Effects of Measurement Error on Data

Alexander Sun

2024-02-27

```
set.seed(12783678)
obs <- rnorm(n = 1000, mean = 1, sd = 1)
obs[901:1000] <- obs[1:100]
negative_indices <- which(obs < 0)
flip <- sample(negative_indices, length(negative_indices) / 2)
obs[flip] <- -obs[flip]
decadj_indices <- which(obs >= 1 & obs < 1.1)
obs[decadj_indices] <- obs[decadj_indices] / 10
cleaned_mean <- mean(obs)
cleaned_mean
```

```
[1] 1.126746
```

We simulated a normally distributed data set with a standard deviation of 1 and a mean of 1. The simulation unfolded in several stages, each introducing a distinct form of contamination or alteration, reflecting real-world challenges researchers might face in the data collection and preparation phases.

Initially, we generated 1,000 observations from the specified Normal distribution, creating a data set that serves as a baseline for our true data generating process. This step was crucial for establishing a reference point against which the effects of subsequent manipulations could be measured. The first manipulation we conducted on the data set was a "measuring equipment error", where the instrument's limitation to a memory of 900 observations led to the overwriting of the final 100 observations with the first 100. This artificial replication not only reduced the diversity of the data set but also introduced a systematic error, skewing the data set towards the characteristics of the initial observations.

In our simulation, further complications arose through the actions of a research assistant tasked with cleaning the data set. The assistant inadvertently altered half of the negative values in the data set to positive, a mistake that significantly skewed the distribution of the data by increasing its mean and reducing the variance of negative values. This was compounded by a

second error in which the decimal place for values between 1 and 1.1 were incorrectly adjusted, effectively dividing the values by 10 and heavily impacting the values within a specific range of the data.

```
install.packages("tidyverse")
```

```
Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
(as 'lib' is unspecified)
```

```
library(ggplot2)
set.seed(0)
data <- rnorm(1000, mean = 1, sd = 1)
df <- data.frame(value = data)
ggplot(df, aes(x=value)) +
  geom_histogram(aes(y=..density..), binwidth = 0.2, fill="blue", color="black", alpha=0.7) +
  labs(title="Histogram of Normally Distributed Data Before Manipulations",
       x="Value",
       y="Density",
       caption="Figure 1: Distribution of Original Observations Before Manipulations") +
  theme_minimal() +
  theme(plot.caption = element_text(hjust = 0.5))
```

```
Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
i Please use `after_stat(density)` instead.
```

