SDS Project Check In 2 - MetroBike

Uploading the packages and data we need

```
#Download the data and store it
  BikeData <- read_csv("Austin_MetroBike_Trips_20240228.csv")
```

Creating our new variables and cleaning the dataset:

```
#Edit the dataset to add in the variables we want to explore - season, time, and day
of the week

#Add Seasons
BikeData <- BikeData |> mutate(Season = case_when(
  Month %in% c(12,1,2) ~ "Winter",
  Month %in% c(3,4,5) ~ "Spring",
  Month %in% c(6,7,8) ~ "Summer",
  Month %in% c(9,10,11) ~ "Fall"
)) |>

#Edit the time to take away the colons, using this source:
  #https://www.statology.org/str_remove-in-r/
  mutate(Time = BikeData$`Checkout Time` |>
  str_remove(":") |>
  str_remove(":")) |>

# Convert Checkout.Date values to days of the week, using this source:
  #https://www.geeksforgeeks.org/convert-date-to-day-of-week-in-r/
  mutate(Weekday = ifelse(weekdays(as.Date(`Checkout Date`)) %in% c("Saturday", "Sund
ay"), "Weekend", "Weekday"))

summary(BikeData)
```

```
##     Trip ID         Membership or Pass Type  Bicycle ID
##  Min.   :20877893   Length:35320           Length:35320
##  1st Qu.:22343674   Class :character       Class :character
##  Median :23854387   Mode  :character       Mode  :character
##  Mean   :24301953
##  3rd Qu.:25580551
##   Bike Type         Checkout Datetime  Checkout Date      Checkout Time
##  Length:35320       Length:35320       Length:35320       Length:35320
##  Class :character   Class :character   Class :character   Class1:hms
##  Mode  :character   Mode  :character   Mode  :character   Class2:difftime
##                                                           Mode  :numeric
##
##  Checkout Kiosk ID Checkout Kiosk     Return Kiosk ID    Return Kiosk
##  Min.   :2494      Length:35320       Length:35320       Length:35320
##  1st Qu.:2566      Class :character   Class :character   Class :character
##  Median :2707      Mode  :character   Mode  :character   Mode  :character
##  Mean   :3121
##  3rd Qu.:3687
##  Trip Duration Minutes     Month            Year         Season
##  Min.   :    2.00      Min.   : 1.000   Min.   :2019   Length:35320
##  1st Qu.:   16.00      1st Qu.: 4.000   1st Qu.:2020   Class :character
##  Median :   30.00      Median : 7.000   Median :2021   Mode  :character
##  Mean   :   64.67      Mean   : 6.829   Mean   :2021
##  3rd Qu.:   53.00      3rd Qu.:10.000   3rd Qu.:2021
##      Time              Weekday
##  Length:35320       Length:35320
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##  [ reached getOption("max.print") -- omitted 1 row ]
```

From the summary, we can see that there aren't any NA values. However, the maximum value for the Trip Duration Minutes is very high compared to the 3rd quartile value, suggesting that there will be outliers and high values we have to take into account when we analyze the data.

```
#Change the var type of time to numerical data
BikeData$Time <- as.numeric(BikeData$Time)
```

Since we won't be working with every column in this dataset, we can create a seperate dataset to manipulate with only the variables we want to explore.

```
#Select the columns that we want to work with
MetroBike <- BikeData |> select(
  TripId = `Trip ID`, BikeId = `Bicycle ID`,
  Duration = `Trip Duration Minutes`,
  Weekday, Time,
  BikeType = `Bike Type`,
  Season, Year)

MetroBike
```

```
## # A tibble: 35,320 × 8
##      TripId BikeId Duration Weekday   Time BikeType Season  Year
##       <dbl> <chr>     <dbl> <chr>    <dbl> <chr>    <chr>  <dbl>
##  1 29503796 288          43 Weekend 151652 classic  Spring  2023
##  2 29529289 21653        14 Weekday 214359 electric Spring  2023
##  3 29538721 21903         5 Weekday 191806 electric Spring  2023
##  4 29537317 19247        18 Weekday 173016 electric Spring  2023
##  5 29537279 19274        21 Weekday 172756 electric Spring  2023
##  6 29542840 19214        14 Weekday 105625 electric Spring  2023
##  7 29532385 19943        77 Weekday 111909 electric Spring  2023
##  8 29532416 19326        73 Weekday 112324 electric Spring  2023
##  9 29533449 19177         7 Weekday 125505 electric Spring  2023
## 10 29533451 16337        76 Weekday 125515 electric Spring  2023
## # ℹ 35,310 more rows
```

