Homework 02

Neal Kar (ink2105)

2/11/2020

Question 0

Add your name and the date to the R Markdown Header.

Insert a floating table of contents to the HTML.

Question 1

This data was collected from www.theramenrater.com. It provides information on different reviews of ramen products, and has variables: Review #, Brand, Variety, Style, Country, Stars, and Top Ten.

(a)

Read in the Ramen data and check it carefully.

```
ramen <- read_csv("data/ramen-ratings.csv")
```

```
## Parsed with column specification:
## cols(
##   `Review #` = col_double(),
##   Brand = col_character(),
##   Variety = col_character(),
##   Style = col_character(),
##   Country = col_character(),
##   Stars = col_character(),
##   `Top Ten` = col_character()
## )
```

(b)

Rename the first variable to be “review_number”, and the last variable to be “top_ten”. Additionally, ensure that the Stars column is saved as a numeric variable, and remove any non-numeric entries.

```
ramen <- rename(ramen, review_number = "Review #", top_ten = "To
p Ten")

ramen_filtered <- ramen %>%
  filter(Stars != "Unrated")

#Check variable type
typeof(ramen_filtered$Stars)
```

```
## [1] "character"
```

```
#Change to numeric
ramen_filtered$Stars <- as.numeric(ramen_filtered$Stars)

#Check variable type again to confirm
typeof(ramen_filtered$Stars)
```

```
## [1] "double"
```

(c)

Filtering just for the the ‘Nissin’ ramen brand, calculate the average rating for each country of this brand. What country has the highest rating of Nissin ramen? What country has the lowest rating?

```
ramen_nissin <- ramen_filtered %>%
  filter(Brand == "Nissin") %>%
  group_by(Country) %>%
  summarize(mean_rating = mean(Stars))

view(ramen_nissin)

#Manually sort mean_rating in ascending and descending order
```

Brazil has the highest rating of Nissin ramen. The Philippines has the lowest rating of Nissin ramen.

(d)

Create a new variable called “popular” which returns a 1 for entries that have a rating above or equal to 4.5 stars, and 0 for those that don’t.

```
ramen_filtered <- mutate(ramen_filtered, popular = if_else(Stars
>= 4.5, 1, 0))
```

(e)

Calculate the average stars for popular and not popular bowl ramen. Explain why (or why not) your results are meaningful.

```
ramen_popular <- ramen_filtered %>%
  group_by(popular) %>%
  summarize(mean_rating_pop = mean(Stars))

view(ramen_popular)
```

The results are meaningful because it suggests ramen popularity may be positively associated with ramen ratings (or vice-versa). In other words, ramen that is popular is rated substantially higher, on average, than unpopular ramen.

Question 2

(a)

Read in NYC Airbnb data, and select only the: host_name, neighborhood_group, room_type, price, minimum_nights, and number_of_reviews. Note, the neighborhood_group variable refers to borough.

```
nyc_airbnb <- read_csv("data/AB_NYC_2019.csv")
```

```
## Parsed with column specification:
## cols(
##   id = col_double(),
##   name = col_character(),
##   host_id = col_double(),
##   host_name = col_character(),
##   neighbourhood_group = col_character(),
##   neighbourhood = col_character(),
##   latitude = col_double(),
##   longitude = col_double(),
##   room_type = col_character(),
##   price = col_double(),
##   minimum_nights = col_double(),
##   number_of_reviews = col_double(),
##   last_review = col_date(format = ""),
##   reviews_per_month = col_double(),
##   calculated_host_listings_count = col_double(),
```

```
##   availability_365 = col_double()
## )
```

```
nyc_airbnb <- select(nyc_airbnb, host_name, neighbourhood_group,
room_type, price, minimum_nights, number_of_reviews)
```

(b)

Create a new variable called minimum_price, which combines the minimum nights and price (per night) to give the minimum amount someone could pay to stay at the Airbnb.

