Homework 5

Neal Kar (ink2105)

3/3/2020

Announcement

Please do not add code folding (code_folding: hide) to your YAML Header or echo = FALSE to your RMD code chunk options. In order to accurately grade your HTML files we need to be able to see all of your code. Thank you!

Question 1

a)

Write a function that returns the fifth element of a vector. If the vector has a length smaller than 5, return a string that says “The input vector is too small!”.

```
#Create the function
return_5th_el <- function(vec) {

  if(is.vector(vec) & !is.na(vec[5])) {
      return(vec[5])
  }

  else if(is.vector(vec) & is.na(vec[5])) {
    return("The input vector is too small!")
  }

}

#Test the function
return_5th_el(c(0,1,2,3,4,5,6,7))
```

```
## [1] 4
```

```
return_5th_el(c("a","b","c","d","e"))
```

```
## [1] "e"
```

```
return_5th_el(c(0,"a",1,"b"))
```

```
## [1] "The input vector is too small!"
```

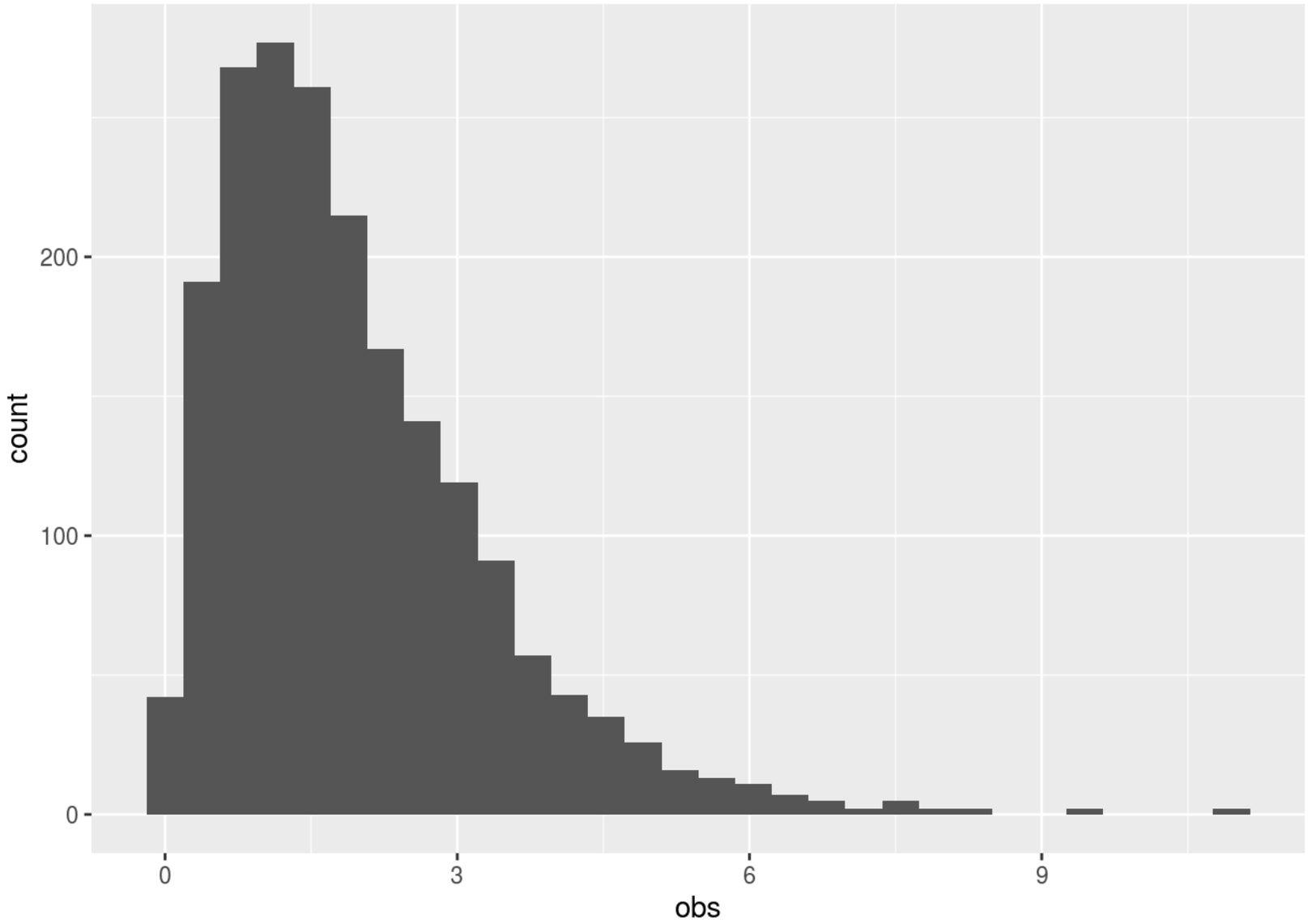
```
return_5th_el(c(100))
```

```
## [1] "The input vector is too small!"
```

