Likelihood of NYC Motor Vehicle Collisions by Time

Sean Andrew Chen (sac820@nyu.edu)

11/14/2017

PUI2017 HW8 - Sean Andrew Chen (sac820@nyu.edu)

Accidents Involving Cyclists and Motorists in New York City

I decided to look at accidents that happen between motorists and cyclists where at least one cyclist is injured. I got the data from the NYC Open Data Portal.

Getting and Cleaning the Data

First we download and clean the data. There was something wrong with NYC's API; it would give a very corrupted CSV. So I decided to host the CSV up on an AWS S3 bucket.

Getting the Data

```
library(RCurl)
```

```
## Loading required package: bitops
```

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
dataFile <- getURL("https://s3.amazonaws.com/aws-website-seanandrewchen-repository-40es3/pui2017_data/N
collisionData <- read.csv(textConnection(dataFile), header = TRUE)
```

Cleaning the Data

One of the main problems was looking at the datetime information. R does not have as advanced a handling of datetime datatypes as Python does. So I had to use some work arounds.

```
collisionData$DATE <- sapply(collisionData$DATE, as.character)
collisionData$TIME <- sapply(collisionData$TIME, as.character)

collisionData$DATETIME <- paste(collisionData$DATE, collisionData$TIME, sep = " ")
collisionData$DATETIME <- strptime(collisionData$DATETIME, "%m/%d/%Y %H:%M")
```

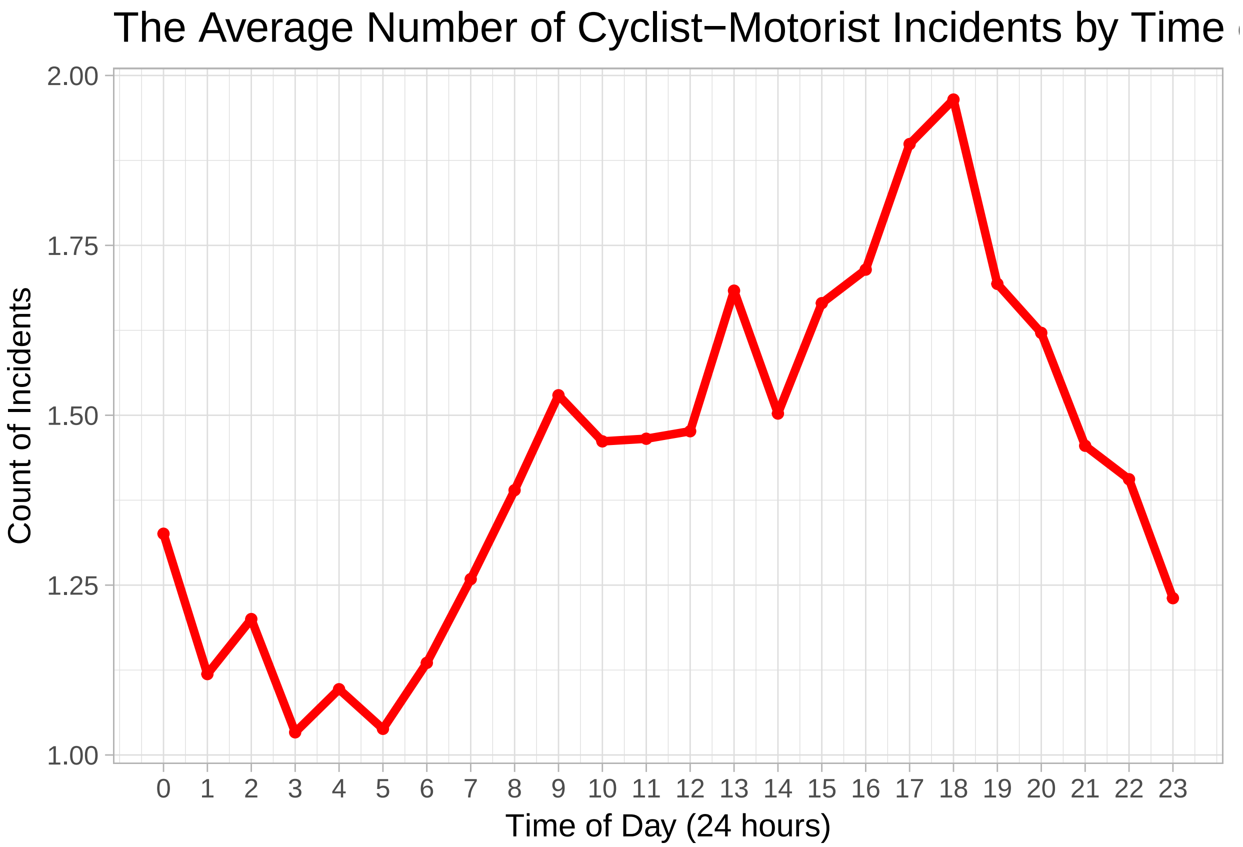
```
collisionData$MONTH <- collisionData$DATETIME$mon+1    #month of year (zero-indexed)
collisionData$YEAR <- collisionData$DATETIME$year+1900 #year (number of years since 1900)
collisionData$DAYOFWEEK <- collisionData$DATETIME$wday #day of week
collisionData$DAYMONTH <- collisionData$DATETIME$mday  #day of month
collisionData$HOUR <- collisionData$DATETIME$hour      #hour
collisionData$MIN <- collisionData$DATETIME$min        #minute
collisionData$TIME <- collisionData$HOUR + (collisionData$MIN/60)