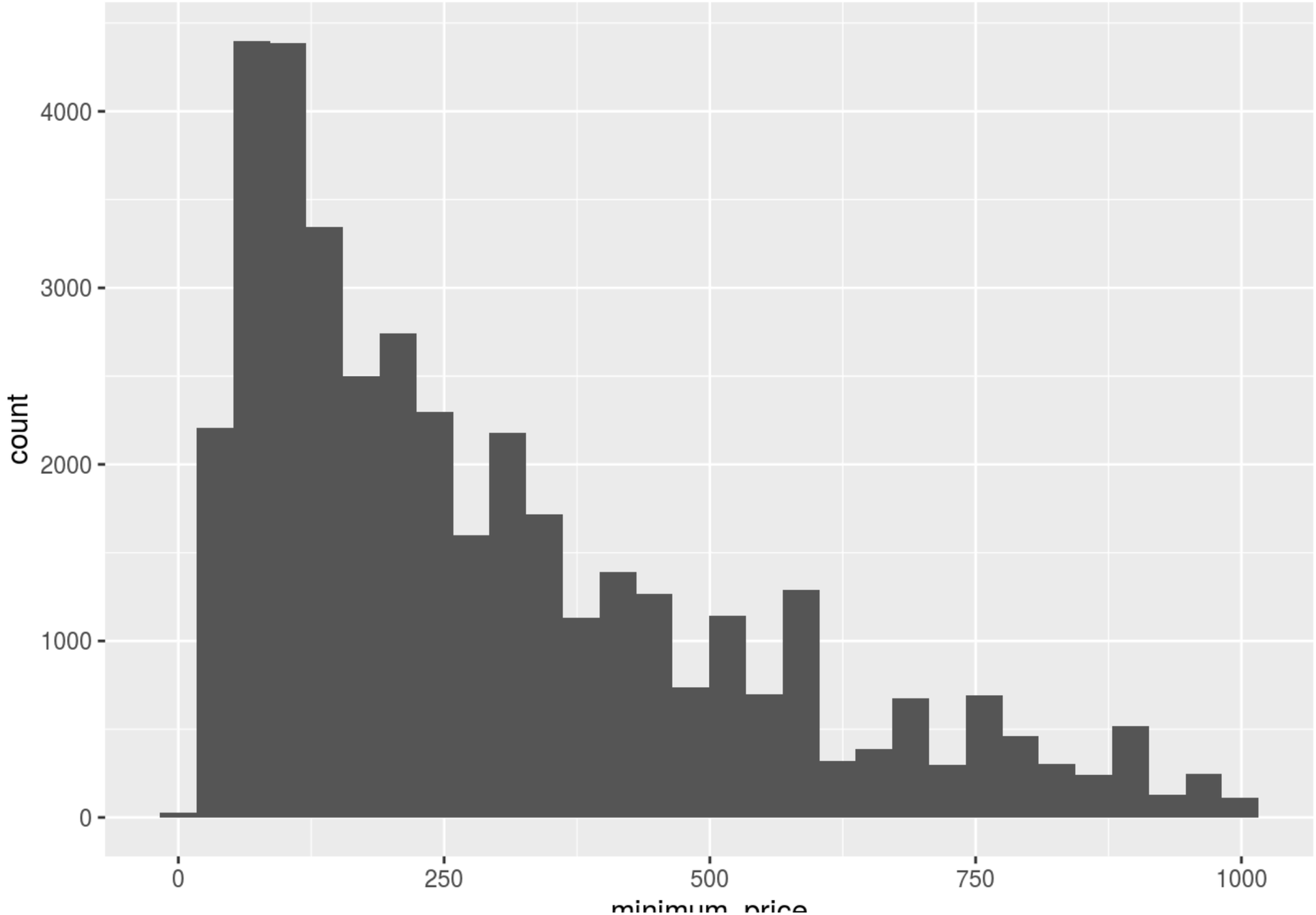
```
nyc_airbnb <- mutate(nyc_airbnb, minimum_price = price*minimum_n
ights)
```

(c)