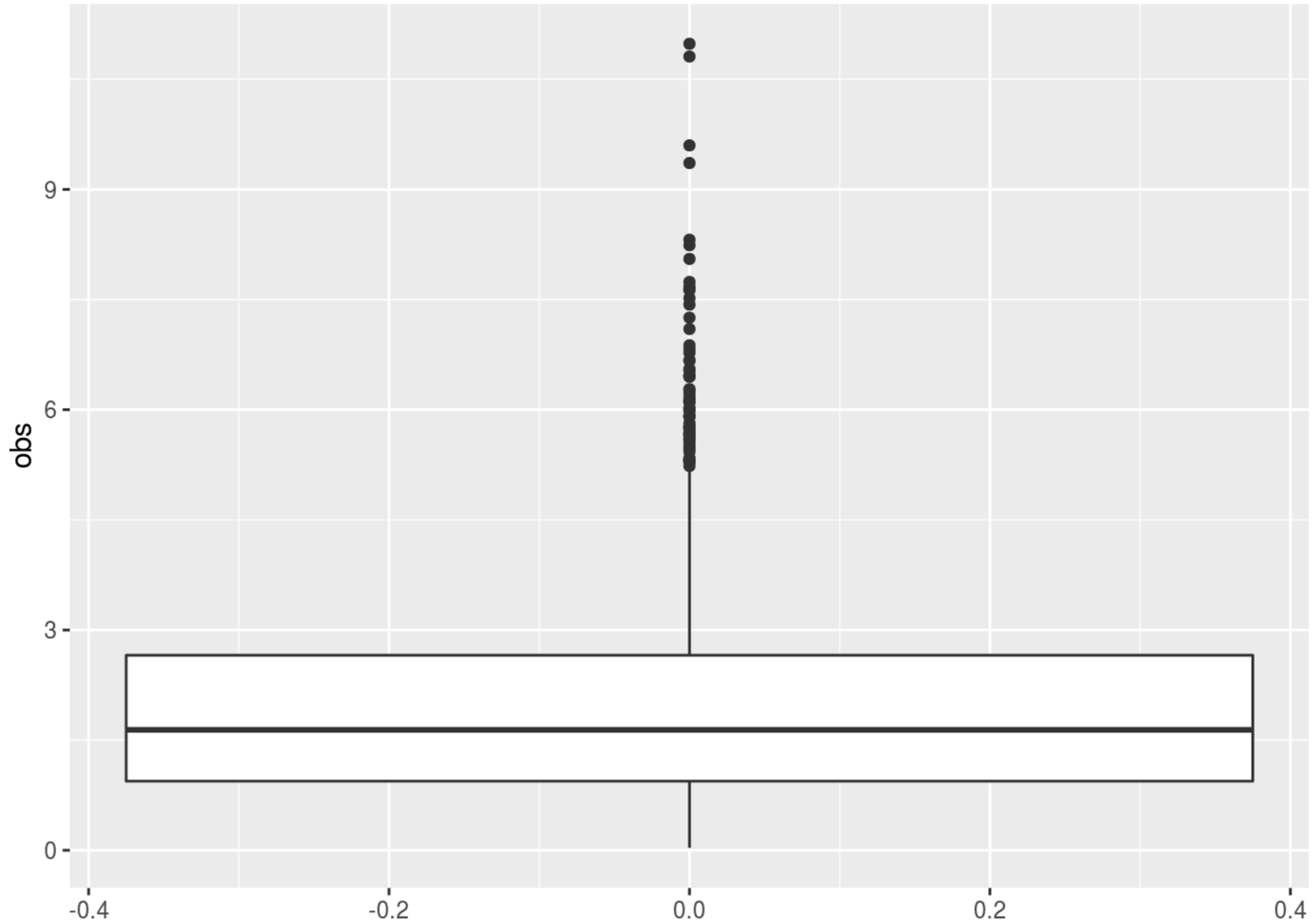
b)

Write a function that takes two arguments, `df` and `graph_type`. If `graph_type` equals "histogram", make a histogram; if graph_type equals "box plot" make a boxplot. The input `df` is a dataframe and the variable to graph will always be `obs`. Test your function by creating a histogram and a box plot of the dataset found in `data/distro_sim.csv`.

```r
#Create the function
create_graph_v1 <- function(df, graph_type) {

  if(graph_type == "histogram") {
    df <- read_csv(df)
      graph <- ggplot(data = df) +
        geom_histogram(aes(x=obs))
    return(graph)
  }

  else if(graph_type == "box plot") {
    df <- read_csv(df)
      graph <- ggplot(data = df) +
        geom_boxplot(aes(y=obs))
    return(graph)
  }

}

#Test the function
create_graph_v1("data/distro_sim.csv", "histogram")
```



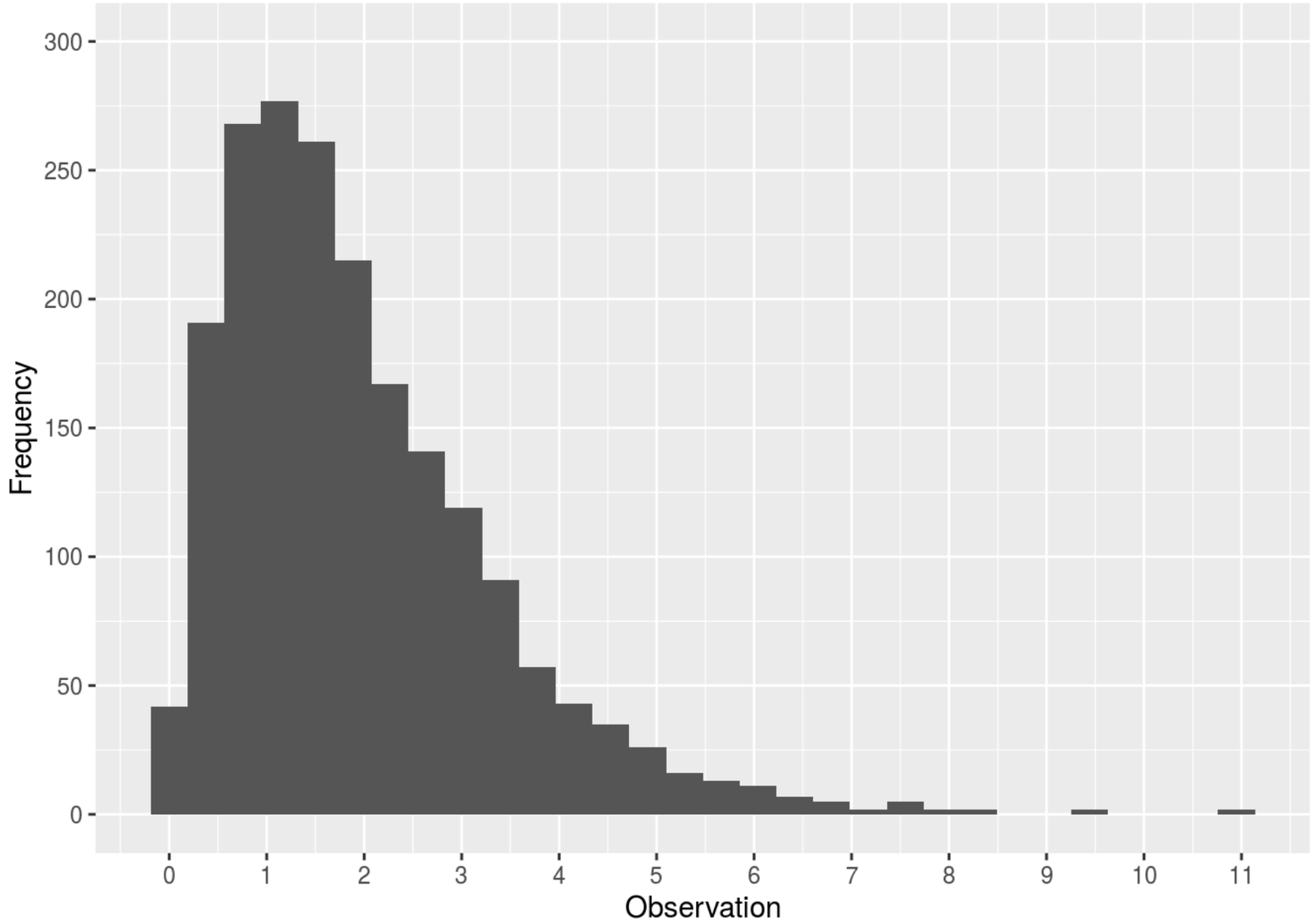

```
create_graph_v1("data/distro_sim.csv", "box plot")
```