collisionData$DAYOFWEEK <- factor(collisionData$DAYOFWEEK)
levels(collisionData$DAYOFWEEK) <- c('Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday',


collisionData <- collisionData[,!names(collisionData) %in% c("DATE", "CONTRIBUTING.FACTOR.VEHICLE.3", "C

cyclistData <- subset(collisionData, VEHICLE.TYPE.CODE.1 == "BICYCLE")
cyclistData <- subset(cyclistData, select = -c(DATETIME))
```

Plotting

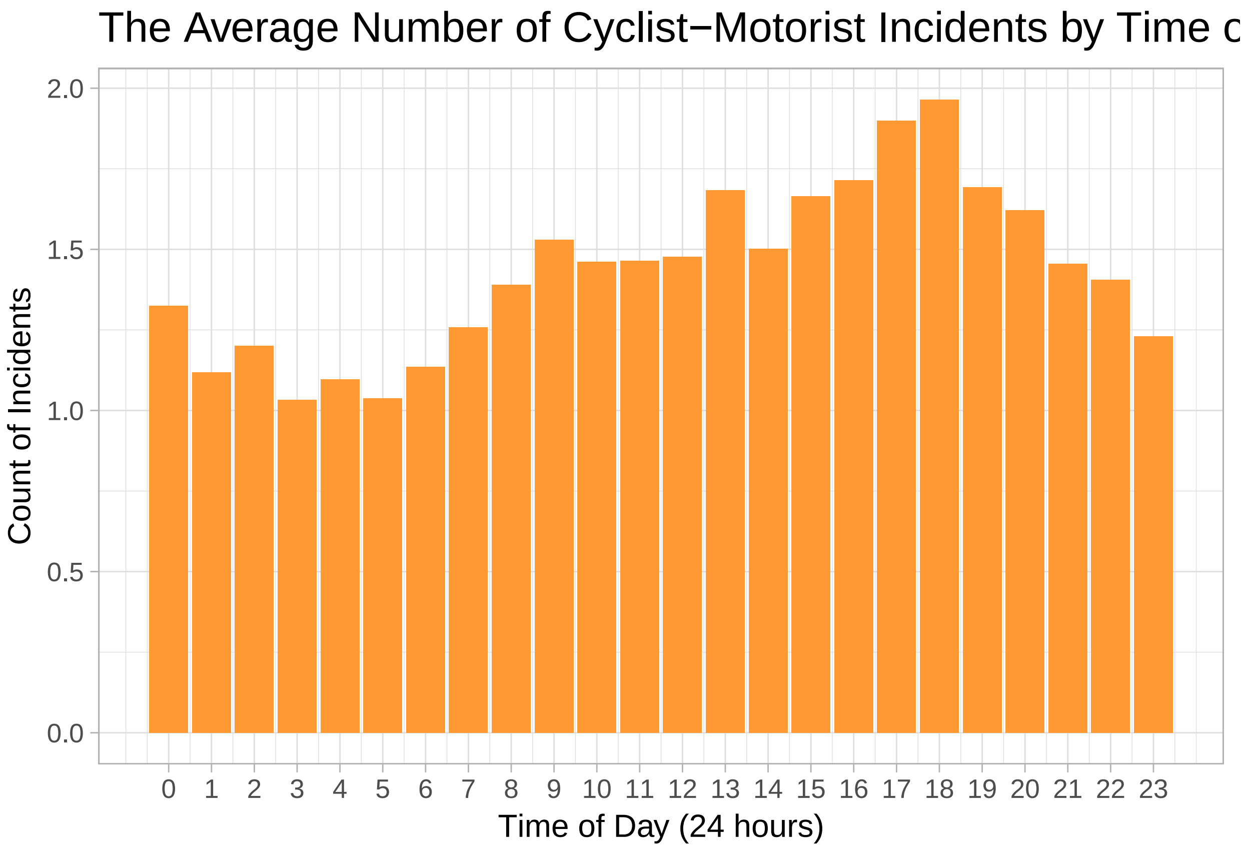
Time of Day

First, let's look at what time of day accidents tend to happen. Is there a specific peak?