Calculate the mean and median minimum_price using ‘summarize’. From these results, estimate whether you expect the data to be left or right skewed? Confirm your hypothesis by creating a histogram. Note, to improve the visualization of your histogram, consider removing very high prices.

```
nyc_airbnb_stats <- nyc_airbnb %>%
  summarize(mean_min_price = mean(minimum_price), med_min_price
= median(minimum_price))

view(nyc_airbnb_stats)

nyc_airbnb_hist <- nyc_airbnb %>%
  filter(minimum_price < 1000)

ggplot(data = nyc_airbnb_hist) +
  geom_histogram(aes(x=minimum_price))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwi
dth`.
```


4000
3000
2000
1000
0
count
0
250
500
750
1000
minimum_price

minimum_price

Since the median is less than the mean, I would expect the data to be right-skewed. The histogram confirms my expectation.

(d)

Are all of the New York City boroughs represented in the data? Prove your conclusion using the summarize function.

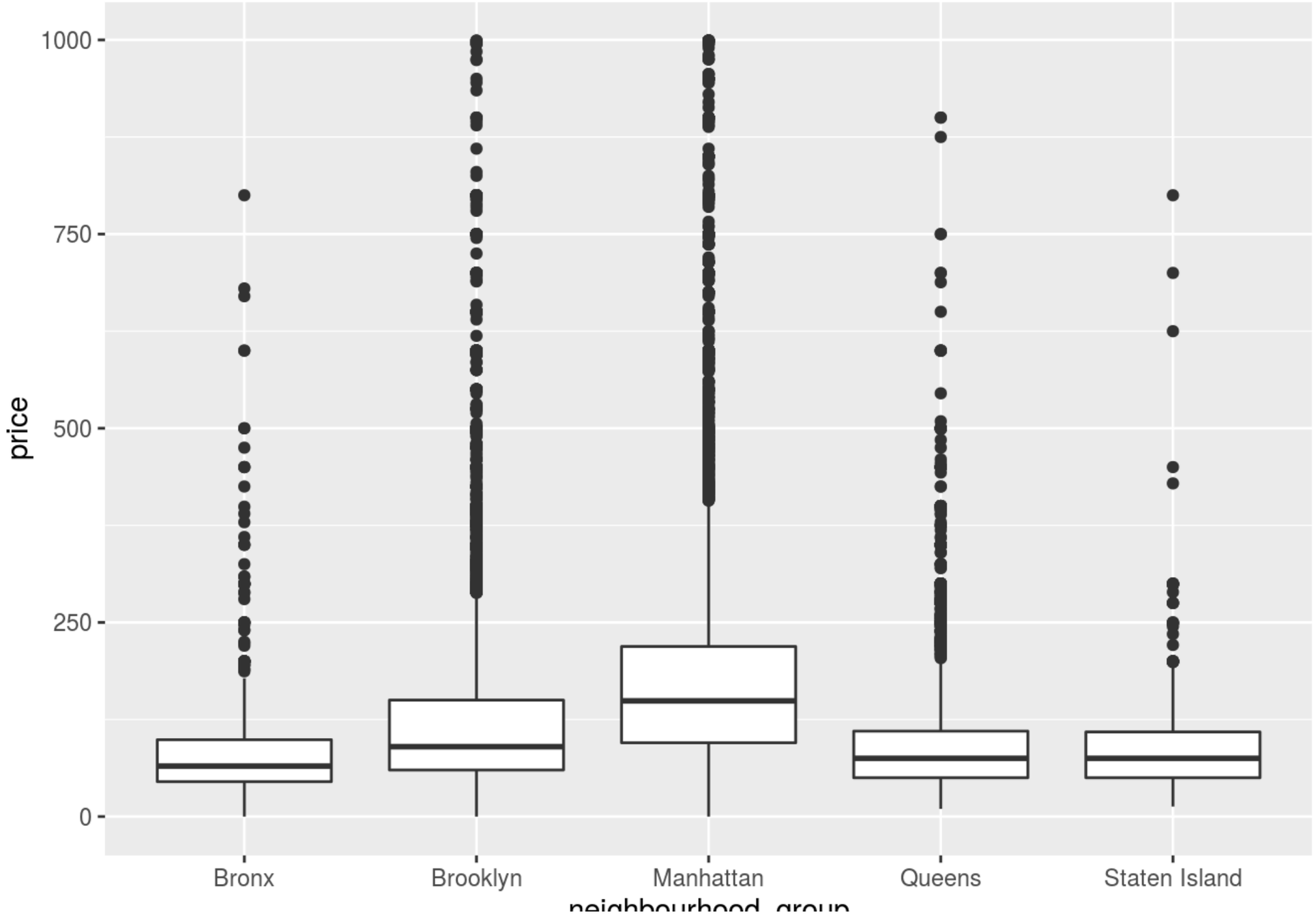
```
nyc_borough <- nyc_airbnb %>%
  group_by(neighbourhood_group) %>%
  summarize(mean_min_price = mean(minimum_price))

view(nyc_borough)
```