obs
9
6
3
0
-0.4
-0.2
0.0
0.2
0.4

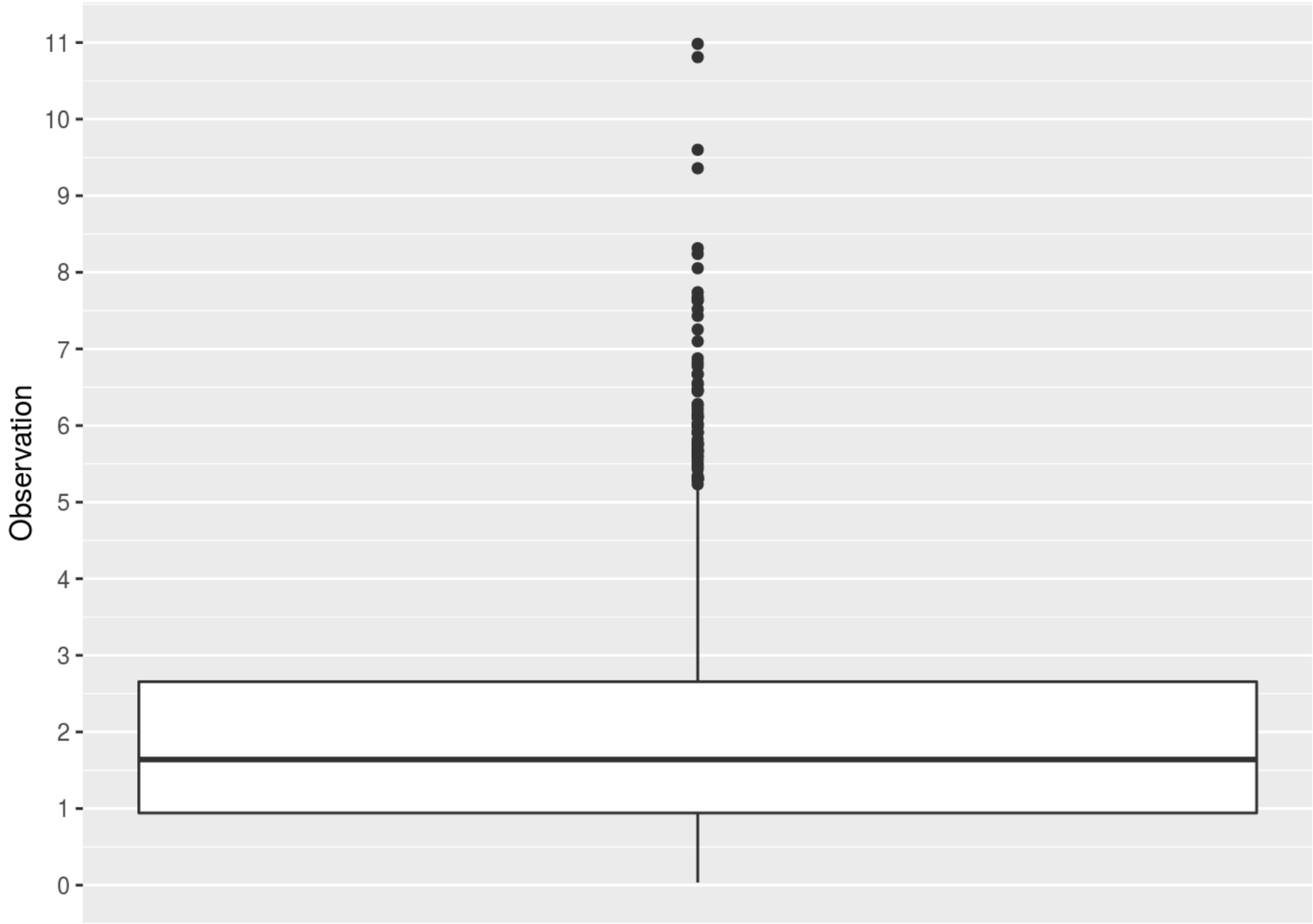
c)

