MSc Business Analytics

"Information Systems and Business Process Management"
Assignment: Data visualization with R - GGPLOT2

Surname : Chatzimoschou
Name : Angeliki
St. code : BAPT1534

Contents:

PART -A-

Parole Dataset

This dataset is about crime rating in USA.

```
# Installing packages
```

install.packages("MASS")
install.packages("ggplot2")

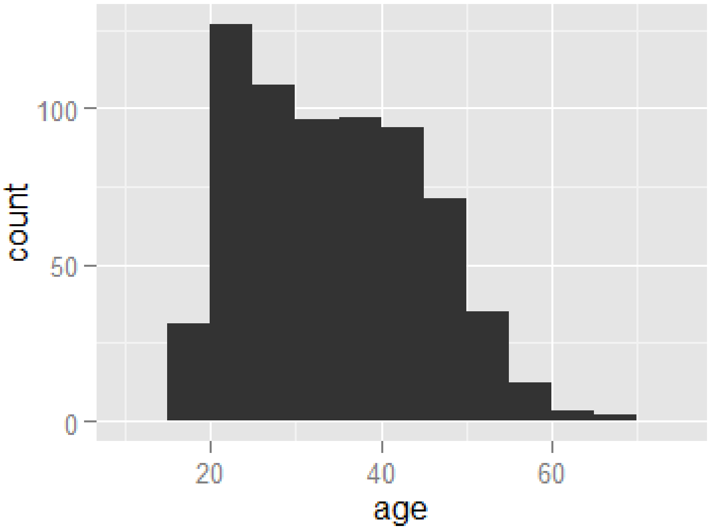
```
# Loading parole dataset
parole<-read.csv("~/parole.csv", stringsAsFactors=FALSE)

# Setting as factor the variables: male, state & crime
parole$male<-as.factor(parole$male)
parole$state<-as.factor(parole$state)
parole$crime<-as.factor(parole$crime)

# Question 1.1 - Count fraction
library(MASS)
fraction<-fractions(nrow(subset(parole,male==0 &
violator==1,select=c(male)))/nrow(subset(parole,violator==1,select=c(ma
le))))
fraction

## [1] 7/39
```

Answer: **7/39**

```
# Question 1.2 - Most frequent crime in the state of Kentucky
a=table(parole$crime[parole$state == 2])
names(a) = c("other", "larceny", "drug", "driving")
names(which.max(a))

## [1] "drug"
```

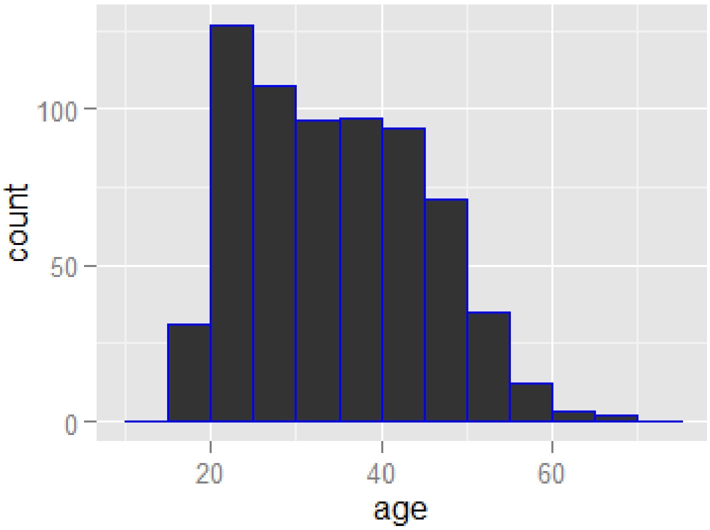
Answer: **b**

```r
# Question 2.1 - Most frequent age of delinquency
library(ggplot2)
ggplot(data=parole, aes(x=age))+geom_histogram(binwidth=5)
```



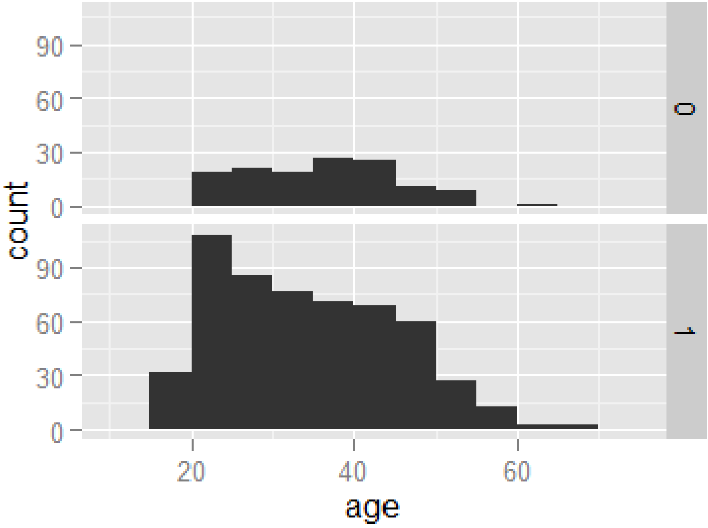
Answer: **a**

```r
# Question 2.2 - Blue colour
ggplot(data=parole, aes(x=age))+geom_histogram(binwidth=5,color="blue")
```



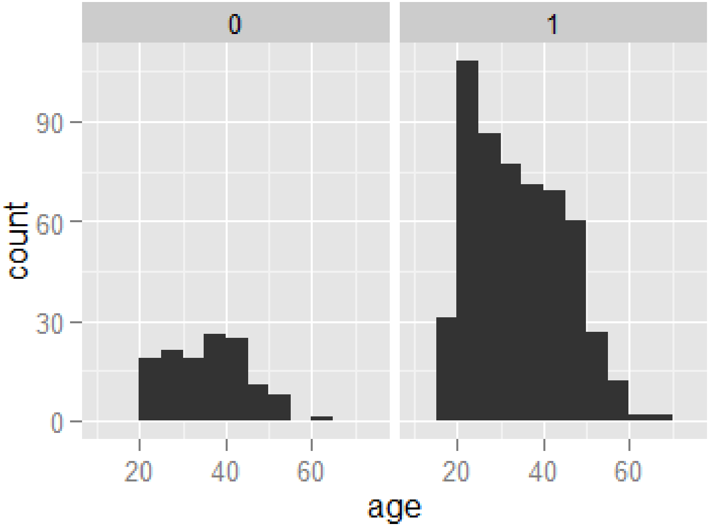
Answer: **c**

```
# Question 3.1 - Age with most female parolees
ggplot(data=parole, aes(x=age))+geom_histogram(binwidth=5) +
facet_grid(male~.)
```



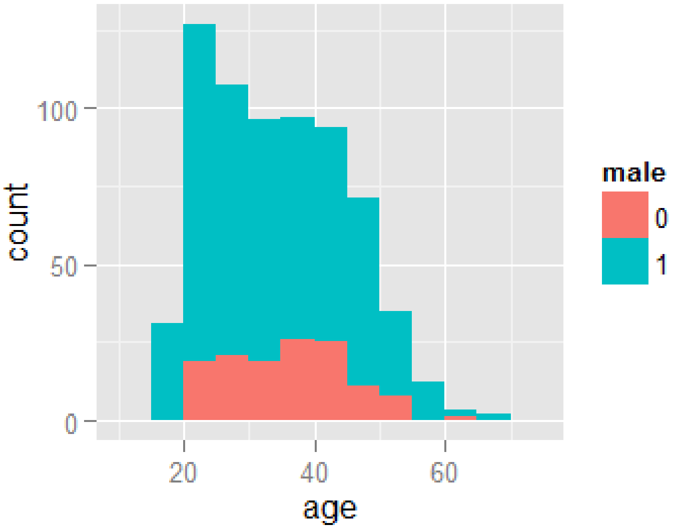
Answer: **d**

```
# Question 3.2 - facet_grid(.~male)
ggplot(data=parole, aes(x=age))
+geom_histogram(binwidth=5)+facet_grid(.~male)
```



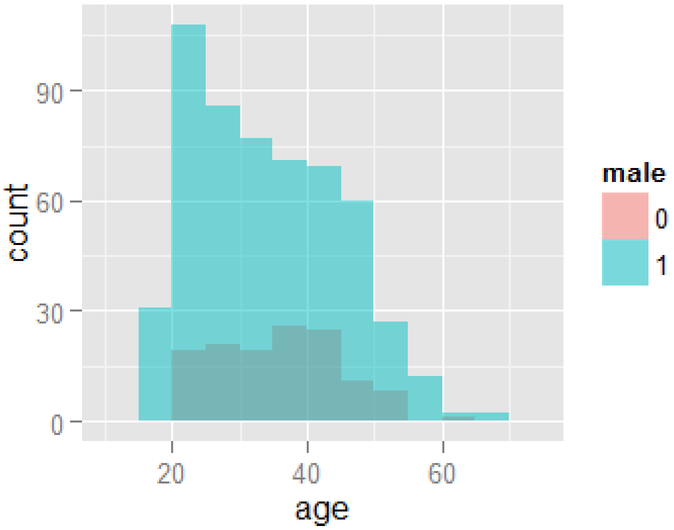
Answer: **b**

```r
# Question 3.3 - Histogram colour for female parolees
ggplot(data=parole,aes(x=age,fill=male))+geom_histogram(binwidth=5)
```



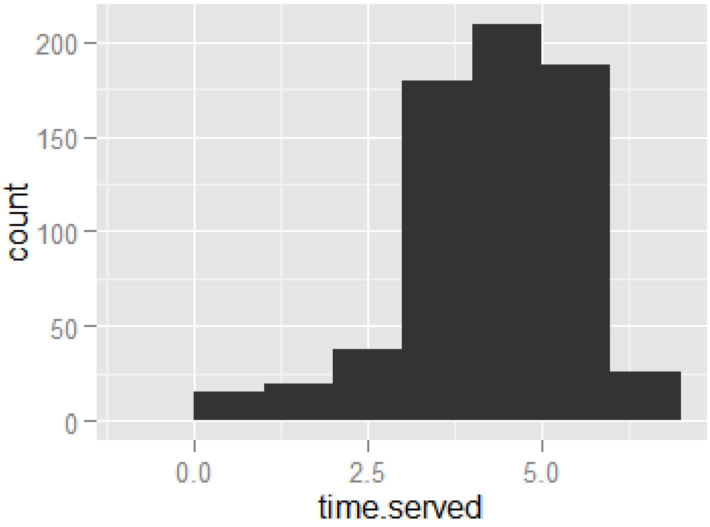
Answer: **b**

```r
# Question 3.4 - Adding transparency and overlaying the two histograms
ggplot(data=parole, aes(x=age,fill=male))
+geom_histogram(binwidth=5,position="identity",alpha=0.5)
```



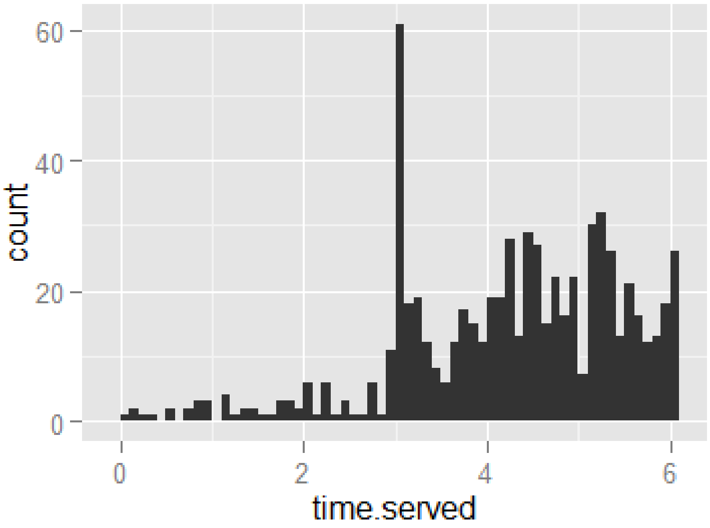
Answer: **a**,**i**,**k**

```
# Question 4.1 - Most common length of time served
ggplot(data=parole,aes(x=time.served))+geom_histogram(binwidth=1)
```



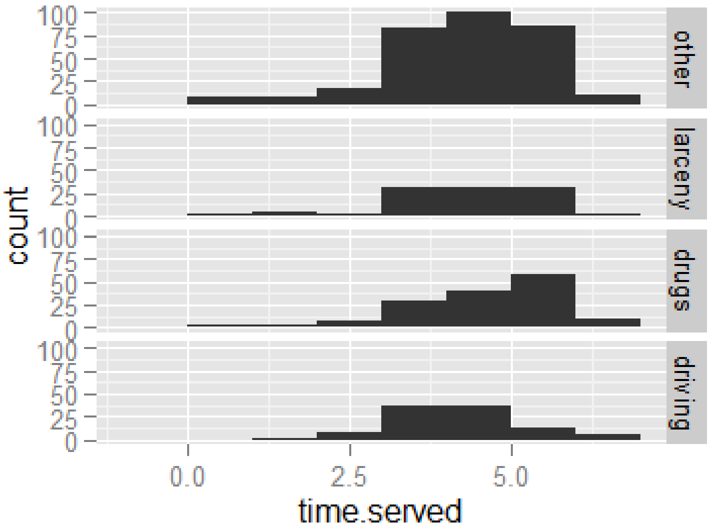
Answer: **c**

```
# Question 4.2 - (binwidth=0.1)
ggplot(data=parole,aes(x=time.served))+geom_histogram(binwidth=0.1)
```



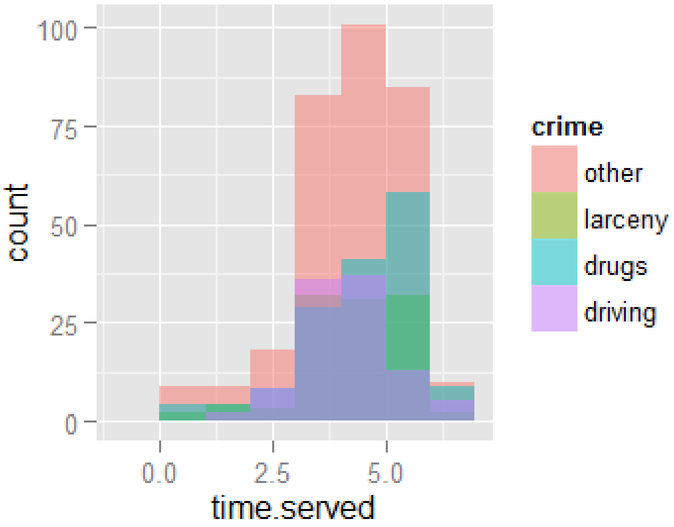
Answer: **b**

```
# Question 4.3 - Histogram for time served concerning each crime
seperately
levels(parole$crime) = c("other", "larceny", "drugs", "driving")
ggplot(data=parole, aes(x=time.served))
+geom_histogram(binwidth=1)+facet_grid(crime~.)
```



Answer 4.3a: **c**   Answer 4.3b: **b**

```
# Question 4.4 - Overlaying the 4 crime histograms
ggplot(data=parole, aes(x=time.served,fill=crime))
+geom_histogram(binwidth=1,position ="identity",alpha=0.5)
```



Answer: **a**

Part -B-

WHO Dataset

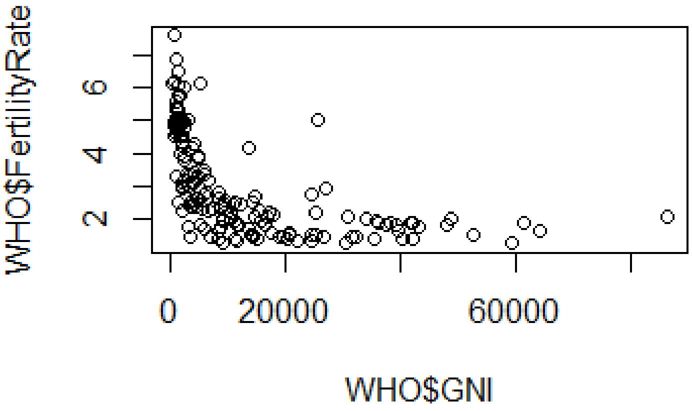
This dataset is about world population and several indexes.

```
# Installing packages
```

```
install.packages("ggplot2")
```

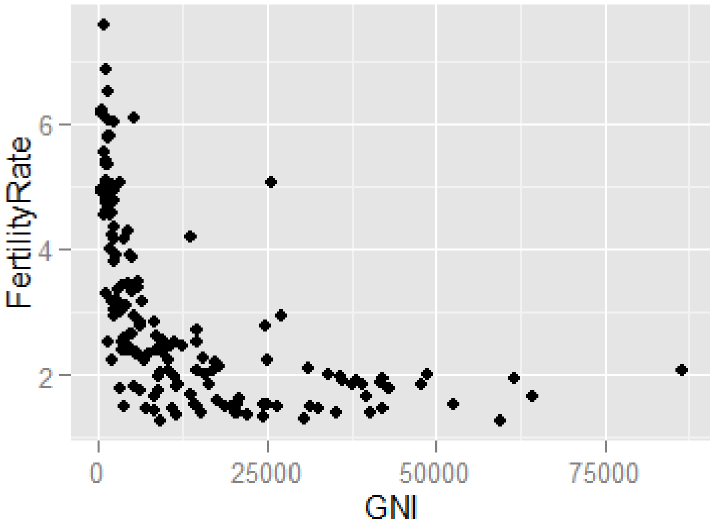
```
# Removing exponential notation
options(scipen=999)

# Loading WHO dataset
WHO<-read.csv("~/WHO.csv")

# Checking data frame structure
str(WHO)

## 'data.frame':    194 obs. of  13 variables:
##  $ Country                       : Factor w/ 194 levels
"Afghanistan",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Region                        : Factor w/ 6 levels
"Africa","Americas",..: 3 4 1 4 1 2 2 4 6 4 ...
##  $ Population                    : int  29825 3162 38482 78 20821 89
41087 2969 23050 8464 ...
##  $ Under15                       : num  47.4 21.3 27.4 15.2 47.6 ...
##  $ Over60                        : num  3.82 14.93 7.17 22.86
3.84 ...
##  $ FertilityRate                 : num  5.4 1.75 2.83 NA 6.1 2.12 2.2
1.74 1.89 1.44 ...
##  $ LifeExpectancy                : int  60 74 73 82 51 75 76 71 82 81
...
##  $ ChildMortality                : num  98.5 16.7 20 3.2 163.5 ...
##  $ CellularSubscribers           : num  54.3 96.4 99 75.5 48.4 ...
##  $ LiteracyRate                  : num  NA NA NA NA 70.1 99 97.8 99.6
NA NA ...
##  $ GNI                           : num  1140 8820 8310 NA 5230 ...
##  $ PrimarySchoolEnrollmentMale   : num  NA NA 98.2 78.4 93.1 91.1 NA
NA 96.9 NA ...
##  $ PrimarySchoolEnrollmentFemale: num  NA NA 96.4 79.4 78.2 84.5 NA
NA 97.5 NA ...
```

```
# Scatterplot GNI-Fertility Rate
plot(WHO$GNI,WHO$FertilityRate)
```

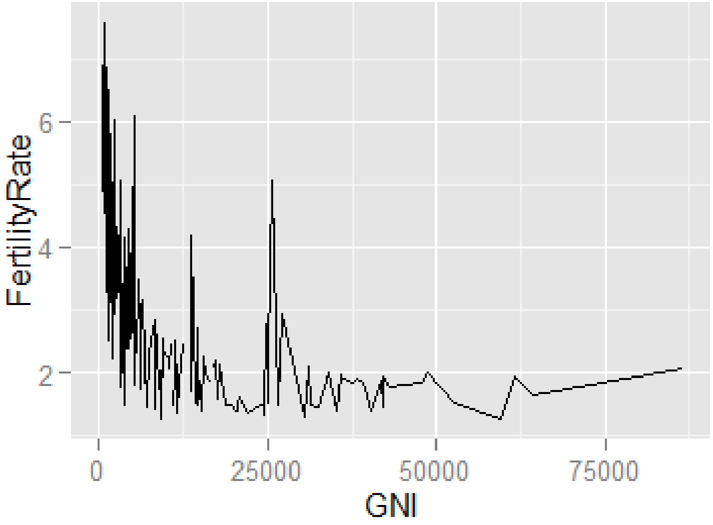


We observe that higher fertility rates are related to low income.

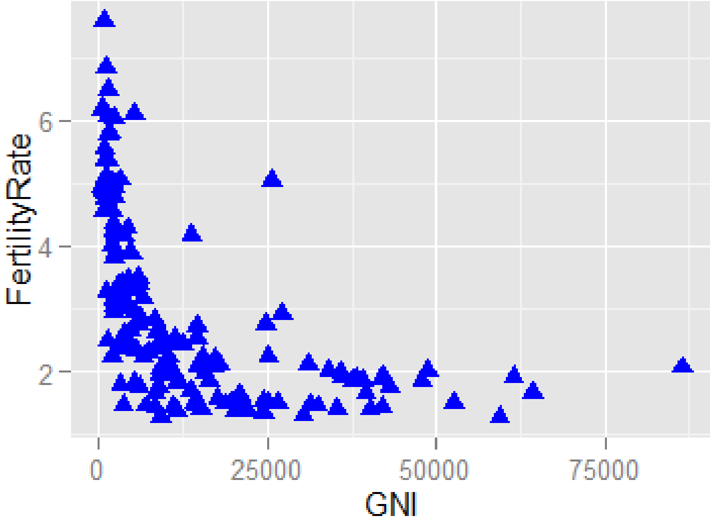
```
# Plot GNI-Fertility Rate with points
library(ggplot2)

scatterplot<-ggplot(WHO,aes(GNI,FertilityRate))
scatterplot+geom_point()

## Warning: Removed 35 rows containing missing values (geom_point).
```



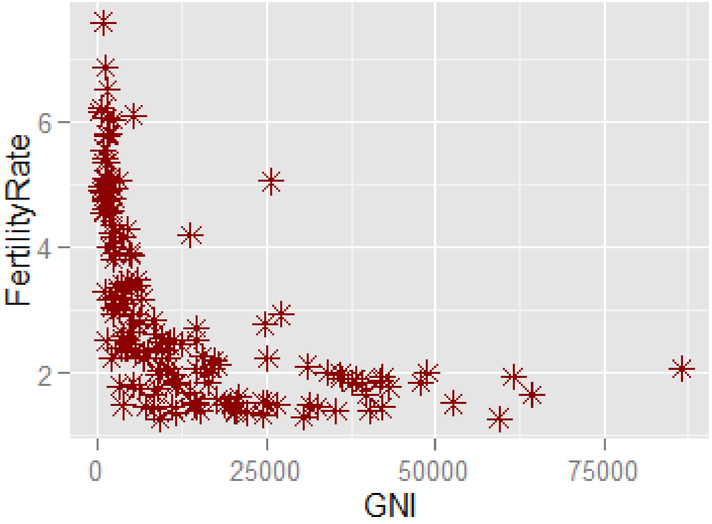
```
# Plot GNI-Fertility Rate with line
scatterplot+geom_line()

## Warning: Removed 32 rows containing missing values (geom_path).
```



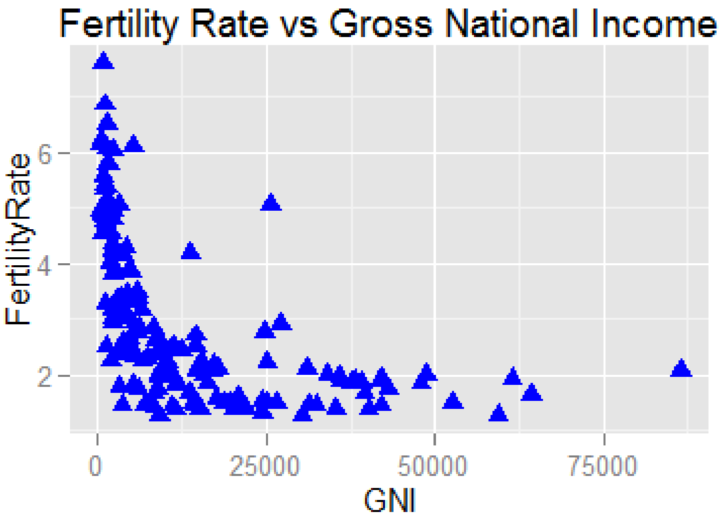
```
# Changing colour, size & shape
scatterplot+geom_point(color="blue",shape=17,size=3)

## Warning: Removed 35 rows containing missing values (geom_point).
```



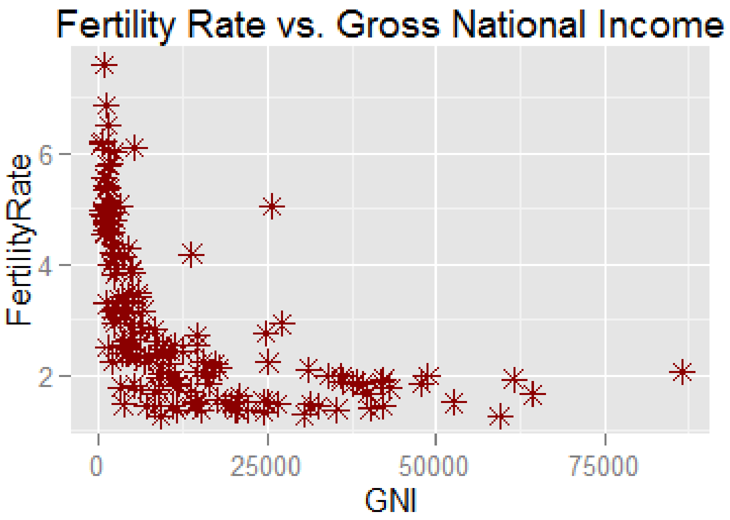
```
scatterplot+geom_point(color="darkred",shape=8,size=3)

## Warning: Removed 35 rows containing missing values (geom_point).
```



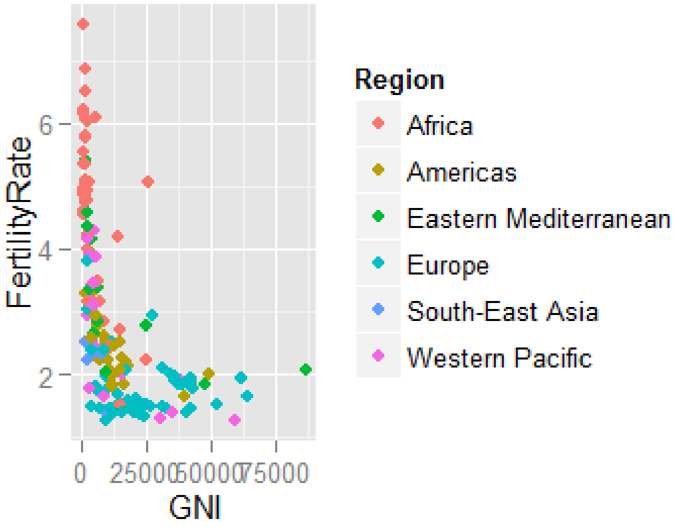
```
# Adding title
scatterplot+geom_point(color="blue",shape=17,size=3)+ggtitle("Fertility
Rate vs Gross National Income")

## Warning: Removed 35 rows containing missing values (geom_point).
```



```
# Save plot as variable
FertilityGNIplot<-
scatterplot+geom_point(color="blue",shape=17,size=3)+ggtitle("Fertility Rate vs Gross National Income")

# Export plot to PDF
pdf("FertilityGNIplot.pdf")
print(FertilityGNIplot)

## Warning: Removed 35 rows containing missing values (geom_point).

dev.off()

## png
##   2

# Export plot to SVG
svg("FertilityGNIplot.svg")
print(FertilityGNIplot)

## Warning: Removed 35 rows containing missing values (geom_point).

dev.off()

## png
##   2

# Plot with dark red colour, stars & title
scatterplot+geom_point(color="darkred",shape=8,size=3)+ggtitle("Fertility Rate vs. Gross National Income")

## Warning: Removed 35 rows containing missing values (geom_point).
```

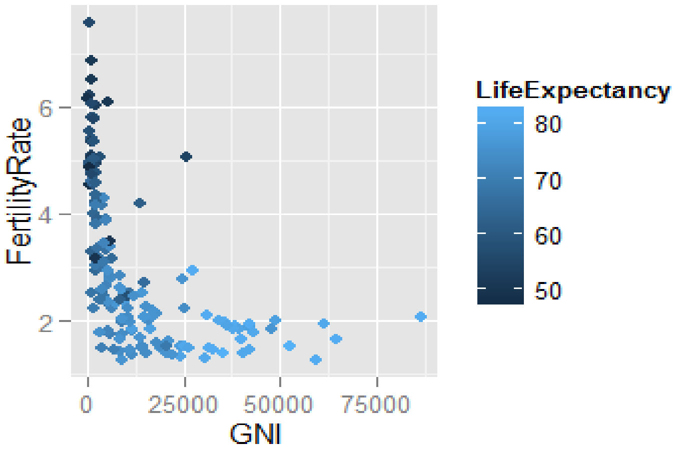


```
# Plot which shows the correlation of GNI-Fertility Rate per Region
ggplot(WHO,aes(x=GNI,y=FertilityRate,color=Region))+geom_point()

## Warning: Removed 35 rows containing missing values (geom_point).
```



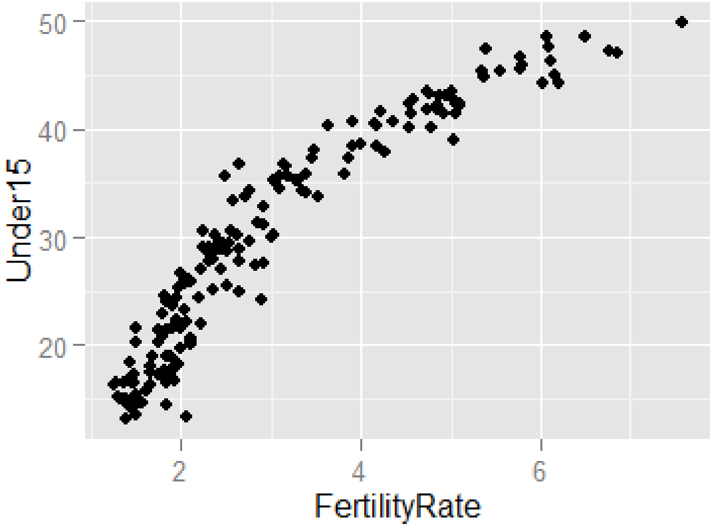
As we see mostly in Africa we find low income correlation and high fertility rate.

```
# Plot which shows the correlation of GNI-Fertility Rate per Life
Expectancy
ggplot(WHO,aes(x=GNI,y=FertilityRate,color=LifeExpectancy))
+geom_point()

## Warning: Removed 35 rows containing missing values (geom_point).
```



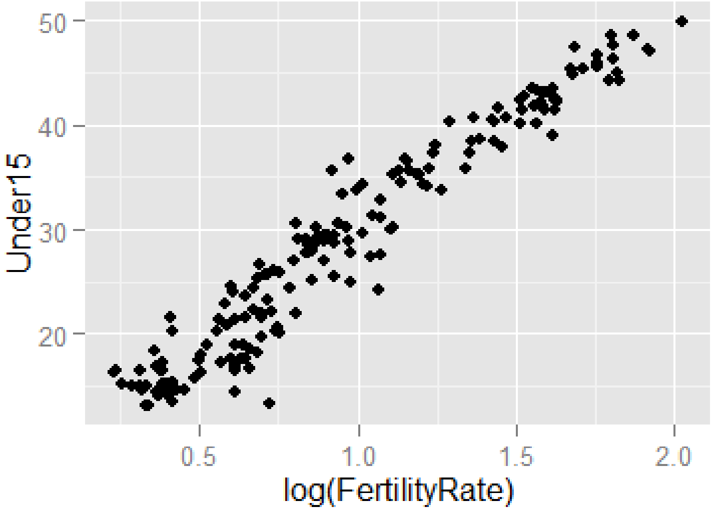
As we see people who have more children and low income tend to live less than people with less children and higher income.

```
# Correlation Plot Fertility Rate - Under 15
ggplot(WHO,aes(x=FertilityRate,y=Under15))+geom_point()

## Warning: Removed 11 rows containing missing values (geom_point).
```



As we see our plot approaches more the pattern of a logistic regression line. This happens because the rate of increase of Under15 variable is smaller than the one of Fertility Rate variable.

```
# Correlation Plot log(Fertility Rate) - Under 15 in order to make our
line more linear
ggplot(WHO,aes(x=log(FertilityRate),y=Under15))+geom_point()

## Warning: Removed 11 rows containing missing values (geom_point).
```



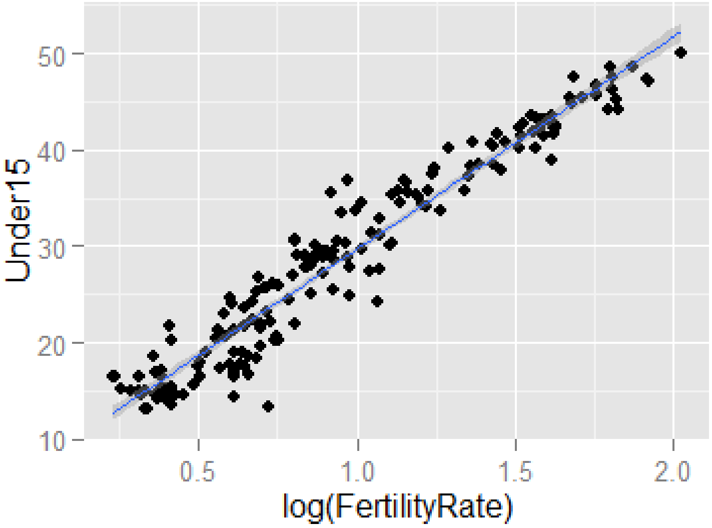
```r
# Constructing mod: linear regression model
mod<-lm(Under15~log(FertilityRate),data = WHO)
```

As we see our model consists of the predicted variable Under15 and the independent variable Fertility Rate (predictor).

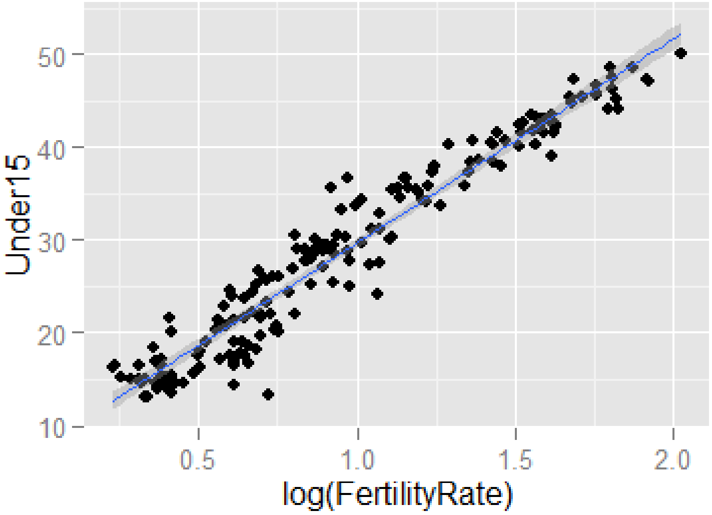
```r
# Model summary
summary(mod)

##
## Call:
## lm(formula = Under15 ~ log(FertilityRate), data = WHO)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.3131  -1.7742   0.0446   1.7440   7.7174
##
## Coefficients:
##                    Estimate Std. Error t value            Pr(>|t|)

## (Intercept)          7.6540     0.4478   17.09 <0.0000000000000002
***
## log(FertilityRate)  22.0547     0.4175   52.82 <0.0000000000000002
***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.65 on 181 degrees of freedom
##   (11 observations deleted due to missingness)
## Multiple R-squared:  0.9391, Adjusted R-squared:  0.9387
## F-statistic:  2790 on 1 and 181 DF,  p-value: < 0.00000000000000022
```

As we see R-squared=0.9391 which means that our predictor variable is of high statistical significance.

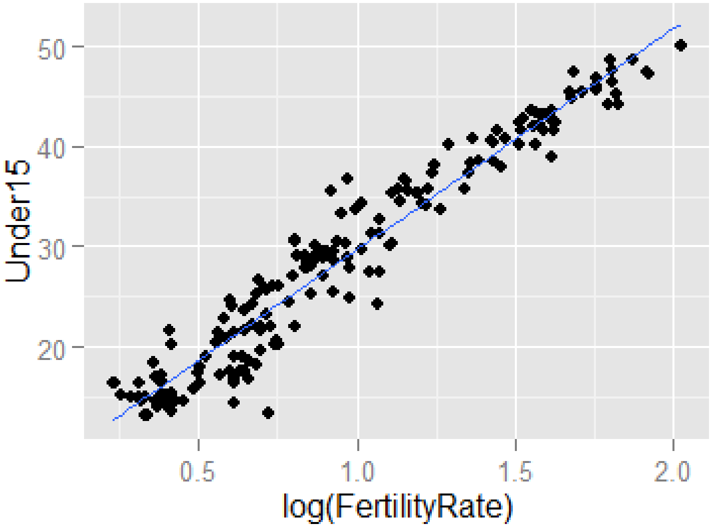
```
# Log(Fertility Rate) - Under 15 Plot with linear regression line
ggplot(WHO,aes(x=log(FertilityRate),y=Under15))+geom_point()
+stat_smooth(method = "lm")

## Warning: Removed 11 rows containing missing values (stat_smooth).

## Warning: Removed 11 rows containing missing values (geom_point).
```



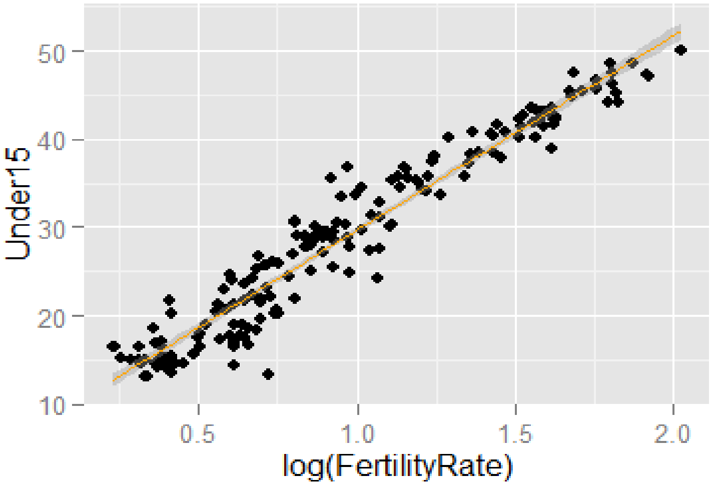
```
# Log(Fertility Rate) - Under 15 Plot with linear regression line and
99% confidence interval
ggplot(WHO,aes(x=log(FertilityRate),y=Under15))+geom_point()
+stat_smooth(method = "lm", level = 0.99)
## Warning: Removed 11 rows containing missing values (stat_smooth).
##Warning: Removed 11 rows containing missing values (geom_point).
```



```
# Correlation Plot log(Fertility Rate) - Under 15 with linear
regression line and NO confidence interval
ggplot(WHO,aes(x=log(FertilityRate),y=Under15))+geom_point()
+stat_smooth(method = "lm", se=FALSE)

## Warning: Removed 11 rows containing missing values (stat_smooth).
## Warning: Removed 11 rows containing missing values (geom_point).
```



```
# Correlation Plot log(Fertility Rate) - Under 15 with orange linear
regression line
ggplot(WHO,aes(x=log(FertilityRate),y=Under15))+geom_point()
+stat_smooth(method = "lm", colour="orange")

## Warning: Removed 11 rows containing missing values (stat_smooth).
## Warning: Removed 11 rows containing missing values (geom_point).
```



MVT Dataset

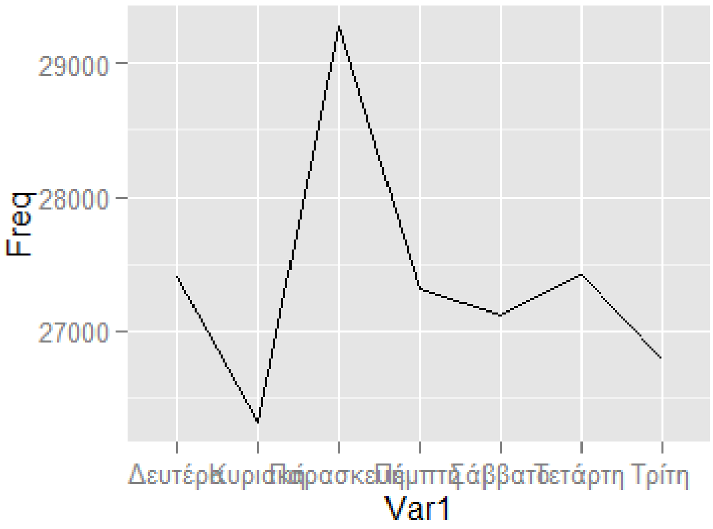
This dataset is about Motor Vehicle Thefts in USA.

```
# Installing packages
```

```
install.packages("ggplot2")
install.packages("maps")
install.packages("ggmap")
```

```
# Loading MVT dataset
mvt<-read.csv("~/MVT.csv", stringsAsFactors=FALSE)

# Checking data frame structure
str(mvt)

## 'data.frame':    191641 obs. of  3 variables:
##  $ Date     : chr  "12/31/12 23:15" "12/31/12 22:00" "12/31/12 22:00" "12/31/12 22:00" ...
##  $ Latitude : num  41.8 41.9 42 41.8 41.8 ...
##  $ Longitude: num  -87.6 -87.7 -87.8 -87.7 -87.6 ...

# Transforming variable Date into readable R format
mvt$Date<-strptime(mvt$Date, format="%m/%d/%y %H:%M")

# Extracting Weekday variable from Date variable
mvt$Weekday<-weekdays(mvt$Date)

# Extracting Hour variable from Date variable
mvt$Hour<-mvt$Date$hour

# re-Checking data frame structure
str(mvt)

## 'data.frame':    191641 obs. of  5 variables:
##  $ Date     : POSIXlt, format: "2012-12-31 23:15:00" "2012-12-31 22:00:00" ...
##  $ Latitude : num  41.8 41.9 42 41.8 41.8 ...
##  $ Longitude: num  -87.6 -87.7 -87.8 -87.7 -87.6 ...
##  $ Weekday  : chr  "Δευτέρα" "Δευτέρα" "Δευτέρα" "Δευτέρα" ...
##  $ Hour     : int  23 22 22 22 21 20 20 20 19 18 ...
```

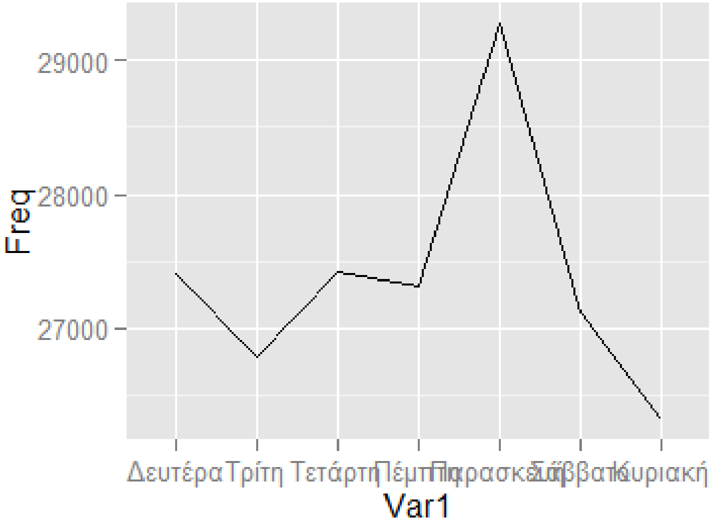
```
# Weekday frequency table
table(mvt$Weekday)

##
##   Δευτέρα   Κυριακή Παρασκευή    Πέμπτη   Σάββατο   Τετάρτη     Τρίτη
##     27397     26316     29284     27319     27118     27416     26791

# Transforming Weekday frequency table into WeekdayCounts data-frame
WeekdayCounts<-as.data.frame(table(mvt$Weekday))
# Checking WeekdayCounts data-frame structure
str(WeekdayCounts)

## 'data.frame':    7 obs. of  2 variables:
##  $ Var1: Factor w/ 7 levels "Δευτέρα","Κυριακή",..: 1 2 3 4 5 6 7
##  $ Freq: int  27397 26316 29284 27319 27118 27416 26791

# Frequency linegraph of total car robberies per day
library(ggplot2)

ggplot(WeekdayCounts,aes(x=Var1,y=Freq))+geom_line(aes(group=1))
```



As we see the days of the week are mixed and there is no chronological day order.

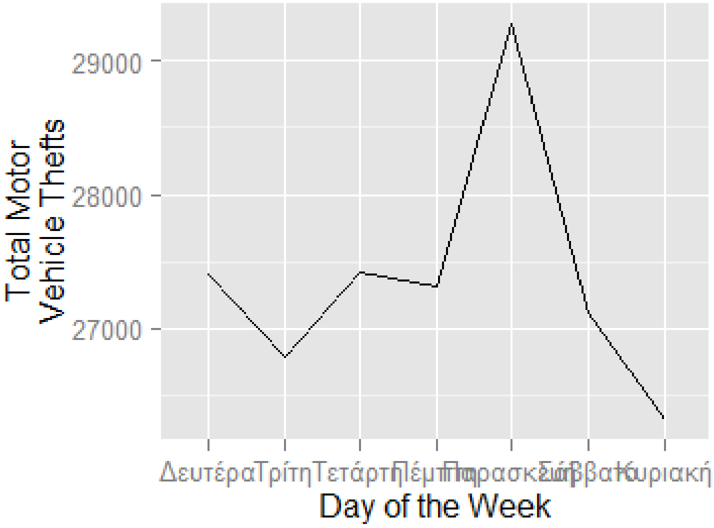
```
# Transforming Var1 into ordered factor
WeekdayCounts$Var1<-
factor(WeekdayCounts$Var1,ordered=TRUE,levels=c("Δευτέρα","Τρίτη","Τετάρτη","Πέμπτη",
                                                 "Παρασκευή","Σάββατο","Κυριακή"))
```

```
# Frequency linegraph of robberies per day with chronological order
ggplot(WeekdayCounts,aes(x=Var1, y=Freq))+geom_line(aes(group=1))
```



As we see from both of our graphs most car robberies take place on Friday, in contrast to Sunday.

```
# Changing both axis label name
ggplot(WeekdayCounts,aes(x=Var1, y=Freq))+geom_line(aes(group=1))+xlab("Day of the
Week")+ylab("Total Motor
Vehicle Thefts")
```



```
# Frequency table of robberries per Weekday - Hour
table(mvt$Weekday,mvt$Hour)
```

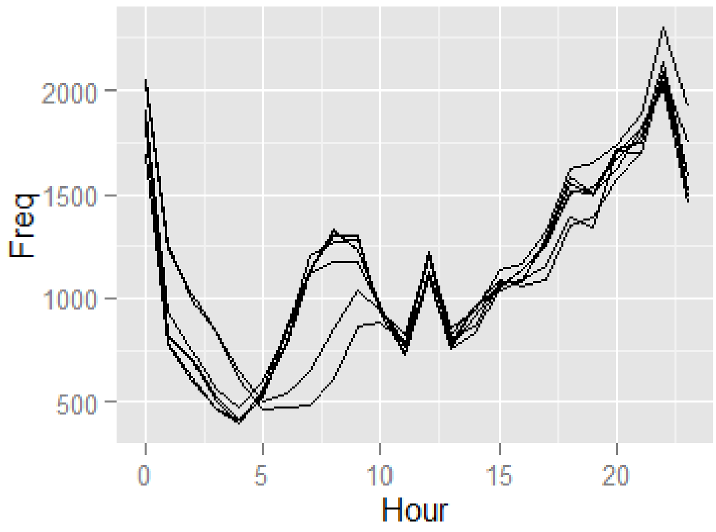
```
##            
##                0    1    2    3    4    5    6    7    8    9   10   11
##   Δευτέρα   1900  825  712  527  415  542  772 1123 1323 1235  971  737
##   Κυριακή   2028 1236 1019  838  607  461  478  483  615  864  884  787
##   Παρασκευή 1873  932  743  560  473  602  839 1203 1268 1286  938  822
##   Πέμπτη    1856  816  696  508  400  534  799 1135 1298 1301  932  731
##   Σάββατο   2050 1267  985  836  652  508  541  650  858 1039  946  789
##   Τετάρτη   1814  790  619  469  396  561  862 1140 1329 1237  947  763
##   Τρίτη     1691  777  603  464  414  520  845 1118 1175 1174  948  786
##            
##               12   13   14   15   16   17   18   19   20   21   22   23
##   Δευτέρα   1129  824  958 1059 1136 1252 1518 1503 1622 1815 2009 1490
##   Κυριακή   1192  789  959 1037 1083 1160 1389 1342 1706 1696 2079 1584
##   Παρασκευή 1207  857  937 1140 1165 1318 1623 1652 1736 1881 2308 1921
##   Πέμπτη    1093  752  831 1044 1131 1258 1510 1537 1668 1776 2134 1579
##   Σάββατο   1204  767  963 1086 1055 1084 1348 1390 1570 1702 2078 1750
##   Τετάρτη   1225  804  863 1075 1076 1289 1580 1507 1718 1748 2093 1511
##   Τρίτη     1108  762  908 1071 1090 1274 1553 1496 1696 1816 2044 1458
```

```
# Transforming Weekday-Hour frequency table into DayHourCounts dataframe
DayHourCounts<-as.data.frame(table(mvt$Weekday, mvt$Hour))

# DayHourCounts dataframe structure
str(DayHourCounts)
```

```
## 'data.frame':    168 obs. of  3 variables:
##  $ Var1: Factor w/ 7 levels "Δευτέρα","Κυριακή",..: 1 2 3 4 5 6 7 1 2 3 ...
##  $ Var2: Factor w/ 24 levels "0","1","2","3",..: 1 1 1 1 1 1 1 2 2 2 ...
##  $ Freq: int  1900 2028 1873 1856 2050 1814 1691 825 1236 932 ...
```

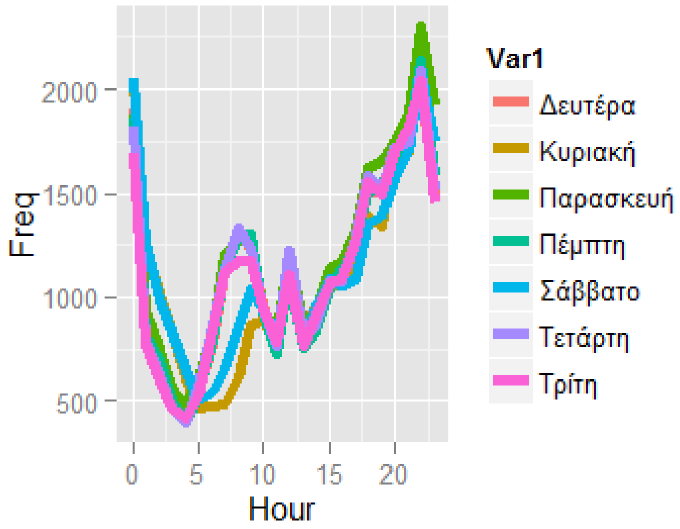
```
# Creating numerical variable Hour by transforming factor variable Var2
DayHourCounts$Hour<-as.numeric(as.character(DayHourCounts$Var2))
```

```
# Robbery linegraph per Day - Hour
ggplot(DayHourCounts,aes(x=Hour,y=Freq))+geom_line(aes(group=Var1))
```



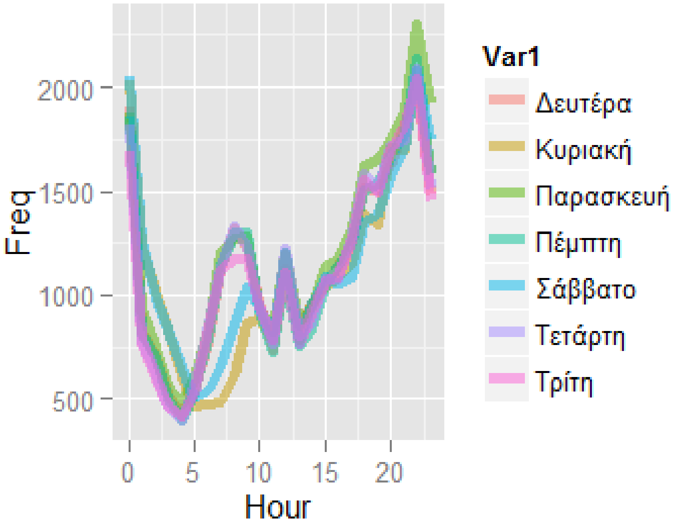
We can't understand much from this graph as all days are represented by the same colour.

```
# Robbery linegraph per Day - Hour with seperate colour for each day and thicker lines
ggplot(DayHourCounts,aes(x=Hour,y=Freq))+geom_line(aes(group=Var1,color=Var1),size=2)
```



As we see most of the car robberies take place during Sunday,Saturday,Monday midnight and Friday at around 10pm. On the other hand, there are less car robberies at around 4am to 5am every day and 5am to 7.5am during weekends.

```
# Same linegraph with more transparent lines
ggplot(DayHourCounts,aes(x=Hour,y=Freq))
+geom_line(aes(group=Var1,color=Var1),size=2,alpha=0.5)
```

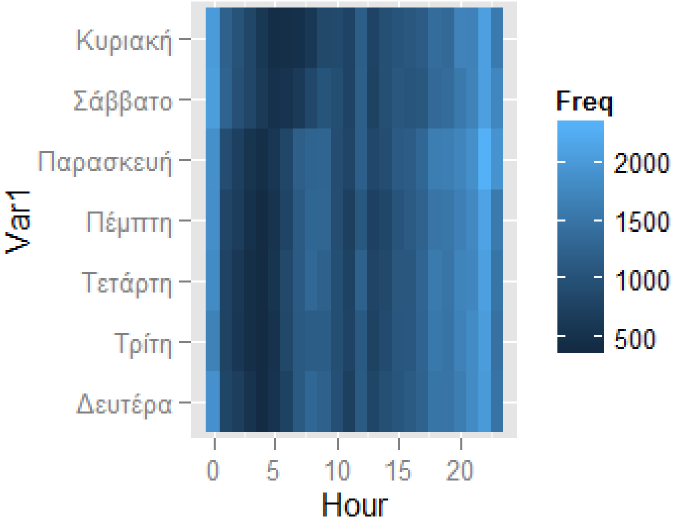


As we see this linegraph isn't of much help.

HEATMAPS

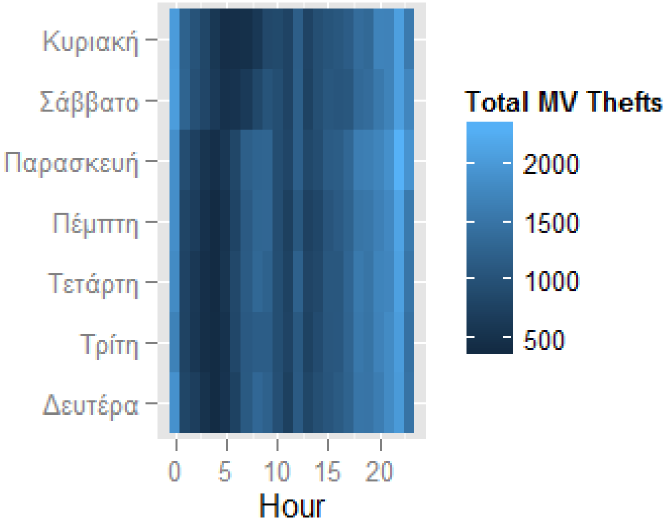
```
# Var1 with ordered levels by chronological day sequence
DayHourCounts$Var1<-factor(DayHourCounts$Var1,ordered=TRUE,
levels=c("Δευτέρα","Τρίτη","Τετάρτη","Πέμπτη","Παρασκευή","Σάββατο","Κυριακή"))
```

```
# Robberies frequency heatmap per Day- Hour
ggplot(DayHourCounts,aes(x=Hour, y=Var1))+geom_tile(aes(fill=Freq))
```

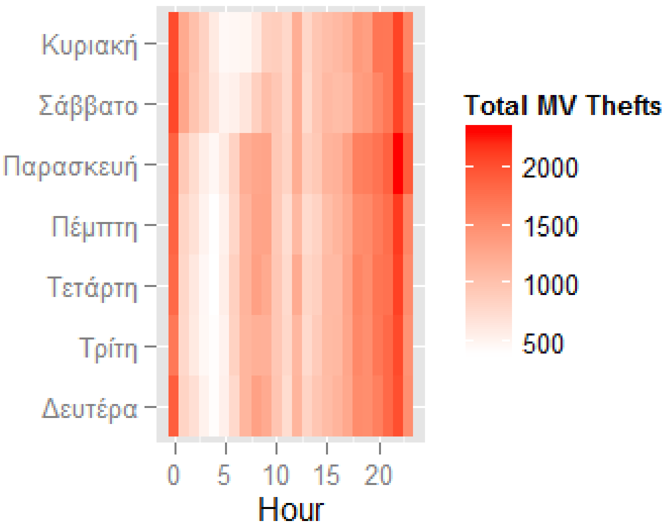


As we see most of the car robberies take place during Sunday,Saturday,Monday midnight and Friday at around 10pm. On the other hand, there are less car robberies at around 4am to 5am every day and 5am to 7.5am during weekends.

```
# Replacing the name of the Heatmap legend with "Total MV Thefts"
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())
```



```
# Changing the colours of the heatmap legend. White for low frequency and red for higher.
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())
```

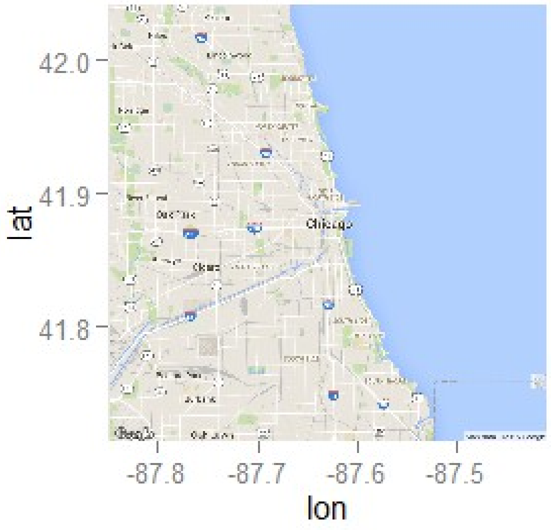


As we see Friday night is a high risk time for car robbery

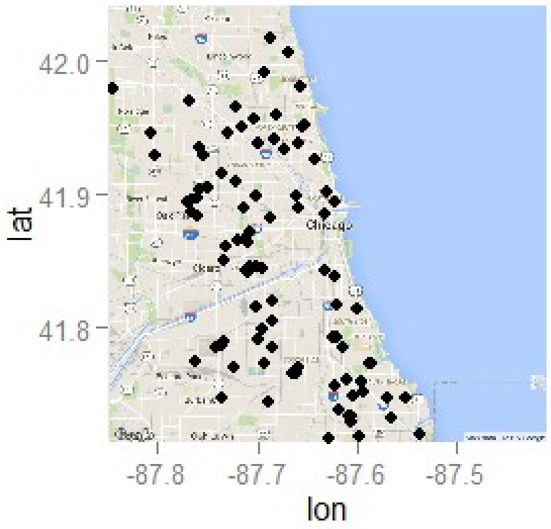
Geospatial HEATMAPS

```
# Loading Chicago map
library(maps)

## Warning: package 'maps' was built under R version 3.2.2

##
## # ATTENTION: maps v3.0 has an updated 'world' map.        #
## # Many country borders and names have changed since 1990. #
## # Type '?world' or 'news(package="maps")'. See README_v3. #

library(ggmap)

## Warning: package 'ggmap' was built under R version 3.2.2

chicago<-get_map(location="chicago",zoom = 11)

## Map from URL : http://maps.googleapis.com/maps/api/staticmap?
center=chicago&zoom=11&size=640x640&scale=2&maptype=terrain&language=en-
EN&sensor=false
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?
address=chicago&sensor=false
```

```
# View Chicago map
ggmap(chicago)
```

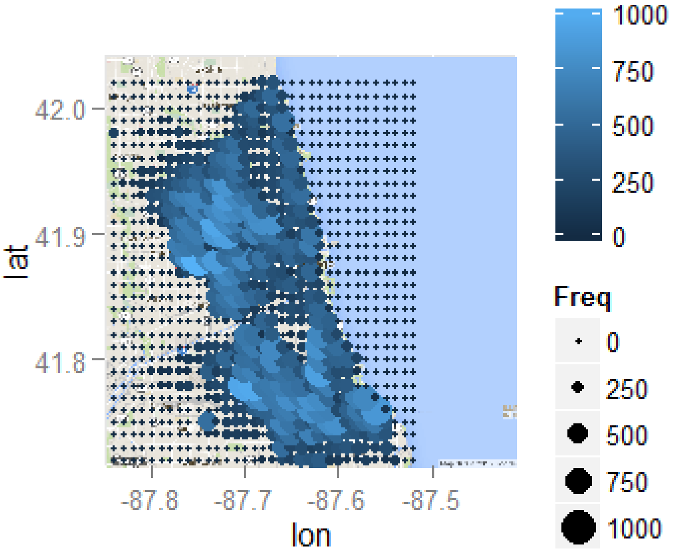


```
# Placing the first 100 robberies on the map of Chicago
ggmap(chicago)+geom_point(data=mvt[1:100,],aes(x=Longitude,y=Latitude))

## Warning: Removed 7 rows containing missing values (geom_point).
```



```
# Creating LatLonCounts dataframe: car robbery frequency dataframe, by rounding up to 2
decimals the Longitude & Latitude variables
LatLonCounts<-as.data.frame(table(round(mvt$Longitude,2),round(mvt$Latitude,2)))
```

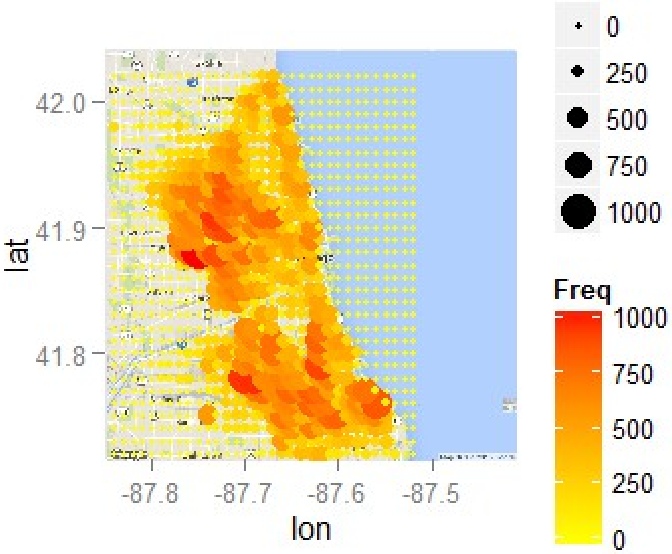
```
# LatLonCounts dataframe structure
str(LatLonCounts)

## 'data.frame':    1638 obs. of  3 variables:
##  $ Var1: Factor w/ 42 levels "-87.93","-87.92",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Var2: Factor w/ 39 levels "41.64","41.65",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Freq: int  0 0 0 0 0 0 0 0 0 0 ...

# Renaming Var1 & Var2 into Long & Lat respectively & turning them into numerical variables
LatLonCounts$Long<-as.numeric(as.character(LatLonCounts$Var1))
LatLonCounts$Lat<-as.numeric(as.character(LatLonCounts$Var2))

# Chicago car robbery map with size and colour of points depending on the frequency of
robberies
ggmap(chicago)+geom_point(data= LatLonCounts,aes(x=Long,y=Lat,color=Freq,size=Freq))

## Warning: Removed 615 rows containing missing values (geom_point).
```



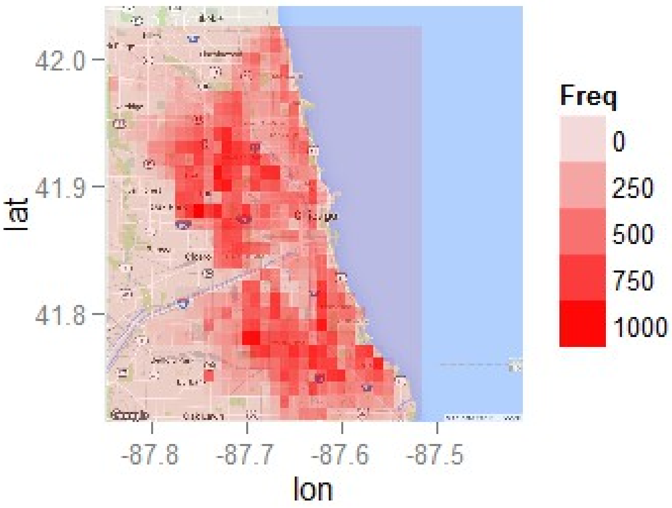
Brighter and Bigger dotpoints on map represent higher car robbery frequency.

```
# Setting yellow colour for low frequency and red colour for high
ggmap(chicago)+geom_point(data=LatLonCounts,aes(x=Long,y=Lat,color=Freq,size=Freq))+
  scale_colour_gradient(low="yellow",high="red")
```

```
## Warning: Removed 615 rows containing missing values (geom_point).
```



```
# Using the argument geom_tile so as to create a more typical heatmap
ggmap(chicago)+geom_tile(data=LatLonCounts,aes(x=Long,y=Lat,alpha=Freq),fill="red")
```



Murders Dataset

This dataset is taken from FBI databases and it's about murders that took place in every state of USA.

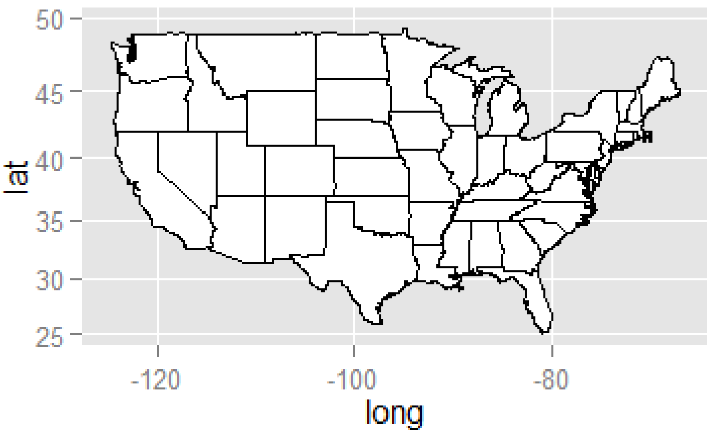
Geospatial HEATMAPS

```
# Installing packages
```

```
install.packages("maps")
install.packages("ggmap")
```

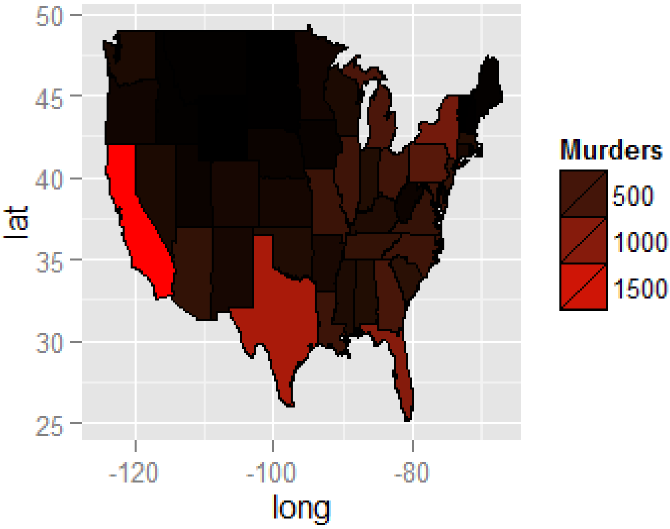
```
# Load murders dataset
murders <- read.csv("~/murders.csv")

# Checking dataset structure
str(murders)

## 'data.frame':    51 obs. of  6 variables:
##  $ State            : Factor w/ 51 levels "Alabama","Alaska",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Population       : int  4779736 710231 6392017 2915918 37253956 5029196 3574097
897934 601723 19687653 ...
##  $ PopulationDensity: num  94.65 1.26 57.05 56.43 244.2 ...
##  $ Murders          : int  199 31 352 130 1811 117 131 48 131 987 ...
##  $ GunMurders       : int  135 19 232 93 1257 65 97 38 99 669 ...
##  $ GunOwnership     : num  0.517 0.578 0.311 0.553 0.213 0.347 0.167 0.255 0.036 0.245 ...

# Creating USA dataset
library(maps)

## Warning: package 'maps' was built under R version 3.2.2

##
##  # ATTENTION: maps v3.0 has an updated 'world' map.        #
##  # Many country borders and names have changed since 1990. #
##  # Type '?world' or 'news(package="maps")'. See README_v3. #

library(ggmap)

## Warning: package 'ggmap' was built under R version 3.2.2
## Loading required package: ggplot2

statesMap<-map_data("state")
```

```
# Checking USA dataset
str(statesMap)

## 'data.frame':    15537 obs. of  6 variables:
##  $ long     : num  -87.5 -87.5 -87.5 -87.5 -87.6 ...
##  $ lat      : num  30.4 30.4 30.4 30.3 30.3 ...
##  $ group    : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ order    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ region   : chr  "alabama" "alabama" "alabama" "alabama" ...
##  $ subregion: chr  NA NA NA NA ...

# Creating USA map
ggplot(statesMap,aes(x=long,y=lat,group=group))+geom_polygon(fill="white",color="black")+
  coord_map("mercator")
```



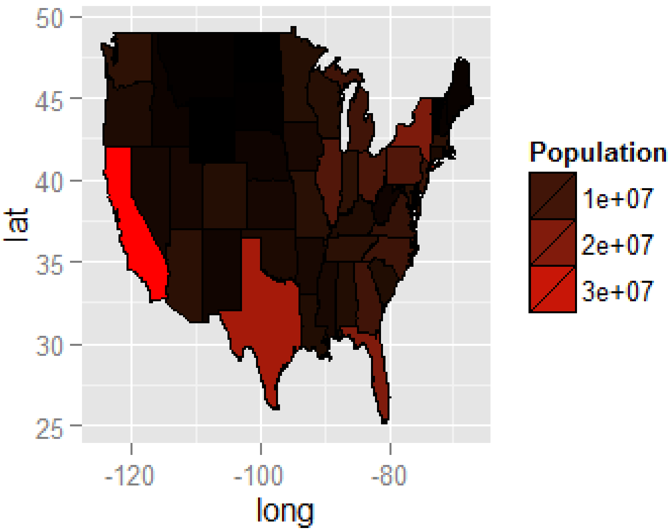
```
# Adding variable region to murder dataframe so as to merge it with statesMap dataframe
murders$region<-tolower(murders$State)

# Merging by common variable region the two datasets: murders & statesMap
murderMap<-merge(statesMap,murders,by="region")
```

```
# Checking murderMap
str(murderMap)

## 'data.frame':    15537 obs. of  12 variables:
##  $ region          : chr  "alabama" "alabama" "alabama" "alabama" ...
##  $ long            : num  -87.5 -87.5 -87.5 -87.5 -87.6 ...
##  $ lat             : num  30.4 30.4 30.4 30.3 30.3 ...
##  $ group           : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ order           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ subregion       : chr  NA NA NA NA ...
##  $ State           : Factor w/ 51 levels "Alabama","Alaska",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Population      : int  4779736 4779736 4779736 4779736 4779736 4779736 4779736
4779736 4779736 4779736 ...
##  $ PopulationDensity: num  94.7 94.7 94.7 94.7 94.7 ...
##  $ Murders         : int  199 199 199 199 199 199 199 199 199 199 ...
##  $ GunMurders      : int  135 135 135 135 135 135 135 135 135 135 ...
##  $ GunOwnership    : num  0.517 0.517 0.517 0.517 0.517 0.517 0.517 0.517 0.517 0.517 ...

# USA heatmap with Murders frequency per state
ggplot(murderMap,aes(x=long,y=lat,group=group,fill=Murders))
+geom_polygon(colour="black")+
 scale_fill_gradient(low="black",high="red",guide = "legend")
```



As we see in the map, most murders take place in California and Texas.

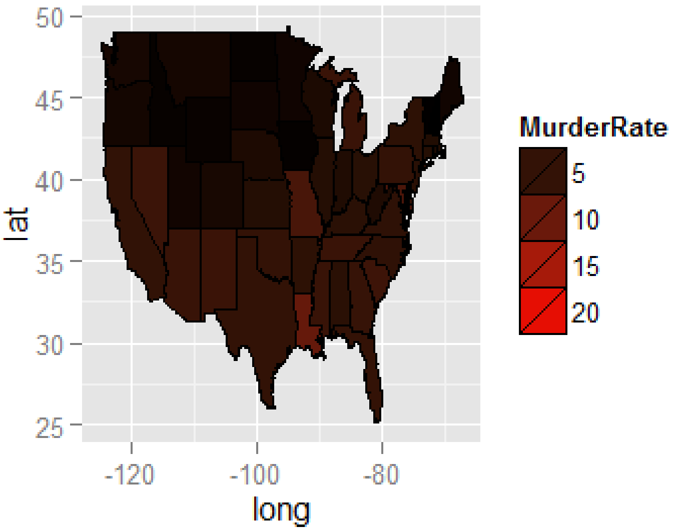
```
# USA heatmap per state population
ggplot(murderMap, aes(x = long, y = lat, group = group, fill = Population)) +
geom_polygon(colour = "black") +scale_fill_gradient(low = "black", high = "red", guide =
"legend")
```



The murder and population heatmaps are almost similar. California and Texas are the most populated states. That's why we will create a heatmap representing the murder rate per population, instead of just the number of murders.

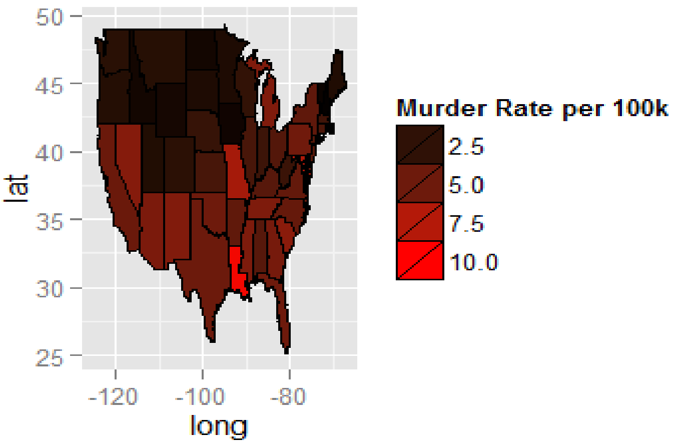
```
# Creating variable MurderRate
murderMap$MurderRate<-murderMap$Murders/murderMap$Population*100000
```

```
# ggplot for USA map with MurderRate frequency per region
ggplot(murderMap,aes(x=long,y=lat,group=group,fill=MurderRate))
+geom_polygon(colour="black")+scale_fill_gradient(low="black",high="red",guide = "legend")
```



Our heatmap shows low murder-rate in every state but this is wrong as Washington DC is an outlier with an extremely high murder rate. That's why we will create a heatmap which contains the states with murder rate =< 10.

```
# ggplot for USA map with MurderRate>10 frequency per region
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))
```



Councluding, Lousiana state has the highest murder rate.

Intlall Dataset

This dataset is about MIT international students and where they come from.

Geospatial HEATMAPS

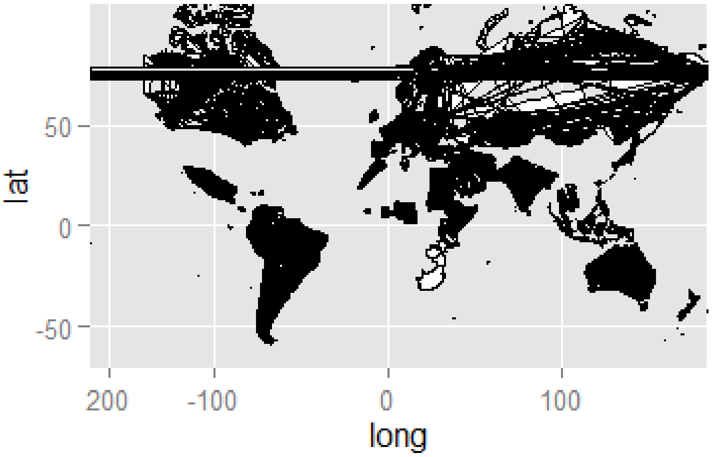
```
# Installing packages
```

```
install.packages("ggplot2")
install.packages("ggmap")
```

```
# Loading libraries
library(ggplot2)
library(ggmap)

## Warning: package 'ggmap' was built under R version 3.2.2

# Loading Intlall dataframe
intlall<-read.csv("~/intlall.csv",stringsAsFactors=FALSE)

# Checking the first 6 rows of intlall dataframe
head(intlall)

##          Citizenship      UG  G   SpecialUG  SpecialG  ExhangeVisiting  Total
## 1           Albania        3  1       0          0           0              4
## 2 Antigua and Barbuda     NA NA      NA          1          NA              1
## 3         Argentina       NA 19      NA         NA          NA             19
## 4           Armenia        3  2      NA         NA          NA              5
## 5         Australia        6 32      NA         NA           1             39
## 6           Austria       NA 11      NA         NA           5             16

# Replacing NAs with 0
intlall[is.na(intlall)]<-0

# re-Checking the first 6 rows of the dataframe
head(intlall)

##          Citizenship     UG  G  SpecialUG  SpecialG  ExhangeVisiting   Total
## 1           Albania       3  1      0          0            0            4
## 2 Antigua and Barbuda     0  0      0          1            0            1
## 3         Argentina       0 19      0          0            0           19
## 4           Armenia       3  2      0          0            0            5
## 5         Australia       6 32      0          0            1           39
## 6           Austria       0 11      0          0            5           16
```

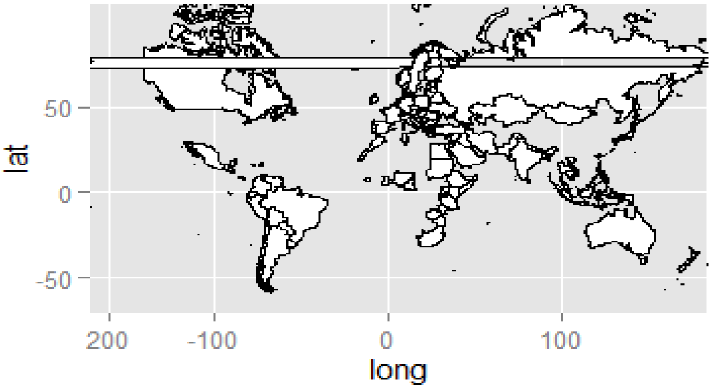
```
# Loading world-map dataframe
world_map<-map_data("world")

# Checking world-map structure
str(world_map)

## 'data.frame':    101913 obs. of  6 variables:
##  $ long     : num  -69.9 -69.9 -69.9 -70 -70.1 ...
##  $ lat      : num  12.5 12.4 12.4 12.5 12.5 ...
##  $ group    : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ order    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ region   : chr  "Aruba" "Aruba" "Aruba" "Aruba" ...
##  $ subregion: chr  NA NA NA NA ...

# Merging intlall into world_map dataframe
world_map<-merge(world_map,intlall,by.x ="region",by.y="Citizenship")

# re-Checking world_map structure
str(world_map)

## 'data.frame':    65153 obs. of  12 variables:
##  $ region        : chr  "Albania" "Albania" "Albania" "Albania" ...
##  $ long          : num  20.5 19.4 20.6 19.4 19.4 ...
##  $ lat           : num  41.3 42.3 40.1 42.1 42.3 ...
##  $ group         : num  6 6 6 6 6 6 6 6 6 6 ...
##  $ order         : int  789 871 813 864 873 818 823 822 874 869 ...
##  $ subregion     : chr  NA NA NA NA ...
##  $ UG            : num  3 3 3 3 3 3 3 3 3 3 ...
##  $ G             : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ SpecialUG     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SpecialG      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ ExhangeVisiting: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Total         : int  4 4 4 4 4 4 4 4 4 4 ...
```

```
# Creating world_map map
ggplot(world_map,aes(x=long,y=lat,group=group))+geom_polygon(fill="white",color="black")+
 coord_map("mercator")
```



This is a wrong world map graph with no meaning due to wrong re-arrangement of observations, which happened because of the merging of the two dataframes.

```
# Re-ordering observations correctly
world_map<-world_map[order(world_map$group,world_map$order),]

# Re-constructing world_map map
ggplot(world_map,aes(x=long,y=lat,group=group))+geom_polygon(fill="white",color="black")+
 coord_map("mercator")
```



As we see there a few missing countries (ie.countries of Africa) This happens because they have different name in our two initial dataframes.

```
# Constructing a student frequency table per country from the intlall dataframe to check which countries
# have different names from the world_map dataframe
table(intlall$Citizenship)

##
##                   Albania         Antigua and Barbuda
##                         1                           1
##                 Argentina                     Armenia
##                         1                           1
##                 Australia                     Austria
##                         1                           1
##                   Bahrain                  Bangladesh
##                         1                           1
##                   Belarus                     Belgium
##                         1                           1
##                   Bolivia          Bosnia-Hercegovina
##                         1                           1
##                    Brazil                    Bulgaria
##                         1                           1
##                  Cambodia                    Cameroon
##                         1                           1
##                    Canada                       Chile
##                         1                           1
## China (People's Republic Of)                Colombia
##                         1                           1
##                Costa Rica               Cote d'Ivoire
##                         1                           1
##                   Croatia                      Cyprus
##                         1                           1
##            Czech Republic                     Denmark
##                         1                           1
##                   Ecuador                       Egypt
##                         1                           1
##               El Salvador                     Estonia
##                         1                           1
##                  Ethiopia                     Finland
##                         1                           1
##                    France                     Georgia
##                         1                           1
##                   Germany                       Ghana
##                         1                           1
##                    Greece                   Guatemala
##                         1                           1
##            Haiti       Hong Kong
##              1             1
##          Hungary        Iceland
##              1             1
##            India       Indonesia
##              1             1
##             Iran          Iraq
##              1             1
##          Ireland        Israel
##              1             1
##            Italy        Jamaica
##              1             1
##            Japan        Jordan
##              1             1
##       Kazakhstan         Kenya
##              1             1
##     Korea, South        Kuwait
##              1             1
##           Latvia       Lebanon
##              1             1
##        Lithuania     Macedonia
##              1             1
##         Malaysia     Mauritius
##              1             1
##           Mexico       Moldova
##              1             1
##         Mongolia    Montenegro
##              1             1
##          Morocco         Nepal
##              1             1
##      Netherlands   New Zealand
##              1             1
##          Nigeria        Norway
##              1             1
##         Pakistan      Paraguay
##              1             1
##             Peru   Philippines
##              1             1
##           Poland      Portugal
##              1             1
##            Qatar       Romania
##              1             1
##           Russia        Rwanda
##              1             1
##                   Saudi Arabia                         Serbia 
##                              1                              1 
##                   Sierra Leone                      Singapore 
##                              1                              1 
##                       Slovakia                        Somalia 
##                              1                              1 
##                   South Africa                          Spain 
##                              1                              1 
##                      Sri Lanka                      St. Lucia 
##                              1                              1 
## St. Vincent & The Grenadines                            Sudan 
##                              1                              1 
##                         Sweden                    Switzerland 
##                              1                              1 
##                          Syria                         Taiwan 
##                              1                              1 
##                       Tanzania                       Thailand 
##                              1                              1 
##              Trinidad & Tobago                        Tunisia 
##                              1                              1 
##                         Turkey                         Uganda 
##                              1                              1 
##                        Ukraine           United Arab Emirates 
##                              1                              1 
##                 United Kingdom                        Unknown 
##                              1                              1 
##                        Uruguay                      Venezuela 
##                              1                              1 
##                        Vietnam                      West Bank 
##                              1                              1 
##                         Zambia                       Zimbabwe 
##                              1                              1
```

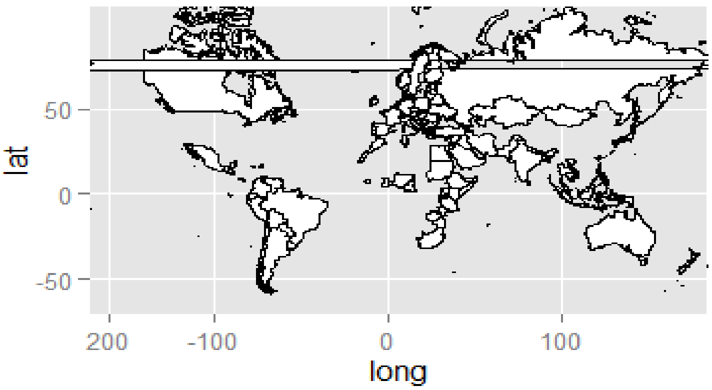
As we see China has a different name in our two datasets.

```
# Re-naming intlall's "China(People's Republic of)" into "China" to match with world_map dataframe
intlall$Citizenship[intlall$Citizenship=="China(People's Republic Of)"]<-"China"

# Re-checking intlall's country names
table(intlall$Citizenship)

## 
##                        Albania            Antigua and Barbuda 
##                              1                              1
##                    Argentina                      Armenia
##                            1                            1
##                    Australia                      Austria
##                            1                            1
##                      Bahrain                   Bangladesh
##                            1                            1
##                      Belarus                      Belgium
##                            1                            1
##                      Bolivia           Bosnia-Hercegovina
##                            1                            1
##                       Brazil                     Bulgaria
##                            1                            1
##                     Cambodia                     Cameroon
##                            1                            1
##                       Canada                        Chile
##                            1                            1
## China (People's Republic Of)                     Colombia
##                            1                            1
##                   Costa Rica                Cote d'Ivoire
##                            1                            1
##                      Croatia                       Cyprus
##                            1                            1
##               Czech Republic                      Denmark
##                            1                            1
##                      Ecuador                        Egypt
##                            1                            1
##                  El Salvador                      Estonia
##                            1                            1
##                     Ethiopia                      Finland
##                            1                            1
##                       France                      Georgia
##                            1                            1
##                      Germany                        Ghana
##                            1                            1
##                       Greece                    Guatemala
##                            1                            1
##                        Haiti                    Hong Kong
##                            1                            1
##                      Hungary                      Iceland
##                            1                            1
##                        India                    Indonesia
##                            1                            1
##                         Iran                         Iraq
##                            1                            1
##        Ireland        Israel
##              1             1
##          Italy       Jamaica
##              1             1
##          Japan        Jordan
##              1             1
##     Kazakhstan         Kenya
##              1             1
##   Korea, South        Kuwait
##              1             1
##         Latvia       Lebanon
##              1             1
##      Lithuania     Macedonia
##              1             1
##       Malaysia     Mauritius
##              1             1
##         Mexico       Moldova
##              1             1
##       Mongolia    Montenegro
##              1             1
##        Morocco         Nepal
##              1             1
##    Netherlands   New Zealand
##              1             1
##        Nigeria        Norway
##              1             1
##       Pakistan      Paraguay
##              1             1
##           Peru   Philippines
##              1             1
##         Poland      Portugal
##              1             1
##          Qatar       Romania
##              1             1
##         Russia        Rwanda
##              1             1
##   Saudi Arabia        Serbia
##              1             1
##   Sierra Leone     Singapore
##              1             1
##       Slovakia       Somalia
##              1             1
##   South Africa         Spain
##              1             1
##                Sri Lanka                St. Lucia
##                        1                        1

## St. Vincent & The Grenadines                    Sudan
##                        1                        1
##                   Sweden              Switzerland
##                        1                        1
##                    Syria                   Taiwan
##                        1                        1
##                 Tanzania                 Thailand
##                        1                        1
##        Trinidad & Tobago                  Tunisia
##                        1                        1
##                   Turkey                   Uganda
##                        1                        1
##                  Ukraine     United Arab Emirates
##                        1                        1
##           United Kingdom                  Unknown
##                        1                        1
##                  Uruguay                Venezuela
##                        1                        1
##                  Vietnam                West Bank
##                        1                        1
##                   Zambia                 Zimbabwe
##                        1                        1
```

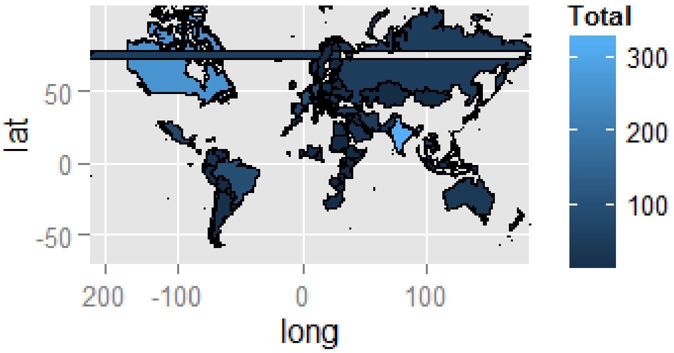
```
# Re-merging both dataframes
world_map<-merge(map_data("world"),intlall,by.x="region",by.y="Citizenship")
```

```
# Re-ordering observations correctly
world_map<-world_map[order(world_map$group,world_map$order),]
```

```
# re-Building map
ggplot(world_map,aes(x=long,y=lat,group=group))+geom_polygon(fill="white",color="black")+
 coord_map("mercator")
```

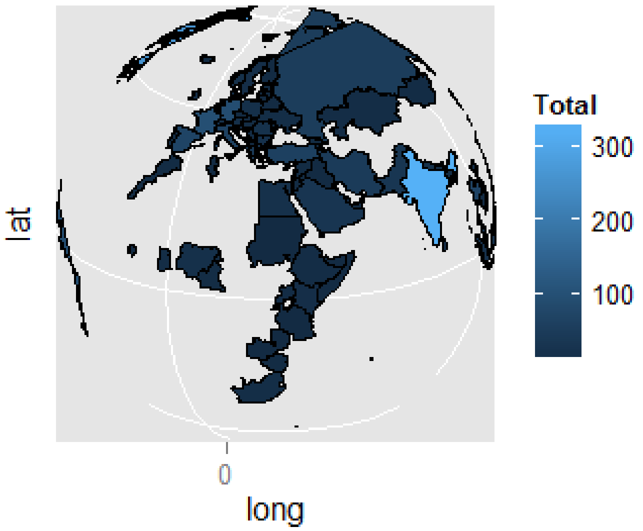


```
# Re-building world_map map by filling it with the number of total students per country
ggplot(world_map,aes(x=long,y=lat,group=group))+geom_polygon(aes(fill=Total),color="black")
+coord_map("mercator")
```

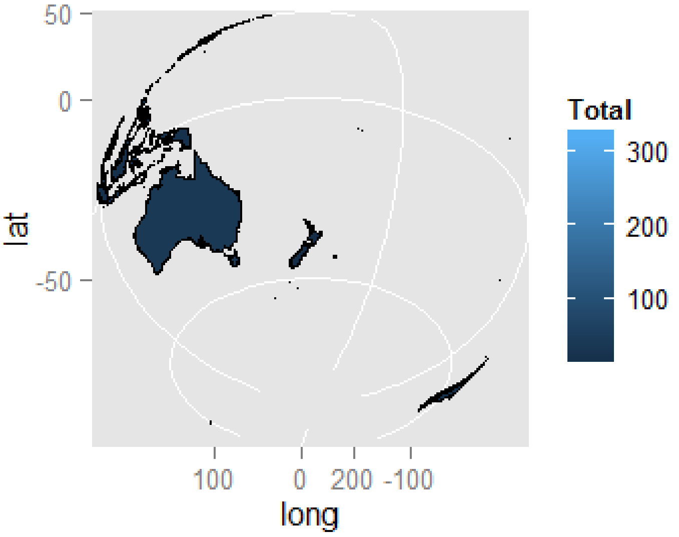


As we see more students come from America and India.

```
# re-building world_map map by using "orthographic" view
ggplot(world_map,aes(x=long,y=lat,group=group))+geom_polygon(aes(fill=Total),color="black")
+coord_map("ortho",orientation=c(20, 30, 0))
```



```
# View of world-map from another side
ggplot(world_map,aes(x=long,y=lat,group=group))+geom_polygon(aes(fill=Total),color="black")
+coord_map("ortho", orientation=c(-37, 175, 0))
```


50
0
-50
lat
100
0
200
-100
long
Total
300
200
100