Explore_bikeshare_data

September 24, 2023

0.0.1 Explore Bike Share Data

For this project, your goal is to ask and answer three questions about the available bikeshare data from Washington, Chicago, and New York. This notebook can be submitted directly through the workspace when you are confident in your results.

You will be graded against the project Rubric by a mentor after you have submitted. To get you started, you can use the template below, but feel free to be creative in your solutions!

```
In [1]: ny = read.csv('new_york_city.csv')
        wash = read.csv('washington.csv')
        chi = read.csv('chicago.csv')
```

```
In [2]: head(ny)
```

X	Start.Time	End.Time	Trip.Duration	Start.Station	End.Statio
5688089	2017-06-11 14:55:05	2017-06-11 15:08:21	795	Suffolk St & Stanton St	W Broadw
4096714	2017-05-11 15:30:11	2017-05-11 15:41:43	692	Lexington Ave & E 63 St	1 Ave & E 7
2173887	2017-03-29 13:26:26	2017-03-29 13:48:31	1325	1 Pl & Clinton St	Henry St &
3945638	2017-05-08 19:47:18	2017-05-08 19:59:01	703	Barrow St & Hudson St	W 20 St & 8
6208972	2017-06-21 07:49:16	2017-06-21 07:54:46	329	1 Ave & E 44 St	E 53 St & 3
1285652	2017-02-22 18:55:24	2017-02-22 19:12:03	998	State St & Smith St	Bond St &

```
In [3]: head(wash)
```

X	Start.Time	End.Time	Trip.Duration	Start.Station
1621326	2017-06-21 08:36:34	2017-06-21 08:44:43	489.066	14th & Belmont St NW
482740	2017-03-11 10:40:00	2017-03-11 10:46:00	402.549	Yuma St & Tenley Circle NW
1330037	2017-05-30 01:02:59	2017-05-30 01:13:37	637.251	17th St & Massachusetts Ave NW
665458	2017-04-02 07:48:35	2017-04-02 08:19:03	1827.341	Constitution Ave & 2nd St NW/DOL
1481135	2017-06-10 08:36:28	2017-06-10 09:02:17	1549.427	Henry Bacon Dr & Lincoln Memorial
1148202	2017-05-14 07:18:18	2017-05-14 07:24:56	398.000	1st & K St SE

```
In [4]: head(chi)
```

X	Start.Time	End.Time	Trip.Duration	Start.Station	End
1423854	2017-06-23 15:09:32	2017-06-23 15:14:53	321	Wood St & Hubbard St	Da
955915	2017-05-25 18:19:03	2017-05-25 18:45:53	1610	Theater on the Lake	She
9031	2017-01-04 08:27:49	2017-01-04 08:34:45	416	May St & Taylor St	Wo
304487	2017-03-06 13:49:38	2017-03-06 13:55:28	350	Christiana Ave & Lawrence Ave	St.
45207	2017-01-17 14:53:07	2017-01-17 15:02:01	534	Clark St & Randolph St	Des
1473887	2017-06-26 09:01:20	2017-06-26 09:11:06	586	Clinton St & Washington Blvd	Ca

0.0.2 Question 1

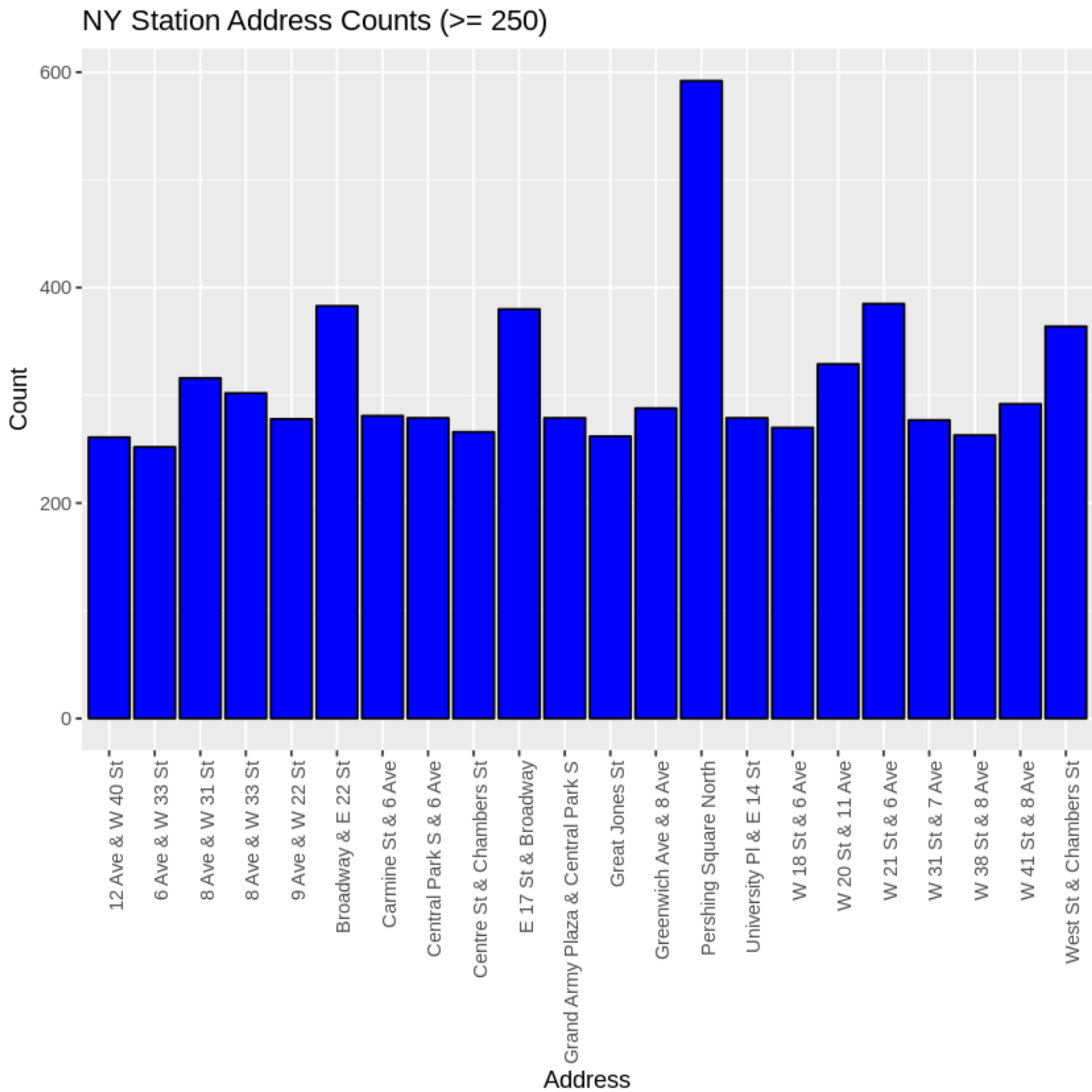
What is the most common start station?

```
In [5]: #created a function to find the max count address
        findMaxCountAddress <- function(data, columnName) {
          # Find the maximum count
          max_count <- max(table(data[[columnName]]))

          # Find the index of the maximum count
          max_count_index <- which.max(table(data[[columnName]]))

          # Find the corresponding address
          address <- names(table(data[[columnName]]))[max_count_index]

          return(address)
        }

        # Finding NY max count address
        addressny <- findMaxCountAddress(ny, "Start.Station")
        addressny
```