Improve the graphs from part (b) by altering their themes, colors, titles, labels, or axes. Make two improvements to each graph and then create a histogram and box plot of the `data/distro_sim.csv` *dataset. Explain your improvements in sentences below your new graphs.*

```r
#Create the function
create_graph_v2 <- function(df, graph_type) {

  if(graph_type == "histogram") {
    df <- read_csv(df)
      graph <- ggplot(data = df) +
        geom_histogram(aes(x=obs)) +
        scale_x_continuous(breaks=0:11, labels=0:11) +
        scale_y_continuous(limits=c(0,300), breaks=seq(from=0,to=300,by=50)) +
        labs(x = "Observation", y = "Frequency")
    return(graph)
  }

  else if(graph_type == "box plot") {
    df <- read_csv(df)
      graph <- ggplot(data = df) +
        geom_boxplot(aes(y=obs)) +
        scale_x_continuous(breaks=NULL, labels=NULL) +
        scale_y_continuous(breaks=0:11, labels=0:11) +
        labs(y = "Observation")
    return(graph)
  }

}

#Test the function
create_graph_v2("data/distro_sim.csv", "histogram")
```




```
create_graph_v2("data/distro_sim.csv", "box plot")
```



For the histogram, I added more axis labels to clarify what the values on each axis represent. In addition, I altered the axes so that each observation value is visible on the x-axis and it’s clear what the frequency of a given observation is on the y-axis.

For the boxplot, I added a y-axis label to clarify what the values on the y-axis represent. In addition, I altered the axes so that each observation value is visible on the y-axis and there are no values on the x-axis since there are no discrete values on the x-axis (i.e. there's only the one box).

Question 2

a)

The code below contains a function which calculates power. This function contains errors which prevent it from behaving correctly and needs debugging. Rename and debug this function so it works correctly. Note: one of the errors IS NOT that it uses the normal distribution to compute power.

```
calculate_power <- function(alt_mu, null_mu, s, n){
  alt_z <-(alt_mu - null_mu)/(s/sqrt(n))
  alt_hypothesis <- rnorm(10000, alt_z, 1)
  alt_tibble <- tibble(obs = alt_hypothesis, scenario = "Alternative Hypothesis") %>%
    mutate(region = if_else(obs < -1.96 | obs > 1.96, "rejection", "non-rejection"))

  power_df <- alt_tibble %>%
    group_by(region) %>%
    summarize(n = n()) %>%
    mutate(proportion = n/sum(n))

  power <- power_df %>%
    filter(region == "rejection") %>%
    pull(proportion) %>%
    round(4)
return(power)
}
```

b)

Test the corrected function from (a) using the following inputs: alt_mu = 45, null_mu = 40, s = 4, n = 15. Report the power in a sentence.

```
power <- calculate_power(45, 40, 4, 15)
power
```

```
## [1] 0.998
```

Power is 99.8%.

Question 3

You have received four separate datasets for a small study on the effectiveness of a new drug on depression and anxiety. The first dataset contains demographic data collected at admission and is saved as `data/study832_demos.csv`. *The second dataset contains randomization information and is saved as* `data/study832_randomization.csv`. *The third dataset contains measurements of HAM-D and HAM-A at baseline and is saved as* `data/study832_baseline.csv". The fourth dataset contains measurements of HAM-D and HAM-A at the end of the study`

a)

Load all four datasets into your R environment. Take a look at each dataset and report the primary key (unique identifier) of each dataset.

```
demos_info <- read_csv("data/study832_demos.csv")
randomiz_info <- read_csv("data/study832_randomization.csv")
baseline_data <- read_csv("data/study832_baseline.csv")
outcome_data <- read_excel("data/study832_study_end.xlsx", sheet="study832_study_end")
```

For the demographic data, the primary key is “study_id”.

For the randomization information, the primary key is “patient_id”.

For the baseline data, the primary key is “id”.

For the end-of-the-study outcome data, the primary key is “study_id”.

b)

Join the demographic and randomization data frames together. Use a join that will drop any individuals who were not randomized.

```
joined_demos_randomiz <- right_join(demos_info, randomiz_info, by=c("study_id" = "patient_id"))
```

c)

Join the data frame from (b) with the baseline HAM-D and HAM-A dataset. Use a join that will drop any individuals who were not in the data frame from (b).

```
joined_demos_randomiz_baseline <- left_join(joined_demos_randomiz, baseline_data, by=c("study_id" = "i
d"))
```

d)

Using the data frame from (c), join the outcome data from the `study832_study_end.xlsx` file and create new change score variables for change in HAM-D and change in HAM-A (the score at the end of the study minus the score at baseline). How many observations are missing values for HAM-A change and how many observations are missing values for HAM-D change?

