Homework 1: Confidence Intervals and Bootstrapping

Name: Alex Ma

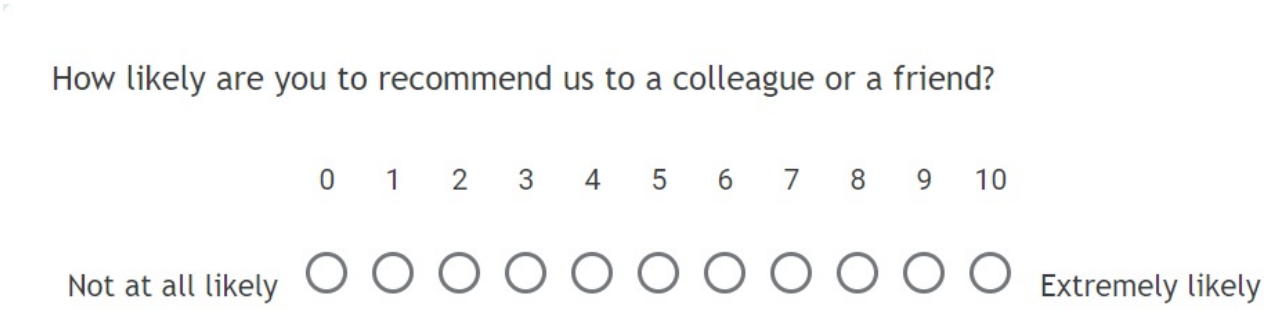
This assignment is due on Gradescope by **Friday January 31 at 5:00PM**. Your solutions to theoretical questions should be done in Markdown directly below the associated question. Your solutions to computational questions should include any specified R code and results as well as written commentary on your conclusions. Remember that you are encouraged to discuss the problems with your classmates, but **you must write all code and solutions on your own**.

NOTES:

- There are 4 total questions on this assignment.
- If you're not familiar with typesetting math directly into Markdown then by all means, do your work on paper first and then typeset it later. Remember that there is a reference guide linked here. **All** of your written commentary, justifications and mathematical work should be in Markdown.
- Because you can technically evaluate notebook cells in a non-linear order, it's a good idea to do Kernel → Restart & Run All as a check before submitting your solutions. That way if we need to run your code you will know that it will work as expected.
- It is **bad form** to make your reader interpret numerical output from your code. If a question asks you to compute some value from the data you should show your code output **AND** write a summary of the results in Markdown directly below your code.
- This probably goes without saying, but... For any question that asks you to calculate something, you **must show all work and justify your answers to receive credit**. Sparse or nonexistent work will receive sparse or nonexistent credit.

Problem 1 (20 Points) Net Promoter Scores

Have you ever seen a survey like this?



One of the most widely used customer satisfaction metrics is NPS - Net Promoter Score. It is extremely popular, especially among the executives -- Wikipedia claims that "versions of the NPS are now used by two-thirds of Fortune 1000 companies". But statisically, it is problematic, and it has to be used with care, as you will hopefully see from this problem!

Here's how the NPS is computed from a survey like the one above:

- We call a specific response a **"promoter"** if the rating given is 9 or 10
- We call a response a **"detractor"** if the rating given is 6 or below
- We call a response neutral if the rating is 7 or 8

$$NPS = \frac{\text{number of promoters} - \text{number of detractors}}{\text{total number of responses}} * 100$$

So for example if the sample of responses to the survey is

```
10, 9, 5, 6, 7, 9, 9, 7, 2, 8
```

then we have 4 promoters (scores `10,9,9,9`) and 3 detractors (scores `5,6,2`), and 3 neutral responses, therefore

$$NPS = \frac{4 - 3}{10} * 100 = 10$$

Notice that NPS can range from -100 (all responses are detractors) to 100 (all responses are promoters). Positive NPS generally signifies positive sentiment, and one of the company's objectives can be to maximize it.

NPS is supposed to measure the general customer sentiment and loyalty towards the product, but only a small subset of people usually fill the survey, so we can use **inference techniques** to estimate what the general sentiment is based on the NPS from the sample.

PART 1.A Let's say we have a sample of 10 responses to the NPS survey `10, 9, 5, 6, 7, 9, 9, 7, 2, 8` and we receive another response.

What is the resulting NPS from the 11 responses if the 11th response is:

a. A promoter

b. A detractor

c. Neutral

Answer

As indicated before, within the original sample of 10 responses, we have 4 promoters and 3 detractors.

a. If the 11th response is a promoter, then we have 5 promoters, 3 detractors and 11 responses. So

$$NPS = \frac{5-3}{11} * 100 = 18.18$$

b. if the 11th response is a detractor, then we have 4 promoters as before, 4 detractors and 11 responses. So

$$NPS = \frac{4-4}{11} * 100 = 0$$

c. if the 11th response is neutral, then we have same as before, 4 promoters, 3 detractors but 11 responses. So

$$NPS = \frac{4-3}{11} * 100 = 9.09$$

PART 1.B

Let's say we have a survey with n responses, and let R_i, $i = 1, \ldots, n$ be a discrete random variable that takes value 1 if the response i is a promoter, value 0 if the response is neutral, and -1 if R_i is a detractor.

In other words, R_i, $i = 1, \ldots, n$ is a random sample of n responses.

If the probability of i'th response being a promoter is p_P, the probability of it being a detractor is p_D, what is the expected value $E[R_i]$ and variance $Var[R_i]$ of R_i?

Answer

Assume P denotes the event that the response i is a promoter, N if the response is neutral and D if R_i is a detractor.

$$E[R_i] = \sum r \cdot p(r) = P \cdot p_P + D \cdot p_D + N \cdot p_N = (1) \cdot p_P + (-1) \cdot p_D + (0) \cdot p_N = p_P - p_D$$

$$E[R_i^2] = \sum r^2 \cdot p(r) = P^2 \cdot p_P + D^2 \cdot p_D + N^2 \cdot p_N = (1)^2 \cdot p_P + (-1)^2 \cdot p_D + (0)^2 \cdot p_N = p_P + p_D$$

$$Var[R_i] = E[R_i^2] - E[R_i]^2 = (p_P + p_D) - (p_P - p_D)^2 = p_P + p_D - p_P^2 + 2p_P \cdot p_D - p_D^2$$

PART 1.C For the remainder of the problem, let's ignore multiplication by 100 in the NPS computation. It's just rescaling, and doesn't really affect conclusion, but it complicates the computations.