'Pershing Square North'

```
In [6]: # install package ggplot2 to plot our histogram
        library(ggplot2)

        #function to create count df
        createCountsDataFrame <- function(data, columnName) {
          # Create a table of counts for each unique value in the specified column
          counts <- table(data[[columnName]])

          # Convert the table to a data frame
          counts_df <- data.frame(
            Address = as.character(names(counts)),
            Count = as.numeric(counts),
            stringsAsFactors = FALSE
          )

          return(counts_df)
        }

        nycounts_df <- createCountsDataFrame(ny, "Start.Station")

        # Filter addresses with counts >= 250
        filtered_address_counts_df <- nycounts_df[nycounts_df$Count >= 250, ]

        # Create a bar plot using ggplot2
        ggplot(filtered_address_counts_df, aes(x = Address, y = Count)) +
```

```
geom_bar(stat = "identity", fill = "blue", color = "black") +
labs(title = "NY Station Address Counts (>= 250)",
     x = "Address",
     y = "Count") +
theme(axis.text.x = element_text(angle = 90, hjust = 1))  # Rotate x-axis labels for r
```

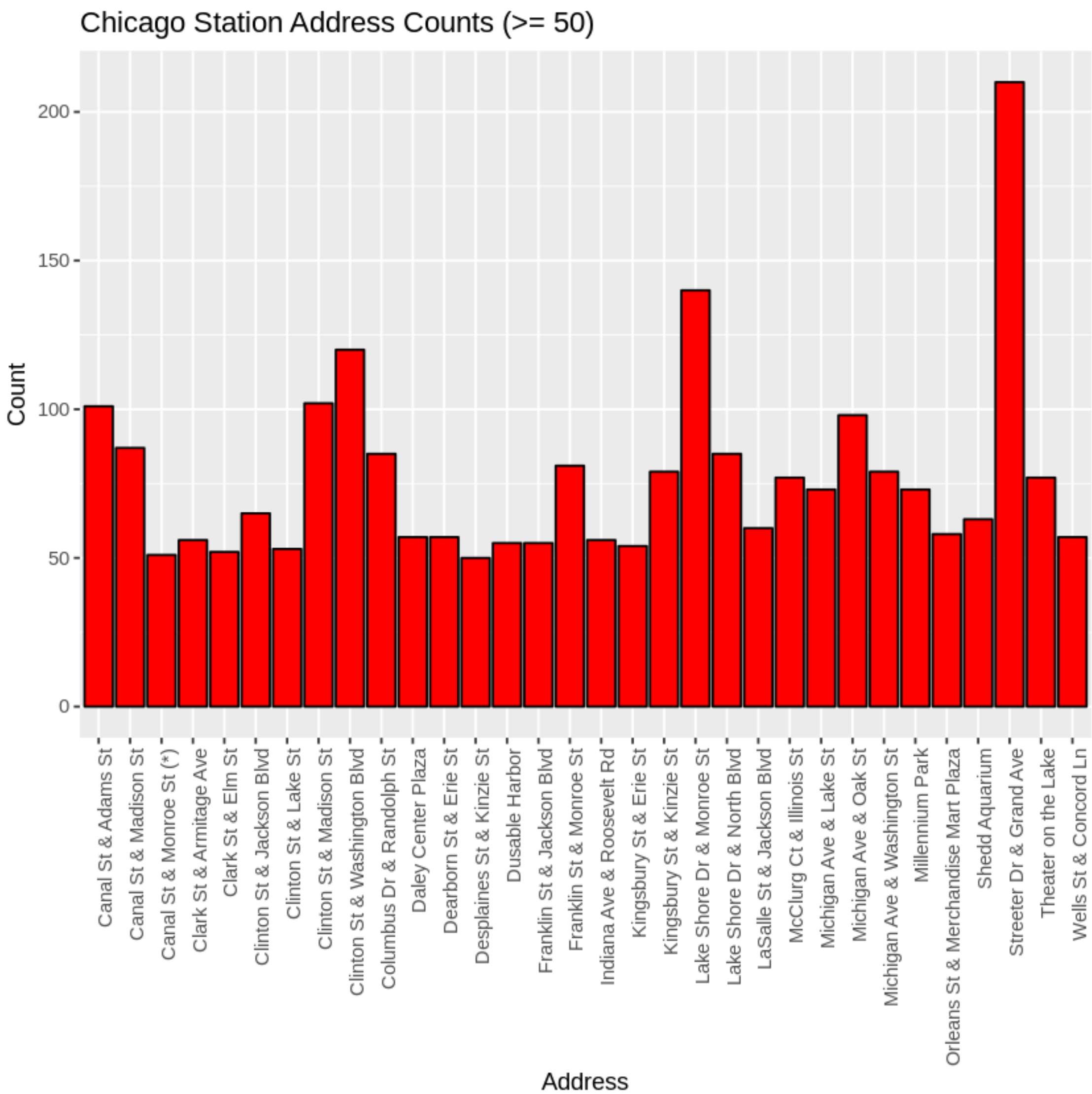



```
In [7]: # Finding Chicago max count address
        addresschi <- findMaxCountAddress(chi, "Start.Station")
        addresschi
```