```
# average counts per hour
daily_group = group_by(cyclistData, MONTH, DAYMONTH, HOUR)
day_hour_counts = summarise(daily_group, count = n())
hour_group = group_by(day_hour_counts, HOUR)
hour_avg_counts = summarise(hour_group, count = mean(count))

# time series: average counts by time of day
ggplot(hour_avg_counts, aes(x = HOUR, y = count)) + geom_point(colour = "red") +
  geom_line(colour = "red", size = 1.5) +
  theme_light(base_size = 12) + xlab("Time of Day (24 hours)") + ylab("Count of Incidents") +
  scale_x_continuous(breaks=c(0:23)) +
  ggtitle("The Average Number of Cyclist-Motorist Incidents by Time of Day") +
  theme(plot.title = element_text(size = 16))
```

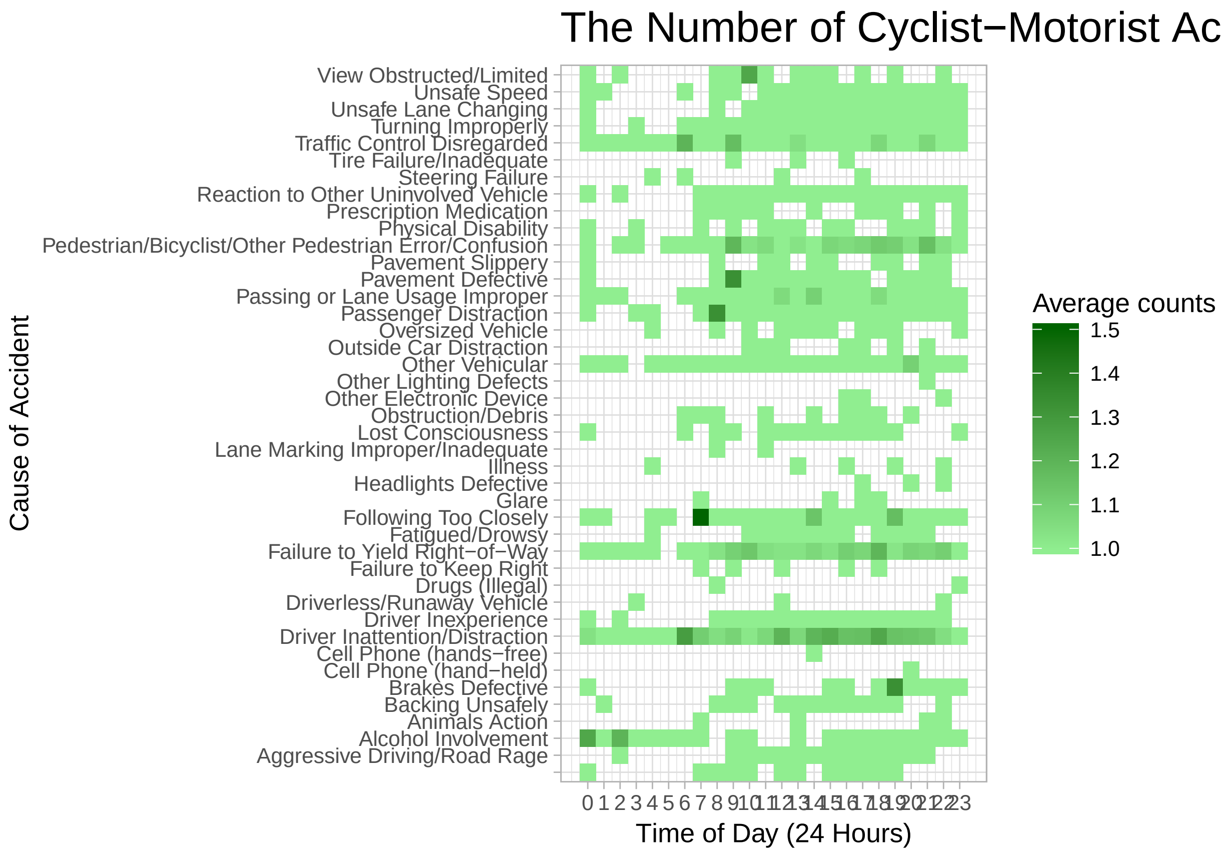