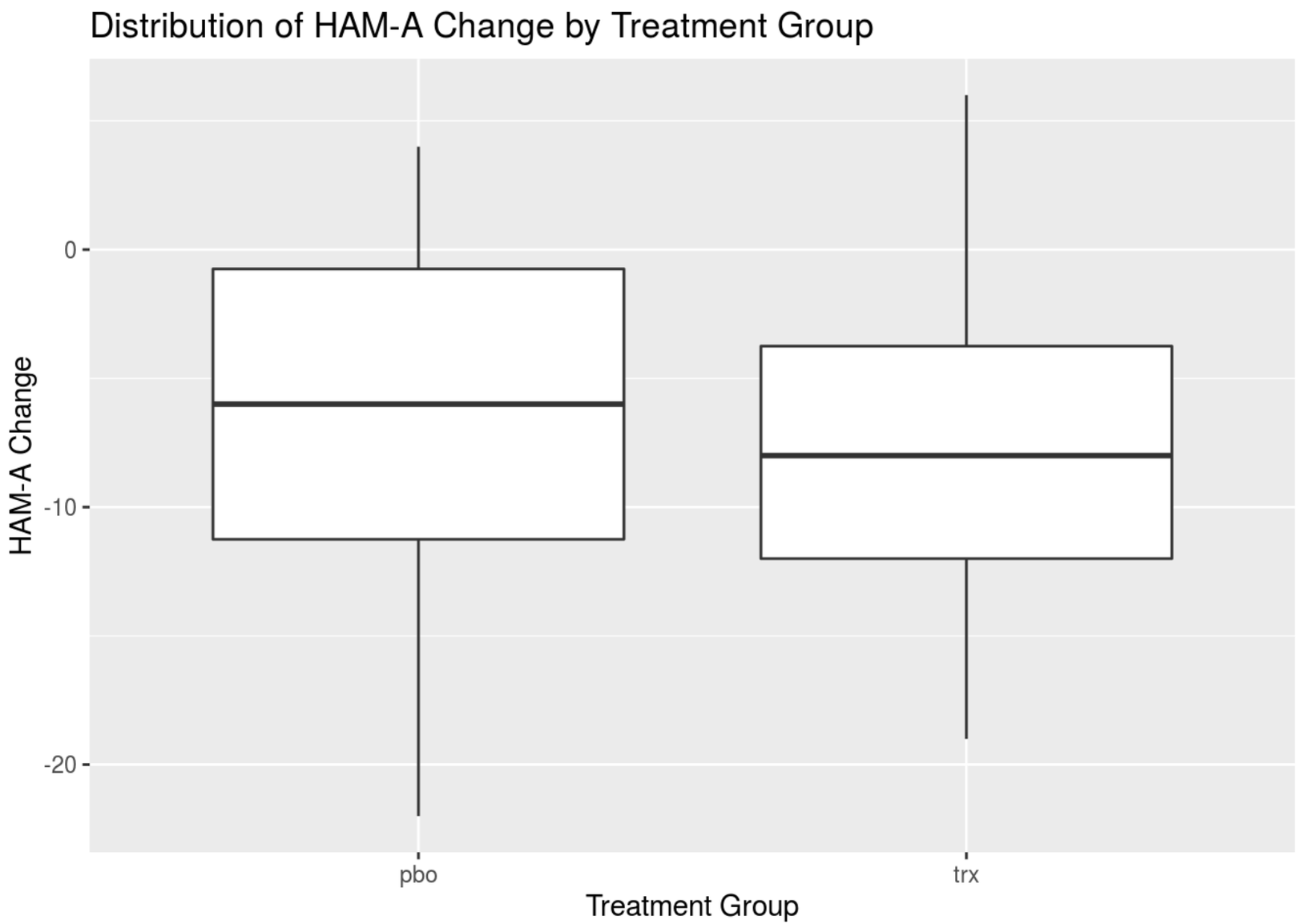
```
joined_all <- full_join(joined_demos_randomiz_baseline, outcome_data, by = "study_id")

joined_all <- joined_all %>%
  mutate(change_hamd = outcome_hamd - baseline_hamd, change_hama = outcome_hama - baseline_hama)

missing_change_hamd <- joined_all %>%
  filter(is.na(change_hamd))

missing_change_hama <- joined_all %>%
  filter(is.na(change_hama))

num_miss_hamd <- nrow(missing_change_hamd)
num_miss_hama <- nrow(missing_change_hama)
```

There are 28 observations missing a value for HAM-A change. There are 28 observations missing a value for HAM-D change.

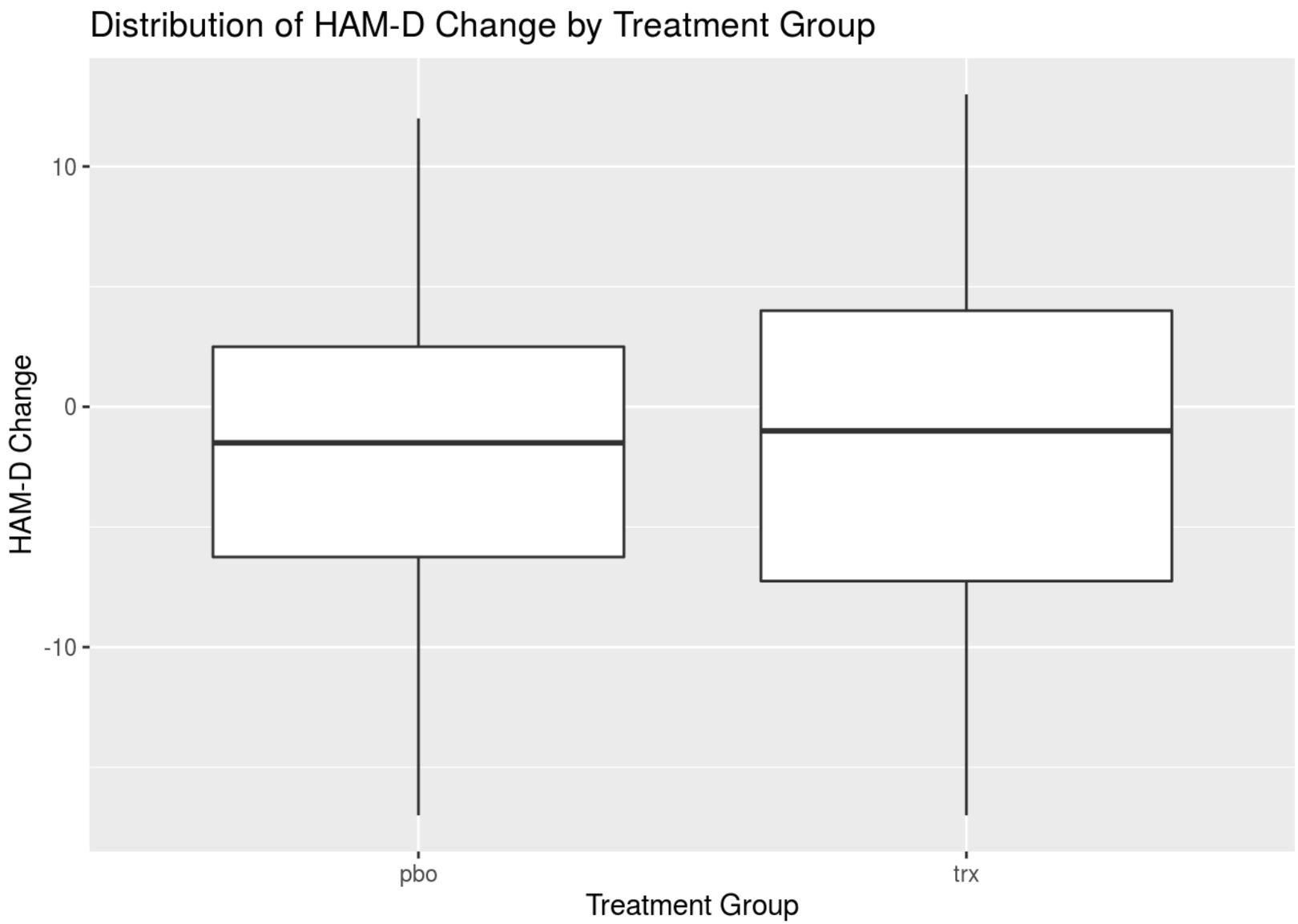
e)