Question 1

A quick description of the dataset(s), reporting the number of rows and columns.

```
#Find the dimensions of our dataset
dim(MetroBike)
```

```
## [1] 35320     8
```

```
MetroBike |> mutate_all(is.na) |> summarize_all(sum)
```

```
## # A tibble: 1 × 8
##   TripId BikeId Duration Weekday  Time BikeType Season  Year
##    <int>  <int>    <int>   <int> <int>    <int>  <int> <int>
## 1      0      0        0       0     0        0      0     0
```

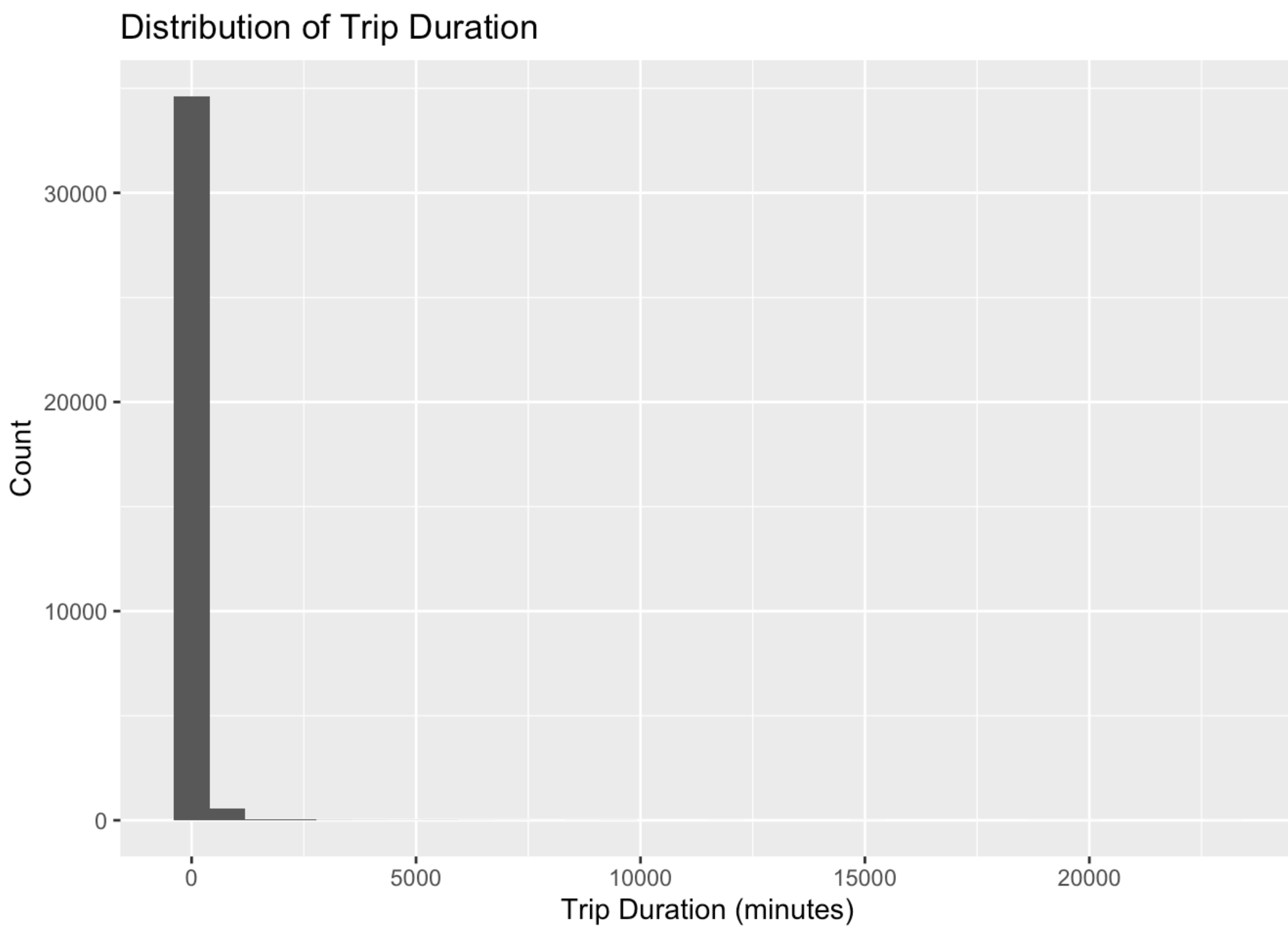
After we successfully added the and selected the variables we need, we have 8 columns with 35320 rows. Also, as we can see, there are 0 NAs in our dataset.