Yes, all 5 NYC boroughs are represented in the data.

(e)

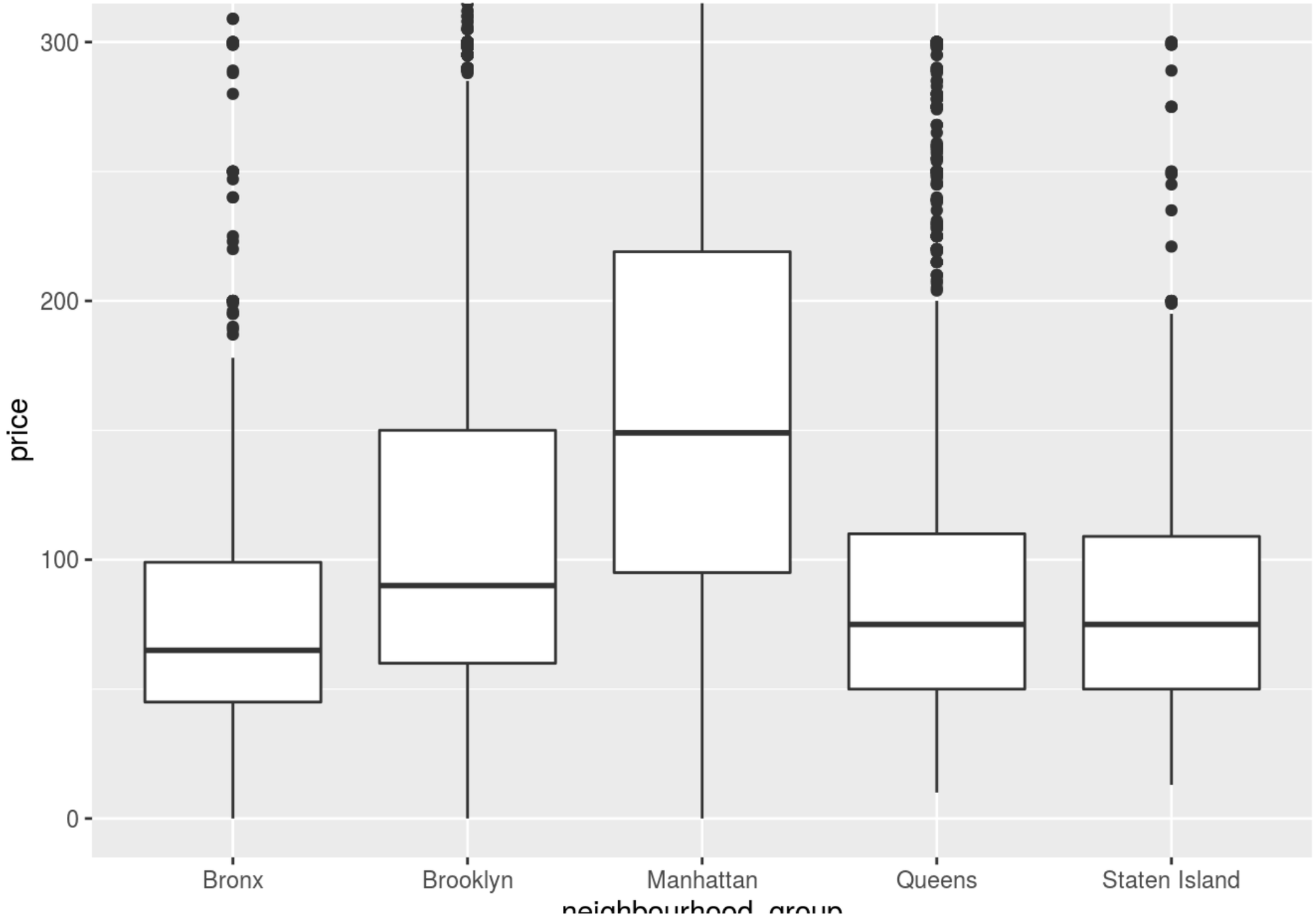
Plot a boxplot of price across boroughs, showing only properties less than 1,000 a night. From the graph, which borough appears to have the highest median price? Which seems to have the lowest median price? Confirm this result using summarize and report the median price by borough.

```
nyc_airbnb_1000 <- filter(nyc_airbnb, price < 1000)

ggplot(data = nyc_airbnb_1000) +
  geom_boxplot(aes(x=neighbourhood_group, y=price))
```