'Streeter Dr & Grand Ave'

```
In [8]: chicounts_df <- createCountsDataFrame(chi, "Start.Station")
```

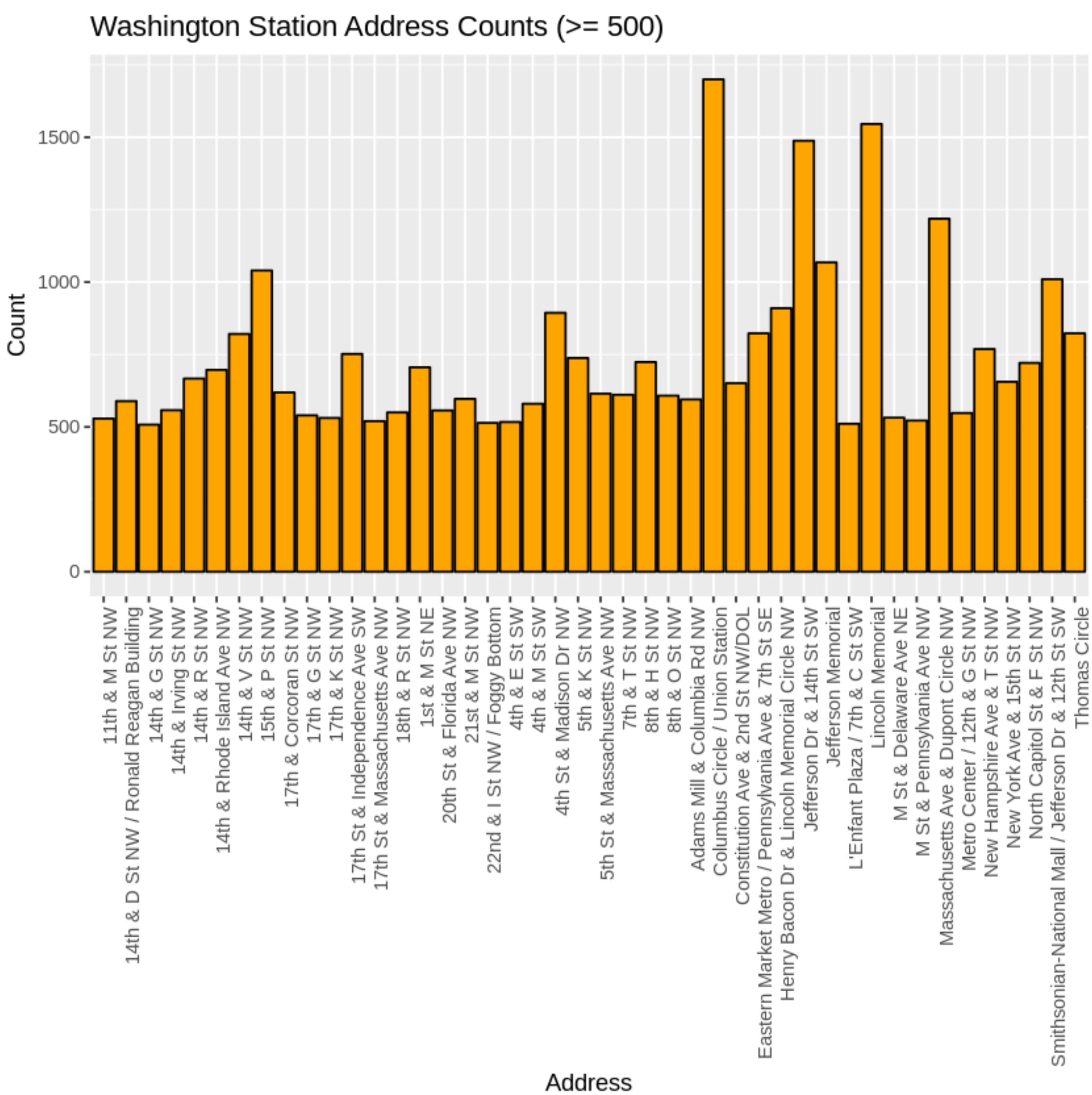
```
# Filter addresses with counts >= 50
filtered_chicounts_df <- chicounts_df[chicounts_df$Count >= 50, ]

# Create a bar plot using ggplot2
ggplot(filtered_chicounts_df, aes(x = Address, y = Count)) +
  geom_bar(stat = "identity", fill = "red", color = "black") +
  labs(title = "Chicago Station Address Counts (>= 50)",
       x = "Address",
       y = "Count") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))  # Rotate x-axis labels for r
```