Question 2

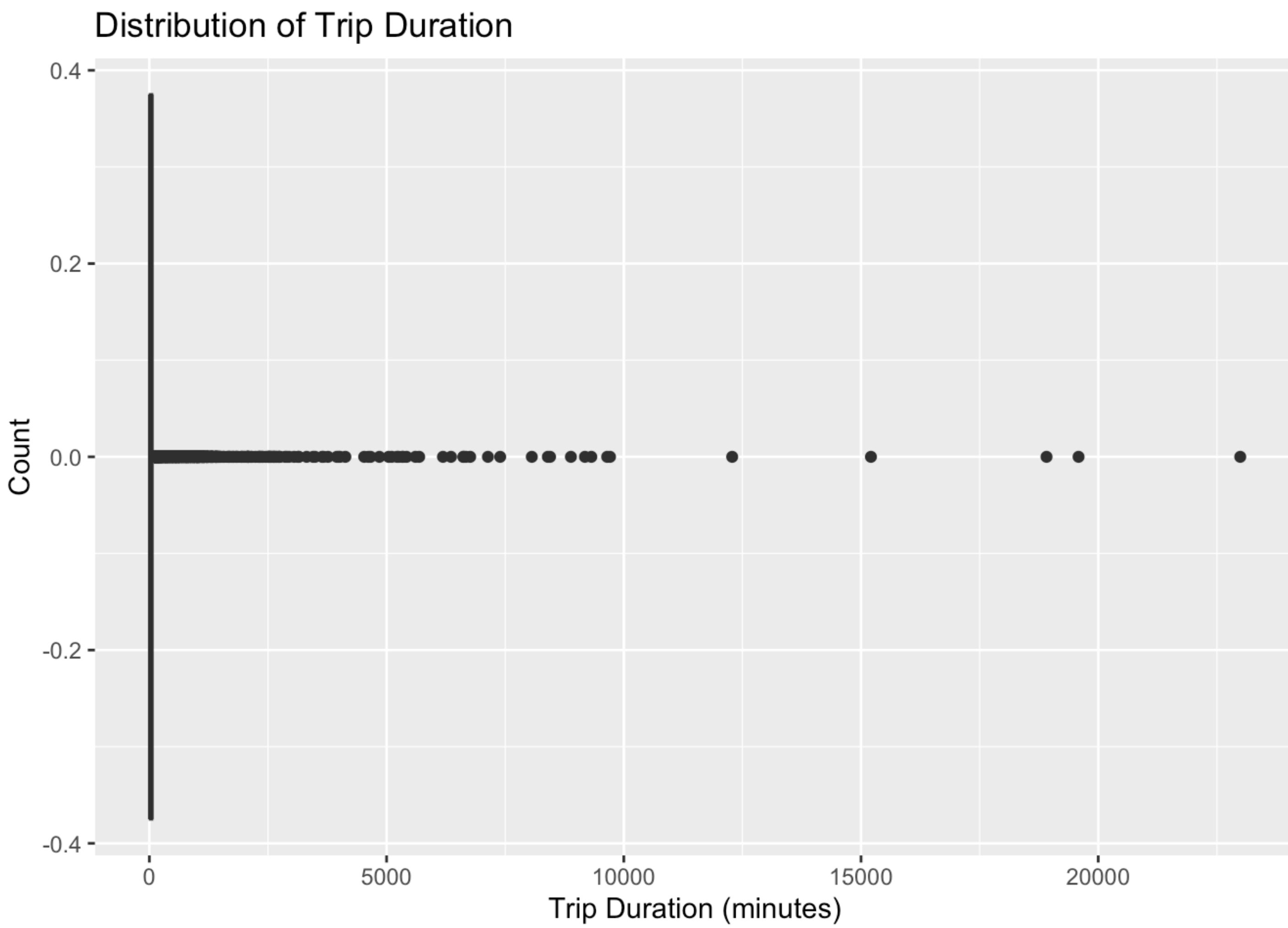
Explore 1 numeric variable in your dataset: include a plot and summary statistics.

```
#Visualize a numeric variable: Trip Duration

#Raw Data
MetroBike |>
  ggplot() +
  geom_histogram(aes(x=Duration)) +
  labs(
    title = "Distribution of Trip Duration",
    x = "Trip Duration (minutes)",
    y="Count"
  )
```