Explain why $NPS = \bar{R}$, where $\bar{R} = \frac{R_1 + \ldots R_n}{n}$, the sample mean of $R_1, \ldots, R_n$

Answer

Ignoring multiplication by 100, by definition we have

$$NPS = \frac{\text{number of promoters} - \text{number of detractors}}{\text{number of responses}}$$

Because $\text{number of responses} = n$, we have

$$
\begin{aligned}
NPS &= \frac{\text{number of promoters} - \text{number of detractors}}{n} \\
&= \frac{\text{number of promoters}}{n} - \frac{\text{number of detractors}}{n} \\
&= \frac{(1) \cdot \text{number of promoters}}{n} + \frac{(-1) \cdot \text{number of detractors}}{n} \\
&= \frac{R_1 + R_2 + \cdots + R_n}{n} = \bar{R}
\end{aligned}
$$

We have second to last line because neutral responses take zero values for the random variable R.

PART 1.D

Considering that NPS can be seen as a sample mean, what is the approximate distribution of NPS for large sample sizes n? Make sure to state the mean and variance of that distribution.

Hint 1. There is a very important, one could even say "central", theorem...

Hint 2. Refer to STAT 5000 to remind yourself what is the mean and variance of a sample mean from an iid sample with population mean μ and variance σ^2

Answer

In class, we have a version of the Central Limit Theorem which states that for $X_1, X_2, \cdots, X_n$ $i.i.d$ samples from a distribution with population mean μ and variance σ, as n becomes large

$$\bar{X} \sim N(\mu, \frac{\sigma^2}{\sqrt{n}})$$

Let $X = R$, then by Part 1.B $\mu = p_P - p_D$, $\sigma^2 = (p_P + p_D) - (p_P - p_D)^2$, so

$$\bar{R} \sim N(p_P - p_D, \sqrt{\frac{(p_P + p_D) - (p_P - p_D)^2}{n}})$$

PART 1.E

Write down the 95% z confidence interval for NPS.

Note 1: Problem 1.4 is what tells us that we can use the z confidence interval in the first place! Remember that "z" refers to!

Note 2: Feel free to just use 1.96 as the critical value

Answer

By definition of CI, A $100 - (1 - \alpha)\%$ confidence interval for unknown μ and known σ is given by:

$$\left[\bar{X} - Z_{\frac{\alpha}{2}}\frac{\sigma}{\sqrt{n}}, \bar{X} + Z_{\frac{\alpha}{2}}\frac{\sigma}{\sqrt{n}}\right]$$

Here, a 95% CI means $\alpha = 0.05$. Assuming $\text{critical value} = Z_{0.025} = 1.96$ the interval is given by

$$\left[\bar{R} - 1.96\sqrt{\frac{(p_P + p_D) - (p_P - p_D)^2}{n}}, \bar{R} + 1.96\sqrt{\frac{(p_P + p_D) - (p_P - p_D)^2}{n}}\right]$$

PART 1.F

Let's say that on a survey with 30 responses, the propotion of promoters was 0.5, and proportion of detractors was 0.3.

Then the NPS (without multiplying by 100) is $0.5 - 0.3 = 0.2$. What is the 95% confidence interval for this score?

Answer

Because NPS (without multiplying by 100) is 0.2 and by Part 1.C $NPC = \bar{R}$, $\bar{R} = 0.2$. Plug $\bar{R} = 0.2$, $p_P = 0.5$, $p_D = 0.3$ and $n = 30$ into the CI expression in Part 1.E, we have

$$CI = \left[0.2 - 1.96 \cdot \sqrt{\frac{(0.5+0.3)-(0.5-0.3)^2}{30}}, 0.2 + 1.96 \cdot \sqrt{\frac{(0.5+0.3)-(0.5-0.3)^2}{30}}\right]$$
$$= [-0.11196, 0.51196]$$

The result is calculated in the cell below.

In [1]:
```
lower <- 0.2 - 1.96*((0.5+0.3-0.2**2)/30)**(1/2)
upper <- 0.2 + 1.96*((0.5+0.3-0.2**2)/30)**(1/2)
cat(lower, upper)
```

```
-0.1119624 0.5119624
```

NPS is widely used in customer analytics, but it doesn't have built-in procedures in R like `t.test` for mean or `prop.test` for proportion, and you might not want to go through the computation above every time you want to report NPS together with the margin of error or its CI. So creating a bootstrap confidence interval can be a good alternative.

Problem 2 (25 Points): Bootstrapping Net Promoter Scores

(For the definition of NPS, see Problem 1)

The code below reads an example dataset of NPS survey responses into the dataframe `data`. The responses are in the `response` column.

In [2]:
```
data <- read.csv('nps.csv')
head(data)
```

response
10
5
8
9
8
10

PART 2.A

Create a function called `nps()` that takes a vector of survey responses as an argument and returns the NPS based on those responses. Demonstrate the function by calling it on the `response` column of `data`.

In [3]:
```
nps <- function(res_vec) {
    res_p <- res_vec[res_vec >= 9]
    res_d <- res_vec[res_vec <= 6]
    nps_score <- (length(res_p) - length(res_d))/length(res_vec)
    return(nps_score)
}
nps_original <- nps(data$response)
nps_original
```

```
0.0831826401446655
```

PART 2.B Create one bootstrap sample from the `response` column of `data`, and compute the NPS for that boostrap sample.

In [4]:
```
bootstrap <- function(samples) {
    return(sample(samples, length(samples), replace = TRUE, prob = NULL))
}
nps(bootstrap(data$response))
```

```
0.0638939119951778
```

If you created the bootstrap sample correctly, the NPS should be different from the NPS you computed from `data$responses`.

PART 2.C

1. Create 30 bootstrap samples of the `response` column of `data`. Save them in a variable.
2. Compute NPS for each sample