In [9]:
```
# Finding Washington max count address
addresswash <- findMaxCountAddress(wash, "Start.Station")
addresswash
```

'Columbus Circle / Union Station'

```
In [10]: washcounts_df <- createCountsDataFrame(wash, "Start.Station")

         # Filter addresses with counts >= 500
         filtered_washcounts_df <- washcounts_df[washcounts_df$Count >= 500, ]

         # Create a bar plot using ggplot2
         ggplot(filtered_washcounts_df, aes(x = Address, y = Count)) +
           geom_bar(stat = "identity", fill = "orange", color = "black") +
           labs(title = "Washington Station Address Counts (>= 500)",
                x = "Address",
                y = "Count") +
           theme(axis.text.x = element_text(angle = 90, hjust = 1))  # Rotate x-axis labels for
```



```
In [13]: #to find summary stats on the new count df
         summary(nycounts_df)
```

```
   Address              Count       
 Length:636         Min.   :  1.00  
 Class :character   1st Qu.: 28.00  
 Mode  :character   Median : 61.00  
                    Mean   : 86.12  
                    3rd Qu.:133.25  
                    Max.   :592.00  
```

```
In [14]: summary(chicounts_df)
```

```
   Address              Count       
 Length:472         Min.   :  1.00  
 Class :character   1st Qu.:  4.00  
 Mode  :character   Median : 12.00  
                    Mean   : 18.28  
                    3rd Qu.: 25.25  
                    Max.   :210.00  
```

```
In [15]: summary(washcounts_df)
```

```
   Address              Count       
 Length:478         Min.   :   1.0  
 Class :character   1st Qu.:  26.0  
 Mode  :character   Median :  87.0  
                    Mean   : 186.3  
                    3rd Qu.: 279.0  
                    Max.   :1700.0  
```

Some interesting insights were observed with this data. While each city has a definitive most common start station the definition takes on a seperate meaning depending on the city in question.

Starting with NYC, the Pershing Square North station won out with 592 instances of use in the dataframe. This number is significant when looking at the summary statistics as it stands out as a clear outlier. It is several hundred counts away from the mean of 86 and median of 61. It is also several 100 counts away from the third quartile value of 133, which is where 75% of the values fall under. Visually the histogram further illustrates much the station's count stands out.

With Chicago we see a similar situatuon. The Streeter Dr & Grand Ave station was the most popular at a 210 count. This number is also an outlier when comparing the summary statistics. With a mean of 18 and median of 12, it falls far above these counts. The third quartile value at 25 further emphasizes this as well as the visual representation through the histogram.

Lastly, Washington had by far the most used station compared to the other cities. The Columbus Circle / Union Station won out with a huge 1700 count. This number is far beyond the mean of 186 and median of 87. It even blazes past the third quartile count of 279.

Visually a couple of other counts go past the 1000 count mark, but not enough to make the most popular station's count insignificant. This could be possibly due to the fact that our data shows Washington DC to be the city thats most popular for the product

Identifying these popular stations and verifying their significance helps to focus the company efforts on expanding the business. These insights can be used to place more bikes at these designated stations, open up more stations closer to the popular stations, etc. It should be noted that all of this analysis is based solely on the three files provided and more data is necessary to diversify these insights.