```
ggplot(hour_avg_counts, aes(x = HOUR, y = count)) +
  geom_bar(position = "dodge", stat = "identity", fill = "#FF9933") +
  theme_light(base_size = 12) + labs(x = "Time of Day (24 hours)", y = "Count of Incidents") +
  scale_x_continuous(breaks=c(0:23)) +
  ggtitle("The Average Number of Cyclist-Motorist Incidents by Time of Day") +
  theme(plot.title = element_text(size = 16))
```



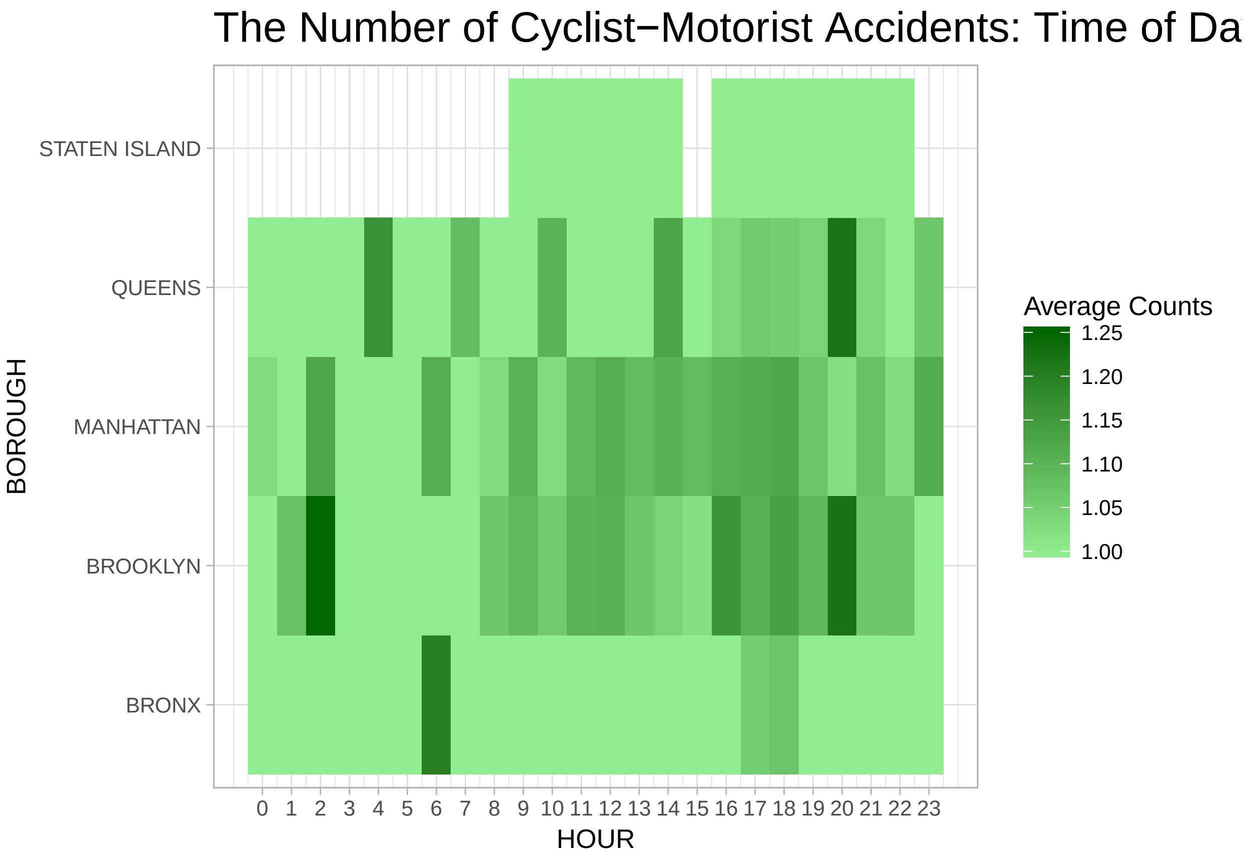
Timing & Cause of Accident