3. Plot the distribution of bootstrapped NPS by plotting a normalized histogram of these scores

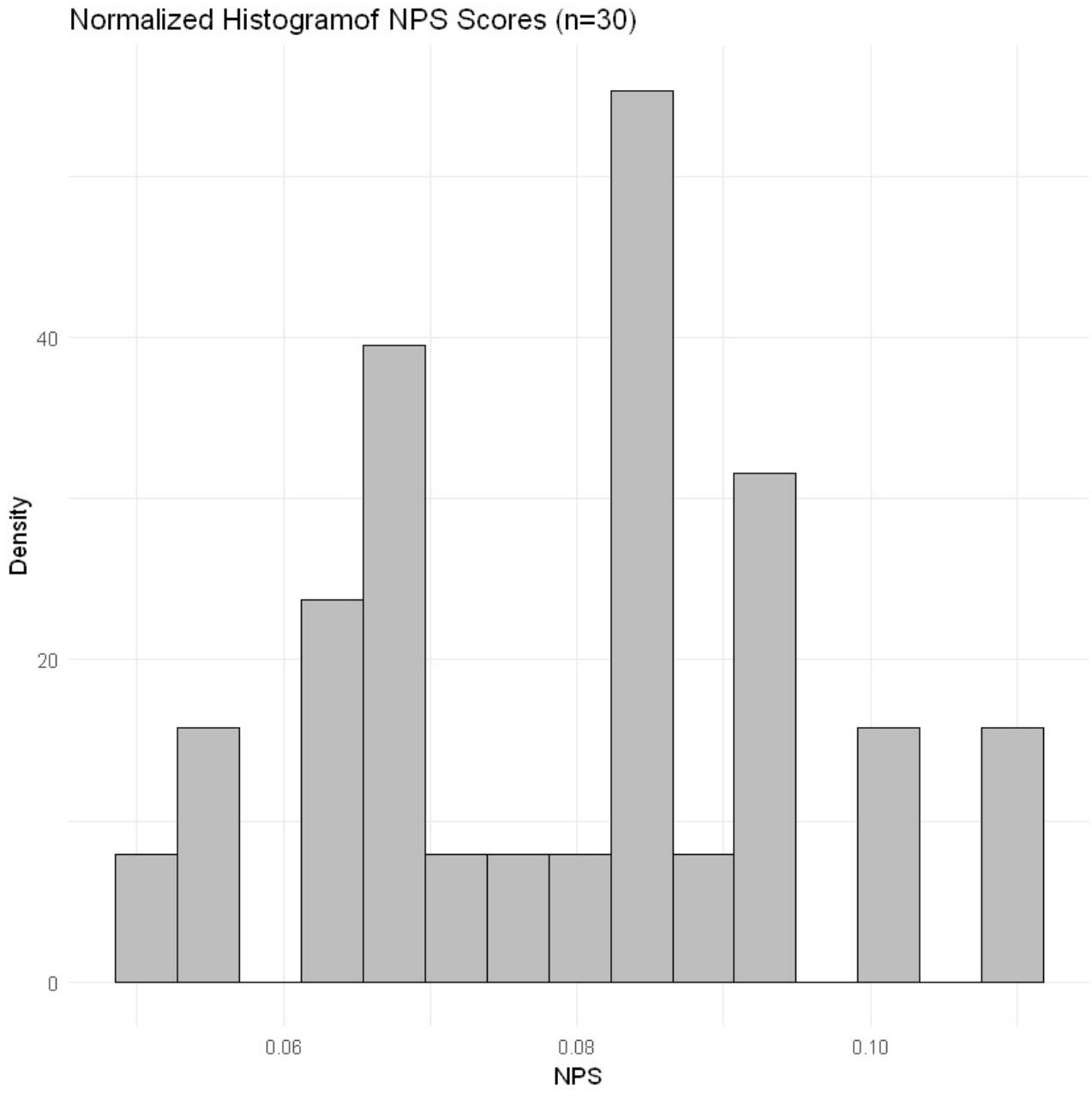
Hint: The function `replicate()` might be helpful

Note: If you're using `ggplot`, you can normalize a histogram by specifying `y = after_stat(density)` inside `aes` in `geom_histogram()`

Notice that 30 is the number of bootstrap samples, not the size of them! The size of any bootstrap sample sample is just the size of the dataset, because the dataset is the sample.

In [5]:
```
n <- 30
bootstrap_samples <- replicate(n, bootstrap(data$response), simplify=FALSE)
nps_list <- lapply(bootstrap_samples, nps)
nps_vec <- unlist(nps_list)

df <- data.frame(values = nps_vec)

library(ggplot2)
ggplot(df, aes(x = values)) +
  geom_histogram(aes(y = ..density..), bins = 15, fill = "grey", color = "black") +
  labs(title = "Normalized Histogramof NPS Scores (n=30)", x = "NPS", y = "Density") +
  theme_minimal()
```

```
Registered S3 methods overwritten by 'ggplot2':
  method         from
  [.quosures     rlang
  c.quosures     rlang
  print.quosures rlang
```



It's very difficult to understand the distribution of scores from this histogram. For example, what is the variance of this distribution? How

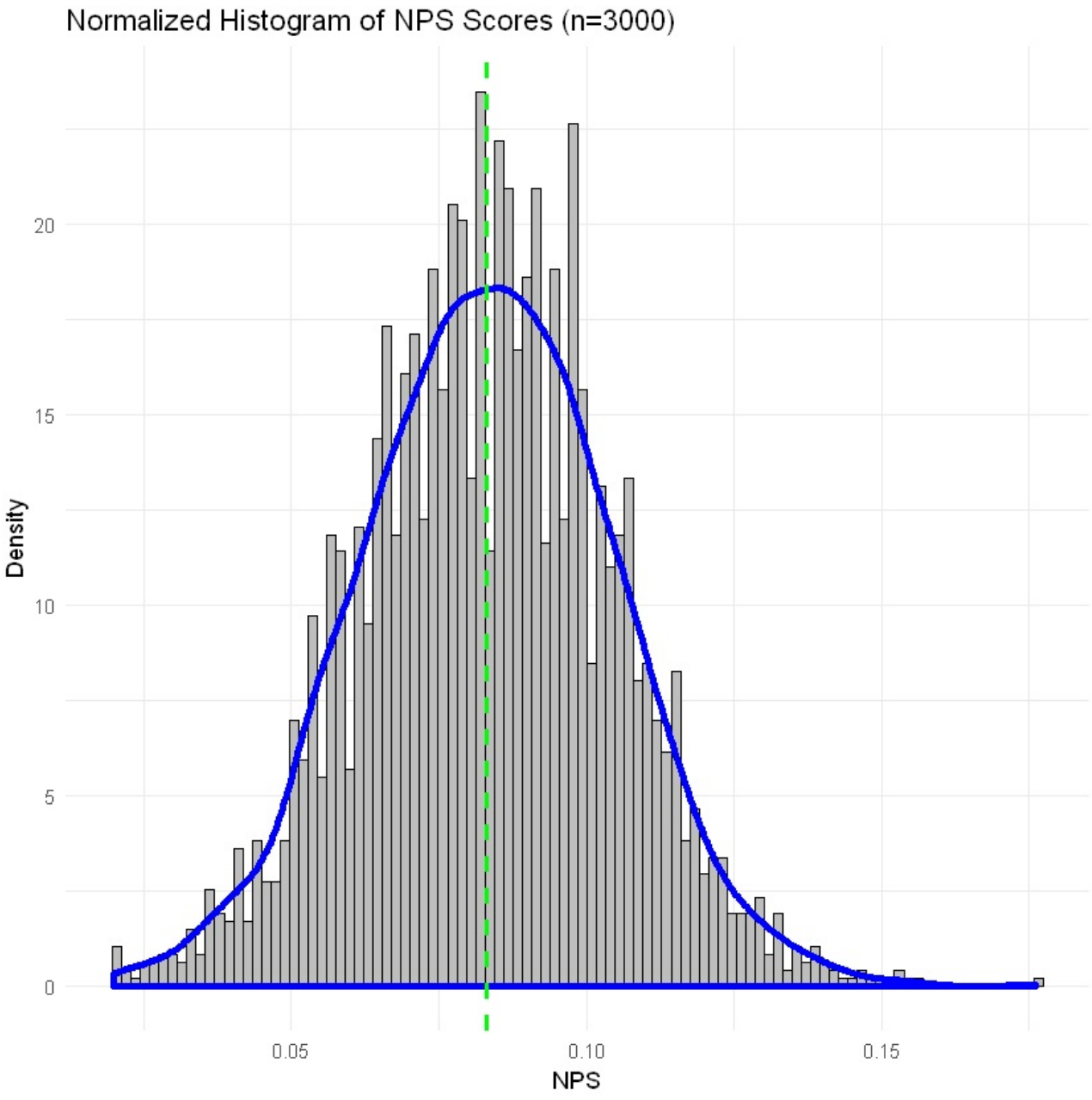
likely are the numbers around 8.31 (the NPS for the original data) to appear?.. To answer questions like this, we need to take more bootstrap samples to get a "smoother" distribution.