1000
750
500
250
0
price
Bronx
Brooklyn
Manhattan
Queens
Staten Island
neighbourhood_group

neighbourhood_group

```
#Without limiting the y-axis range, it's difficult to tell which boroughs have the highest and lowest median prices, so limit the y-axis range to 0-300.

ggplot(data = nyc_airbnb_1000) +
  geom_boxplot(aes(x=neighbourhood_group, y=price)) +    coord_cartesian(ylim = c(0, 300))
```


300
200
100
0
price
Bronx
Brooklyn
Manhattan
Queens
Staten Island
neighbourhood_group

neighbourhood_group

```
#That provides a much clearer visual depiction to determine the
boroughs with the highest and lowest median prices

nyc_borough_prices <- nyc_airbnb_1000 %>%
  group_by(neighbourhood_group) %>%
  summarize(med_price = median(price))

view(nyc_borough_prices)
```

Based on the histogram, Manhattan appears to have the highest median price per night, and the Bronx appears to have the lowest median price per night. These guesses are confirmed by the data on median prices per night by borough, as shown below:

Bronx: $65; Brooklyn: $90; Manhattan: $149; Queens: $75; Staten Island: $75.

(f)

What is the most commonly occuring host name? Re-write the code below using pipes and then report the answer in a sentence.

```
nyc_host_names <- nyc_airbnb %>%
  group_by(host_name) %>%
  summarise(n = n(), -n) %>%
  arrange

view(nyc_host_names)

#Manually sort dataframe in descending order of variable "n."
```

“Michael” is the most commonly occurring host name.

Question 3

(a)

You have been given a dataset for a study of a potential depression drug. It includes treatment and placebo status, dose, age, along with HAM-D and HAM-A scores at baseline (`baseline_hamd` and `baseline_hama`) and at the end of the study (`outcome_hamd` and `outcome_hama`). Read in this dataset (`STUDYDAT12014.csv`) and report how many variables and how many observations are in this dataset.

```
drug_study <- read_csv("data/STUDYDAT12014.csv")
```

```
## Parsed with column specification:
## cols(
##   id = col_double(),
##   trx = col_character(),
##   dose = col_character(),
##   age = col_double(),
##   baseline_hama = col_double(),
##   baseline_hamd = col_double(),
##   outcome_hama = col_double(),
##   outcome_hamd = col_double()
## )
```

(b)

The investigators are interested in analyzing subjects who are at least 30 years old but less than 40 years old, so create a dataset which contains only observations from people within this age range.

```
drug_study_age30 <- filter(drug_study, age >= 30 & age < 40)
```

(c)

The treatment groups are currently labeled `pbo` and `trx`, but the researchers would like to have them labeled `Placebo` and `Drug 13XA`. Please make these changes to the dataset.

```
drug_study_age30 <- mutate(drug_study_age30, trx_group = if_else
(trx=="pbo", "Placebo", "Drug 13XA"))

drug_study_age30 <- select(drug_study_age30, id, trx_group, ever
ything())

drug_study_age30 <- select(drug_study_age30, -trx)

drug_study_age30 <- rename(drug_study_age30, trx = trx_group)
```