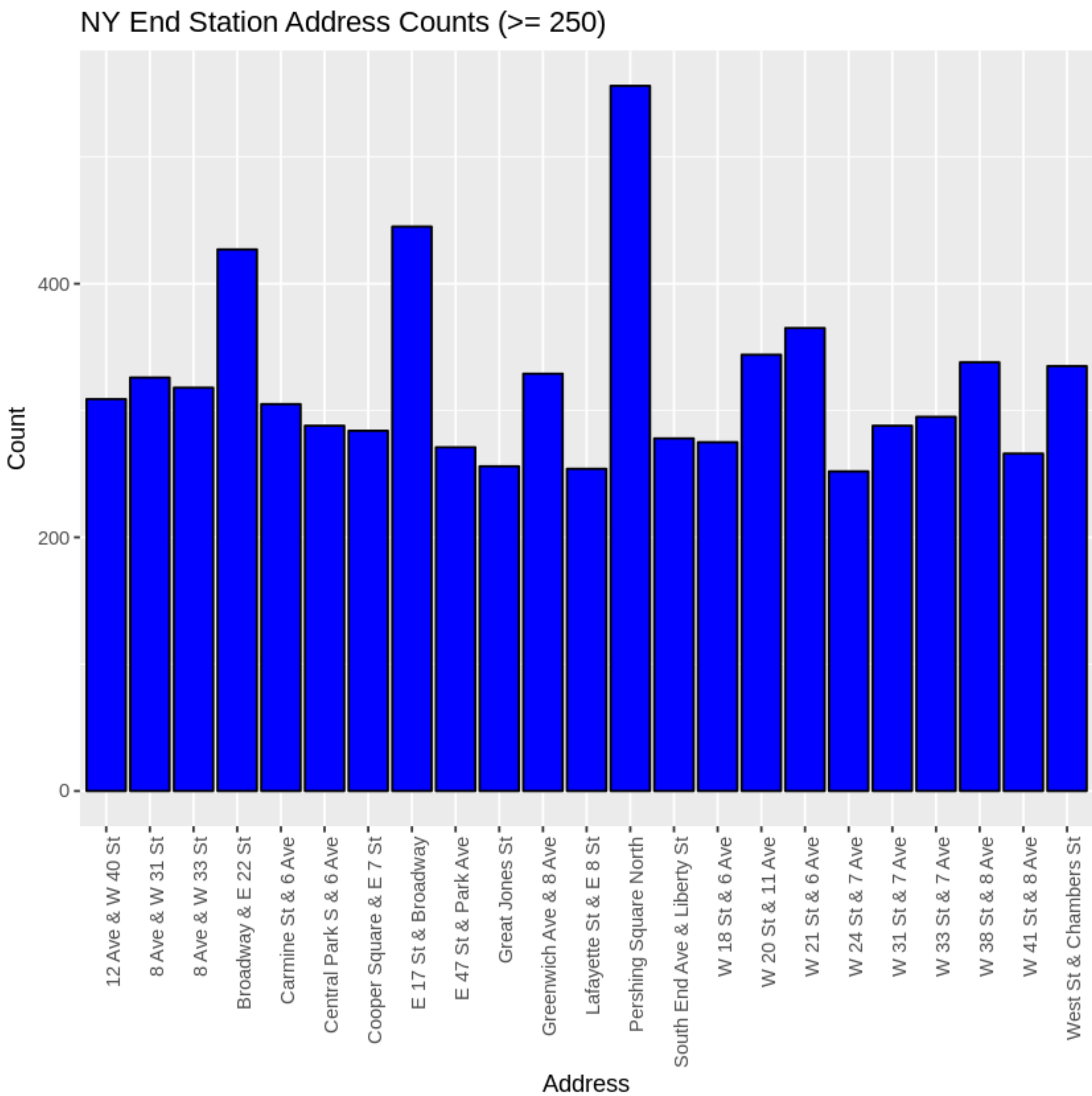
0.0.3 Question 2

What is the most common end station?

```
In [16]: #using the same function as before to find the max count end station
         ny_endaddress <- findMaxCountAddress(ny, "End.Station")
         ny_endaddress
         #curious that it's the same station as the start
```

'Pershing Square North'

```
In [25]: #using function from previous question to create a count df
         nyend_df <- createCountsDataFrame(ny, "End.Station")

         # Filter addresses with counts >= 250
         filtered_nyend_df <- nyend_df[nyend_df$Count >= 250, ]

         # Create a bar plot using ggplot2
         ggplot(filtered_nyend_df, aes(x = Address, y = Count)) +
           geom_bar(stat = "identity", fill = "blue", color = "black") +
           labs(title = "NY End Station Address Counts (>= 250)",
                x = "Address",
                y = "Count") +
           theme(axis.text.x = element_text(angle = 90, hjust = 1))  # Rotate x-axis labels for
```