PART 2.D

1. Repeat the process in the previous question, but generate 3000 samples this time. Save the 3000 NPS results in a variable -- you'll use them in the next question.
2. Plot the normalized histogram of the 3000 samples, and a density plot in a different color.
3. Also, add a vertical line at the NPS value for the original dataset (in a third color)

The distribuion should look much smoother now. What can you say about the shape of the bootstrap distribution, and the role of the NPS of the original dataset in it?

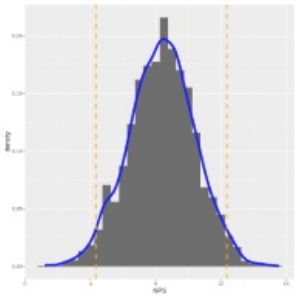
In [6]:
```
n <- 3000
bootstrap_samples <- replicate(n, bootstrap(data$response), simplify=FALSE)
nps_list <- lapply(bootstrap_samples, nps)
nps_vec <- unlist(nps_list)

df <- data.frame(values = nps_vec)

library(ggplot2)
ggplot(df, aes(x = values)) +
  geom_histogram(aes(y = ..density..), bins = 100, fill = "grey", color = "black") +
  geom_density(color = "blue", size = 1.5) +
  geom_vline(xintercept = nps_original, color = "green", linetype = "dashed", size = 1) +
  labs(title = "Normalized Histogram of NPS Scores (n=3000)", x = "NPS", y = "Density") +
  theme_minimal()
```



When reporting business metrics (like NPS scores) that are based on samples, it's a good idea to also specify the margin of error or the the confidence interval, so that the audience of the report understands how to interpret the metric. For example, it can look something like 20.3 ± 3.56, or $(52.4, 61.3)$

Given the bootstrap distribution that you just obtained, we'll compute the 95% confidence interval for the NPS of the original dataset.

There are multiple methods of obtaining CIs from bootstrap samples. We'll use a method called **percentile bootstrap**: to determine the 95% CI, we'll find such values that 95% of the bootstrap distribution lies within those values:



For this symmeric distribution, this means we need to find one value such that 2.5% (half of 5%) of the bootstrap distribution is to the left of it, and another value such that 2.5% of the distribution is to the right.

PART 2.E

Use the built in `quantile()` function to find the lower and upper limits of the 95% confidence interval for the bootstrap distribution that you created in 2.4.

In [7]:
```
lower <- quantile(nps_vec, 0.025)
upper <- quantile(nps_vec, 0.975)
cat("Lower limit:", lower, " Upper limit:", upper)
```

```
Lower limit: 0.04157625  Upper limit: 0.124774
```

PART 2.F

Compute the differences between the original sample NPS and the lower/upper limits for the confidence interval.

Are they equal? Should we expect them to be? Why or why not?

Hint: is the distribution *perfectly, completely symmetric*?

In [8]:
```
lower_diff <- nps_original - lower
upper_diff <- upper - nps_original
cat("Lower Diff:", lower_diff, " Upper Diff:", upper_diff, " Equal?:", lower_diff == upper_diff)
```

```
Lower Diff: 0.04160639  Upper Diff: 0.04159132  Equal?: FALSE
```

They are not equal and we shouldn't expect them to be. Looking at the blue density line on the $n = 3000$ plot, the distribution is skewed slightly to the left, which justified why the difference between the original NPS score and the lower limit is smaller than that of the upper limit.

Problem 3 (25 Points): How old are cats in animal shelters?

Austin city government regularly publishes data about animals in the city-run animal shelter. The code below loads information about a *sample* of cats from that shelter into the variable `cats`:

In [9]:
```
cats <- read.csv('cats.csv')
head(cats)
```

X	animal_id	name	outcome_type	animal_type	breed	color	age_days
1	A846341		Adoption	Cat	Domestic Shorthair	Blue Tabby	55
2	A772759	*Bella	Adoption	Cat	Domestic Medium Hair Mix	Blue	63
3	A666893	Pristy	Transfer	Cat	Domestic Shorthair Mix	Black	2237
4	A753626	*Mooney	Adoption	Cat	Domestic Shorthair Mix	Black	92
5	A747591		Transfer	Cat	Domestic Longhair Mix	Brown Tabby	369
6	A703272	*Catelyn	Adoption	Cat	Domestic Shorthair Mix	Cream Tabby/White	88

This data contains information about each cat, like their name, age and breed, as well as the outcome for them (e.g. "Adoption").

In this problem, we'll be working with `age_days` column, which contains the estimated age of the cat (at the moment of the outcome, so e.g. at the moment of adoption).

This granularity - days instead of for example years - is important, as we'll see shortly.

PART 3.A

Do you think most of the cats in the shelter are kittens? Adults? Seniors? Everyone in equal measure?

You don't need to analyze data, and there's no incorrect answer to this question. Just write what your intuition tells you!