(d)

Create two new variables `hamd_diff` and `hama_diff` that are changes between baseline and outcome measurements for HAM-A and HAM-D.

```
drug_study_age30 <- mutate(drug_study_age30, hama_diff = outcome
_hama - baseline_hama, hamd_diff = outcome_hamd - baseline_hamd)
```

(e)

The investigators are interested in assessing whether there is a difference in (1) mean HAM-A changes between treatment groups and (2) mean HAM-D changes between treatment groups. Use t-tests (unequal variance) to test the difference between treatment groups for both of these outcomes. Be sure to report the test statistic, p-value, and degrees of freedom for each test in your write-up.

```
ttest_hama <- t.test(hama_diff ~ trx, data = drug_study_age30)

ttest_hamd <- t.test(hamd_diff ~ trx, data = drug_study_age30)
```

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

1. Test statistic: -1.0012236
2. Degrees of freedom: 38.9376515
3. p-value: 0.3229005

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

1. Test statistic: 0.8025073
2. Degrees of freedom: 45.1303189
3. p-value: 0.4264656

Since $p > 0.05$ for both mean differences, among participants 30-39 years old, the mean changes in HAM-A and HAM-D were not significantly different between the treatment group and control group.

Question 4

(a)

You have been provided with a dataset from a student-run cafe. This dataset contains data from a cafe, called Executive Express, run by undergraduate business students at a Midwestern public university. It was collected over a ten-week period from January to April 2010. Use what you have learned to read the data in, prepare it for analysis, and then calculate the necessary summary statistics to fill out the paragraph below, replacing the X’s with appropriate results.

```
cafe_data <- read_excel("data/cafedata.xls", sheet="cafedata")

cafe_data <- cafe_data %>%
  rename(wraps_sold = "Wraps Sold", muffins_sold = "Muffins Sol
d", cookies_sold = "Cookies Sold") %>%
  mutate(profit = if_else(Sales >= 160, 1, 0), muff_cook = muffi
ns_sold + cookies_sold)

#Find number of profitable days
cafe_profitable <- cafe_data %>%
  filter(profit == 1)

view(cafe_profitable)

#Find number of unprofitable days
cafe_unprofitable <- cafe_data %>%
  filter(profit == 0)

view(cafe_unprofitable)
```

```r
#Find mean and sd of wraps sold on profitable days
wraps_prof <- cafe_profitable %>%
  pull(wraps_sold)
  mean_wraps_sold_prof <- mean(wraps_prof)
  sd_wraps_sold_prof <- sd(wraps_prof)


#Find mean and sd of wraps sold on unprofitable days
wraps_unprof <- cafe_unprofitable %>%
  pull(wraps_sold)
  mean_wraps_sold_unprof <- mean(wraps_unprof)
  sd_wraps_sold_unprof <- sd(wraps_unprof)


#Find mean and sd of muffins and cookies sold on profitable days
muff_cook_prof <- cafe_profitable %>%
  pull(muff_cook)
```

```
  mean_muffcook_sold_prof <- mean(muff_cook_prof)
  sd_muffcook_sold_prof <- sd(muff_cook_prof)


#Find mean and sd of muffins and cookies sold on unprofitable da
ys
muff_cook_unprof <- cafe_unprofitable %>%
  pull(muff_cook)
  mean_muffcook_sold_unprof <- mean(muff_cook_unprof)
  sd_muffcook_sold_unprof <- sd(muff_cook_unprof)


#Find mean and sd of coffee sold on profitable days
coffee_prof <- cafe_profitable %>%
  pull(Coffees)
  mean_coffee_prof <- mean(coffee_prof)
  sd_coffee_prof <- sd(coffee_prof)
```

```
#Find mean and sd of coffee sold on unprofitable days
coffee_unprof <- cafe_unprofitable %>%
  pull(Coffees)
  mean_coffee_unprof <- mean(coffee_unprof)
  sd_coffee_unprof <- sd(coffee_unprof)


#Ttest of wraps sold by profitability days
ttest_wraps <- t.test(wraps_sold ~ profit, data=cafe_data)

#Ttest of muffins and cookies sold by profitability days
ttest_muff_cook <- t.test(muff_cook ~ profit, data=cafe_data)

#Ttest of coffee sold by profitability days
ttest_coffee <- t.test(Coffees ~ profit, data=cafe_data)
```