Create two boxplots: one which shows the distribution of HAM-A changes scores by treatment group, and one that shows HAM-D change scores by treatment group.

```
#Create boxplot for HAM-A change
ggplot(data = joined_all) +
  geom_boxplot(aes(x=trx, y=change_hama)) +
  labs(title= "Distribution of HAM-A Change by Treatment Group", x = "Treatment Group", y = "HAM-A Chan
ge")
```


Distribution of HAM-A Change by Treatment Group
HAM-A Change
0
-10
-20
pbo
trx
Treatment Group

```
#Create boxplot for HAM-D change
ggplot(data = joined_all) +
  geom_boxplot(aes(x=trx, y=change_hamd)) +
  labs(title= "Distribution of HAM-D Change by Treatment Group", x = "Treatment Group", y = "HAM-D Chan
ge")
```


Distribution of HAM-D Change by Treatment Group
HAM-D Change
10
0
-10
pbo
trx
Treatment Group

f)

Use t-tests (unequal variance) to test the difference between treatment groups for change in HAM-D scores and change in HAM-A scores. Be sure to report the test statistic, p-value, and degrees of freedom for each test in your write-up.

```
ttest_hama <- t.test(change_hama ~ trx, data = joined_all)
tstat_hama <- ttest_hama$statistic
df_hama <- ttest_hama$parameter
p_hama <- ttest_hama$p.value

ttest_hamd <- t.test(change_hamd ~ trx, data = joined_all)
tstat_hamd <- ttest_hamd$statistic
df_hamd <- ttest_hamd$parameter
p_hamd <- ttest_hamd$p.value
```

Below are the statisics examing the difference in mean HAM-A change between treatment groups:

1. Test statistic: 0.8052282
2. Degrees of freedom: 34.7938813
3. p-value: 0.4261567

Since p > 0.05, there was no significant difference in mean HAM-A change between the treatment groups.

Below are the statisics examing the difference in mean HAM-D change between treatment groups:

1. Test statistic: -0.1205093
2. Degrees of freedom: 37.964089
3. p-value: 0.9047153

Since p > 0.05, there was no significant difference in mean HAM-D change between the treatment groups.

Question 4

Investigators are interested in whether overall milk production in lame dairy cows depends on the breed of the cows (Local Zebu, or Boran). During a small study, they followed 5 lame cows from each breed for 1 year and recorded their milk production in liters/month:

Local Zebu cows: 552, 616, 637, 563, 541 liters/month

Boran cows: 704, 739, 770, 748, 750 liters/month

a)

Enter this data into a tibble so that you will be able to manipulate it easily. There should be one column for the breed of cow, and one column for the milk measurements.

```
zebu <- c("Local Zebu", "Local Zebu", "Local Zebu", "Local Zebu", "Local Zebu")

boran <- c("Boran", "Boran", "Boran", "Boran", "Boran")

milk_production_data <- tibble(cow_breed = c(zebu, boran), milk_measurement = c(552, 616, 637, 563, 54
1, 704, 739, 770, 748, 750))
```

b)

*Perform a **two-sided** permutation test to test whether there are mean differences in lame milk production between Local Zebu and Boran cow breeds. Perform 1000 permutations for your test. Make sure to:*

(1) Graph the distribution of your simulated test statistics for the 1000 permutations.

(2) Include a dashed vertical line marking the observed test statistic.

(3) Report the p-value and the result of your permutation test in a sentence.