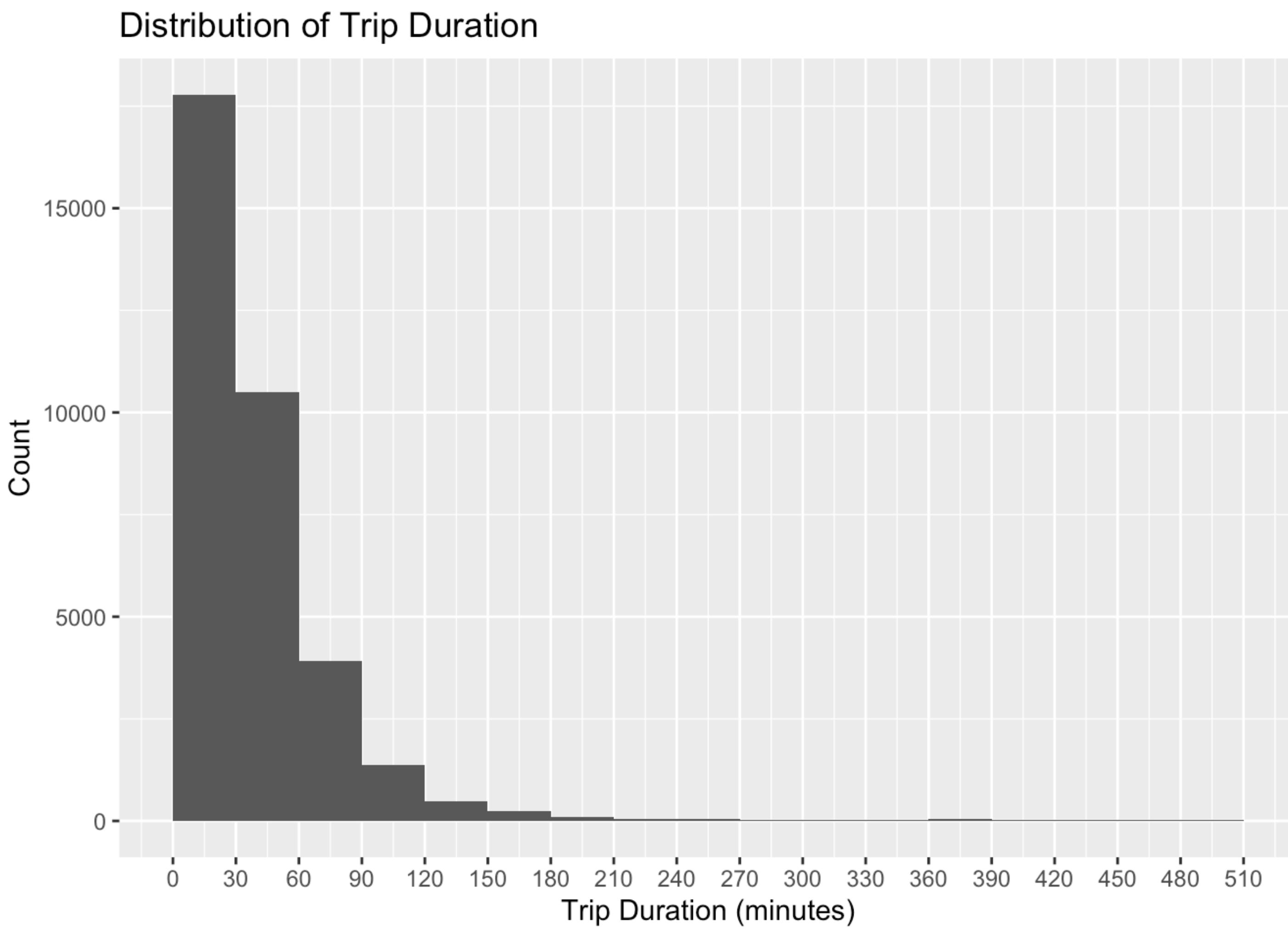


```
#Raw Data with boxplot to show outliers
MetroBike |>
  ggplot() +
  geom_boxplot(aes(x=Duration)) +
  labs(
    title = "Distribution of Trip Duration",
    x = "Trip Duration (minutes)",
    y = "Count"
  )
```