Students called a day a ‘profitable day’, when they had at least $160 in sales. There were 18 ‘profitable’ days and 29 ‘unprofitable’ days.

On ‘profitable’ days, the mean number of wraps sold was 16.1111111 with a standard deviation of 6.8245999. On ‘unprofitable’ days, the mean number of wraps sold was 11.3103448 with a standard deviation of 4.3760994.

On ‘profitable days’, on average they sold 13.7777778 muffins and cookies combined, with a standard deviation of 7.352915. On ‘unprofitable days’, on average they sold 10.3103448 muffins and cookies combined, with a standard deviation of 6.5416438.

The mean number of coffees sold on ‘profitable’ days was 25.6111111 with a standard deviation of 8.4445519 and the mean number of coffees sold on ‘unprofitable’ days was 18.9655172 with a standard deviation of 11.8697778.

When comparing profitable to nonprofitable days, there was a significant difference in coffee sold (p-value = 0.0303483, t=-2.2376848, df=44.0449661), and in wraps sold (p-value = 0.0131412, t=-2.6638622, df=25.7654393). However, the sales of muffins and cookies did not significantly differ between profitable and nonprofitable days (p-value = 0.1108585, t=-1.6383443, df=32.9745634), at a level of significance of 5%.

(b)

Create a graph that displays the distribution of coffee sales by day. Make sure the graph has the days ordered properly, as we would expect to see them on a calendar.

