20] LASKOWSKI K. L.: What to do when you don't trust your data anymore – Laskowski Lab at UC Davis, Jan. 2020.
- [Law21] LAWRENCE J.: Why Was a Major Study on Ivermectin for COVID-19 Just Retracted?, July 2021.
- [LH19] LI H., HUANG J.: Localization of Deep Inpainting Using High-Pass Fully Convolutional Network. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (Seoul, Korea (South), Oct. 2019), IEEE, pp. 8300–8309. doi:10.1109/ICCV.2019.00839.
- [LMD*20] LASKOWSKI K. L., MODLMEIER A. P., DEMARCO A. E., COLEMAN A., ZHAO K., BRITTINGHAM H. A., MCDERMOTT D. R., PRUITT J. N.: Retraction: Persistent social interactions beget more pronounced personalities in a desert-dwelling social spider. *Biology Letters* 16, 2 (Feb. 2020), 20200062. doi:10.1098/rsbl.2020.0062.
- [LMP20] LASKOWSKI K. L., MONTIGLIO P.-O., PRUITT J. N.: Retraction: Individual and Group Performance Suffers from Social Niche Disruption. *The American Naturalist* 195, 2 (Feb. 2020), 393–393. doi:10.1086/708066.
- [LP20] LASKOWSKI K. L., PRUITT J. N.: Retraction: Evidence of social niche construction: Persistent and repeated social interactions generate stronger personalities in a social spider. *Proceedings of the Royal Society B: Biological Sciences* 287, 1919 (Jan. 2020), 20200077. doi:10.1098/rspb.2020.0077.
- [LWX*19a] LI D., WANG H., XU C., SHI F., MA X., LU J.: WARDER: Refining Cell Clustering for Effective Spreadsheet Defect Detection via Validity Properties. In *2019 IEEE 19th International Conference on Software Quality, Reliability and Security (QRS)* (July 2019), pp. 139–150. doi:10.1109/QRS.2019.00030.
- [LWX*19b] LI D., WANG H., XU C., ZHANG R., CHEUNG S.-C., MA X.: SGUARD: A Feature-Based Clustering Tool for Effective Spreadsheet Defect Detection. In *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)* (Nov. 2019), pp. 1142–1145. doi:10.1109/ASE.2019.00122.

- [Mil15] MILLER S. J.: *Benford's Law*. Princeton University Press, 2015.
- [Nig12] NIGRINI M. J.: *Benford's Law: Applications for forensic accounting, auditing, and fraud detection*, vol. 586. John Wiley & Sons, 2012.
- [NM09] NIGRINI M. J., MILLER S. J.: Data diagnostics using second-order tests of benford's law. *Auditing: A Journal of Practice & Theory* 28, 2 (2009), 305–324.
- [NSH*18] NIEDERER C., STITZ H., HOURIEH R., GRASSINGER F., AIGNER W., STREIT M.: TACO: Visualizing Changes in Tables Over Time. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (Jan. 2018), 677–686. doi:10.1109/TVCG.2017.2745298.
- [Ora10] ORANSKY A. I.: Why write a blog about retractions?, Aug. 2010.
- [PBL08] POWELL S., BAKER K., LAWSON B.: A critical review of the literature on spreadsheet errors. *Decision Support Systems* 46 (Dec. 2008), 128–138. doi:10.1016/j.dss.2008.06.001.
- [Pil22] PILLER C.: Potential fabrication in research images threatens key theory of Alzheimer's disease. *Science* 377, 6604 (2022), 358–363.
- [PRA*16] PASHLER H., ROHRER D., ABRAMSON I., WOLFSON T., HARRIS C. R.: A Social Priming Data Set With Troubling Oddities. *Basic and Applied Social Psychology* 38, 1 (Jan. 2016), 3–18. doi:10.1080/01973533.2015.1124767.
- [PSL21] PARK T., SONG H., LEE S. J.: Detecting and Recovering Integer Data Manipulated by Multiplication With a Nonintegral Real Number and a Rounding Operation. *IEEE Access* 9 (2021), 57149–57164. doi:10.1109/ACCESS.2021.3071794.
- [RC94] RAO R., CARD S. K.: The table lens: Merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 1994), CHI '94, Association for Computing Machinery, pp. 318–322. doi:10.1145/191666.191776.
- [RNE16] RIFE S., NUIJTEN M., EPSKAMP S.: Statcheck: Extract statistics from articles and recompute p-values [web application], 2016.
- [Shn96] SHNEIDERMAN B.: The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings of the IEEE Symposium on Visual Languages (VL '96)* (1996), pp. 336–343. doi:10.1109/VL.1996.545307.
- [SKV*20] SOSSO F. A. E., KUSS D. J., VANDELANTOTTE C., JASSO-MEDRANO J. L., HUSAIN M. E., CURCIO G., PAPADOPOULOS D., ASEEM A., BHATI P., LOPEZ-ROSALES F., BECERRA J. R., D'AURIZIO G., MANSOURI H., KHOURY T., CAMPBELL M., TOTH A. J.: Retraction Note: Insomnia, sleepiness, anxiety and depression among different types of gamers in African countries. *Scientific Reports* 10, 1 (June 2020), 9256. doi:10.1038/s41598-020-66798-w.
- [SMWH17] SATYANARAYAN A., MORITZ D., WONG-SUPHASAWAT K., HEER J.: Vega-Lite: A Grammar of Interactive Graphics. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (Jan. 2017), 341–350. doi:10.1109/TVCG.2016.2599030.
- [SSBW12] SCHULZ M.-A., SCHMALBACH B., BRUGGER P., WITT K.: Analysing Humanly Generated Random Number Sequences: A Pattern-Based Approach. *PLOS ONE* 7, 7 (July 2012), e41531. doi:10.1371/journal.pone.0041531.
- [SSN21] SIMONSOHN U., SIMMONS J., NELSON L.: [98] Evidence of Fraud in an Influential Field Experiment About Dishonesty. <https://datacolada.org/98>, Aug. 2021.
- [SWA03] SCHLEIMER S., WILKERSON D. S., AIKEN A.: Winnowing: Local algorithms for document fingerprinting. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data* (New York, NY, USA, June 2003), SIGMOD '03, Association for Computing Machinery, pp. 76–85. doi:10.1145/872757.872770.
- [The18] THE CENTER FOR SCIENTIFIC INTEGRITY: The Retraction Watch Database, 2018.
- [Tho22] THORP H. H.: Editorial Retraction. *Science* 377, 6608 (Aug. 2022), 826–826. doi:10.1126/science.ade2691.
- [TLB14] TOWSE J. N., LOETSCHER T., BRUGGER P.: Not all numbers are equal: Preferences and biases among children and adults when generating random sequences. *Frontiers in Psychology* 5 (Jan. 2014), 19. doi:10.3389/fpsyg.2014.00019.
- [Vig20] VIGLIONE G.: 'Avalanche' of spider-paper retractions shakes behavioural-ecology community. *Nature* 578, 7794 (Feb. 2020), 199–200. doi:10.1038/d41586-020-00287-y.
- [Wan19] WANG S.: Retraction: Glioma Gene Therapy Using Induced Pluripotent Stem Cell

- Derived Neural Stem Cells. *Molecular Pharmacology* 16, 9 (Sept. 2019), 4088. doi:10.1021/acs.molpharmaceut.9b00837.
- [WWO*19] WANG S.-Y., WANG O., OWENS A., ZHANG R., EFROS A. A.: Detecting Photoshopped Faces by Scripting Photoshop. *arXiv:1906.05856 [cs]* (Sept. 2019). arXiv:1906.05856.
- [YLL*20] YANG C., LI H., LIN F., JIANG B., ZHAO H.: Constrained R-CNN: A general image manipulation detection model. *arXiv:1911.08217 [cs]* (Mar. 2020). arXiv:1911.08217.
- [ZDZ*20] ZHANG Y., DOU W., ZHU J., XU L., ZHOU Z., WEI J., YE D., YANG B.: Learning to detect table clones in spreadsheets. In *Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis* (New York, NY, USA, July 2020), ISSTA 2020, Association for Computing Machinery, pp. 528–540. doi:10.1145/3395363.3397384.

Ferret: Reviewing Tabular Dataset for Manipulations

Supplementary Material

Devin Lange, Shaurya Sahai, Jeff M. Phillips, Alexander Lex

Case Study: DS-Driving

Retraction: <https://doi.org/10.1073/pnas.2115397118>

Blog: <http://datacolada.org/98>

This psychology study claims that signing an honesty pledge at the top of a document leads to more honest reporting than at the bottom. This dataset is from an experiment that asked participants to report the odometer mileage of their car both before and after some period of time.

The dataset is sorted into rows. Each row corresponds to an insurance policy number. Each policy can have between 1 and 4 cars on it. As a result, there are four sets of before/after odometer columns.

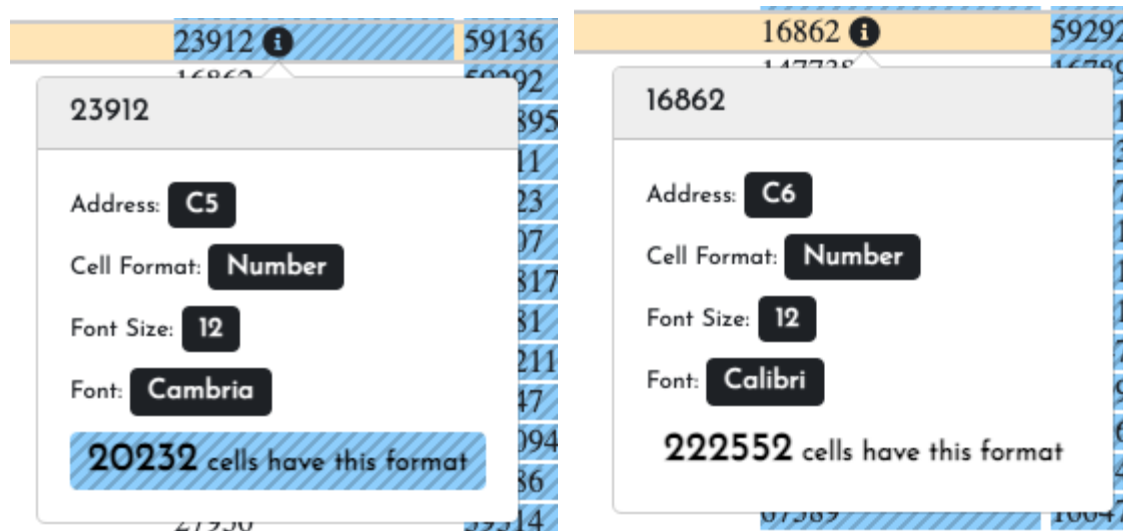
Half of the **rows** appear to be generated by adding a small amount of noise to the original values. In addition, “after” **columns** appear to be generated by adding a random number between 0 and 50,000 to the “before” number.

Formatting artifacts indicate fabricated rows

On initial load, it is evident that there is some different styling in the odometer reading for the first two cars.

OMR Version	Policy # (masked)	Odom Reading 1 ...	Odom Reading 1 ...	Odom Reading 2...	Odom Reading 2...
↓ ↑ ***	↓ ↑ ***	↓ ↑ ***	↓ ↑ ***	↓ ↑ ***	↓ ↑ ***
Sign Top	1	896	39198		
Sign Bottom	2	21396	63511	32659	47605
Sign Bottom	3	21340	37460	44998	59002
Sign Bottom	4	23912	59136		
Sign Bottom	5	16862	59292		
Sign Top	6	147738	167895	125820	164688
Sign Bottom	7	18780	49811	45402	54824
Sign Top	8	41930	80323	181416	229852
Sign Top	9	28993	63707	13291	28165
Sign Bottom	10	78382	127817		
Sign Top	11	58500	81081		
Sign Bottom	12	99417	149211	48770	95179
Sign Bottom	13	93231	98047		
Sign Bottom	14	83443	105094		
Sign Bottom	15	22008	26486		
Sign Bottom	16	27950	59514	95883	126309
Sign Bottom	17	67589	100475	27617	74443
Sign Bottom	18	32753	76724		
Sign Top	19	33044	70775		
Sign Bottom	20	104857	109961	19548	47796
Sign Bottom	21	121699	137849		
Sign Top	22	16094	45489	159167	200316
Sign Top	23	78182	122739	21730	37863
Sign Bottom	24	21735	58693	39666	77258
Sign Top	25	47473	68971	4502	47293
Sign Bottom	26	121416	123661	53987	86852
Sign Bottom	27	4616	30597		
Sign Bottom	28	13604	31750		
Sign Bottom	29	125000	139801		
Sign Top	30	40463	70095		
Sign Top	31	34823	78262		
Sign Top	32	120296	137162	138617	169196
Sign Top	33	79173	125933		
Sign Bottom	34	110372	142100		
Sign Bottom	35	3586	41419		
Sign Top	36	26017	60347	12996	17179
Sign Top	37	149000	177873		
Sign Top	38	15939	17025		

On closer inspection, the difference is due to a difference in font between Cambria (Blue), and Calibri (White, no highlight).

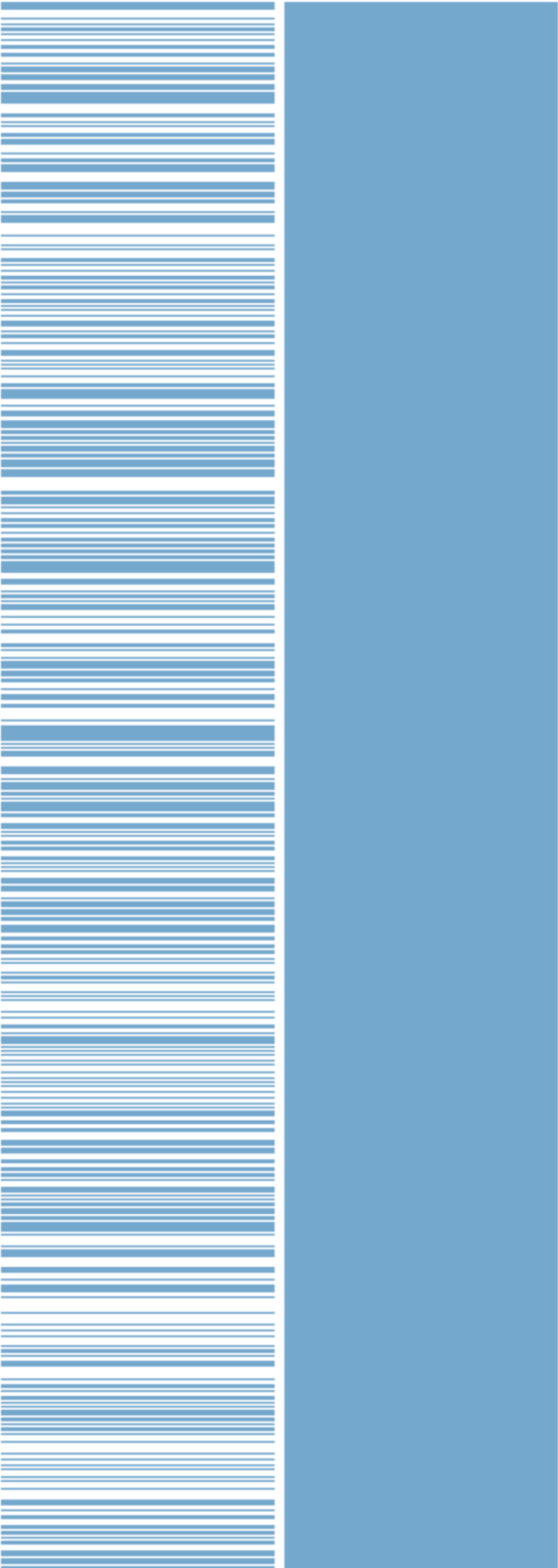


After switching to the Structural Overview, the pattern appears to continue through the entire 13,488 rows of the table:

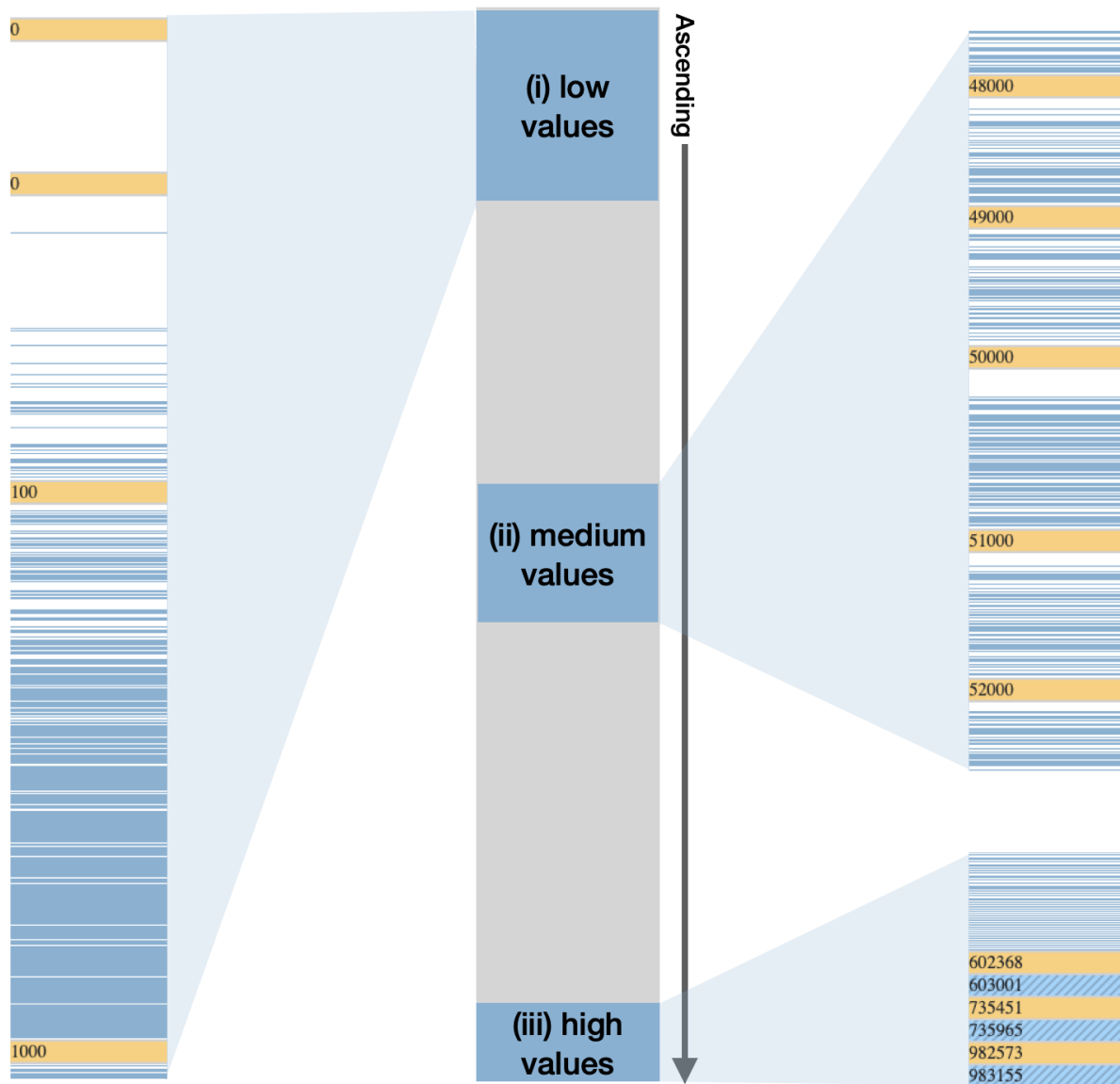
Odom Reading 1 ... Odom Reading 1 ...

↓ $\frac{1}{9}$ ***

↓ $\frac{1}{9}$ ***



However, sorting by the odometer 1 reading results in several interesting patterns:



The values under 100 are almost entirely **Calibri font**. The values between 100 and 1000 are predominantly in **Cambria**.

Throughout the column, there are regular chunks of **Calibri** only font. These appear around round numbers. **Cambria** does not have the same large repeated regions around these large numbers. Larger runs of **Cambria** appear to be spurious (see below), and do not contain repeated numbers.

The largest values of the column you see rows altering back and forth between **Calibri** and **Cambria**.

[illegible]

Expanding the rows provides more detail. On closer inspection, it appears that every **Calibri** value has a **Cambria** equivalent that is within 1000 values.

Odom Reading 1 ... ↓ ↑ ***	Odom Reading 2... ↓ ↑ ***	Odom Reading 3... ↓ ↑ ***	Odom Reading 4... ↓ ↑ ***
299987			
300000		6000	
300525		6987	
304000			
304448			
309142	900	66000	
309564	1487	66917	
313668	5270	85217	
314174	6219	86073	
318102			
318458			
327774	55914		
328396	56341		
328549			
328981			
340232			
340638			
343000			
343935			
348285			
348702			
358236	111823	40000	
358544	112660	40845	
359641			
359700			
364774	112123	48472	
365387	112247	49086	
394482			
395272			
402847			
403733			
409515	31134	95000	
409663	31578	95013	
416537	48813	118579	
417041	48826	119477	
443920			
444290			
463090			
463284			
602368	152327	130210	152600
603001	153284	130947	153254
735451	99735	163390	
735965	100512	163756	
982573			
983155			

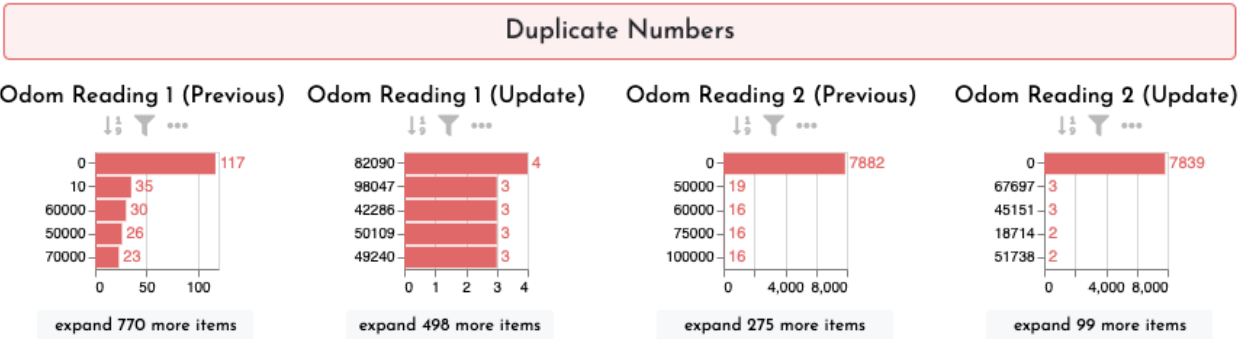
This is especially strange if you sort by the number of cars in the policy so you can easily see all of the policies with four cars. For each row with four cars in **Calibri**, there is an equivalent in **Cambria** where the odometer reading is within 1000 miles for each car.

Odom Reading 1 ... 2 ↓ 1/9 ***	Odom Reading 2... ↓ 1/9 ***	Odom Reading 3... ↓ 1/9 ***	Odom Reading 4... ↓ 1/9 ***
735965	100512	163756	
0	120000	125000	146000
13	130240	37910	80791
845	131045	38591	80980
935	120126	125099	146367
1053	134778	175000	132000
1195	135427	175847	132596
8907	104849	35094	91640
9058	105406	35642	92607
10111	145650	176230	147569
10991	145902	176424	148268
11652	71000	13938	17911
12633	71384	13946	18711
14437	13640	17879	33864
14846	13821	18864	33985
17330	106000	43218	104591
18235	106253	43457	104916
18904	13024	103791	96954
19827	13425	103939	97538
34114	64000	34000	98885
34840	64523	34667	99180
41279	73641	45283	112415
41588	73757	46236	112457
47600	6500	15000	39000
47951	6901	15105	39364
49675	17709	27357	64428
50350	18421	27714	64784
51016	244058	120336	172906
51046	244307	120958	173372
57000	123663	16000	90000
57640	123666	16469	90026
58826	244390	122407	176373
59132	127063	26508	105626
59535	245203	123282	176947
59977	127872	27090	106443
89027	30	169777	143537
89625	1006	170410	143617
90367	14781	170958	147750
91079	15425	171105	147822
128392	124477	87000	14255
128516	124659	87127	14862
128628	132997	88688	145681
129585	133119	89193	146241
602368	152327	130210	152600
603001	153284	130947	153254

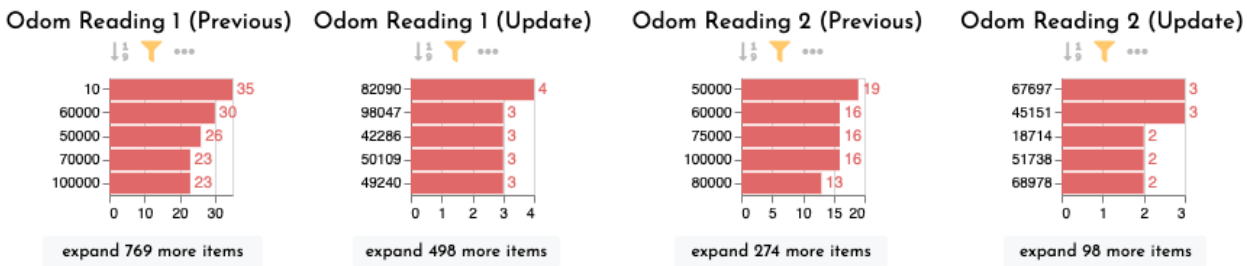
These attributes indicate that rows were copied to a temporary worksheet, then increased by a random noise function between zero and 1000, and added back into the original worksheet.

Numerical artifacts and deviations from domain expectations indicate fabricated columns

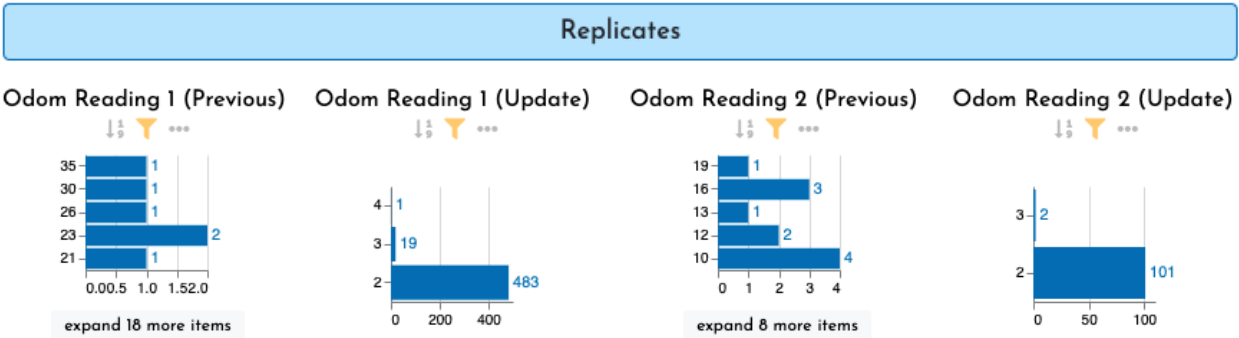
Moving beyond the formatting to the numerical artifacts. There appears to be a difference in how many duplicates are in a column in the **Previous** column compared to the **Update** column. The previous column contains many duplicates of round numbers such as 50000.



Since some blank fields are appearing as zero here, we can choose to ignore zero from our analysis to make the relevant data more clear.

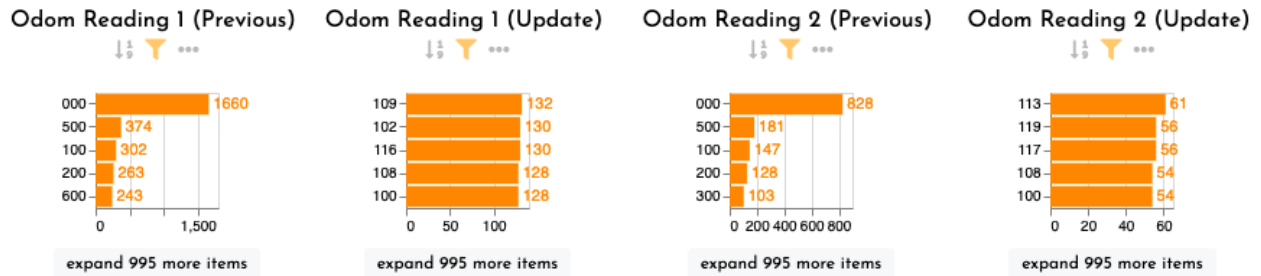


In addition, to which numbers are duplicated a lot, we can see how many times a number has been duplicated in the replicates chart.



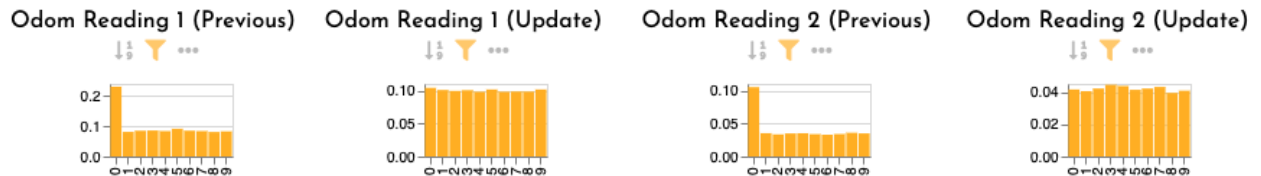
In addition, to duplicate numbers, looking at duplicate sequences of digits again reveals that the **Previous** column includes the digits “000” much more frequently than the **Update** column.

Duplicate Digits



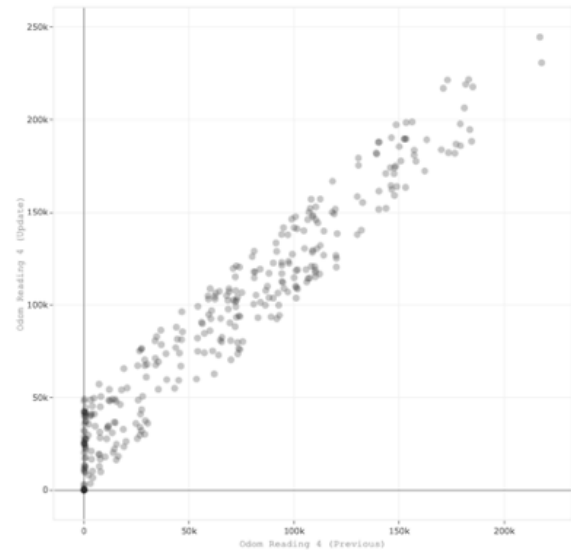
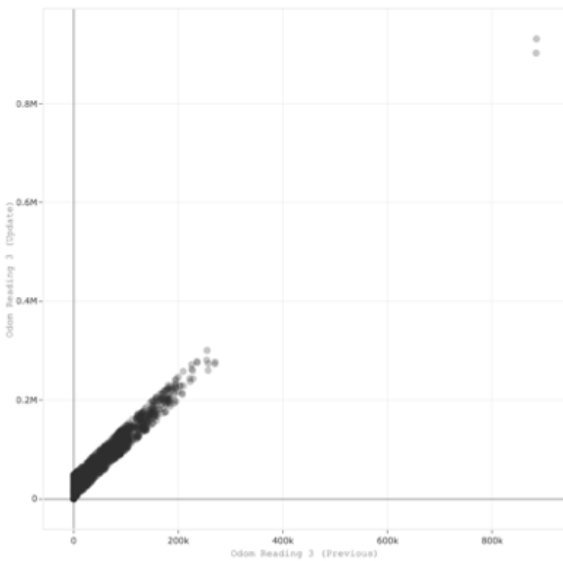
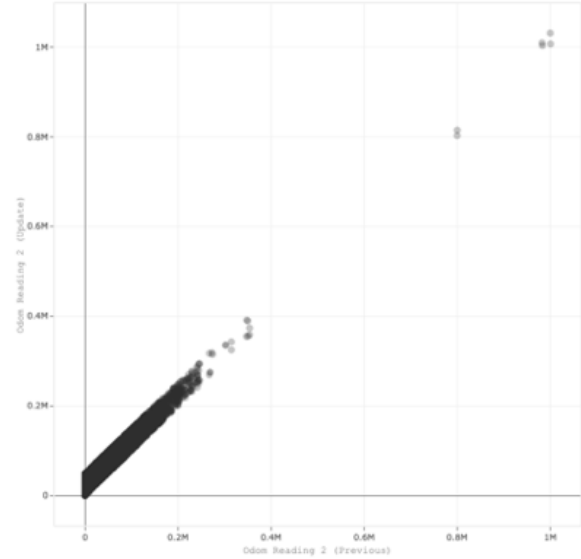
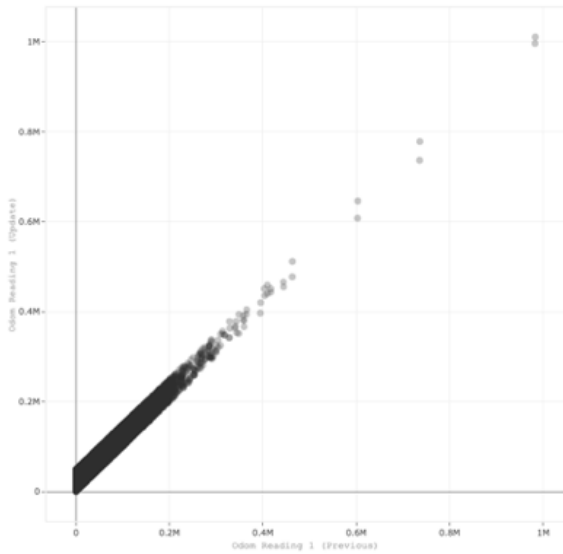
Finally, if we look at just the final trailing digit of each number, you can again see the much higher frequency of zero in the **Previous** column compared to the **Update** column.

Trailing Digits



With all of these charts, it is clear that there is a rounding effect present in the **Previous** column but not in the **Update** column. This may lead you to question the relationship between the Previous and Update column.

With the general visualization analysis tool in Ferret, it is easy to plot scatter plots of these two columns for the four different pairs of columns.



These plots reveal a strange correlation between the Previous and Update columns. That is, the miles driven falls below 50,000 miles.

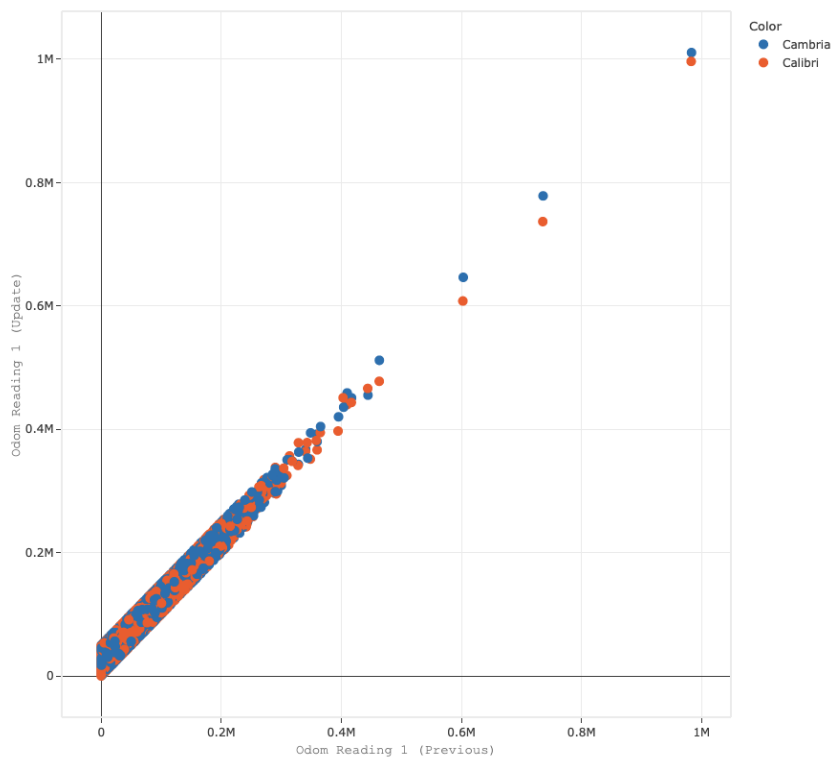
It is believed that the Update column was generated by adding a random number between zero and 50,000 miles.

Interestingly, this plot also hints at the duplicated rows. Examining the tails closely will reveal that points are always grouped in pairs. There is always at least one point within 1000 miles in the x-axis.

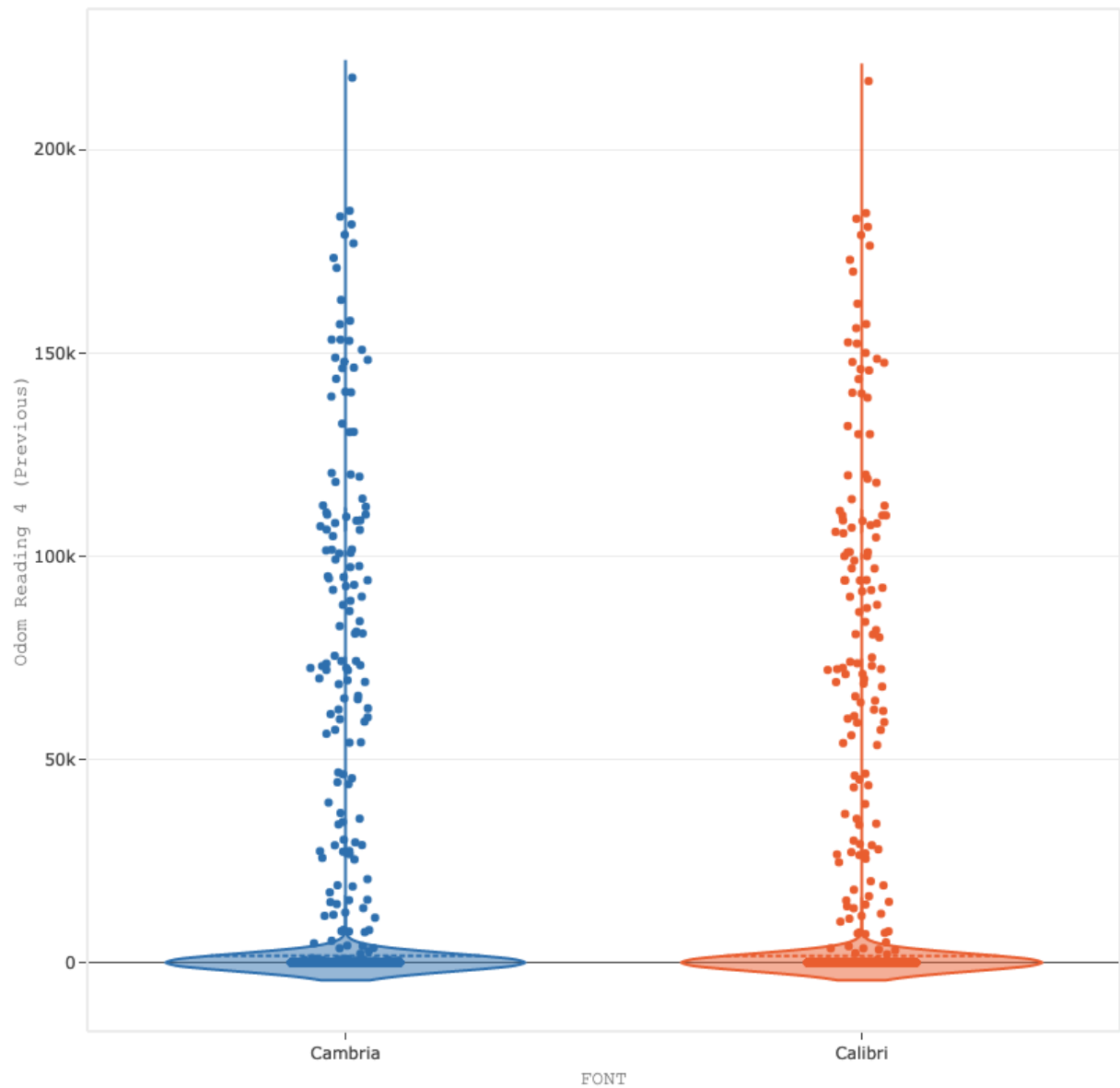
Bonus, including font as a column

All analysis prior to this, was done with the original excel data sheet within Ferret. Adding a categorical column to track the font and loading it back into Ferret provides a few more options.

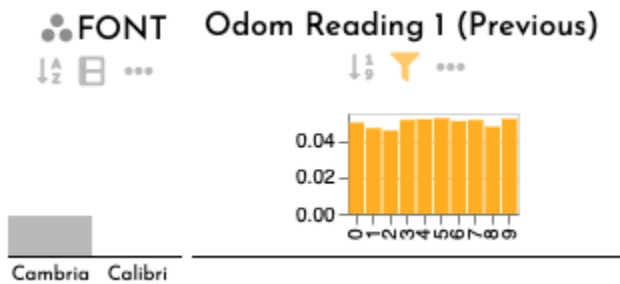
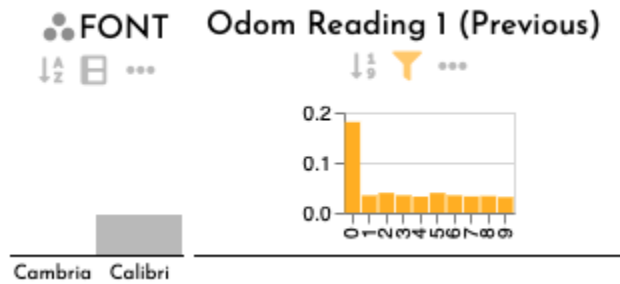
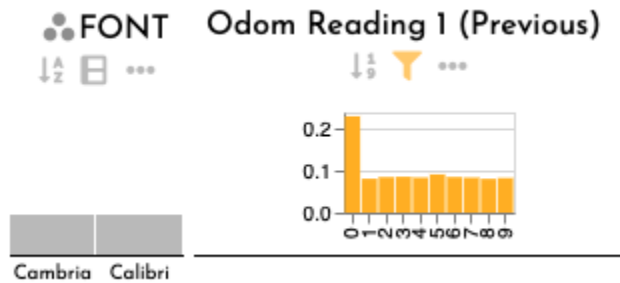
First, the observations of pairs at the extreme are strengthened by the fact that the pairs always include one **Calibri** formatted cell and one **Cambria** formatted cell.



Furthermore, if we plot a violin plot of the previous column faceted by the font, you see they are extremely similar, again strengthening the hypothesis that these data are nearly copies.



Lastly, we can also observe the lack of rounding effects in the Cambria plot using the trailing digit frequency visualization combined with dynamically filtering out one font compared to another.



Case Study: DS-Gaming

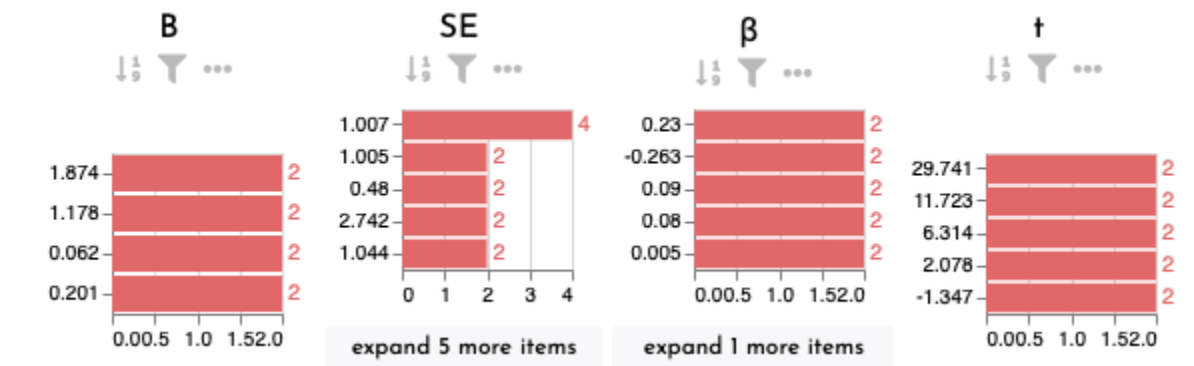
Retraction: <https://doi.org/10.1038/s41598-020-66798-w>

Blog: <http://steamtraen.blogspot.com/2020/04/some-issues-in-recent-gaming-research.html>

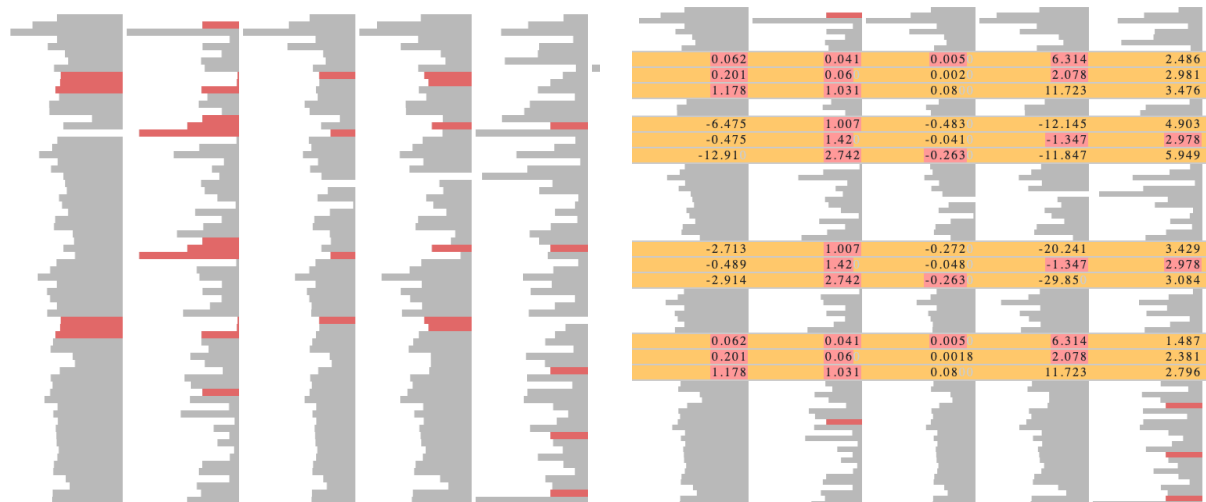
This study looked for a relationship between video gaming habits and sleep habits. A survey was sent over email asking about video gaming habits, demographic information, and sleeping habits. The paper contains a table with summary statistics based on survey responses. Nick Brown, the author of the blog associated with this dataset converted the table into an excel file which we have utilized.

Repeated Regions

Since this table only contains 68 rows in total, the amount of duplicate numbers is a bit high, though it may not be enough to be conclusive on its own.



After highlighting some numbers, however, repeated regions become more clear.



Case Study: DS-Covid

Retraction: <https://grfr.news/why-was-a-major-study-on-ivermectin-for-covid-19-just-retracted/>

Blog: <http://steamtraen.blogspot.com/2021/07/Some-problems-with-the-data-from-a-Covid-study.html>

This dataset collected data on how effective and safe ivermectin is for testing Covid-19.

Unexpected Formatting

This dataset contains many instances of unexpected formatting. Excluding the cell data format from the formatting highlighting makes it easier to identify formatting discrepancies, such as outlier fonts:

220	discharged	SFM	arged	SFM	F
221	discharged	AG	arged	AG i	F
222	discharged	MA	arged	MA	M
223	discharged	MAG	arged	AG	F
224	discharged i	AKE	arged		F
225	discharged	EEE	arged		M
226	discharged		arged		M
227	discharged		arged		M
228	discharged		arged		F
229	discharged		arged		F
230	discharged		arged		M
231	discharged		arged		M
232	discharged		arged		M
233	discharged		arged		F
234	discharged		arged		F
235	discharged		arged		F
236	discharged		arged		F
237	discharged		arged		F
238	discharged		arged		F
239	discharged		arged		F
240	discharged		arged		F
241	discharged		arged		F
242	discharged		arged		F
243	discharged		arged		F
244	discharged		arged		F
245	discharged		arged		F
246	discharged		arged		F
247	discharged		arged		F
248	discharged		arged		F
249	discharged		arged		F
250	discharged		arged		F
251	discharged		arged		F
252	discharged		arged		F
253	discharged		arged		F
254	discharged		arged		F
255	discharged		arged		F
256	discharged		arged		F
257	discharged		arged		F
258	discharged		arged		F
259	discharged		arged		F
260	discharged		arged		F
261	discharged		arged		F
262	discharged		arged		F
263	discharged		arged		F
264	discharged		arged		F
265	discharged		arged		F
266	discharged		arged		F
267	discharged		arged		F
268	discharged		arged		F
269	discharged		arged		F
270	discharged		arged		F
271	discharged		arged		F
272	discharged		arged		F
273	discharged		arged		F
274	discharged		arged		F
275	discharged		arged		F
276	discharged		arged		F
277	discharged		arged		F
278	discharged		arged		F
279	discharged		arged		F
280	discharged		arged		F
281	discharged		arged		F
282	discharged		arged		F
283	discharged		arged		F
284	discharged		arged		F
285	discharged		arged		F
286	discharged		arged		F
287	discharged		arged		F
288	discharged		arged		F
289	discharged		arged		F
290	discharged		arged		F
291	discharged		arged		F
292	discharged		arged		F
293	discharged		arged		F
294	discharged		arged		F
295	discharged		arged		F
296	discharged		arged		F
297	discharged		arged		F
298	discharged		arged		F
299	discharged		arged		F
300	discharged		arged		F

By including the data format of cells in the styling criteria it is easier to spot issues with the actual malformed data. Such errors and inconsistencies are immediately obvious in the detailed view of the table.

TLC (X 103 ↓ ₉ ...	lymph. % ↓ ₉ ...	symptoms date&... ↓ ₉ ...	recovery date & -... ↓ ₉ ...
4	0.15	Wed Aug 05 2020 ...	Sat Dec 05 2020 1...
4.7	0.14	Wed Aug 05 2020 ...	Thu Nov 05 2020 ...
5.2	0.16	Sat Sep 05 2020 1...	Sat Dec 05 2020 1...
Fri Jan 05 1900 00...	0.15	Sat Sep 05 2020 1...	14/6/2020
6.8	0.14	Sat Sep 05 2020 1...	13/6/2020
5.2	0.15	Mon Oct 05 2020 ...	13/6/2020
6.7	0.2	Mon Oct 05 2020 ...	14/6/2020 5 days
7.2	0.17	Mon Oct 05 2020 ...	15/6/2020
8.9	0.15	Thu Nov 05 2020 ...	14/6/2020
7.3	0.16	Thu Nov 05 2020 ...	16/6/2020
6.4	0.17	Thu Nov 05 2020 ...	16/6/2020
6.4	0.18	Thu Nov 05 2020 ...	15/6/2020
5.8	0.2	Thu Nov 05 2020 ...	15/6/2020
6.2	0.17	Sat Dec 05 2020 1...	17/6/2020 6 days
7.4	0.15	Sat Dec 05 2020 1...	17/6/2020
9.0	0.18	Sat Dec 05 2020 1...	17/6/2020
8.4	0.16	Sat Dec 05 2020 1...	16/6/2020
6.4	0.2	13/6/2020	16/6/2020
7.2	0.19	13/6/2020	17/6/2020
5.9	0.16	13/6/2020	16/6/2020
7.4	0.15	13/6/2020	18/6/2020
8.3	0.18	14/6/2020	17/6/2020 4 days
7.5	0.2	14/6/2020	17/6/2020
8.4	0.2	14/6/2020	19/6/2020
5.9	0.17	13/6/2020	18/6/2020
8.4	0.18	14/6/2020	19/6/2020
7.9	0.18	15/6/2020	20/6/2020
8.4	0.16	15/6/2020	20/6/2020
7.2	0.15	15/6/2020	19/6/2020
7.4	0.14	15/6/2020	19/6/2020
8.2	0.19	15/6/2020	18/6/2020
8.4	0.2	15/6/2020	18/6/2020
7.9	0.16	16/6/2020	19/6/2020
8.4	0.15	16/6/2020	19/6/2020
4.9	0.17	16/6/2020	20/6/2020
5.9	0.18	16/6/2020	20/6/2020
7.3	16.00%	17/6/2020	20/6/2020
6.8	0.2	17/6/2020	20/6/2020
7.2	0.18	17/6/2020	22/6/2020

The overview mode can also be helpful in reviewing how many errors of this kind exist. For instance, here, the left column is recording date values. Orange rows are correctly formatted as a date in Excel, whereas white and yellow rows are strings. In this figure, roughly half of the rows are recorded correctly.

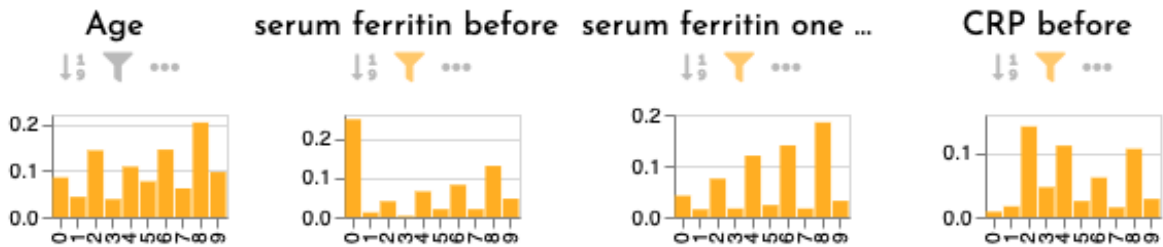
The right column is recording numerical values. The teal rows are correctly formatted as numbers, and the white ones are invalid numbers formatted as strings, such as “9.0%”



These types of errors are likely the result of entering data into a spreadsheet manually.

Unexpected Trailing Digit

There are four columns within this dataset that show a strange preference for even numbers over odd ones. This can be in the following Trailing Digit visualizations. The blog for this post dataset does mention asymmetry in odd/even values for the **Age** column, **however, it does not mention it for the other three columns identified by Ferret.**



DS-Spider: Dataset Description

Blog: <https://laskowskilab.faculty.ucdavis.edu/2020/01/29/retractions/>

The three spider studies share some common authors and were retracted in the same wave. The three datasets are related to each other but have different structures and attributes.

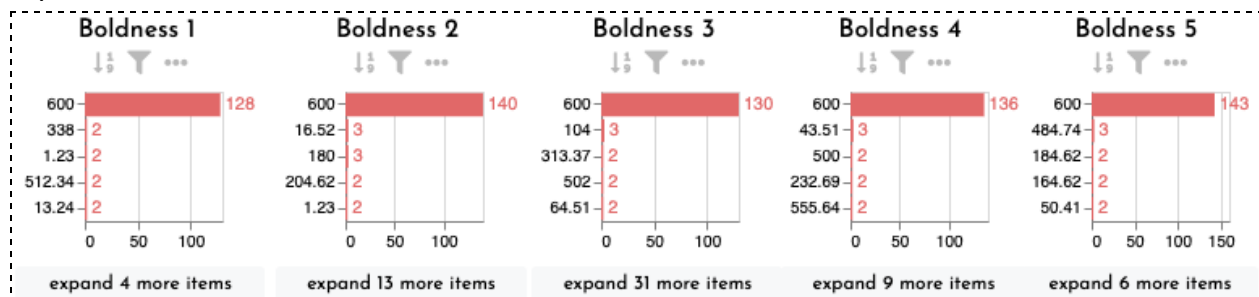
All of the datasets include a “boldness” of spiders. This “boldness” was measured by recording how long it will take spiders to reemerge from their enclosure after a simulated predator attack.

Case Study: DS-Spider-E

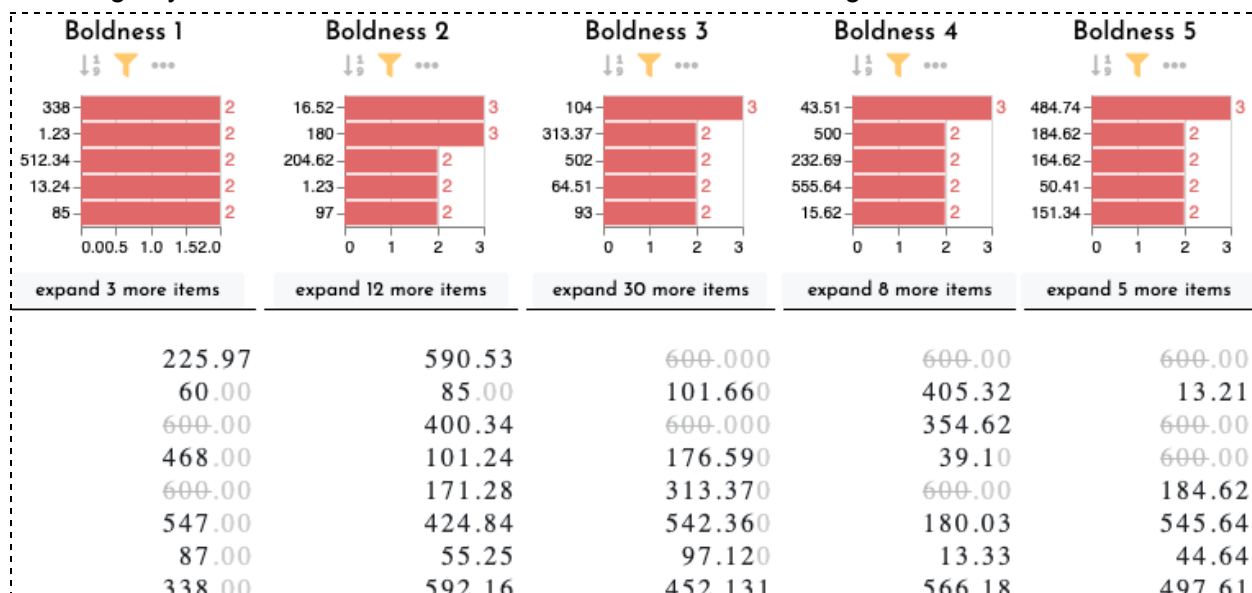
Retraction: <https://doi.org/10.1098/rspb.2020.0077>

Duplicate Numbers Artifact

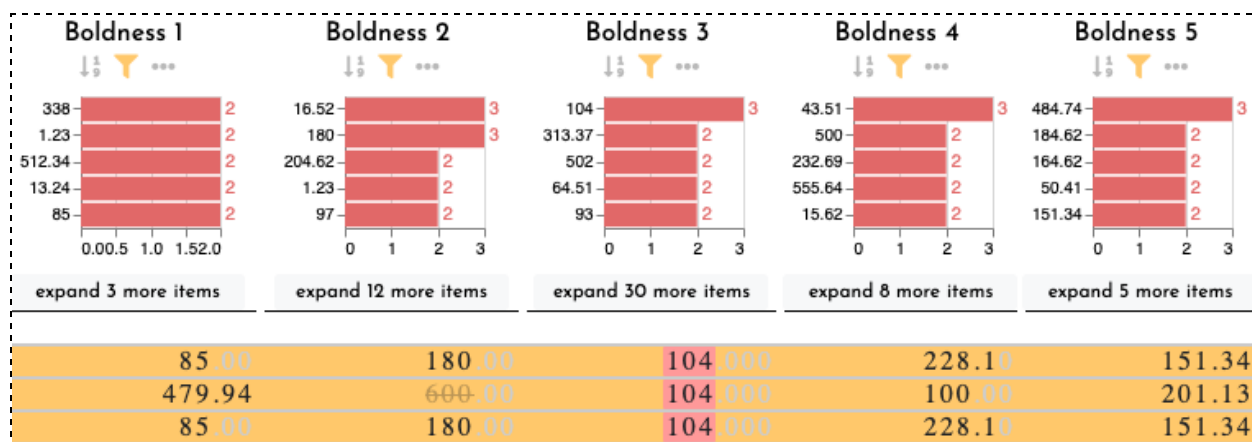
The five **Boldness** columns in this dataset, all show a very large number of 600s. Since these are time measurements, 600 seconds corresponds to 10 minutes, the maximum amount of time they waited for a spider to reemerge. In other words, there is a reasonable explanation for these duplicates.



Their presence, however makes it difficult to look for other datasets. Ignoring 600 globally removes it from the analysis and strikes out the 600s in the table view. This is different from removing any row that contains a 600 as that would remove a large relevant data to examine.

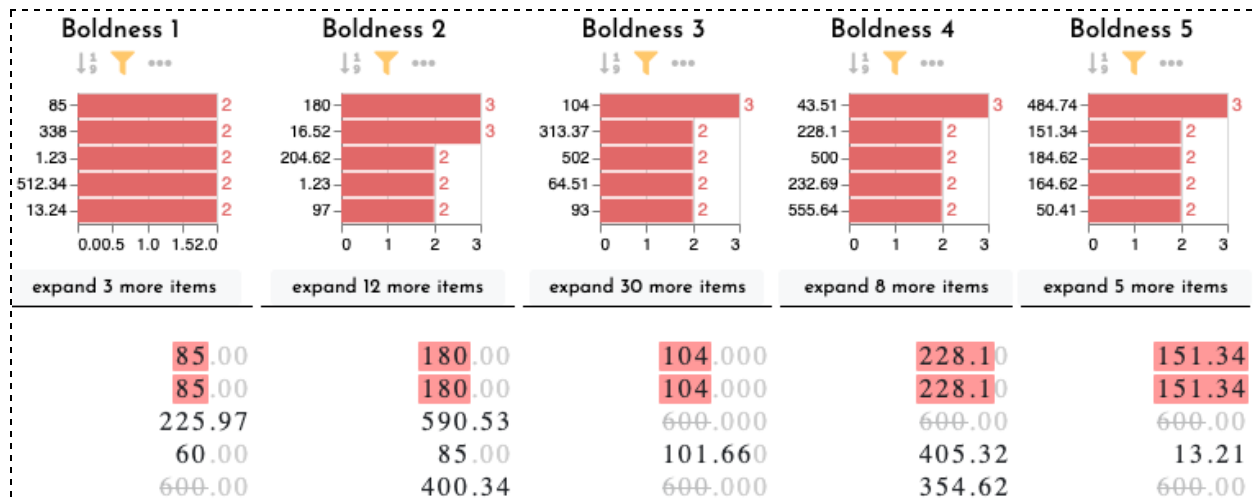


The amount of remaining duplicates is still large for this dataset of 350 rows, especially with the two degrees of precision listed. Highlighting 104 in the **Boldness 3** column makes it easy to examine the neighborhood of cells.



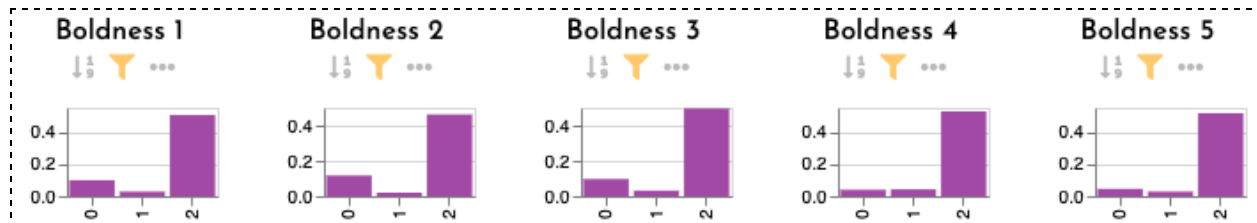
Repeated Regions Artifacts

Highlighting more values makes it more obvious that at least one duplicated row exists in this dataset.



Unexpected Varied Precision Artifacts

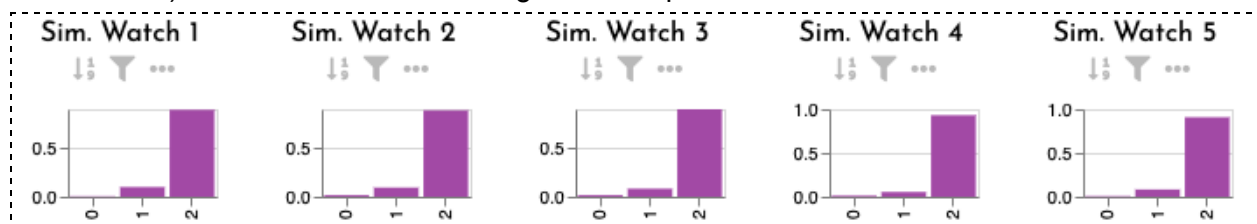
The precision analysis reveals an interesting pattern. There are more values with 0 digits of precision than there are with 1 digit of precision. Since these are time measurements, you would expect most values to have two digits of precision (e.g. 3.12 seconds), less with one digit of precision (e.g. 3.1 seconds), and very few to have zero digits of precision (e.g. 3 seconds).



This anomaly is not due to the large numbers of 600s. The chart above is ignoring 600s. The chart below is what it would look like with 600s included.



The expected results are easy to simulate. The charts below are created from an excel spreadsheet that used **round(rand(),2)** in five columns by 350 rows (the same dimensions as the real data) to simulate the last two digits of a stopwatch.



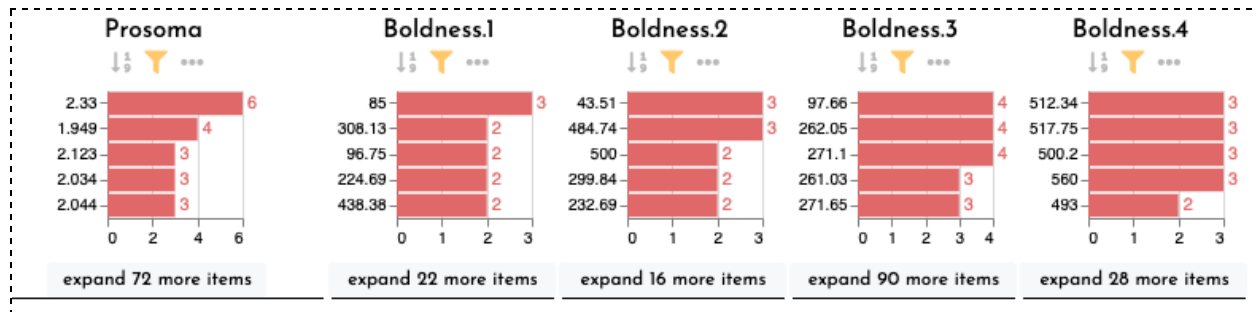
This unexpected distribution of precision is not mentioned in any blog posts.