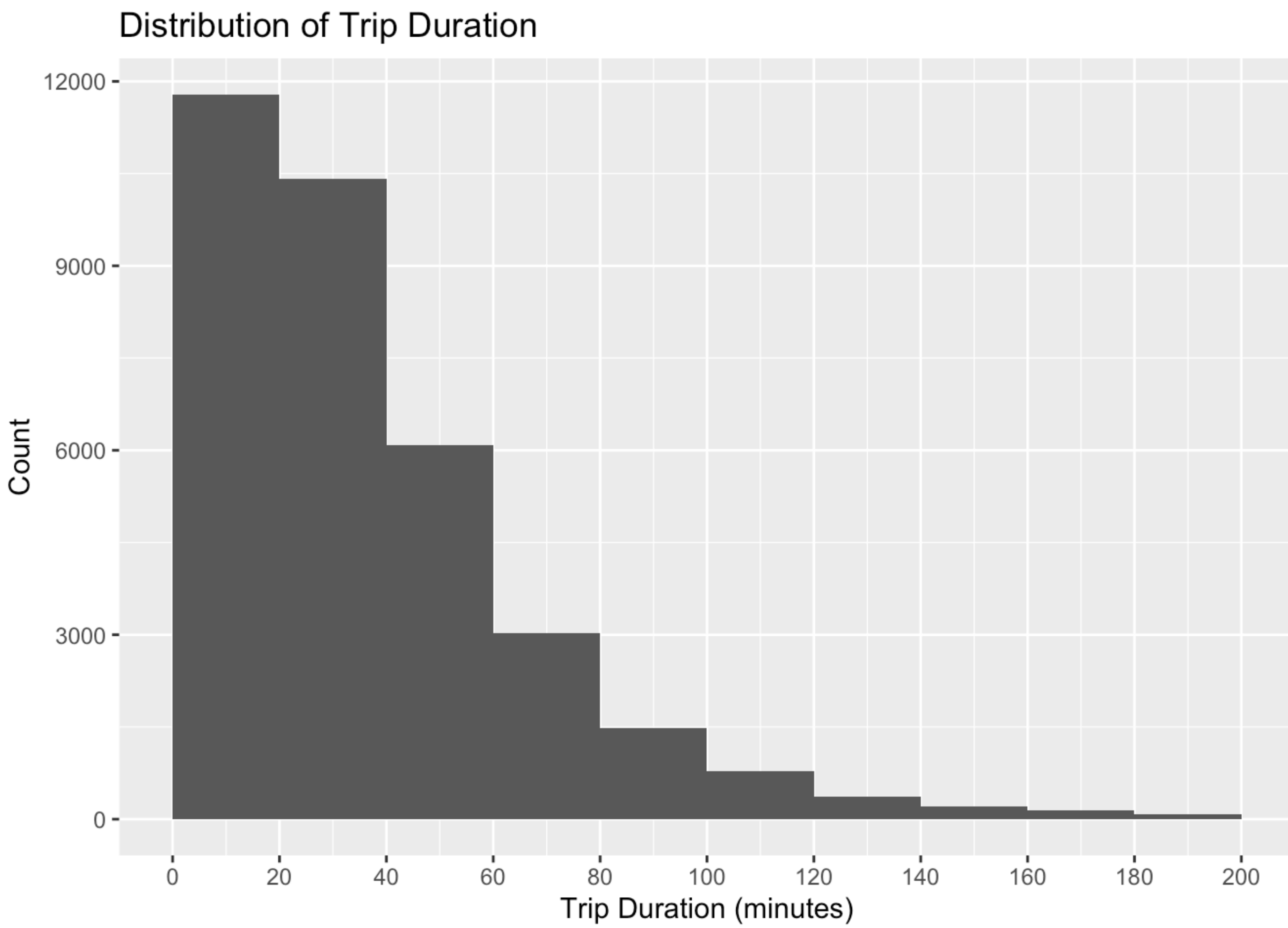


As we can see from both visualizations, there are *many* outliers with very high values in this variable. Without removing the outliers, there is no way to easily visualize the distribution of trip duration for the general population.