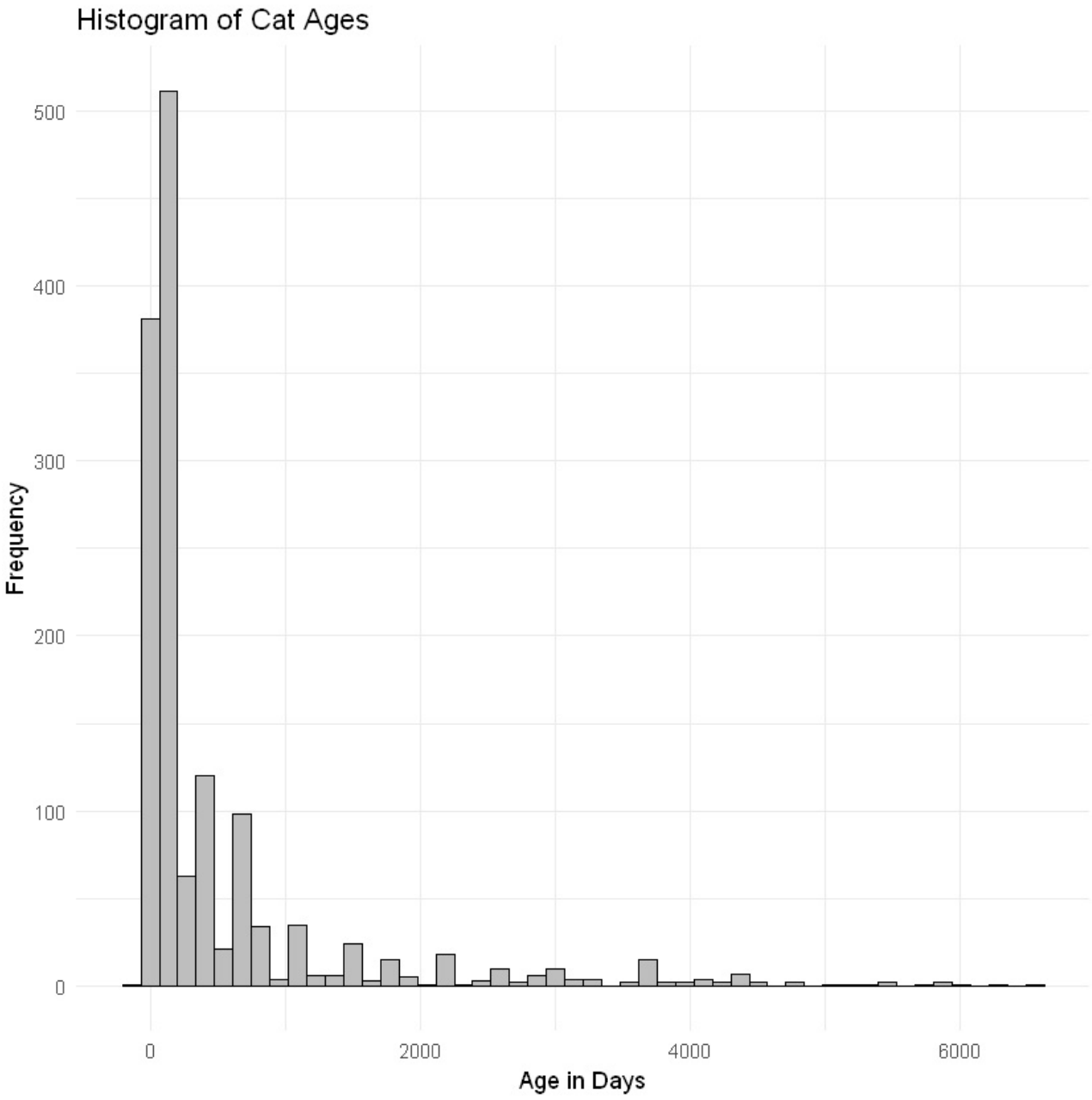
Intuition

Most of the cats should be kittens because adult cats are usually already owned by people. Then adults and then seniors, because senior stray cats usually don't end up well and are few.

PART 3.B

Plot the histogram of the ages of the cats in this sample (the `age_days` column).

In [10]:
```
ggplot(cats, aes(x = age_days)) +
  geom_histogram(bins = 50, fill = "grey", color = "black") +
  labs(title = "Histogram of Cat Ages", x = "Age in Days", y = "Frequency") +
  theme_minimal()
```



PART 3.C

Compute the mean and the median of cat ages in the `cats` sample.

In [11]:
```
age_mean <- mean(cats$age_days, na.rm = TRUE)
```

```
age_median <- median(cats$age_days, na.rm = TRUE)
cat("Cat Ages: mean", age_mean, "days,", "median", age_median, "days")
```

```
Cat Ages: mean 518.7756 days, median 110 days
```

PART 3.D

Considering the shape of the distribution that you plotted in 3.2, which of the two measures - mean or median - is a more meaningful measure of "centrality" for this data? Why?