```
cyclistData <- subset(cyclistData, CONTRIBUTING.FACTOR.VEHICLE.1 != "Unspecified")
cyclistData$CAUSEOFACCIDENT <- cyclistData$CONTRIBUTING.FACTOR.VEHICLE.1

hourly_group = group_by(cyclistData, CAUSEOFACCIDENT, MONTH, DAYMONTH, HOUR)
category_day_hour_counts = summarise(hourly_group, count = n())
category_hourly_group = group_by(category_day_hour_counts, CAUSEOFACCIDENT, HOUR)
category_hour_avg_counts = summarise(category_hourly_group, count = mean(count))

ggplot(category_hour_avg_counts, aes(x = HOUR, y = CAUSEOFACCIDENT)) +
  geom_tile(aes(fill = count)) +
  scale_fill_gradient(name = "Average counts", low = "lightgreen", high = "darkgreen") +
  scale_x_continuous(breaks=c(0:23)) +
  theme(axis.title.y = element_blank()) + theme_light(base_size = 10) +
  theme(plot.title = element_text(size=16)) +
  ylab("Cause of Accident") +
  xlab("Time of Day (24 Hours)") +
  ggtitle("The Number of Cyclist-Motorist Accidents: Time of Day vs. Cause of Accident")
```

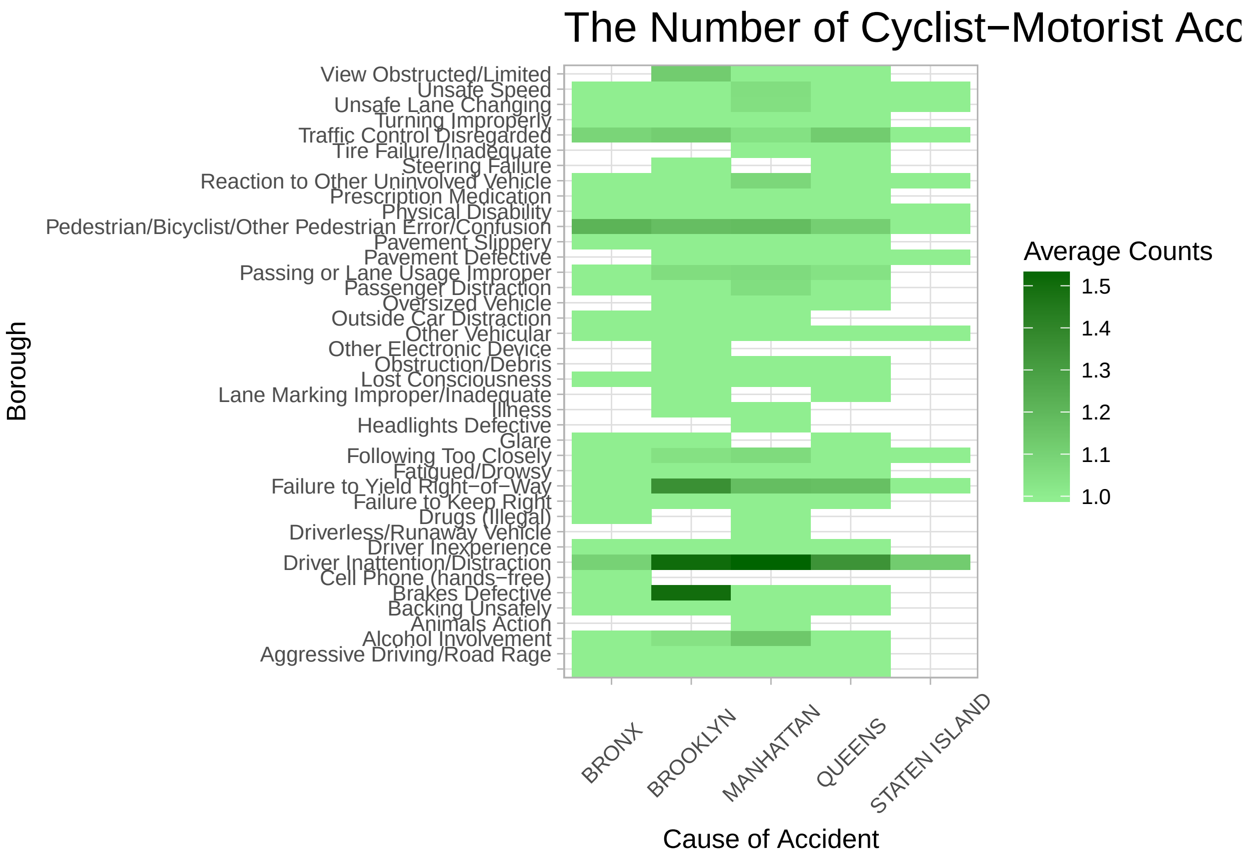


Timing, Cause, & Borough

```
#First timing and borough
cyclistData <- cyclistData[-which(cyclistData$BOROUGH == ""), ]
hourly_group = group_by(cyclistData, BOROUGH, MONTH, DAYMONTH, HOUR)
district_day_hour_counts = summarise(hourly_group, count = n())
district_hourly_group = group_by(district_day_hour_counts, BOROUGH, HOUR)
district_hour_avg_counts = summarise(district_hourly_group, count = mean(count))