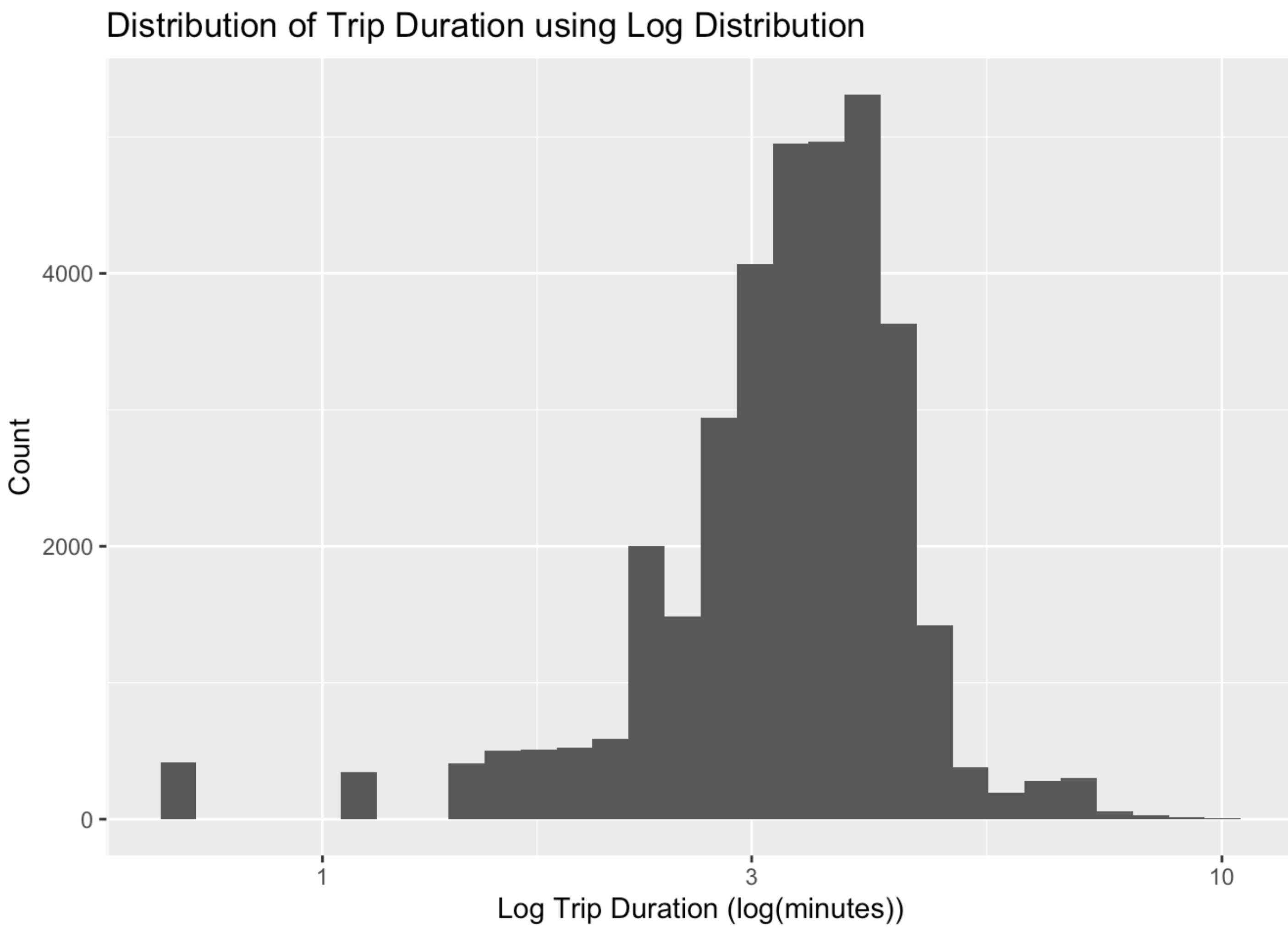
```
#Filtered data:
MetroBike |>
  filter(Duration < 500) |>
  ggplot() +
  geom_histogram(aes(x=Duration), binwidth = 30, center=15) +
  scale_x_continuous(limits = c(0, 510), breaks = seq(0, 510, 30)) +
  labs(
    title = "Distribution of Trip Duration",
    x = "Trip Duration (minutes)",
    y="Count"
  )
```



```
#Filtered data:
MetroBike |>
  filter(Duration < 200) |>
  ggplot() +
  geom_histogram(aes(x=Duration), binwidth = 20, center=10) +
  scale_x_continuous(limits = c(0, 200), breaks = seq(0, 200, 20)) +
  labs(
    title = "Distribution of Trip Duration",
    x = "Trip Duration (minutes)",
    y="Count"
  )
```



```
#Log data with all of our data instead of filtering to exclude data
MetroBike |>
  ggplot() +
  geom_histogram(aes(x=log(Duration))) +
  labs(
    title = "Distribution of Trip Duration using Log Distribution",
    x = "Log Trip Duration (log(minutes))",
    y="Count"
  )+
  scale_x_continuous(trans = "log10")
```




```
#Summary Statistics:
MetroBike |> summarize(Mean=mean(Duration), Median = median(Duration), IQR = IQR(Dura
tion), Max=max(Duration))
```

```
## # A tibble: 1 × 4
##    Mean Median   IQR   Max
##   <dbl>  <dbl> <dbl> <dbl>
## 1  64.7     30    37 22993
```

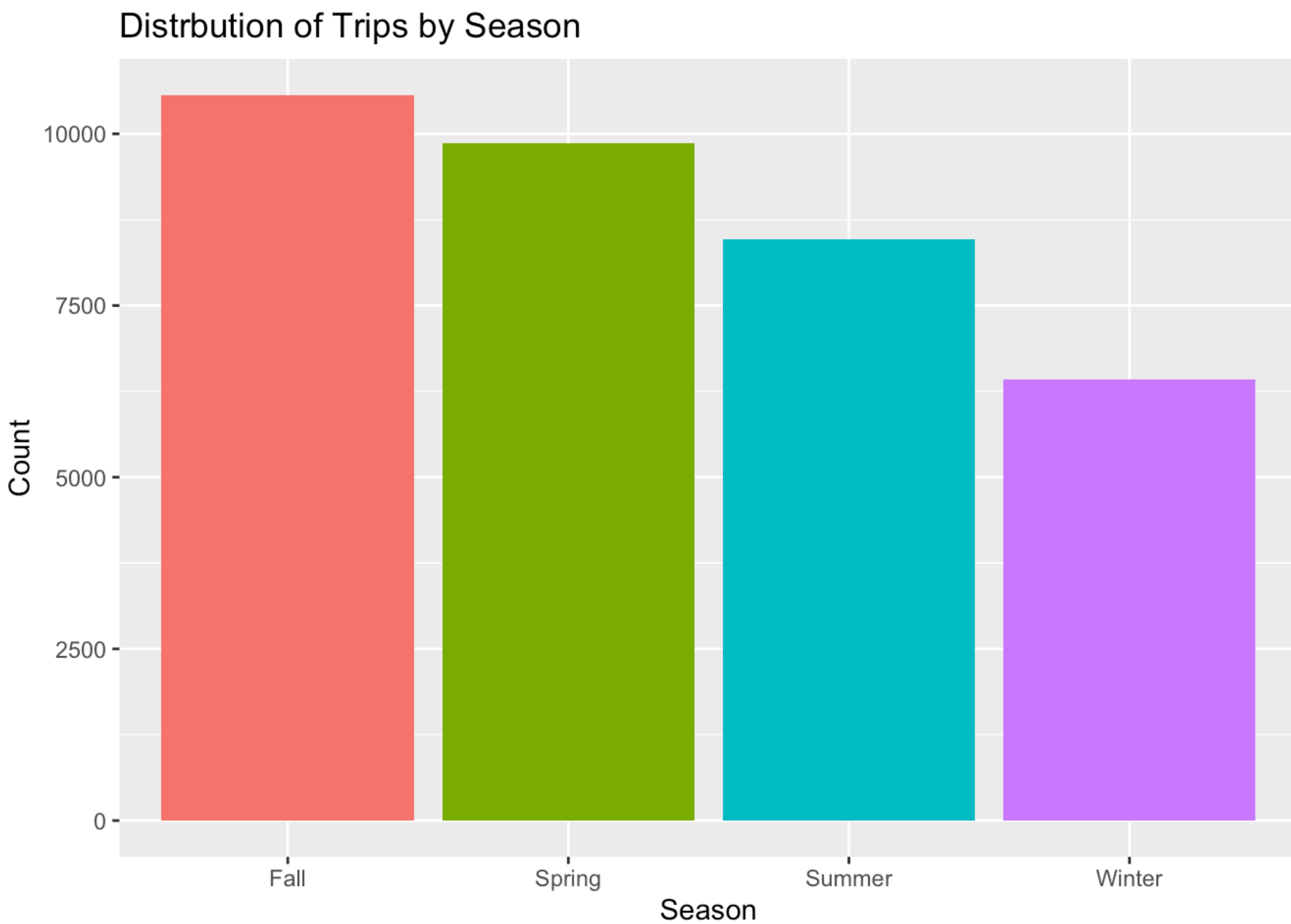
We tried out many different distributions to visualize our data. In the first two graphs, we just graphed the raw data without any additional settings, which clearly showed the many high outliers in our data set. Then, in our next two graphs, we tried to correct the visualization by filtering the trip duration to a specific range of minutes, producing a much better looking graph. However, this method does exclude a lot of important data, so lastly we tried looking at a histogram of the log data, which gives almost a bell-shaped distribution.

From our graphs, we can see the data is heavily skewed right - most of the trips taken in Austin were less than 1 hr long However, as we have seen from the previous graphs, there are many high outliers as well, up to 22993 minutes - which is a little less than 16 days! From the summary statistics, we can see that the median trip duration is 30 minutes, with an IQR of 37 minutes. (Also from the summary statistics, we can see the extreme right skew, as the mean of 64.7 minutes is more than double the median of 30 minutes.)

Question 3

Explore 1 categorical variable in your dataset: include a plot and summary statistics.

```
#Explore a cateogrical variable: season
MetroBike |>
  ggplot() +
  geom_bar(aes(x=Season, fill = Season)) +
  labs(
    title = "Distrbution of Trips by Season",
    x = "Season",
    y = "Count"
  ) +
  guides(fill="none")
```




```
MetroBike |> group_by(Season) |> count()
```