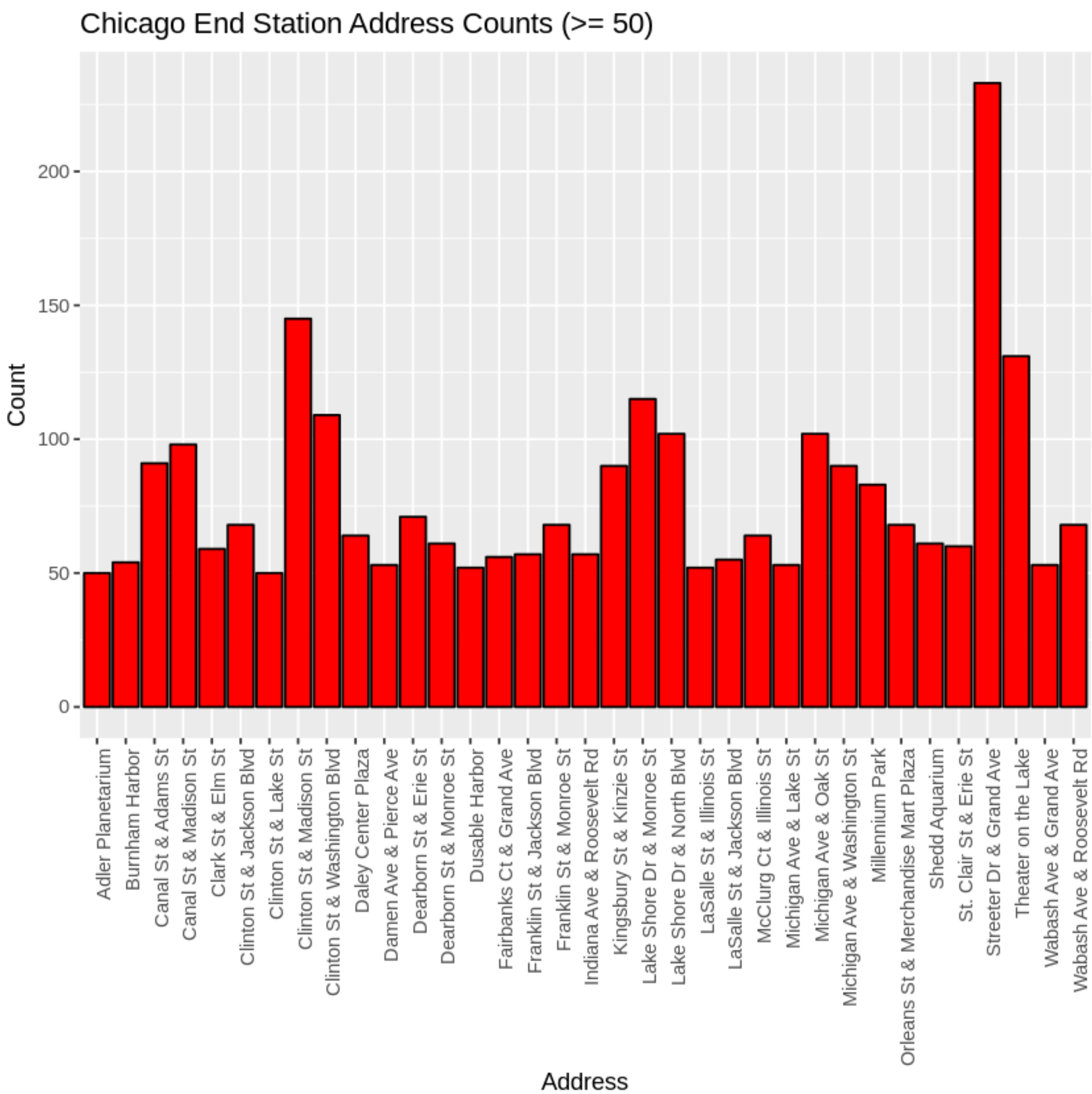



```
In [26]: chi_endaddress <- findMaxCountAddress(chi, "End.Station")
         chi_endaddress
```

'Streeter Dr & Grand Ave'

```
In [32]: chiend_df <- createCountsDataFrame(chi, "End.Station")

         # Filter addresses with counts >= 50
         filtered_chiend_df <- chiend_df[chiend_df$Count >= 50, ]

         # Create a bar plot using ggplot2
         ggplot(filtered_chiend_df, aes(x = Address, y = Count)) +
           geom_bar(stat = "identity", fill = "red", color = "black") +
```

```
labs(title = "Chicago End Station Address Counts (>= 50)",
     x = "Address",
     y = "Count") +
theme(axis.text.x = element_text(angle = 90, hjust = 1))  # Rotate x-axis labels for
```

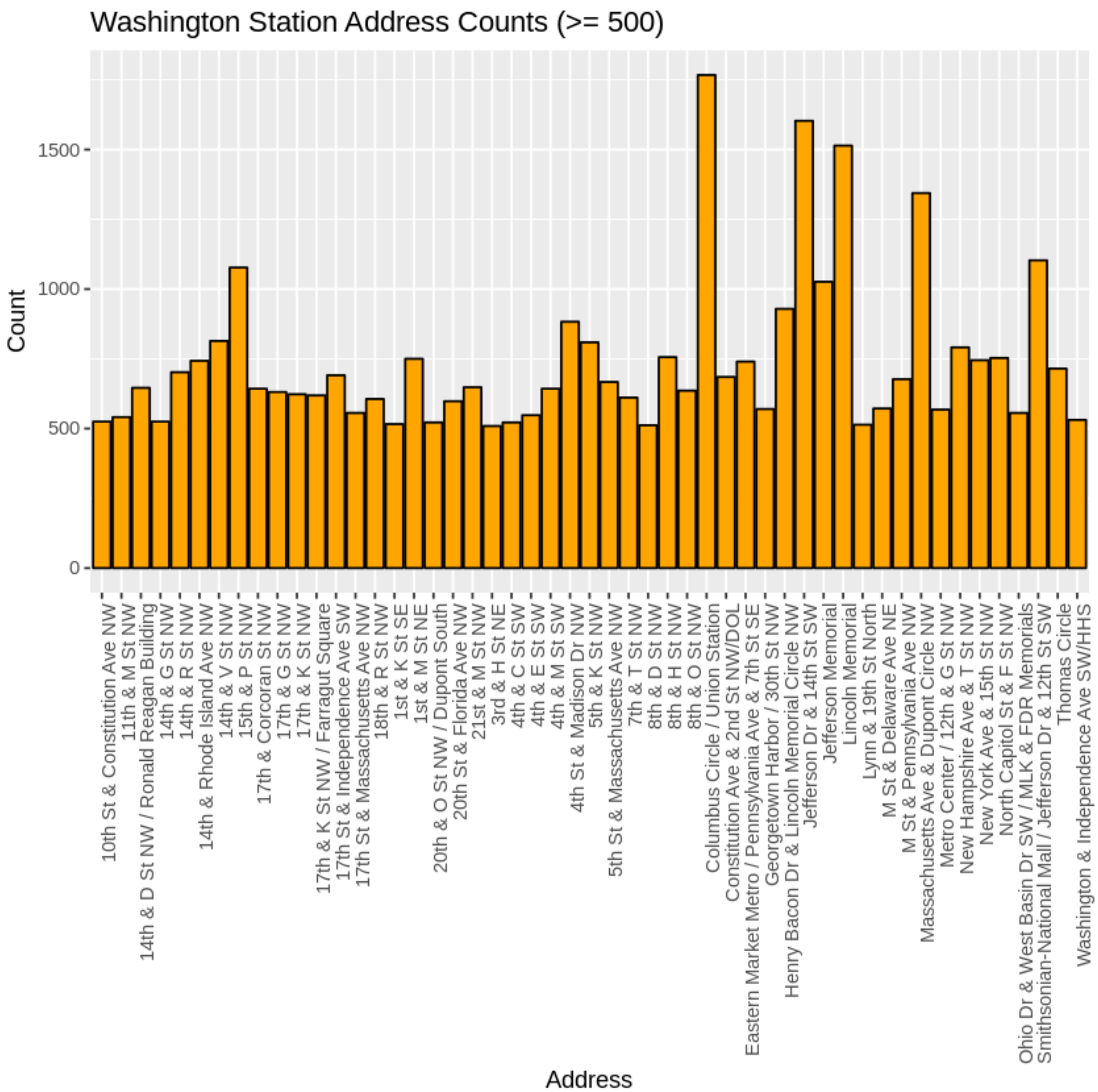



```
In [27]: wash_endaddress <- findMaxCountAddress(wash, "End.Station")
         wash_endaddress
```