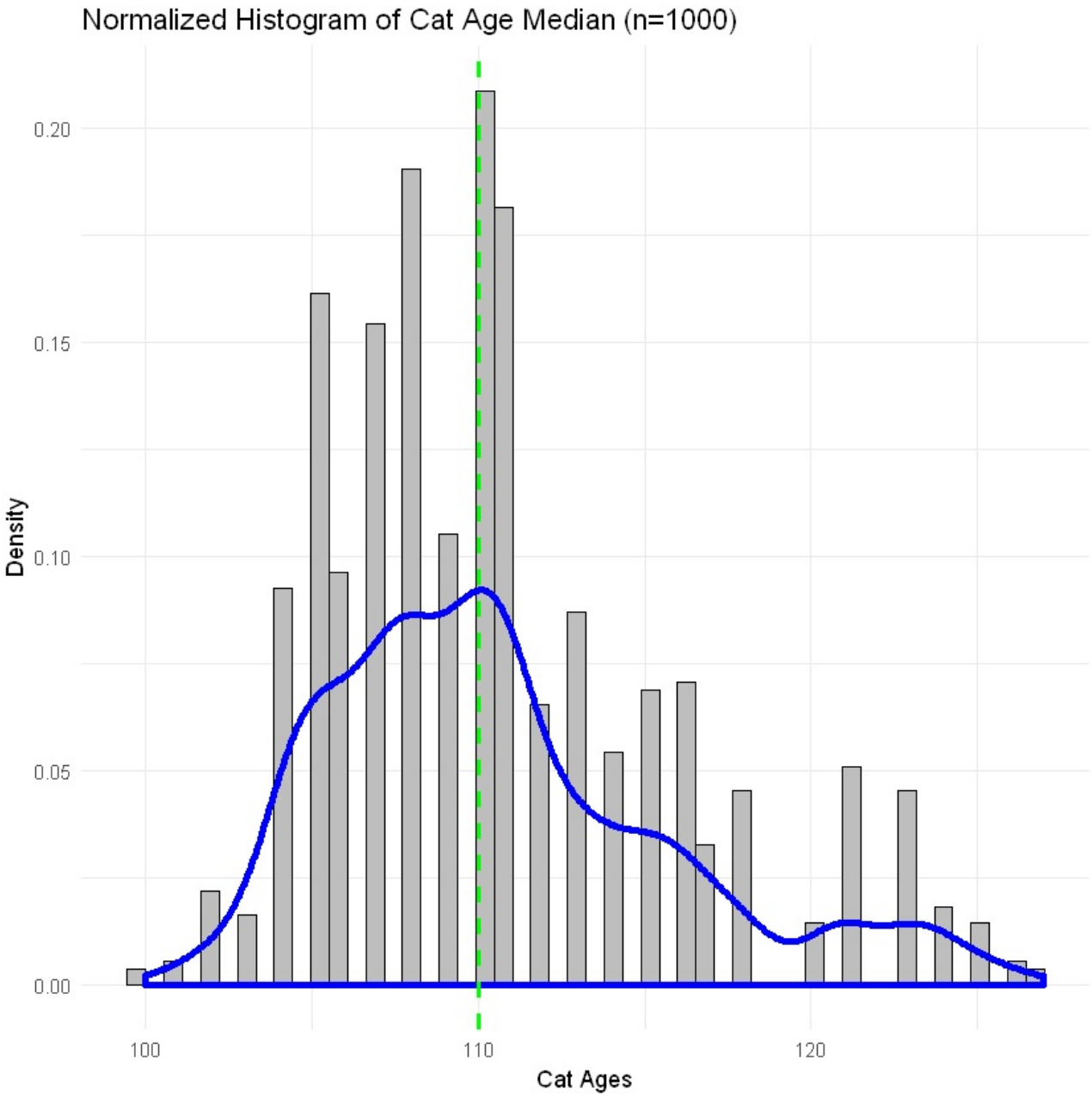
Answer

Median is a more meaningful measure of "centrality" of this data. The data follows a power law curve that have extremely high frequency on the left and a long tail to the right. Because of this imbalance in distribution the mean will skew significantly to the left and is not a meaningful measure of the degree of centrality. By definition, median only considers the rank of data magnitude instead of the magnitude itself, therefore will not be significantly skewed by the power law distribution.

PART 3.E

1. Generate 1000 bootstrap samples of cat ages, and compute the median for each sample
2. Plot the histogram and density plot (in different colors) of the distribution of the 1000 bootstrapped medians
3. Add a vertical line (in a different color) at the value of the sample median of `cats$age_days`

In [12]:
```
n <- 1000
bootstrap_samples <- replicate(n, bootstrap(cats$age_days), simplify=FALSE)
medians_list <- lapply(bootstrap_samples, median)
medians_vec <- unlist(medians_list)

df <- data.frame(values = medians_vec)

library(ggplot2)
ggplot(df, aes(x = values)) +
  geom_histogram(aes(y = ..density..), bins = 50, fill = "grey", color = "black") +
  geom_density(color = "blue", size = 1.5) +
  geom_vline(xintercept = age_median, color = "green", linetype = "dashed", size = 1) +
  labs(title = "Normalized Histogram of Cat Age Median (n=1000)", x = "Cat Ages", y = "Density") +
  theme_minimal()
```



PART 3.F

Do you think this distribution is well approximated by a normal distribution? What deviations from normality can you detect, if any?

Make sure your reasoning is consistent with your plot.

Answer

It's somehow approximated but not very well. Based on the density line in blue, the distribution clearly skew to the left and has a long tail to the right, deviating from a normal distribution.

---

*We'll use **percentile bootstrap** to compute the 95% confidence interval: we'll find the 2.5th and 97.5th percentiles of the bootstrap distribution, which will determine the lower and upper limits of the CI. This way, 95% of the distribution will be within the confidence interval.*

PART 3.G

Take a look at the density plot of bootstrapped medians that you created before. Do you think the percentile bootstrap CI will be symmetric around the median of the original sample? Which of the lower/upper limits do you think will be closer to the sample median, and why?

Note: the answer completely depends on your specific bootstrap samples! It's can be different for different people, and even for different executions of the same code.

Don't forget to justify your answer! You can use the density plot you generated before to eyeball where the percentiles will be, based on

the "mass" of the distribution.

Answer

The percentile bootstrap CI will not be symmetric and the lower limit will be closer to the sample mean. Because the bootstrapped medians have a distribution that skew to the left, the distribution mean is closer to the lower end which is the lower limit of the bootstrap CI.

PART 3.H

Compute the lower and upper limits of the 95% bootstrap confidence interval using the percentile method. Additionally, compute the difference of both lower and upper limits of the CI with the sample median

In [13]:
```
lower <- quantile(medians_vec, 0.025)
upper <- quantile(medians_vec, 0.975)
lower_diff <- age_median - lower
cat("Lower limit:", lower, "days,", " Upper limit:", upper, "days.", '\n')
upper_diff <- upper - age_median
cat("Lower Diff:", lower_diff, "days,", " Upper Diff:", upper_diff, "days,", " Equal?:", lower_diff == upper_di
```

```
Lower limit: 103 days,  Upper limit: 123 days.
Lower Diff: 7 days,  Upper Diff: 13 days,  Equal?: FALSE
```

PART 3.I

Please note that this problem part is OPTIONAL for STAT 4010 students and is REQUIRED for STAT 5010 students.

If you want a better idea about the distribution of bootstrap statistics, you need more bootstrap samples. Now imagine if your dataset is huge -- hundreds of millions of records, and to generate even one bootstrap sample, you'll need to sample those hundreds of millions of records with replacement.

If you want to generate a million bootstrap samples, you need to sample hundreds of millions of records a million times. You can see how this can get computationally prohibitive very quick!

There are many methods to make the bootstrap more computationally efficient. One of them was developed at Spotify. The linked article proposes a fast bootstrap algorithm for computing quantiles (like a median). It also gives Python code examples of applying that algorithm to simulated data.

Your task in this question is to reimplement the algorithm in R, and apply it to the dataset you've been working with to compute the 95% bootstrap confidence for the median of cat ages, using 500 000 bootstrap samples.