ggplot(district_hour_avg_counts, aes(x = HOUR, y = BOROUGH)) +
  geom_tile(aes(fill = count)) +
  scale_fill_gradient(name = "Average Counts", low = "lightgreen", high = "darkgreen") +
  scale_x_continuous(breaks=c(0:23)) +
  theme(axis.title.y = element_blank()) + theme_light(base_size = 10) +
  theme(plot.title = element_text(size = 16)) +
  ggtitle("The Number of Cyclist-Motorist Accidents: Time of Day vs. Borough")
```



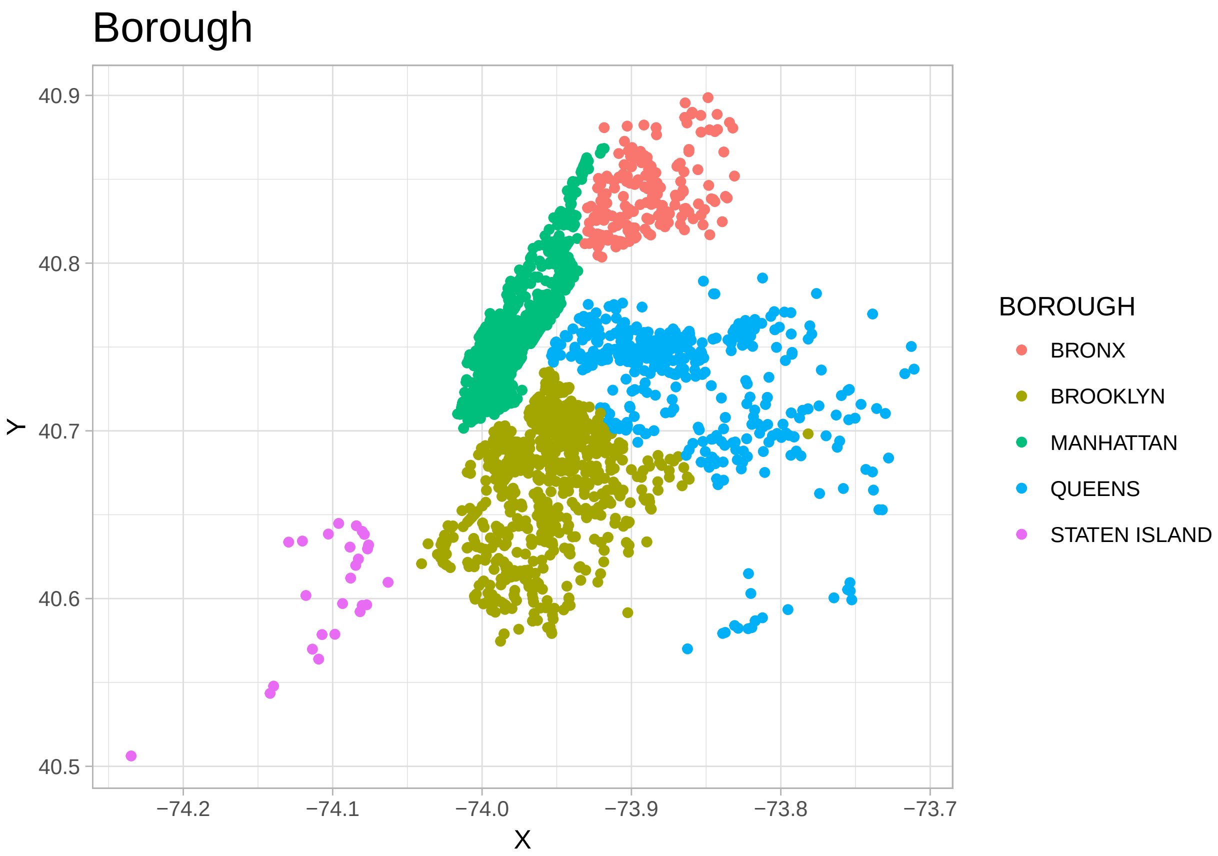
```
#Second borough and cause
category_group = group_by(cyclistData, MONTH, DAYMONTH, BOROUGH, CONTRIBUTING.FACTOR.VEHICLE.1)
day_district_category_counts = summarise(category_group, count = n())
district_category_group = group_by(day_district_category_counts, BOROUGH, CONTRIBUTING.FACTOR.VEHICLE.1
district_category_avg_counts = summarise(district_category_group, count = mean(count))