Case Study: DS-Spider-P

Retraction: <https://doi.org/10.1098/rsbl.2020.0062>

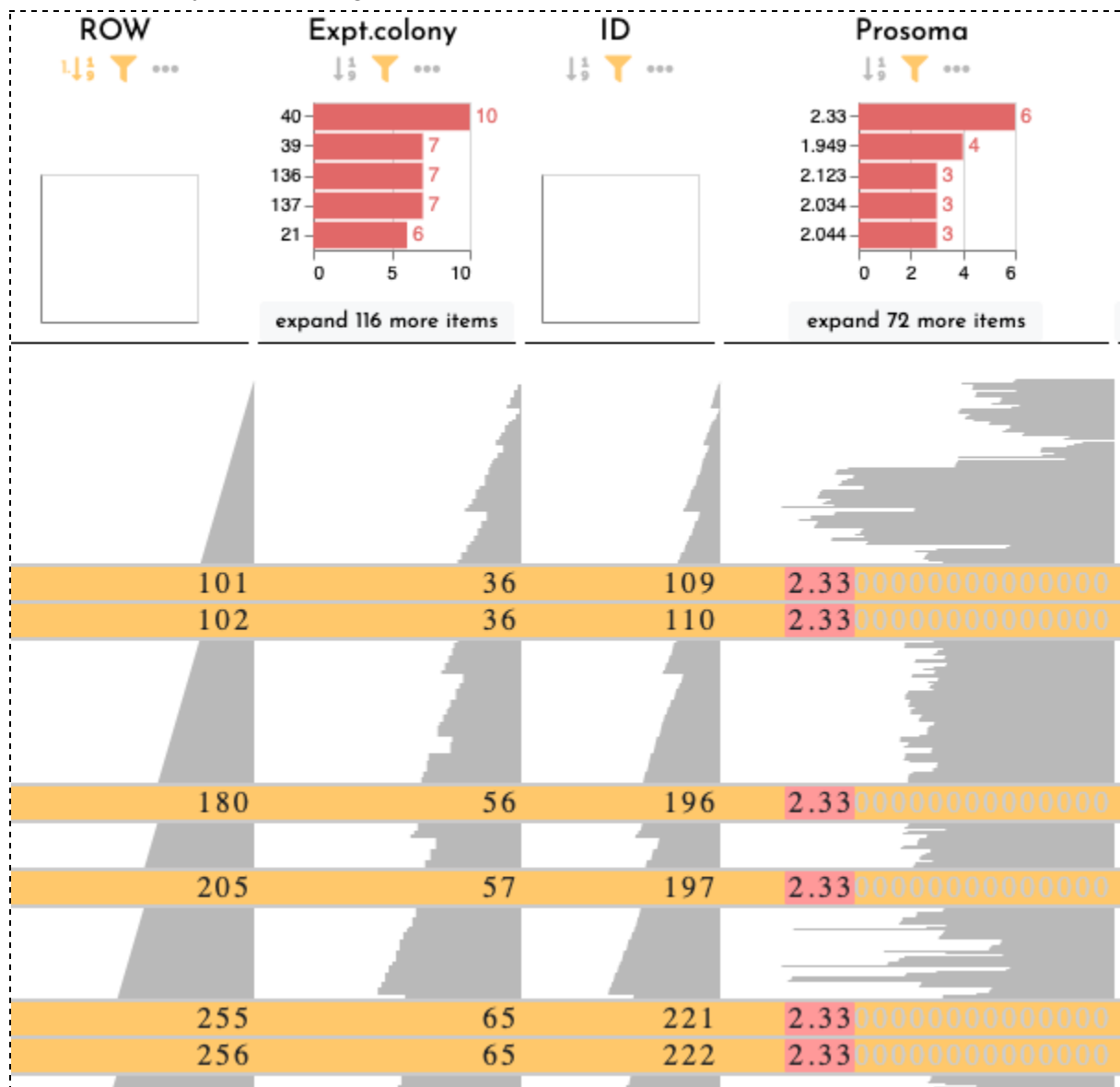
Duplicate Numbers Artifact

Similar to **DS-Spider-E** there is a duplicate numbers artifact in this dataset of 479 rows. Again 600 is ignored from the analysis.



Repeated Regions Artifact

Highlighting the most duplicated value (2.33) in the **Prosoma** columns shows that these values are from actually repeated regions.

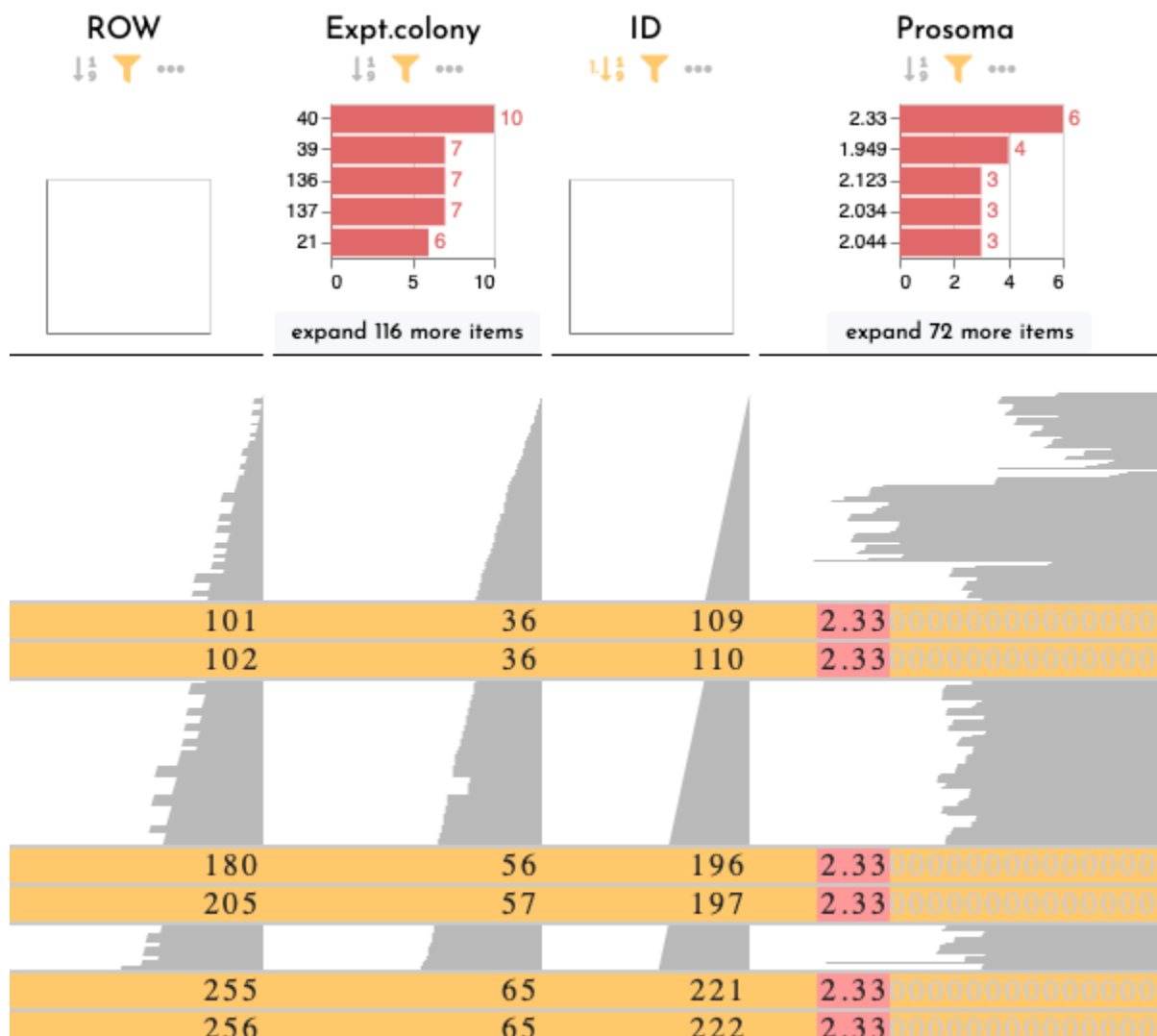


Ordering Artifacts.

The image above also illustrates an ordering artifact in this dataset. The **ROW** column is the original row value in the dataset (automatically inserted by Ferret). The **Expt.colony** and **ID** columns share a strange relationship with **ROW**.

Sorting by the ID column reveals that 2.33 appears even more like a repeated region with the data sorted by ID (which it is likely to have been at some point). In addition, this sort shows that

ID and **Expt.colony** also share a strange relationship with each other. **This ordering artifact is not mentioned in any blog posts.**



Case Study: DS-Spider-I

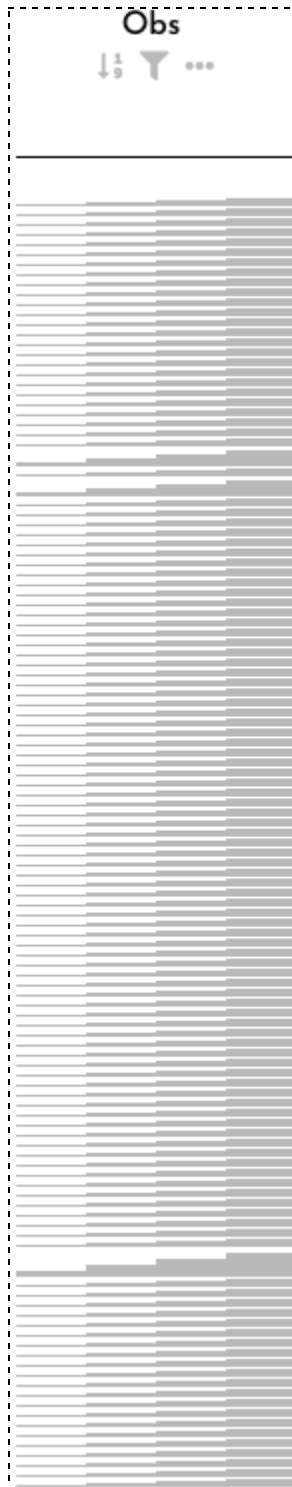
Retraction: <https://doi.org/10.1086/708066>

This dataset is still related to the same boldness measurement of spiders. However, it is formatted differently than the previous two — it is in long format. Before each row contained all of the boldness scores for a single spider. In this long format, each row corresponds to one observation. So there are 5 rows of observation for each spider.

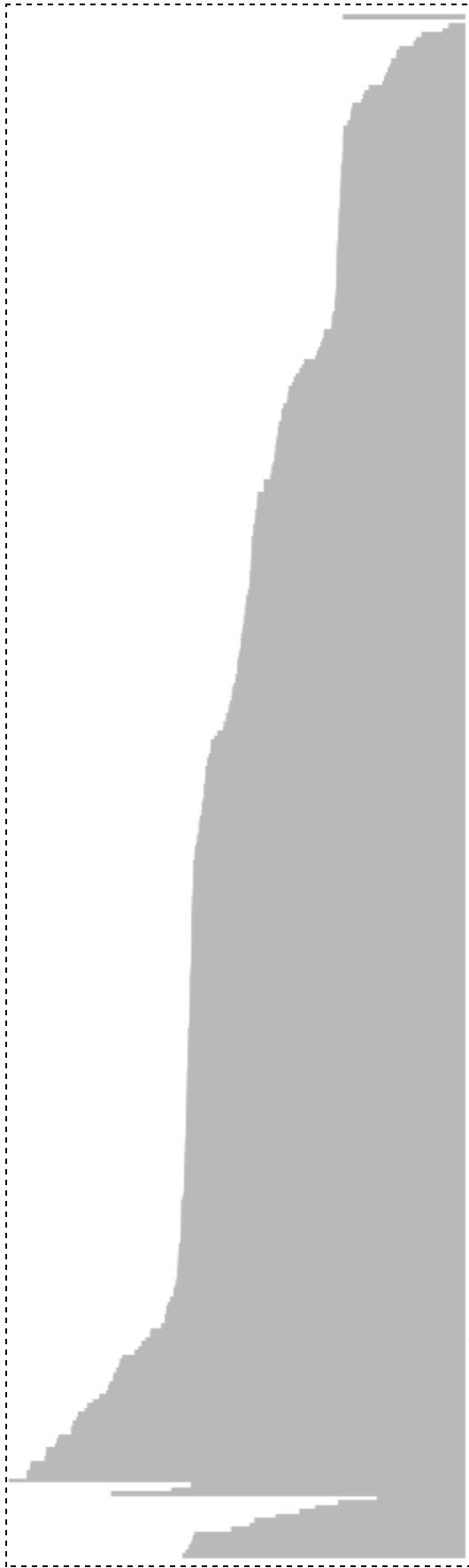
Ordering Artifacts

There are a few unusual ordering patterns in this dataset.

The **obs** column generally follows a repeated structure, of 1,2,3,4,5. Alternating through the 5 observations for each individual spider. However, this pattern is broken several times.



There is also unexplained ordering in the **Percent.mass.change** column.



This column is monotonically increasing for a majority of the dataset with the exception of the very first value, and the last few values. It is not clear how this ordering could occur. **Neither of these ordering artifacts are mentioned in any blog post.**

Duplicate Numbers Artifacts

Similar to the other spider datasets, the boldness columns contain many duplicates. Since this dataset is in long format, there are only two columns. **Pre.boldness** and **Post.boldness**. With 1745 rows.



Repeated Regions

We know that this dataset contains repeated regions thanks to the blog post written about it by one of the co-authors of the retracted paper. That blog mentions the duplicate numbers found in the data in this form, then describes how when the wide formatted version of this data contained repeated regions. Since Ferret does not support the ability to convert between long and wide data formats we were not able to identify this known artifact in this dataset.

Case Study: DS-Glioma

Retraction: <https://doi.org/10.1021/acs.molpharmaceut.9b00837>

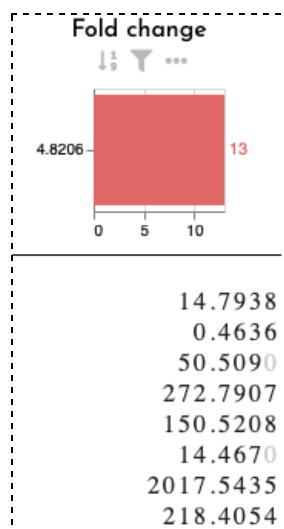
Blog: NA - found via retraction watch.

This dataset contains numerical measurements of **fold change** of mouse embryonic stem cells.

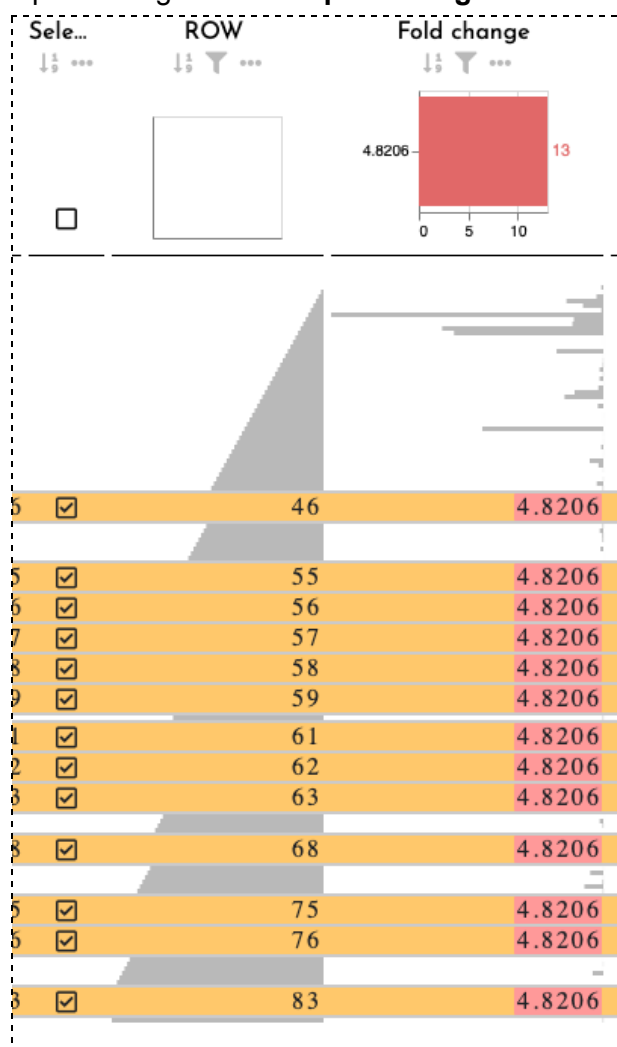
When loading this dataset, it is evident there are a few outlier cells with regard to formatting.

Formatting artifacts

Agg...	Rank	Sele...	ROW	Accession	Symbol	Description	Fold change	Functional Gene ...	Functional Gene Grouping (II)
...
		<input type="checkbox"/>							
1		<input type="checkbox"/>	1	NM_010425	Foxd3	Forkhead box D3	14.7938	ESC-specific genes	Transcription factors maintaining "stem-ness"
2		<input type="checkbox"/>	2	NM_010258	Gata6	GATA binding prot...	0.4636	ESC-specific genes	Transcription factors maintaining "stem-ness"
3		<input type="checkbox"/>	3	NM_010262	Gbx2	Gastrulation brain ...	50.509	ESC-specific genes	Transcription factors maintaining "stem-ness"
4		<input type="checkbox"/>	4	NM_028016	Nanog	Nanog homeobox	272.7907	ESC-specific genes	Transcription factors maintaining "stem-ness"
5		<input type="checkbox"/>	5	NM_030676	Nr5a2	Nuclear receptor s...	150.5208	ESC-specific genes	Transcription factors maintaining "stem-ness"
6		<input type="checkbox"/>	6	NM_010264	Nr6a1	Nuclear receptor s...	14.467	ESC-specific genes	Transcription factors maintaining "stem-ness"
7		<input type="checkbox"/>	7	NM_013633	Pou5f1	POU domain, class...	2017.5435	ESC-specific genes	Transcription factors maintaining "stem-ness"
8		<input type="checkbox"/>	8	NM_011443	Sox2	SRY-box containin...	218.4054	ESC-specific genes	Transcription factors maintaining "stem-ness"
9		<input type="checkbox"/>	9	NM_023755	Tcfcp2l1	Transcription facto...	228.8357	ESC-specific genes	Transcription factors maintaining "stem-ness"
10		<input type="checkbox"/>	10	NM_009482	Utr1	Undifferentiated e...	1194.5975	ESC-specific genes	Transcription factors maintaining "stem-ness"
11		<input type="checkbox"/>	11	NM_009556	Zfp42	Zinc finger protein 42	1106.6193	ESC-specific genes	Transcription factors maintaining "stem-ness"
12		<input type="checkbox"/>	12	NM_147778	Commd3	COMM domain co...	0.3192	ESC-specific genes	Signalling molecules required for pluripotency and self-Renewal
13		<input type="checkbox"/>	13	NM_007759	Crabp2	Cellular retinoic ac...	0.3381	ESC-specific genes	Signalling molecules required for pluripotency and self-Renewal
14		<input type="checkbox"/>	14	NM_007904	Ednrb	Endothelin recepto...	0.1798	ESC-specific genes	Signalling molecules required for pluripotency and self-Renewal
15		<input type="checkbox"/>	15	NM_010202	Fgf4	Fibroblast growth f...	346.5816	ESC-specific genes	Signalling molecules required for pluripotency and self-Renewal
16		<input type="checkbox"/>	16	NM_010203	Fgf5	Fibroblast growth f...	0.7147	ESC-specific genes	Signalling molecules required for pluripotency and self-Renewal
17		<input type="checkbox"/>	17	NM_008071	Gabrb3	Gamma-aminobut...	2.8777	ESC-specific genes	Signalling molecules required for pluripotency and self-Renewal
18		<input type="checkbox"/>	18	NM_010253	Gal	Galanin	5.0343	ESC-specific genes	Signalling molecules required for pluripotency and self-Renewal
19		<input type="checkbox"/>	19	NM_010346	Grb7	Growth factor rece...	27.4283	ESC-specific genes	Signalling molecules required for pluripotency and self-Renewal
20		<input type="checkbox"/>	20	NM_010407	Hck	Hemopoietic cell k...	2.9869	ESC-specific genes	Signalling molecules required for pluripotency and self-Renewal
21		<input type="checkbox"/>	21	NM_026820	Ifitm1	Interferon induced ...	27.1546	ESC-specific genes	Signalling molecules required for pluripotency and self-Renewal
22		<input type="checkbox"/>	22	NM_010560	Il6st	Interleukin 6 signa...	0.9224	ESC-specific genes	Signalling molecules required for pluripotency and self-Renewal
23		<input type="checkbox"/>	23	NM_021099	Kit	Kit oncogene	36.3721	ESC-specific genes	Signalling molecules required for pluripotency and self-Renewal
24		<input type="checkbox"/>	24	NM_010094	Lefty1	Left right determin...	212.8439	ESC-specific genes	Signalling molecules required for pluripotency and self-Renewal
25		<input type="checkbox"/>	25	NM_177099	Lefty2	Left-right determin...	287.4606	ESC-specific genes	Signalling molecules required for pluripotency and self-Renewal
26		<input type="checkbox"/>	26	NM_013584	Lifr	Leukemia inhibitor...	0.1562	ESC-specific genes	Signalling molecules required for pluripotency and self-Renewal
27		<input type="checkbox"/>	27	NM_013611	Nodal	Nodal	40.6486	ESC-specific genes	Signalling molecules required for pluripotency and self-Renewal
28		<input type="checkbox"/>	28	NM_008711	Nog	Noggin	0.1023	ESC-specific genes	Signalling molecules required for pluripotency and self-Renewal
29		<input type="checkbox"/>	29	NM_010949	Numb	Numb gene homol...	0.5352	ESC-specific genes	Signalling molecules required for pluripotency and self-Renewal
30		<input type="checkbox"/>	30	NM_008960	Pten	Phosphatase and te...	0.4161	ESC-specific genes	Signalling molecules required for pluripotency and self-Renewal
31		<input type="checkbox"/>	31	NM_009144	Sfrp2	Secreted frizzled-r...	0.1661	ESC-specific genes	Signalling molecules required for pluripotency and self-Renewal
32		<input type="checkbox"/>	32	NM_011562	Tdfr1	Teratocarcinoma-d...	897.2167	ESC-specific genes	Signalling molecules required for pluripotency and self-Renewal
33		<input type="checkbox"/>	33	NM_026396	Bxdc2	Brix domain contai...	1.2078	ESC-specific genes	other ESC-specific genes
34		<input type="checkbox"/>	34	NM_007657	Cd9	CD9 antigen	0.5989	ESC-specific genes	other ESC-specific genes
35		<input type="checkbox"/>	35	NM_017398	Diap2	Diaphanous homol...	1.1919	ESC-specific genes	other ESC-specific genes
36		<input type="checkbox"/>	36	NM_010068	Dnmt3b	DNA methyltransf...	15.6366	ESC-specific genes	other ESC-specific genes
37		<input type="checkbox"/>	37	NM_030694	Ifitm2	Interferon induced ...	0.8363	ESC-specific genes	other ESC-specific genes
38		<input type="checkbox"/>	38	NM_183029	Igf2bp2	Insulin-like growth...	0.8695	ESC-specific genes	other ESC-specific genes
39		<input type="checkbox"/>	39	NM_145833	Lin28	Lin-28 homolog (...)	100.8125	ESC-specific genes	other ESC-specific genes
40		<input type="checkbox"/>	40	NM_013723	Podxl	Podocalyxin-like	35.1346	ESC-specific genes	other ESC-specific genes
41		<input type="checkbox"/>	41	NM_011263	Rest	RE1-silencing tran...	1.5946	ESC-specific genes	other ESC-specific genes



By highlighting the frequent value and switching to overview mode, we can clearly see the repeated region. **This repeated region artifact has not been mentioned in any blog post.**



Case Study: DS-Fly

Editor's Note: <https://www.doi.org/10.1098/rspb.2021.0505>

Blog: <https://pubpeer.com/publications/70DDAFDEA32DD2D9181998DBF1EECB>

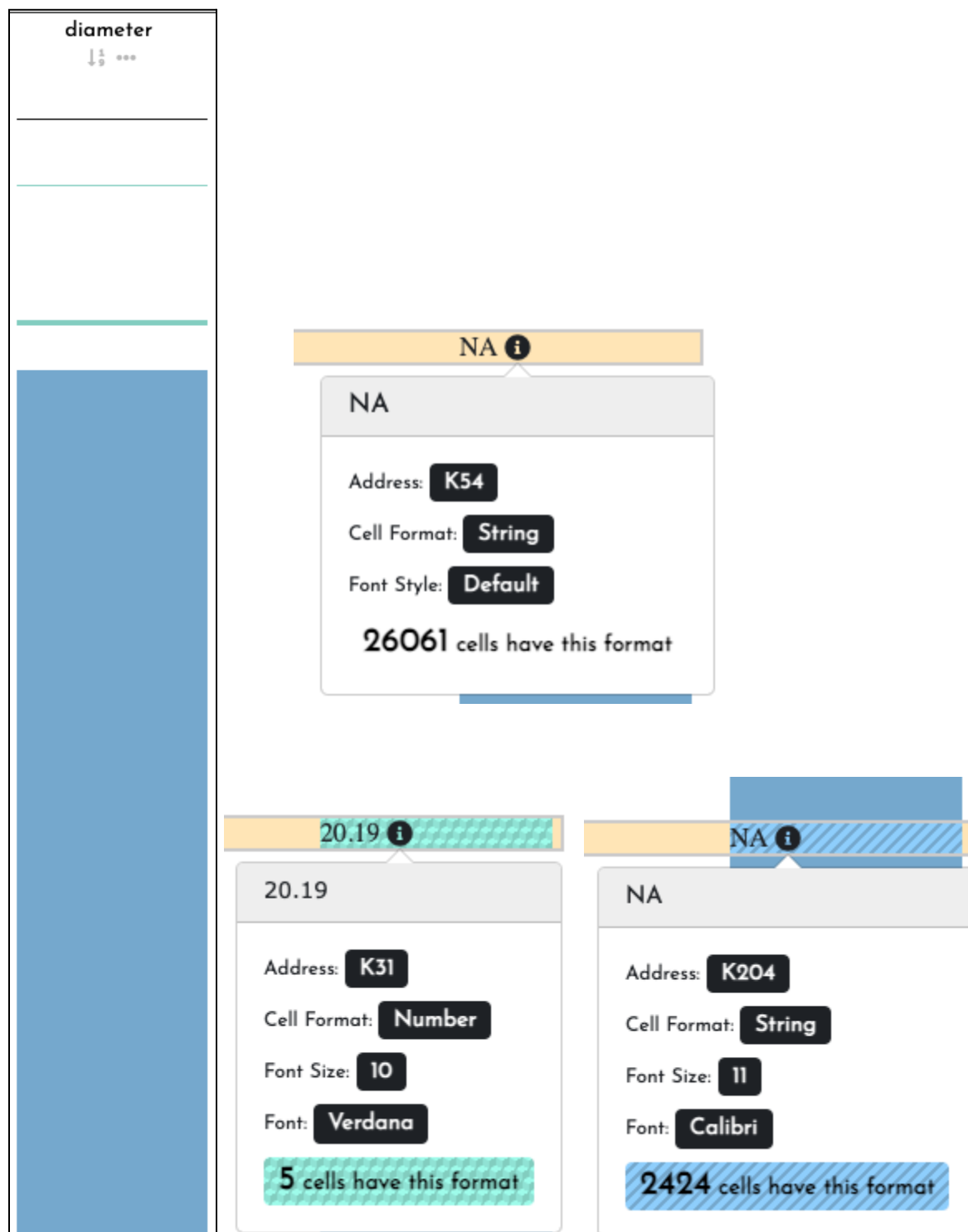
This biology experiment was studying the behavior of flies. This dataset contained two separate sheets. The first contains a column named **diameter**, which is the diameter of the flies. The second sheet includes **dispersal**, which measures the distance a fly has flown.

Formatting artifacts

In the first analysis of Sheet 1, an outlier in formatting can be quickly spotted:

distance	stems	gall	fly	larvae	bird	mordellistena	eury.obs	eury.gig	diameter
1:1 ***	1:1 ***	1:1 ***	1:1 ***	1:1 ***	1:1 ***	1:1 ***	1:1 ***	1:1 ***	1:1 ***
78	7	0	0	NA	NA	NA	NA	NA	NA
78	7	0	0	NA	NA	NA	NA	NA	NA
78	7	0	0	NA	NA	NA	NA	NA	NA
78	7	0	0	NA	NA	NA	NA	NA	NA
78	7	0	0	NA	NA	NA	NA	NA	NA
78	7	0	0	NA	NA	NA	NA	NA	NA
78	7	0	0	NA	NA	NA	NA	NA	NA
87	9	0	0	NA	NA	NA	NA	NA	NA
87	9	0	0	NA	NA	NA	NA	NA	NA
87	9	0	0	NA	NA	NA	NA	NA	NA
87	9	0	0	NA	NA	NA	NA	NA	NA
87	9	0	0	NA	NA	NA	NA	NA	NA
87	9	0	0	NA	NA	NA	NA	NA	NA
87	9	0	0	NA	NA	NA	NA	NA	NA
87	9	0	0	NA	NA	NA	NA	NA	NA
87	9	0	0	NA	NA	NA	NA	NA	NA
87	9	0	0	NA	NA	NA	NA	NA	NA
82	12	0	0	NA	NA	NA	NA	NA	NA
82	12	0	0	NA	NA	NA	NA	NA	NA
82	12	0	0	NA	NA	NA	NA	NA	NA
82	12	0	0	NA	NA	NA	NA	NA	NA
82	12	0	0	NA	NA	NA	NA	NA	NA
82	12	0	0	NA	NA	NA	NA	NA	NA
82	12	0	0	NA	NA	NA	NA	NA	NA
82	12	0	0	NA	NA	NA	NA	NA	NA
82	12	0	0	NA	NA	NA	NA	NA	NA
82	12	0	0	NA	NA	NA	NA	NA	NA
82	12	0	0	NA	NA	NA	NA	NA	NA
82	12	0	0	NA	NA	NA	NA	NA	NA
74	22	4	1	1	0	0	0	0	20.19
74	22	4	1	1	0	0	0	0	22.22
74	22	4	1	1	0	0	0	0	19.99
74	22	4	1	1	0	0	0	0	21.42
74	22	4	0	NA	NA	NA	NA	NA	NA
74	22	4	0	NA	NA	NA	NA	NA	NA
74	22	4	0	NA	NA	NA	NA	NA	NA
74	22	4	0	NA	NA	NA	NA	NA	NA
74	22	4	0	NA	NA	NA	NA	NA	NA
74	22	4	0	NA	NA	NA	NA	NA	NA
74	22	4	0	NA	NA	NA	NA	NA	NA
74	22	4	0	NA	NA	NA	NA	NA	NA

Switching to the overview mode reveals a bigger picture. This column contains three different kinds of formatting (ignoring the cell data format). **This formatting artifact has not been mentioned in any blog post.**



In the second sheet we see that dispersal is formatted differently than the rest of the sheet. On closer investigation, it has the same font as the green cells from sheet 1 (Verdana, 10-point). This type of artifact

gall	time	dispersal
4		2.56 ⓘ

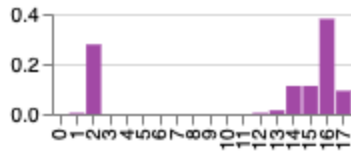
2.56
Address: G50
Cell Format: Number
Font Size: 10
Font: Verdana
205 cells have this format

Stepping through the numerical artifacts in **dispersal** column appears to have some unexpected variation in precision.



dispersal

↓ ↕ ⚡ ...



0.94792489838110930
1.01000000000000000
1.62319303830208830
1.62000000000000000
0.13212690720821374
1.80511228341283210
1.82000000000000000
1.16594676271109000
1.23613609877601260
2.84276271996890500
2.89912519843103930
1.24394910287012600
2.01634300440641030
2.31088659789645600
2.77029445937031360
0.79322363328396730
1.95632838106376950
2.34186656380859300
0.46747358616961127
1.17609170796926630
1.88947546233840050
1.95000000000000000
0.64197374897472770
1.17132357341018430
1.22560528909542970
0.07369836242426997

Ordering artifact combined with precision artifact

In Sheet 1, the **diameter** column has this same variation in precision. There also appear to be some ordering effects with respect to if values have 2 degrees of precision, or many.

The first three regions of non-null diameters all have 2 digits of precision.

ROW	diameter
...	↓ ↕ 🔍 ⚙ ...
28	—
29	—
30	20.19
31	22.22
32	19.99
33	21.42
34	—
35	—
75	—
76	15.68
77	18.95
78	21.02
79	16.78
80	19.11
81	19.85
82	16.52
83	—
84	—

101	—	
102		16.23
103		18.25
104		14.75
105		19.22
106		17.24
107		16.98
108	—	
109	—	

In the next region, every number has high precision:

129	—	
130	19.182055669222805	
131	19.250352060765866	
132	19.225638308856986	
134	—	

After this, there are larger regions of non null values. These vary, but they are predominantly filled with high precision numbers, with a few low precision numbers near the bottom of the region.

213	—	
214	12.233923184938636	
215	17.023932759034206	
216	18.282996426388973	
217	20.688780448623383	
218	20.71885488335425	
219	20.88786368060483	
220	21.626116559670525	
221	23.375222757418683	
225		19.5
226		16.24
227	—	

319	—
320	17.665179804485
321	19.682002079175124
322	21.050236949953526
327	16.24
328	—

512	—
513	13.590875280868142
514	15.021849729490205
515	15.45706703080656
516	15.58543819176719
517	16.591257829707036
518	16.70496590594684
519	22.170371221034777
522	15.45
523	17.8
524	—

580	—
581	13.273803525640387
582	15.357050841163028
583	20.31891757217873
584	22.41311229861569
589	15.42
590	20.4
591	—

Case Study: DS-Priming

Retraction: <https://link.springer.com/article/10.1007/s11002-016-9401-6>

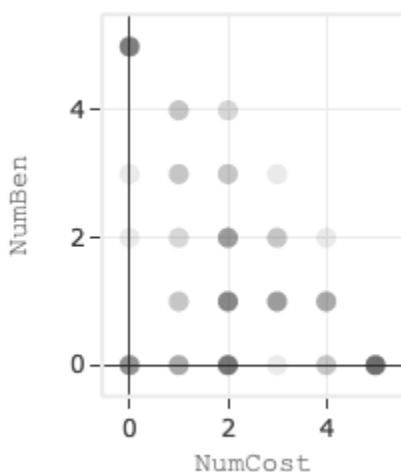
Blogs:

- <https://blog.openmktg.org/2021/07/retracted-article-why-money-meanings.html>
- <https://www.tandfonline.com/doi/full/10.1080/01973533.2015.1124767>

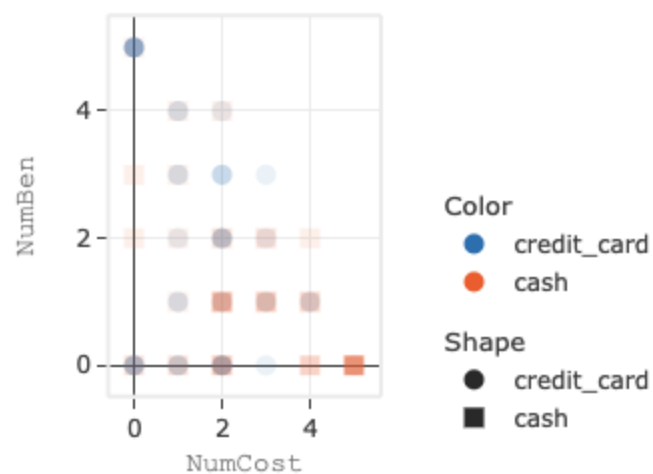
This psychology study measured different priming effects on charitable behavior. Specifically, if there was a difference in being primed with words related to cash vs. credit cards. They provided word completion tasks and counted the number of words the participant responded with that indicated a benefit of volunteering (**NumBen**) and the number of words that correspond to a cost of volunteering (**NumCost**). The original dataset included an experimental condition with four values. For convenience we have added a new column to the beginning of the dataset (**CashOrCredit_Ferret**)

Deviations from domain expectations in response distribution

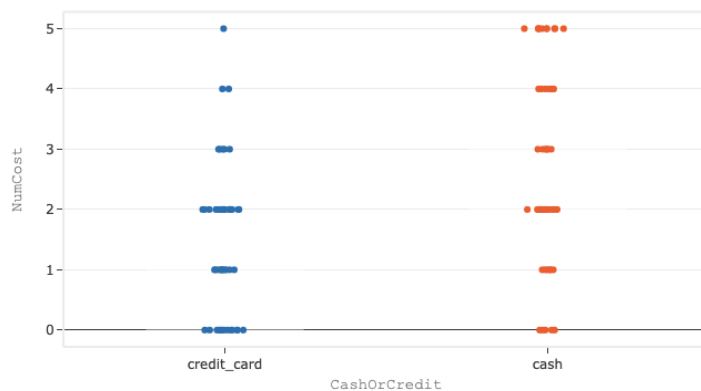
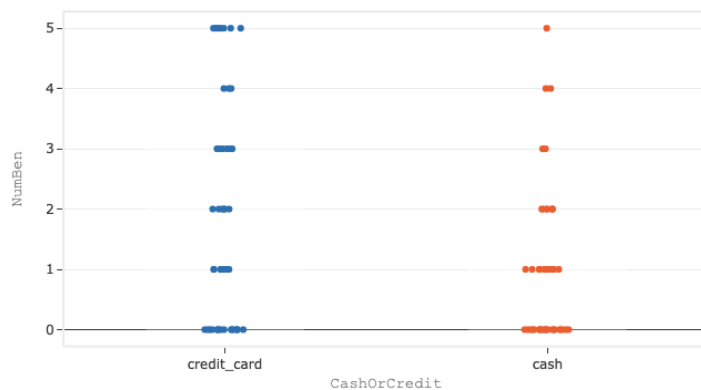
The primary numerical data from this dataset is the number of cost words and benefit words selected by each participant. One way to view this data is through a scatterplot



Due to overplotting, opacity is used to distinguish points with a few participants (light gray) and many participants (dark gray). In the first scatterplot you can many participants near the boundaries (5 benefit words, 0 cost words), and (0 benefit, 5 cost words). Applying a color and shape encoding reveals more information, that most of the (5,0) participants are in the credit condition and the (0,5) are mostly in the cash condition.



Switching to faceted strip plots reveals that in (5,0) and (0,5) there is only one outlier.



Repeated Regions of word responses

Back in the tabular visualization sorting by **NumBen** and **NumCost** is a convenient method to surface the (0,5) group in the table. Examining the word responses reveals that most of the participants in this group produced the same word across multiple different trials.

FOOT	NAP	SPOOK	RECOVERY	0	5
FOOT	NAP	SPOOK	RECOVERY	0	5
FOG	NAG	SPOKEN	RECOMMEND	0	5
FOND	NAP	SPOOK	RECOVERY	0	5
FOOT	NAP	SPOOK	RECOVERY	0	5
FOOT	NAP	SPOOK	RECOVERY	0	5
FOOT	NAP	SPOOK	RECOVERY	0	5
FOOT	NAP	SPOOK	RECOVERY	0	5
FOOL	NAP	SPOOK	RECOVERY	0	5
FORT	NAP	SPOT	RECONSTRUCT	0	5
FOOL	NAP	SPOOK	RECOVERY	0	5