Here are some pointers:

- You'll need the single-sample bootstrap example, which is the first code sample in the article.You'll need to figure out what each line of code does, and apply it to your case. (But please read the rest of the article too, to understand what's happening! The most exciting contribution of the article is actually the second example).
- Note that the code sample uses simulated data from a normal distribution! You'll need to change it to use the data from `cats$age_days` instead. In particular, the `sample_size` will be determined by the data.
- The code uses 1 000 000 bootstrap samples. You'll need to change that. (What other parts of the code will you need to change, if any?)
- You'll need to expore what the Python function `binom.ppf()` does. The function `qbinom()` in R accomplishes something similar, but make sure you read the documentation for both to see how they compare.
- Depending on your prior knowledge, you might also need to do some additional research to see how to do some array operations in R. This is part of the assignment!
- Python uses 0-based indexing, but R uses 1-based indexing of arrays! FYI Julia also uses 1-based indexing (there's a link to Julia code later in the article).

Note 1: Most of you should be familiar with Python by now, and should be able to read the code in the linked article. If you are not familiar with Python, you can try one of the followifing options: (1) read through the algorithm description and just implement the code yourself, (2) click on the link in the article that leads to a repository containing Julia code and try to read the Julia code instead (Julia is often much more user-friendly for novices), or (3) ask a classmate or a Course Assistant for help.

Note 2: Implementing new algorithms from papers that use a different programming language, and applying these new algorithms to your data, is something that you are very likely going to be doing in your professional day-to-day life very often. This specific problem is good practice for such workflow because the algorithm is very simple, and the code snipped provided by authors is short but complete. Many methods that you'll work with will have such simple setup, unfortunately.

Answer

The quantile of interest = 0.5 because this in the original blog represents the position in the ordered list of our target statistic. Because we're looking at the median, this means we want the element in exactly middle of the list, so $q = 0.5$.

In the original code, the outcomes are simulated using sampling from a normal distribution. The real meaning of the outcome list should be the ordered sample list, so here we sort the age column and strore it as the outcome list and name the variable something closer to the true meaning like `obs_sorted`.

The rest of the code is only different from the original code in syntax. `qbinom()` allows us to get the percentile indices, and get the observations in the sample list that sit in the given indices. Here, we used `c(a, b)` to create a vector instead of simply using brackets like Python.

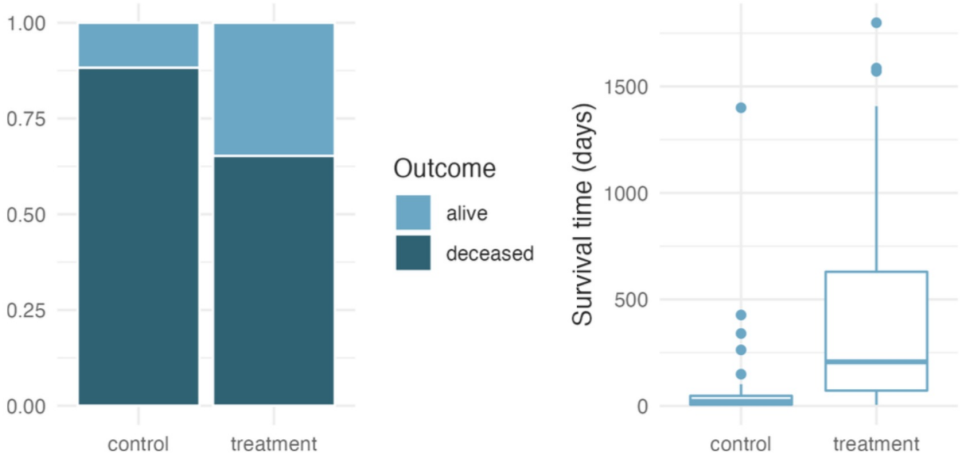
In [14]:
```
alpha <- .05
quantile_of_interest <- 0.5 # doesn't need to change because we want medians
number_of_bootstrap_samples <- 500000
obs_sorted <- sort(cats$age_days) # by the blog we get the observation on the (N+1)q index of the ordered sampl

ci_indexes <- qbinom(c(alpha/2,1-alpha/2), length(cats$age_days)+1, quantile_of_interest)
bootstrap_confidence_interval <- obs_sorted[c(as.integer(floor(ci_indexes[1])), as.integer(ceiling(ci_indexes[2
bootstrap_confidence_interval
```

1. 102
2. 124

Problem 4 (30 Points): Hypothesis Testing with Randomization

For this problem, we will be performing a hypothesis test with randomization.

The Stanford University Heart Transplant Study was conducted to determine whether an experimental heart transplant program increased lifespan. Each patient entering the program was designated an official heart transplant candidate, meaning that they were gravely ill and would most likely benefit from a new heart. Some patients got a transplant and some did not. The variable `transplant` indicates which group the patients were in; patients in the treatment group got a transplant and those in the control group did not. Of the 34 patients in the control group, 30 died. Of the 69 people in the treatment group, 45 died. Another variable called `survived` was used to indicate whether or not the patient was alive at the end of the study.



Part 4.A: Does the stacked bar plot indicate that survival is independent of whether or not the patient got a transplant? Explain your reasoning?

Answer

It seems to indicate that survival is dependent on if the patient got a transplant, though further data-driven testing is needed. There could be enough signal because it seems like the ratio of surviving is higher for the group that got treatments.

Part 4.B: What do the box plots above suggest about the efficacy (effectiveness) of the heart transplant treatment?

Answer

The box plots suggest that the heart transplant is somewhat effective. Although the treatment group has a clear long tail towards longer survival time, the mean suggest that most of the patient still skew towards a lower survival time. Otherwise, the mean of survival time of the treatment group is clearly longer than the control group.

Part 4.C: What proportion of patients in the treatment group and what proportion of patients in the control group died?

Answer

$\frac{45}{69} = 65.22\%$ of patients in the treatment group died and $\frac{30}{34} = 88.24\%$ of patients in the control group died.

Part 4.D: Now we will perform a hypothesis test using randomization. State the null and alternative hypotheses for the test.

Answer

H_0: Null hypothesis. The variables getting treatment and survival are independent. They have no relationship, and the observed difference between the survival rate between treatment and control group, 23%, was due to the natural variability inherent in the population.