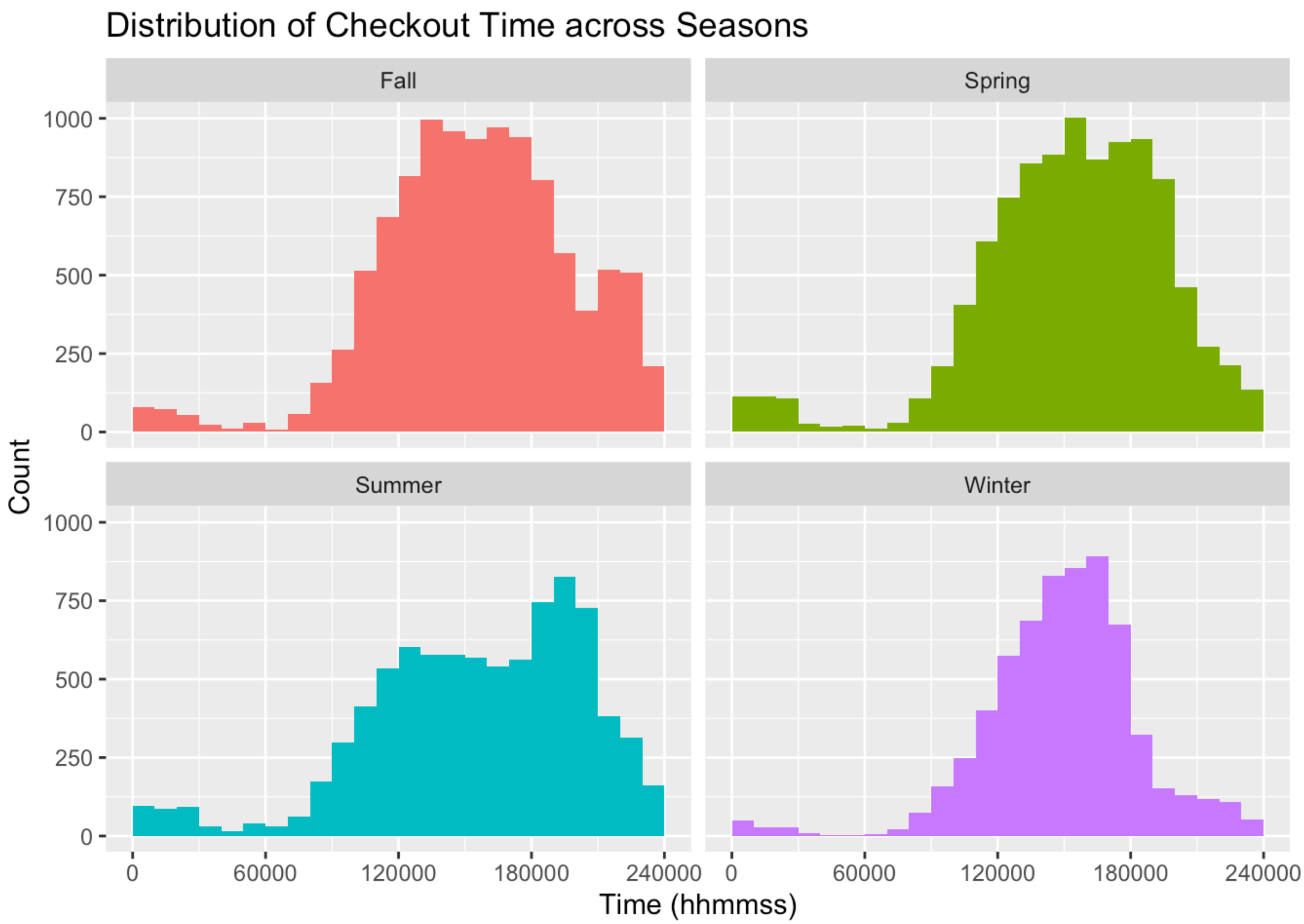
```
## # A tibble: 4 × 2
## # Groups:   Season [4]
##   Season     n
##   <chr>  <int>
## 1 Fall   10565
## 2 Spring  9869
## 3 Summer  8461
## 4 Winter  6425
```

From the data, we can see that fall is the most popular season for tourists to come use the metrobike - there was a total of *10565* Single Trips bought - while winter was the least popular season. There were 9869 bought in spring, 8461 bought in summer, and 6425 bought in winter.

Relationship between 2 variables

Research Question: Is there a relationship between checkout time and the season in which the trip takes place?

```
#Show the relationship between Season and the time bikes were checked out.
MetroBike |>
  ggplot() +
  geom_histogram(aes(x = Time, fill = Season), binwidth = 10000, center = 5000) +
  facet_wrap(~Season) +
  scale_x_continuous(limits = c(0, 240000), breaks = seq(0, 240000, 60000)) +
  guides(fill = "none") +
  labs(
    title = "Distribution of Checkout Time across Seasons",
    x = "Time (hhmmss)",
    y = "Count"
  )
```



From the graph, we can see how during the winter months, there is a steep decline later during the day after around 6pm, as it would be too cold. In summer, there was an increase in the number of users at around 8-10pm, which is when the temperature is cooler. In spring, the use peaked at around 5 to 6pm where more people used a MetroBike, as the weather is generally nicer then. The dip in the left middle side of the graph also makes sense - while it isn't uncommon for people to be out late until 2-3 am, not many people will be out from the 3-7am time.

Question 4

Let us know if there is anything you have questions about to manipulate your dataset(s)!

Is there a better way for us to represent time, especially on a graph? For now, we just have our time as hhmmss, so it can be represented on a graph as a numerical variable, but it is not very intuitive.