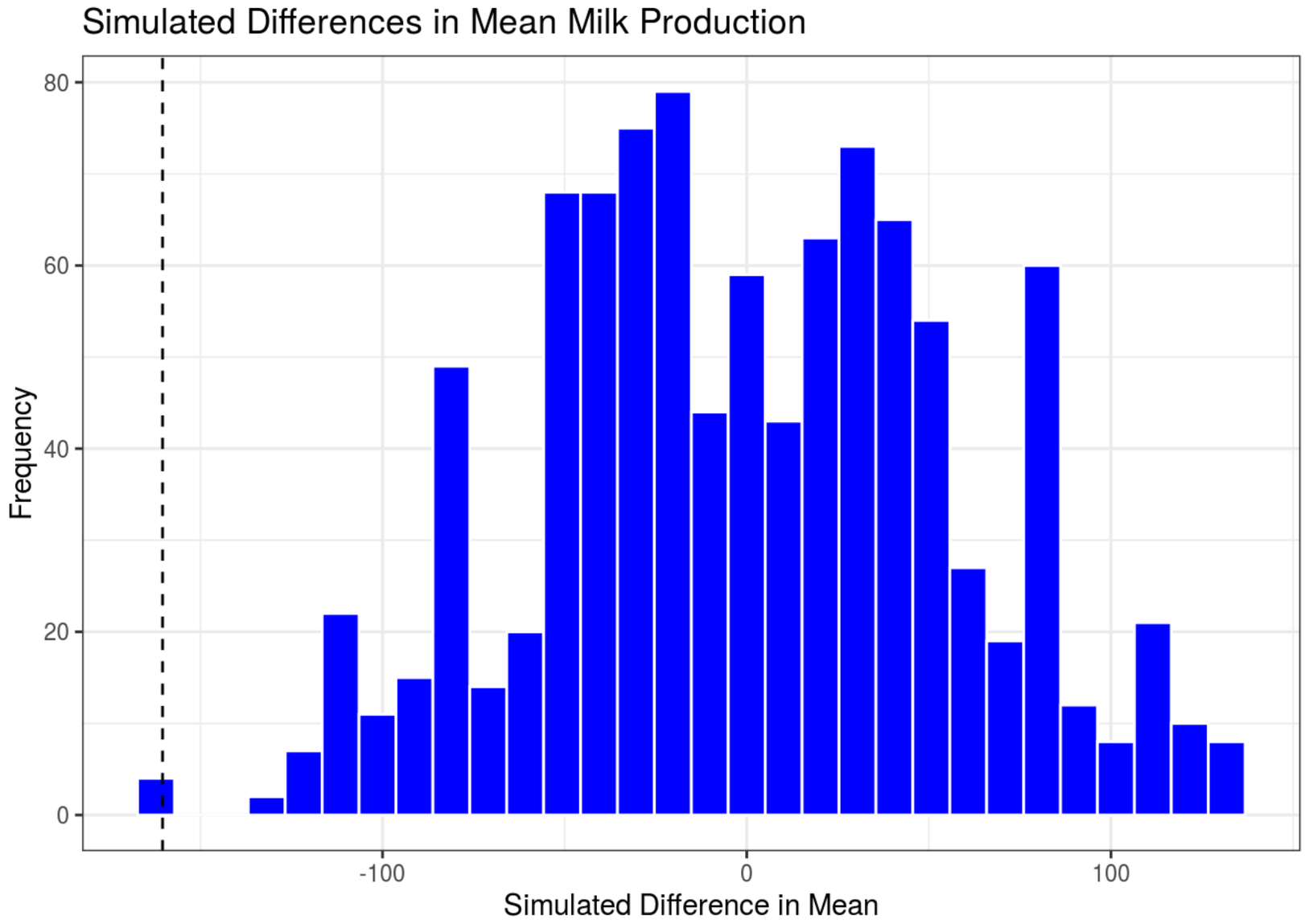
```
# Create a function to calculate the test statistic

calculate_ts <- function(df){
  summary <- df %>%
    group_by(cow_breed) %>%
    summarize(mean_milk_prod = mean(milk_measurement))

  zebu_prod <- summary %>%
    filter(cow_breed == "Local Zebu") %>%
    pull(mean_milk_prod)

  boran_prod <- summary %>%
    filter(cow_breed == "Boran") %>%
    pull(mean_milk_prod)

  difference <- zebu_prod - boran_prod
  return(difference)
}

obs_stat <- calculate_ts(milk_production_data)
obs_stat
```

```
## [1] -160.4
```

```r
# Create a function that performs a single permutation.

perform_permutation <- function(df){
  permuted <- df %>%
    mutate(milk_measurement = sample(milk_measurement))
  return(permuted)
}

permuted <- perform_permutation(milk_production_data)


# Combine the permutation function and the calculation of the test statistic
do_permutation_test <- function(df){
  permed <- perform_permutation(df)
  ts <- calculate_ts(permed)
  return(ts)
}

do_permutation_test(milk_production_data)
```

```
## [1] -76.8
```

```r
# Now we want to do this 1000 times

results <- map_dbl(1:1000, function(x) do_permutation_test(milk_production_data))

# Put into a tibble for graphing:
res_tibble <- tibble(sim_stat = results)



# Graph all of these permuted test statistics to the observed difference in means

ggplot(data = res_tibble) +
  geom_histogram(aes(x = sim_stat), bins = 30, fill = "blue", color = "white") +
  geom_vline(aes(xintercept = obs_stat), linetype = "dashed") +
  theme_bw() +
  labs(title="Simulated Differences in Mean Milk Production", x="Simulated Difference in Mean", y="Freq
uency")
```


Simulated Differences in Mean Milk Production
Frequency
80
60
40
20
0
-100
0
100
Simulated Difference in Mean

```
# Compare simulated test statistics to observed test statistic to calculate p-value. The proportion of simulated test statistics whose absolute value is greater than or equal to the observed test statistic will be our p-value:

greater_than <- res_tibble %>%
  mutate(abs_val_greater = if_else(abs(sim_stat) >= abs(obs_stat), 1, 0)) %>%
  filter(abs_val_greater == 1) %>%
  summarize(n = n()) %>%
  pull(n)

permute_pval <- greater_than/1000

permute_pval
```

```
## [1] 0.004
```

The observed difference in mean milk production between the cow breeds was 160.4 liters/month (with Boran cows producing more than Local Zebu cows), and the absolute value of 4 simulated differences in means (out of 1000) were greater than the observed difference, giving us a p-value of 0.004, indicating there was a significant difference in mean milk production between the two cow breeds.

c)

*Perform a **one-sided** permutation test to test whether there are median differences in lame milk production between Local Zebu and Boran cow breeds. Perform 5000 permutations for your test. Make sure to:*

(1) Graph the distribution of your simulated test statistics for the 5000 permutations.

(2) Include a dashed vertical line marking the observed test statistic.

(3) Report the p-value and the result of your permutation test in a sentence.