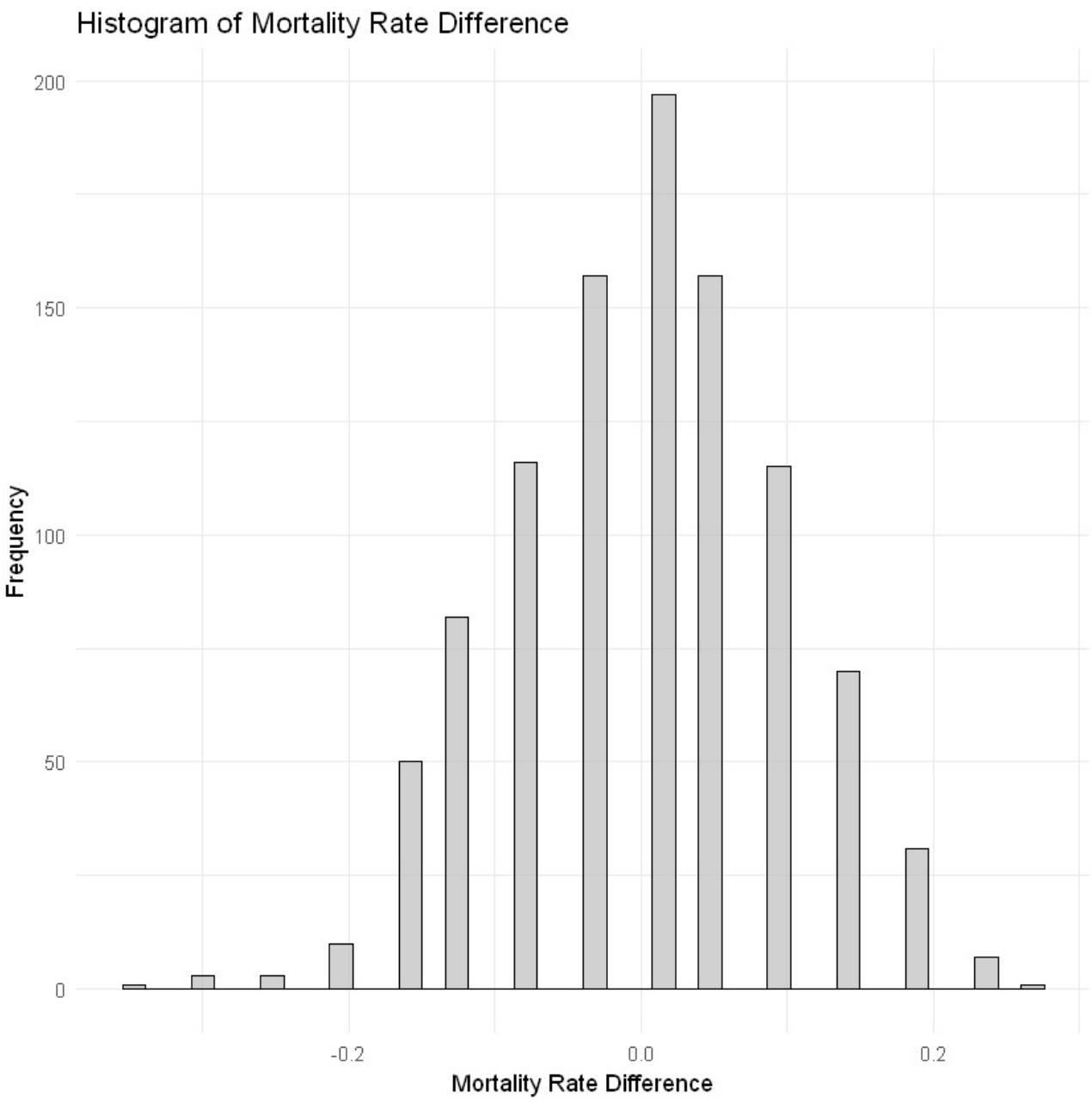
H_A: Alternative hypothesis. The variables getting treatment and survival are not independent. The observed difference between the survival rate between treatment and control group, 23%, wasn't due to the natural variability and treatment helps patients live longer.

Part 4.E: Run 1000 simulations for a randomization test and compile the results in a histogram. Display your histogram below. Please see pages 215-216 in "Introduction to Modern Statistics" as a reference for this. Note that this problem is taken directly from the textbook. We are trying to see if we can replicate the results. Due to random sampling, it is expected that your histograms should be slightly different from the textbook and from each other.

Note, you will need to create synthetic data. There were $34 + 69 = 103$ Total people in this study. You will need to create a process for labeling the data points as "survivor" or "non-survivor" and making random draws from this labeled data. Hint: You could make a list or a csv file with this data and then sample it randomly. Or you could compute the rate of survivors vs non-survivors within these 103 data values and simulate the data using the survival rate as a probability distribution. Or you could come up with your own process. The first way I described it a bit closer to the classic set up we have been discussing in class. But I think the second way is a reasonable approximation. Either is fine for the sake of this homework assignment.

In [15]:
```
# Within the 103 total patients, 30 + 45 = 75 patients died.
# Create a list of 103 patients where 75 of them died and 103 - 75 = 28 patients survived.
patients <- c(rep("non-survivor", 75), rep("survivor", 28))

# Sample n_treatment number of patients from all patients, 1000 times.
# Calculate survival rate diff.
mortality_rate_diff <- function() {

    # calculate number of treatment and control patients
    n_pop <- 103
    n_treatment <- 69
    n_control <- n_pop - n_treatment

    n_non_survivor <- 75

    # sample treatment group
    samples <- sample(patients, n_treatment, replace = FALSE)

    # get number of non-survivors in treatment group
    n_non_survivor_treatment <- sum(samples == "non-survivor")

    # get number of non-survivors in control group
    n_non_survivor_control <- n_non_survivor - n_non_survivor_treatment

    # calculate treatment and control mortality rate
    treatment_rate <- n_non_survivor_treatment / n_treatment
    control_rate <- n_non_survivor_control / n_control

    # return mortality rate difference
    return(control_rate - treatment_rate)
}

mortality_rate_diffs <- replicate(1000, mortality_rate_diff())
```

In [16]:
```
data <- data.frame(mortality_rate_diffs)

ggplot(data, aes(x = mortality_rate_diffs)) +
  geom_histogram(bins=40, fill = "grey", color = "black",
                 alpha = 0.7) +
  labs(title = "Histogram of Mortality Rate Difference",
       x = "Mortality Rate Difference",
       y = "Frequency") +
  theme_minimal()
```



Part F: What is your conclusion and why? (In other words, will you reject your null hypothesis or fail to reject your null hypothesis?)

Answer

Because only in 0.002% of the simulations the mortality rate difference is larger than 23.02%, the original sample statistics, there's strong evidence to reject the null hypothesis and conclude that the heart transplant treatment increases patients' survival rate.

In [17]:
```
num_rates_larger <- sum(mortality_rate_diffs > 0.2302)
percentage_rates_larger <- num_rates_larger / 1000
cat("In", num_rates_larger,  "or,",percentage_rates_larger, "% of the simulations the mortality rate difference
```

```
In 1 or, 0.001 % of the simulations the mortality rate difference is larger than 23.02%.
```