ggplot(district_category_avg_counts, aes(x = BOROUGH, y = CONTRIBUTING.FACTOR.VEHICLE.1)) +
  geom_tile(aes(fill = count)) +
  scale_fill_gradient(name="Average Counts", low="lightgreen", high="darkgreen") +
  theme(axis.title.y = element_blank()) + theme_light(base_size = 10) +
  theme(plot.title = element_text(size = 16)) +
  ylab("Borough") +
  xlab("Cause of Accident") +
  ggtitle("The Number of Cyclist-Motorist Accidents: Borough vs. Cause of Accident") +
  theme(axis.text.x = element_text(angle = 45,size = 8, vjust = 0.5))
```



Location and Day of Week

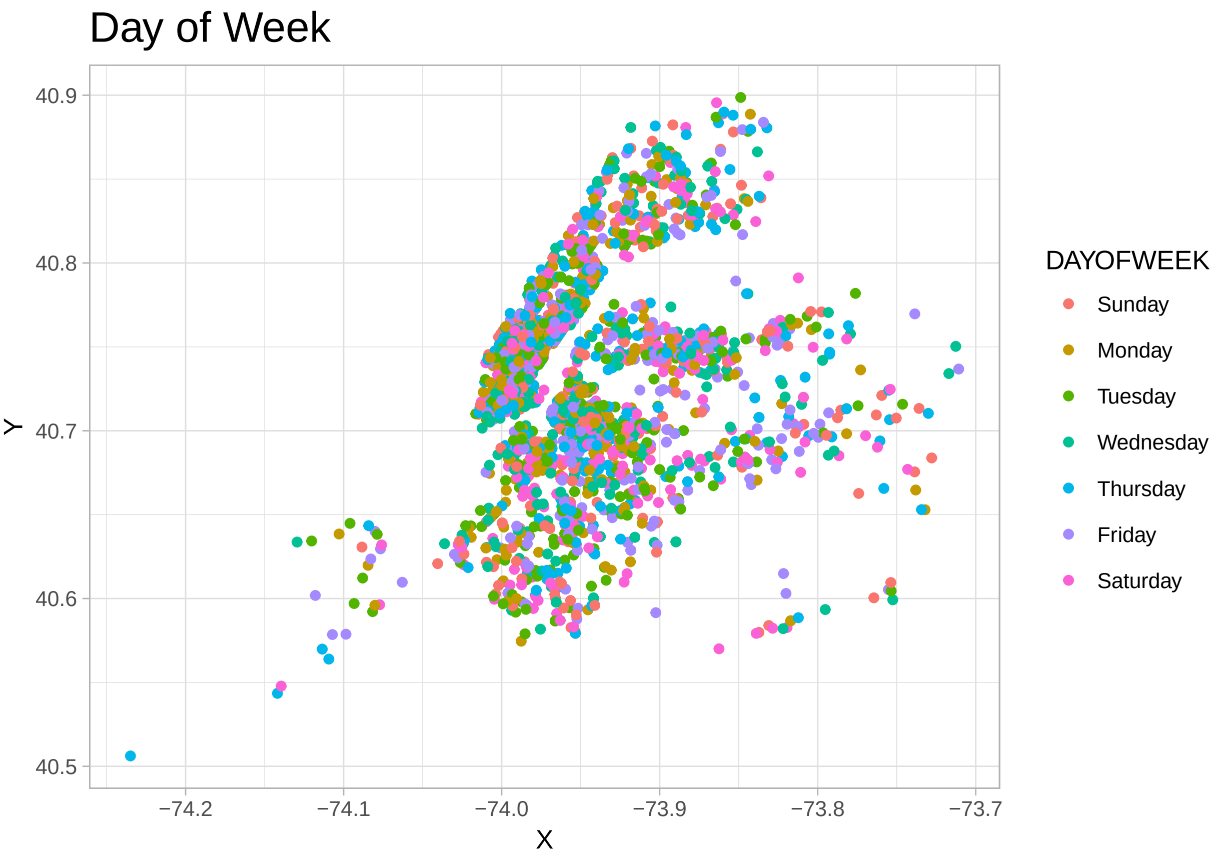
```
# scatter plot
cyclistData <- cyclistData[!with(cyclistData,is.na("LONGITUDE")& is.na("LATITUDE")),]
cyclistData <- cyclistData[-which(cyclistData$LONGITUDE == 0 & cyclistData$LATITUDE == 0), ]
ggplot(cyclistData, aes(x = LONGITUDE, y = LATITUDE)) + geom_point(aes(colour = BOROUGH), size = 1.25)
  theme_light(base_size = 10) + xlab("X") + ylab("Y") +
  ggtitle("Borough") + theme(plot.title=element_text(size = 16))
```

```
## Warning: Removed 129 rows containing missing values (geom_point).
```



```
# location by day of week
ggplot(cyclistData, aes(x = LONGITUDE, y = LATITUDE)) + geom_point(aes(colour = DAYOFWEEK), size = 1.25
  theme_light(base_size = 10) + xlab("X") + ylab("Y") +
  ggtitle("Day of Week") + theme(plot.title=element_text(size = 16))
```

```
## Warning: Removed 129 rows containing missing values (geom_point).
```


Day of Week
40.9
40.8
40.7
40.6
40.5
Y
−74.2
−74.1
−74.0
−73.9
−73.8
−73.7
X
DAYOFWEEK
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday