```r
# Week 8 - Lecture 2, Predictive Policing


# VIDEO 3 - A Basic Line Plot

# Load our data:
mvt = read.csv("mvt.csv", stringsAsFactors=FALSE)

str(mvt)

# Convert the Date variable to a format that R will recognize:
mvt$Date = strptime(mvt$Date, format="%m/%d/%y %H:%M")

# Extract the hour and the day of the week:
mvt$Weekday = weekdays(mvt$Date)
mvt$Hour = mvt$Date$hour

# Let's take a look at the structure of our data again:
str(mvt)

# Create a simple line plot - need the total number of crimes on each
 day of the week. We can get this information by creating a table:
table(mvt$Weekday)

# Save this table as a data frame:
WeekdayCounts = as.data.frame(table(mvt$Weekday))

str(WeekdayCounts)


# Load the ggplot2 library:
library(ggplot2)

# Create our plot
ggplot(WeekdayCounts, aes(x=Var1, y=Freq)) + geom_line(aes(group=1))

#geom_line(aes(group=1), linetype=2) to make the line dashed
#geom_line(aes(group=1),alpha=0.3) to make the line lighter in color

# Make the "Var1" variable an ORDERED factor variable
WeekdayCounts$Var1 = factor(WeekdayCounts$Var1, ordered=TRUE,
 levels=c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",
 "Friday","Saturday"))

# Try again:
ggplot(WeekdayCounts, aes(x=Var1, y=Freq)) + geom_line(aes(group=1))

# Change our x and y labels:
```

```r
ggplot(WeekdayCounts, aes(x=Var1, y=Freq)) + geom_line(aes(group=1))
 + xlab("Day of the Week") + ylab("Total Motor Vehicle Thefts")



# VIDEO 4 - Adding the Hour of the Day

# Create a counts table for the weekday and hour:
table(mvt$Weekday, mvt$Hour)

# Save this to a data frame:
DayHourCounts = as.data.frame(table(mvt$Weekday, mvt$Hour))

str(DayHourCounts)

# Convert the second variable, Var2, to numbers and call it Hour:
DayHourCounts$Hour = as.numeric(as.character(DayHourCounts$Var2))

# Create out plot:
ggplot(DayHourCounts, aes(x=Hour, y=Freq)) +
 geom_line(aes(group=Var1))

# Change the colors
ggplot(DayHourCounts, aes(x=Hour, y=Freq)) +
 geom_line(aes(group=Var1, color=Var1), size=2)

# Separate the weekends from the weekdays:
DayHourCounts$Type = ifelse((DayHourCounts$Var1 == "Sunday") |
 (DayHourCounts$Var1 == "Saturday"), "Weekend", "Weekday")

# Redo our plot, this time coloring by Type:
ggplot(DayHourCounts, aes(x=Hour, y=Freq)) +
 geom_line(aes(group=Var1, color=Type), size=2)


# Make the lines a little transparent:
ggplot(DayHourCounts, aes(x=Hour, y=Freq)) +
 geom_line(aes(group=Var1, color=Type), size=2, alpha=0.5)



# Fix the order of the days:
DayHourCounts$Var1 = factor(DayHourCounts$Var1, ordered=TRUE,
 levels=c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
 "Saturday", "Sunday"))

# Make a heatmap:
ggplot(DayHourCounts, aes(x = Hour, y = Var1)) + geom_tile(aes(fill =
 Freq))

# Change the label on the legend, and get rid of the y-label:

ggplot(DayHourCounts, aes(x = Hour, y = Var1)) + geom_tile(aes(fill =
 Freq)) + scale_fill_gradient(name="Total MV Thefts") +
 theme(axis.title.y = element_blank())

# Change the color scheme
ggplot(DayHourCounts, aes(x = Hour, y = Var1)) + geom_tile(aes(fill =
 Freq)) + scale_fill_gradient(name="Total MV Thefts", low="white",
 high="red") + theme(axis.title.y = element_blank())




# VIDEO 5 - Maps

# Install and load two new packages:
install.packages("maps")
install.packages("ggmap")
library(maps)
library(ggmap)

# Load a map of Chicago into R:
chicago = get_map(location = "chicago", zoom = 11)

# Look at the map
ggmap(chicago)

# Plot the first 100 motor vehicle thefts:
ggmap(chicago) + geom_point(data = mvt[1:100,], aes(x = Longitude, y
 = Latitude))

# Round our latitude and longitude to 2 digits of accuracy, and
 create a crime counts data frame for each area:
LatLonCounts = as.data.frame(table(round(mvt$Longitude,2), round(mvt
 $Latitude,2)))

str(LatLonCounts)

# Convert our Longitude and Latitude variable to numbers:
LatLonCounts$Long = as.numeric(as.character(LatLonCounts$Var1))
LatLonCounts$Lat = as.numeric(as.character(LatLonCounts$Var2))

# Plot these points on our map:
ggmap(chicago) + geom_point(data = LatLonCounts, aes(x = Long, y =
 Lat, color = Freq, size=Freq))
```

```
# Change the color scheme:
ggmap(chicago) + geom_point(data = LatLonCounts, aes(x = Long, y =
 Lat, color = Freq, size=Freq)) + scale_colour_gradient(low="yellow",
 high="red")

# We can also use the geom_tile geometry
ggmap(chicago) + geom_tile(data = LatLonCounts, aes(x = Long, y =
 Lat, alpha = Freq), fill="red")



# VIDEO 6 - Geographical Map on US

# Load our data:
murders = read.csv("murder.csv")

str(murders)

# Load the map of the US
statesMap = map_data("state")

str(statesMap)

# Plot the map:
ggplot(statesMap, aes(x = long, y = lat, group = group)) +
 geom_polygon(fill = "white", color = "black") +
 coord_map("mercator")

# Create a new variable called region with the lowercase names to
 match the statesMap:
murders$region = tolower(murders$State)

# Join the statesMap data and the murders data into one dataframe:
murderMap = merge(statesMap, murders, by="region")
str(murderMap)

# Plot the number of murder on our map of the United States:
ggplot(murderMap, aes(x = long, y = lat, group = group, fill =
 Murders)) + geom_polygon(colour = "black") + scale_fill_gradient(low
 = "black", high = "red", guide = "legend")

# Plot a map of the population:
ggplot(murderMap, aes(x = long, y = lat, group = group, fill =
 Population)) + geom_polygon(colour = "black") +
 scale_fill_gradient(low = "black", high = "red", guide = "legend")

# Create a new variable that is the number of murders per 100,000
```

```
 population:
murderMap$MurderRate = murderMap$Murders / murderMap$Population *
 100000

# Redo our plot with murder rate:
ggplot(murderMap, aes(x = long, y = lat, group = group, fill =
 MurderRate)) + geom_polygon(colour = "black") +
 scale_fill_gradient(low = "black", high = "red", guide = "legend")

# Redo the plot, removing any states with murder rates above 10:
ggplot(murderMap, aes(x = long, y = lat, group = group, fill =
 MurderRate)) + geom_polygon(colour = "black") +
 scale_fill_gradient(low = "black", high = "red", guide = "legend",
 name = "Murder Rate per 100k", limits = c(0.9,10))
```