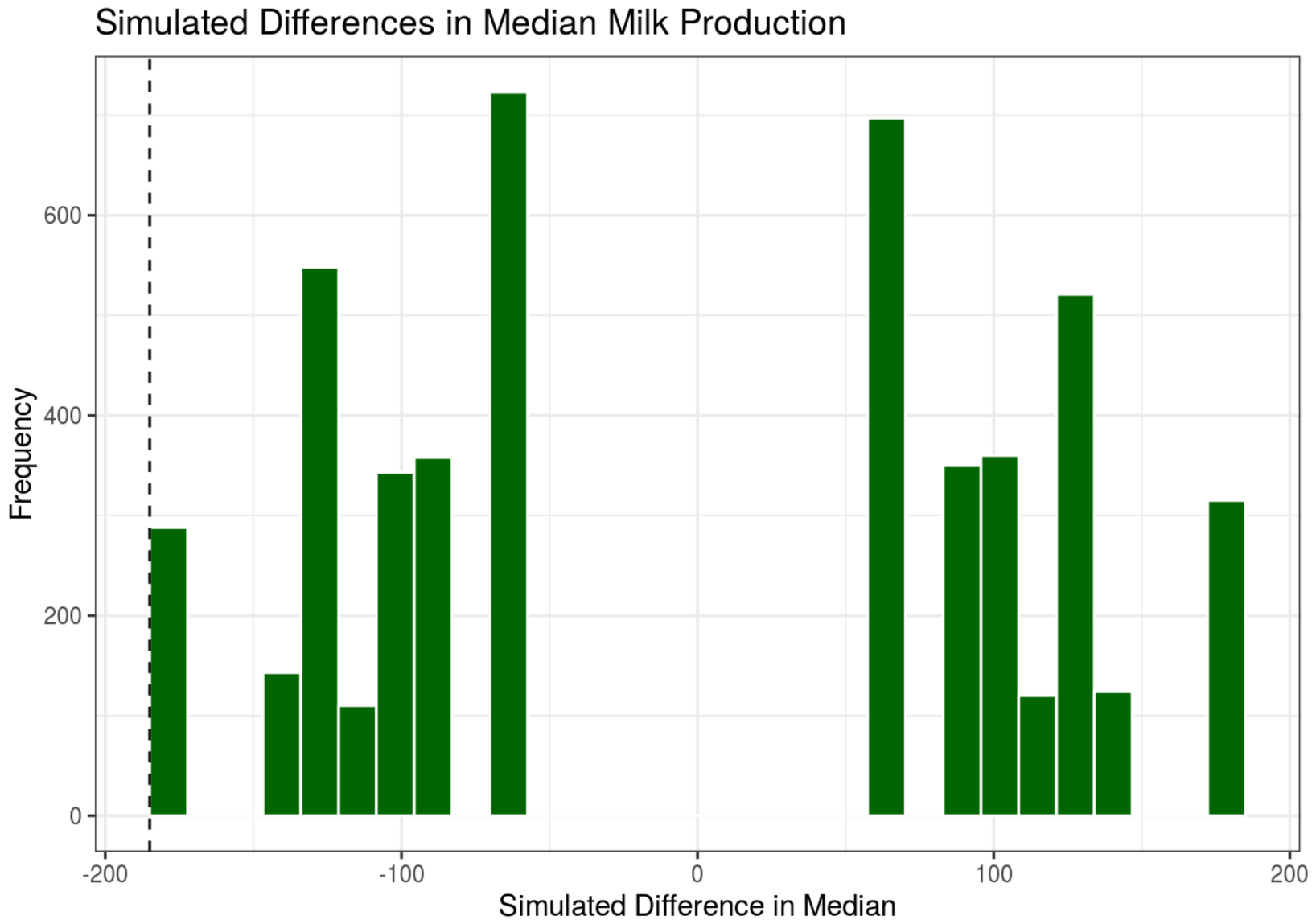
```
calculate_ts_med <- function(df){
  summary <- df %>%
    group_by(cow_breed) %>%
    summarize(milk_prod_med = median(milk_measurement))

  zebu_med <- summary %>%
    filter(cow_breed == "Local Zebu") %>%
    pull(milk_prod_med)

  boran_med <- summary %>%
    filter(cow_breed == "Boran") %>%
    pull(milk_prod_med)

  difference <- zebu_med - boran_med
  return(difference)
}

# Observed variance test statistic:
obs_stat_med <- calculate_ts_med(milk_production_data)
obs_stat_med
```

```
## [1] -185
```

```r
# Now create a function that performs a single permutation.

perform_permutation_med <- function(df){
  permuted <- df %>%
    mutate(milk_measurement = sample(milk_measurement))
  return(permuted)
}

permuted_med <- perform_permutation_med(milk_production_data)



# Combining the permutation function and the calculation of the test statistic
do_permutation_test_med <- function(df){
  permed <- perform_permutation_med(df)
  ts <- calculate_ts_med(permed)
  return(ts)
}

do_permutation_test_med(milk_production_data)
```

```
## [1] -123
```

```
# Now do this 5000 times:

# Full results:
results_med <- map_dbl(1:5000, function(x) do_permutation_test_med(milk_production_data))

# Put into a tibble for graphing:
res_tibble_med <- tibble(sim_stat_med = results_med)

# Plot results
ggplot(data = res_tibble_med) +
  geom_histogram(aes(x = sim_stat_med), bins = 30, fill = "darkgreen", color = "white") +
  geom_vline(aes(xintercept = obs_stat_med), linetype = "dashed") +
  theme_bw() +
  labs(title="Simulated Differences in Median Milk Production", x="Simulated Difference in Median", y
="Frequency")
```


Simulated Differences in Median Milk Production
Frequency
600
400
200
0
-200
-100
0
100
200
Simulated Difference in Median

```
# Compare simulated test statistics to observed test statistic to calculate p-value. The proportion of simulated test statistics whose value is greater than or equal to the observed test statistic will be our p-value. Note that we are not using absolute value because this is a one-sided test.

greater_than_med <- res_tibble_med %>%
  mutate(med_greater = if_else(abs(sim_stat_med) >= abs(obs_stat_med), 1, 0)) %>%
  filter(med_greater == 1) %>%
  summarize(n = n()) %>%
  pull(n)

permute_pval_med <- greater_than_med/5000

permute_pval_med
```

```
## [1] 0.047
```

The observed difference in median milk production between the cow breeds was 185 liters/month (with Boran cows producing more than Local Zebu cows), and the absolute value of 235 simulated differences in medians (out of 5000) were greater than the observed difference, giving us a p-value of 0.047, indicating there was a significant difference in median milk production between the two cow breeds.