'Columbus Circle / Union Station'

```
In [37]: washend_df <- createCountsDataFrame(wash, "End.Station")

         # Filter addresses with counts >= 500
```

```
filtered_washend_df <- washend_df[washend_df$Count >= 500, ]

# Create a bar plot using ggplot2
ggplot(filtered_washend_df, aes(x = Address, y = Count)) +
  geom_bar(stat = "identity", fill = "orange", color = "black") +
  labs(title = "Washington Station Address Counts (>= 500)",
       x = "Address",
       y = "Count") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))  # Rotate x-axis labels for
```




```
In [24]: summary(nyend_df)
```

```
    Address              Count
```

```
 Length:638         Min.   :  1.00  
 Class :character   1st Qu.: 28.00  
 Mode  :character   Median : 59.00  
                    Mean   : 85.85  
                    3rd Qu.:126.00  
                    Max.   :556.00  
```

```
In [38]: summary(chiend_df)
```

```
   Address              Count       
 Length:471         Min.   :  1.00  
 Class :character   1st Qu.:  4.00  
 Mode  :character   Median : 12.00  
                    Mean   : 18.32  
                    3rd Qu.: 24.00  
                    Max.   :233.00  
```

```
In [39]: summary(washend_df)
```

```
   Address              Count       
 Length:479         Min.   :   1.0  
 Class :character   1st Qu.:  23.5  
 Mode  :character   Median :  69.0  
                    Mean   : 185.9  
                    3rd Qu.: 265.0  
                    Max.   :1767.0  
```

These are some interesting insights. I won't explore the numbers as in depth as the previous questions since the numbers are similar in every aspect(count, mean, median, third quartile). The most popular end stations, Columbus Circle / Union Station, Streeter Dr & Grand Ave, and Pershing Square North are the same popular start stations. Their counts are still clear outliers visually and compared to their summary statistics making them significant.

This could mean a multitude of things as far as the business is concerned. If users are starting and ending at the same stations most of the time it could mean that the customer base is highly centered around these key stations. It could also mean the bikes are primarily used for just leisure rides, or commutes to work. Perhaps a popular attraction or bike trails are nearest these stations? All of these insights could prove valuable for advertising purposes or inventory.

It should be noted that all of this analysis is based solely on the three files provided and more data is necessary to diversify these insights. More information is needed to solidify these insights

0.0.4 Question 3

What is the average travel time for users in different cities?

In [45]:
```
#to pull up the summary statistics, including mean
nystats <- summary(ny$Trip.Duration)
nystats
```

```
  Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
  61.0     368.0     610.0     903.6    1051.0 1088634.0         1
```