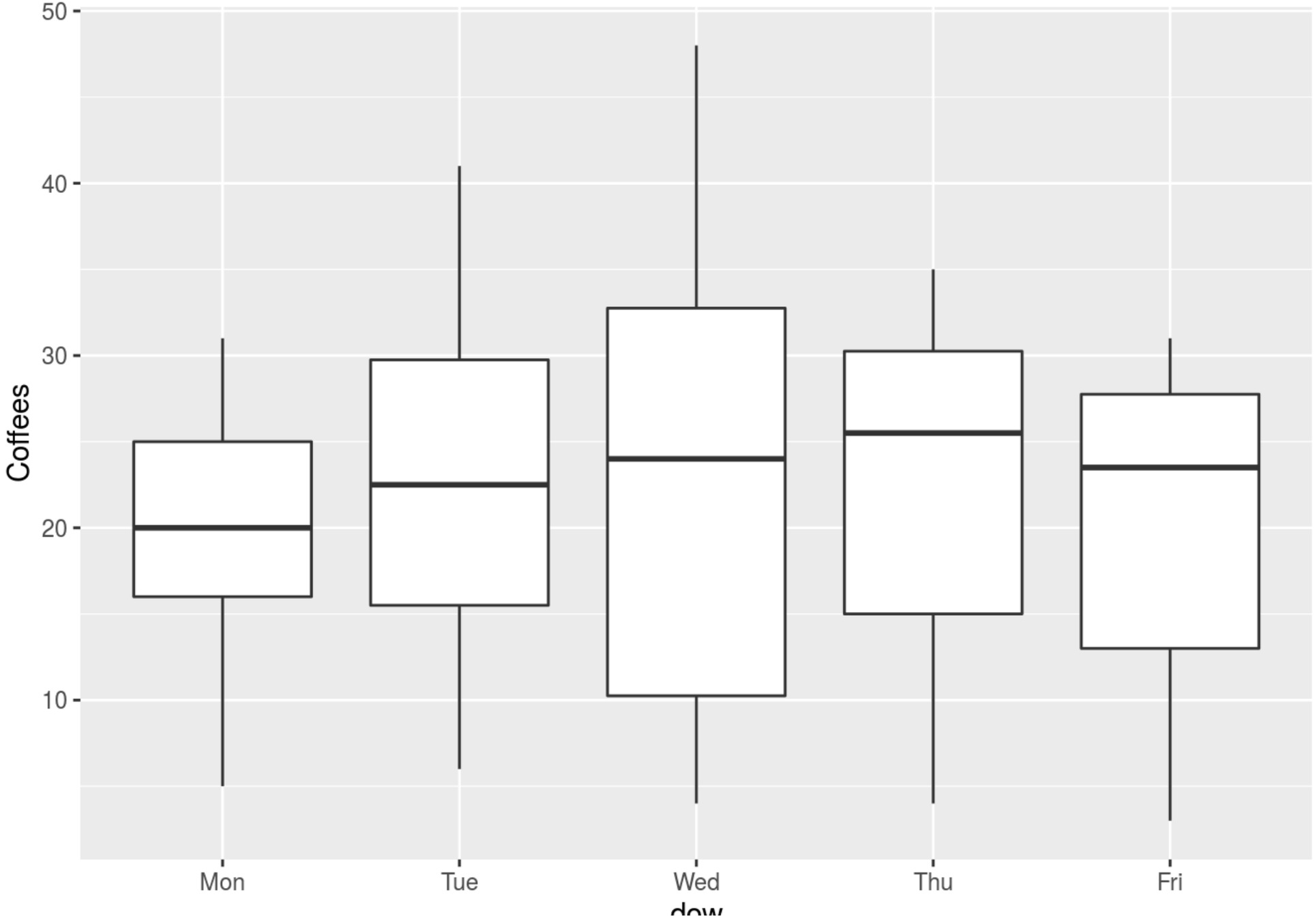
```
cafe_data <- rename(cafe_data, dow_code = "Day Code", dow = "Day
of Week")

cafe_data$dow <- factor(cafe_data$dow, levels= c("Mon", "Tue",
"Wed", "Thu", "Fri"))

cafe_data[order(cafe_data$dow), ]
```

```
## # A tibble: 48 x 24
##        t Date                dow_code dow   `Bread Sand Sol…
`Bread Sand Was…
##    <dbl> <dttm>                 <dbl> <fct>            <dbl>
<dbl>
##  1     5 2010-01-25 00:00:00        1 Mon                3
0
##  2    10 2010-02-01 00:00:00        1 Mon                2
6
##  3    15 2010-02-08 00:00:00        1 Mon                3
5
##  4    20 2010-02-15 00:00:00        1 Mon                3
3
##  5    25 2010-02-22 00:00:00        1 Mon                6
0
##  6    30 2010-03-01 00:00:00        1 Mon                3
2
##  7    35 2010-03-15 00:00:00        1 Mon                6
0
```

```
##  8    40 2010-03-22 00:00:00         1 Mon                      4
2
##  9    45 2010-03-29 00:00:00         1 Mon                      2
4
## 10     1 2010-01-19 00:00:00         2 Tue                      5
3
## # … with 38 more rows, and 18 more variables: wraps_sold <dbl
>, `Wraps
## #   Waste` <dbl>, muffins_sold <dbl>, `Muffins Waste` <dbl>,
## #   cookies_sold <dbl>, `Cookies Waste` <dbl>, `Fruit Cup Sol
d` <dbl>, `Fruit
## #   Cup Waste` <dbl>, Chips <dbl>, Juices <dbl>, Sodas <dbl>,
Coffees <dbl>,
## #   `Total Soda and Coffee` <dbl>, Sales <dbl>, `Max Daily Te
mperature
## #   (F)` <dbl>, `Total Items Wasted` <dbl>, profit <dbl>, muf
f_cook <dbl>
```

```
ggplot(data = cafe_data) +
  geom_boxplot(aes(x=dow, y=Coffees))
```

```
## Warning: Removed 1 rows containing non-finite values (stat_bo
xplot).
```


50
40
30
20
10
Coffees
Mon
Tue
Wed
Thu
Fri
dow

dow