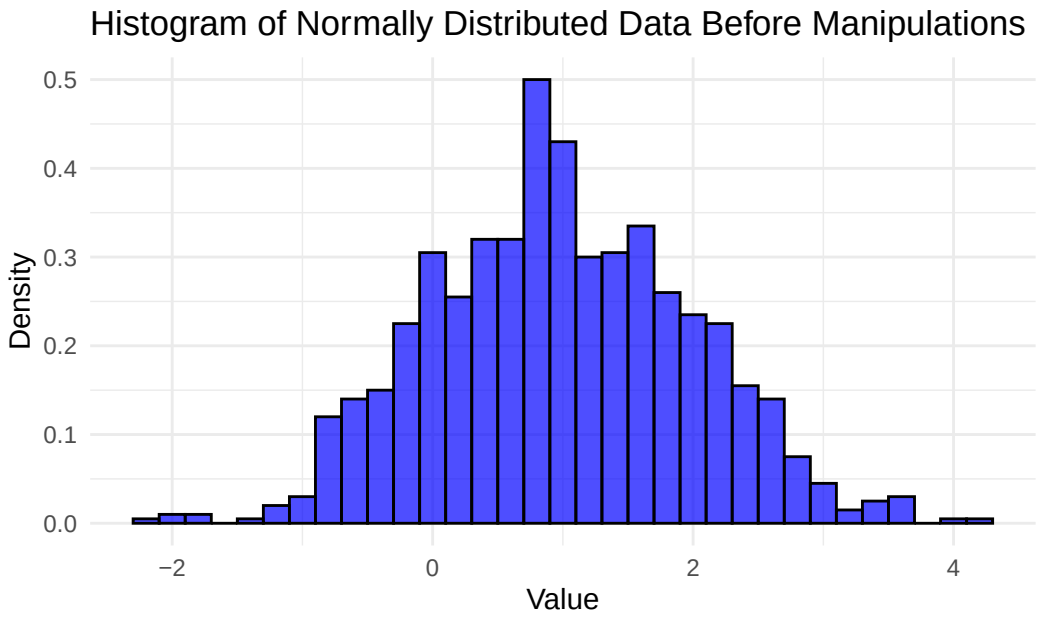

Figure 1: Distribution of Original Observations Before Manipulations

Our original data set before any manipulations occured is depicted above in Figure 1. We can see that the data follows a rough normal distribution, with a mean of approximately 1 and a standard deviation of 1. This will serve as a baseline graphic that helps us visualize any effects the errors may have on our data set. Hopefully, it will also illustrate the degree to which these issues influence our end result.

```
library(ggplot2)
set.seed(12783678)
data <- rnorm(1000, mean = 1, sd = 1)
data[901:1000] <- data[1:100]
negative_indices <- which(data < 0)
flip_indices <- sample(negative_indices, length(negative_indices) / 2)
data[flip_indices] <- -data[flip_indices]
decadj_indices <- which(data >= 1 & data < 1.1)
data[decadj_indices] <- data[decadj_indices] / 10
df_manipulated <- data.frame(value = data)
ggplot(df_manipulated, aes(x=value)) +
  geom_histogram(aes(y=..density..), binwidth = 0.2, fill="blue", color="black", alpha=0.7) +
  labs(title="Histogram of Manipulated Data",
       x="Value",
       y="Density",
       caption="Figure 2: Distribution of Manipulated Observations") +
  theme_minimal() +
  theme(plot.caption = element_text(hjust = 0.5))
```

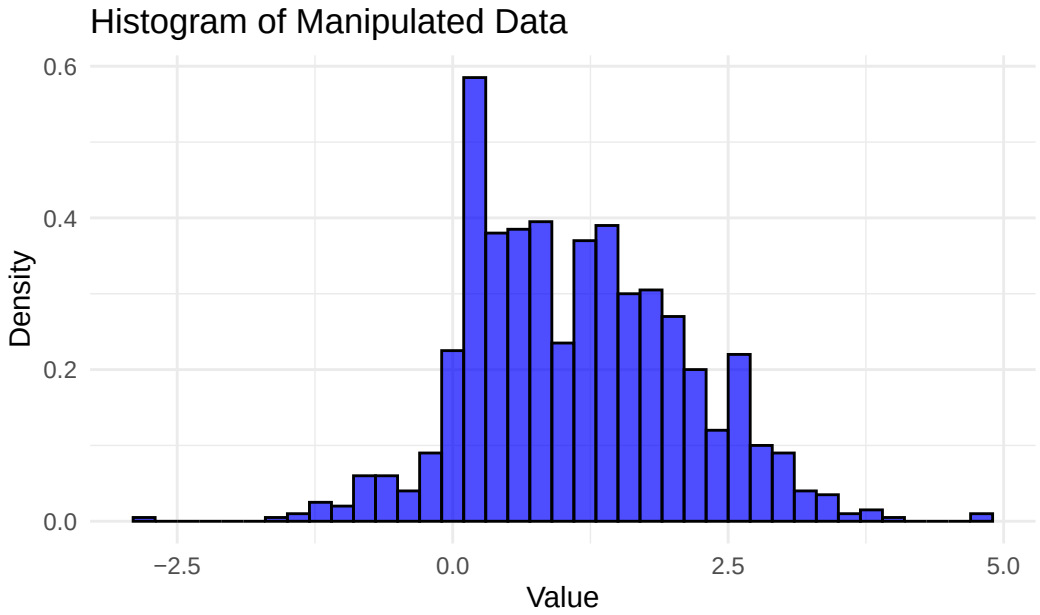

Figure 2: Distribution of Manipulated Observations

Upon analysis of the cleaned data set, we found its mean to be approximately 1.12, marginally more than the true mean but significantly above zero. This outcome suggests that despite the layered errors introduced through data handling and processing, the mean of the data set still reflected a central tendency greater than zero, indicative of the underlying data generating process.

The cumulative effect of these issues—ranging from the instrument's memory limitation to the research assistant's mistakes—demonstrates the ability of errors to influence experiment results. The instrument's memory issue artificially inflated the similarity within the data set, the first mistake by the assistant introduced a bias towards positive values, and the second mistake disproportionately affected observations near the true mean, each altering the data set in non-insignificant ways.

Addressing such challenges requires a thorough approach to ensure data integrity. Systematic checks throughout the duration of an experiment can help identify unusual patterns or duplicate observations. Analyzing the range and distribution of data regularly can uncover unexpected shifts indicative of accidental manipulation. Implementing protocols for data cleaning, utilizing anomaly detection methods, and training awareness among data handlers are critical steps toward mitigating the impact of such errors. Through the potential application of these strategies, researchers can safeguard against data integrity issues, thereby enhancing the robustness of their analyses and the validity of their conclusions. While errors in data handling are inevitable, their adverse effects can be significantly reduced through careful planning, transparent processing, and proper worker training.