In [80]:
```
chistats <- summary(chi$Trip.Duration)
chistats
```

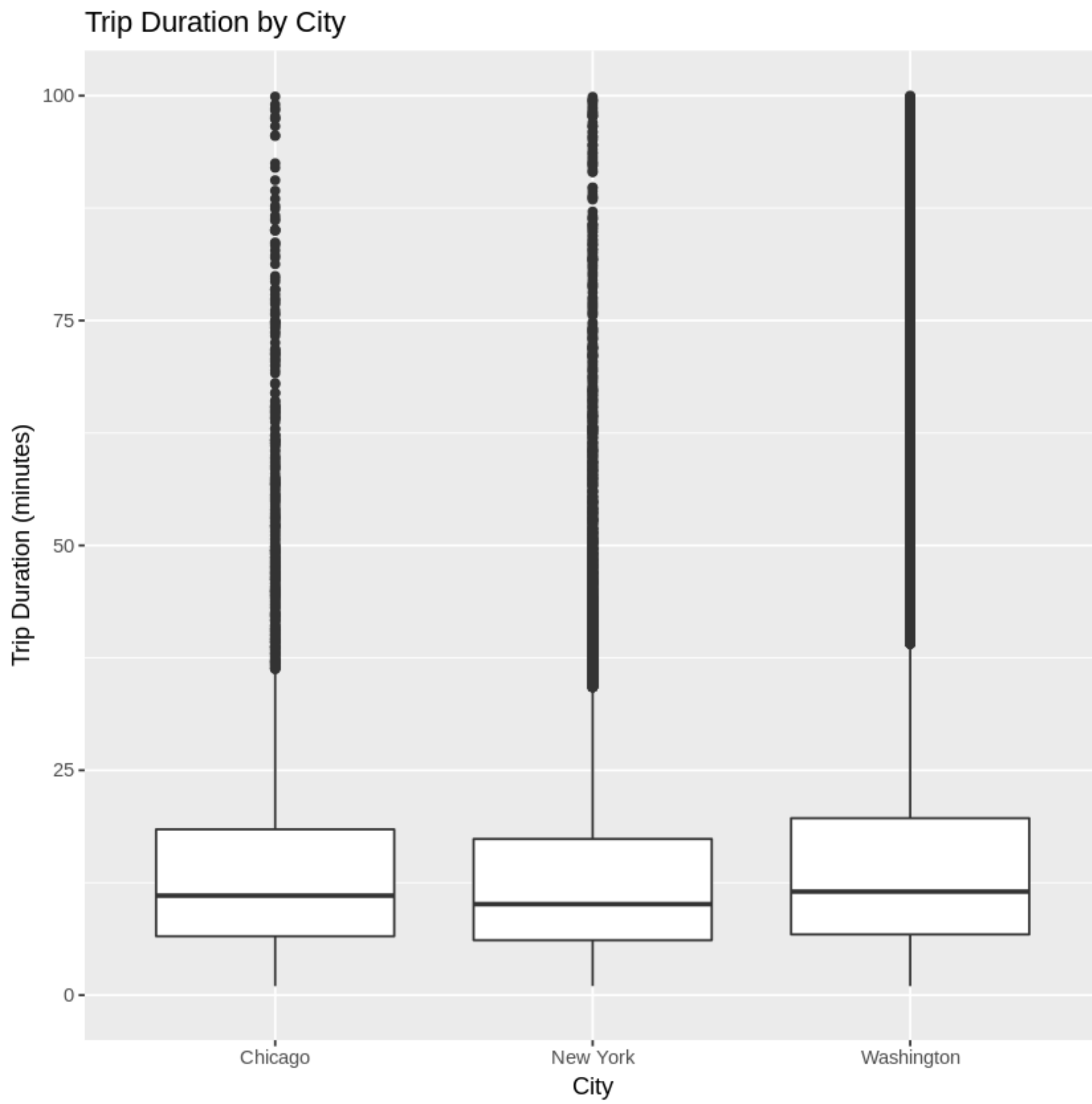
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   60.0   394.2   670.0   937.2  1119.0 85408.0
```

In [81]:
```
washstats <- summary(wash$Trip.Duration)
washstats
```

```
    Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
    60.3     410.9     707.0    1234.0    1233.2  904591.4         1
```

In [83]:
```
#Drawn inspiration from the link below for these lines of code
#https://github.com/josehoras/Exploratory-Data-Analysis-with-R/blob/master/Explore_bike
#Creates a function that returns a new df with the trip duration and city name
average_df <- function(Column, Name,  Cities){
    new_df <- data.frame(Column, Cities, stringsAsFactors = FALSE)
    names(new_df) <- c(Name, 'City')
    return(new_df)
}

#binds these dfs into one master df named trips
trips <- rbind(average_df(chi$Trip.Duration, 'Trip.Duration', 'Chicago'),
               average_df(ny$Trip.Duration, 'Trip.Duration', 'New York'),
               average_df(wash$Trip.Duration, 'Trip.Duration', 'Washington'))

#creates box plots
qplot(x=City, y=Trip.Duration/60,
    data=subset(trips, !is.na(Trip.Duration)),
    geom='boxplot',
    ylim=c(0,100),
    ylab='Trip Duration (minutes)',
    main='Trip Duration by City')
```

```
Warning message:
Removed 2313 rows containing non-finite values (stat_boxplot).
```



The average travel time for users in NYC is 15 minutes. For users in Chicago it's also 15 minutes. For users in D.C. it's 20 minutes. All averages are similar to each other, so a safe overall average of 16.6 minutes seems to be what the data is displaying. It should be noted that there are extreme outliers if the max numbers are considered, but the 3rd quartile values still show to be only a few minutes larger than the means so the data can still be reliable.

If three different cities show similar statistics in trip duration, these numbers can be significant to the business for a variety of purposes. It should be noted that all of this analysis is based solely on the three files provided and more data is necessary to diversify these insights.

0.1 Finishing Up

Congratulations! You have reached the end of the Explore Bikeshare Data Project. You should be very proud of all you have accomplished!

0.2 Directions to Submit

Before you submit your project, you need to create a .html or .pdf version of this notebook in the workspace here. To do that, run the code cell below. If it worked correctly, you should get a return code of 0, and you should see the generated .html file in the workspace directory (click on the orange Jupyter icon in the upper left).

Alternatively, you can download this report as .html via the **File** > **Download as** submenu, and then manually upload it into the workspace directory by clicking on the orange Jupyter icon in the upper left, then using the Upload button.

Once you've done this, you can submit your project by clicking on the "Submit Project" button in the lower right here. This will create and submit a zip file with this .ipynb doc and the .html or .pdf version you created. Congratulations!

```
In [1]: system('python -m nbconvert Explore_bikeshare_data.ipynb')
```