Final INSH

Sanjan Gorlla

Dec/12/22

Final Exam

```r
# loading required libraries
library(readr) # to read data
library(dplyr) # to tidy data
```

```
##
## 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
```

```r
library(GGally) # to make correlation matrix
```

```
## Loading required package: ggplot2

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```r
library(lmtest) # for Breusch-Pagan/heteroscedasticity test
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(car) # for multicollinearity test
```

```
## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode
```

```r
library(corrplot) # for correlation
```

```
## corrplot 0.92 loaded
```

```r
library(ggplot2) # plots
library(skimr) # for skim() function
library(knitr) # for kable() function
library(psych) # for pairs.plot()
```

```
##
## Attaching package: 'psych'

## The following object is masked from 'package:car':
##
##     logit

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```r
library(reshape) # for melt()
```

```
##
## Attaching package: 'reshape'

## The following object is masked from 'package:dplyr':
##
##     rename
```

```r
library(scales) # for percent()
```

```
##
## Attaching package: 'scales'

## The following objects are masked from 'package:psych':
##
##     alpha, rescale
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```r
library(leaps) # To check all possible regression
```

Problem 1: You roll five six-sided dice. Write a script in R to calculate the probability of getting between 15 and 20 (inclusive) as the total amount of your roll (ie, the sum when you add up what is showing on all five dice). Exact solutions are preferable but approximate solutions are ok as long as they are precise (10pts)

```r
## Defining the function which does calculation
side_dice <- function(n){
  dice <- expand.grid(1:n, 1:n, 1:n, 1:n, 1:n) # five 'n' sided dice
  # the probability of getting between 15 and 20 (inclusive)
  # as the total amount of your roll (ie, the sum when you add up what is showing on all five dice)
  return (mean(15 <= rowSums(dice) & rowSums(dice) <=20))
}

## Implementation of  the function
# Therefore the probability of getting between
# 15 and 20 (inclusive) as the total amount of your roll
# (ie, the sum when you add up what is showing on all five dice) is 0.5570988
side_dice(6)
```

```
## [1] 0.5570988
```

```r
# "The probability of getting between 15 and 20 (inclusive)
# as the total amount of five six-sided dice is:56%
```

```r
set.seed(1)
# create some simulated data
# x is a random normal variable
x <- rnorm(n = 100, mean = 0, sd = 1)
# epsilon is also a random normal error with mean 0 and sd 1
e <- rnorm(n = 100, mean = 0, sd = 1)
# The final equation
y <- 0.1 + 2 * x + e # vectorize
```

Problem 2: Create a simulated dataset of 100 observations, where x is a random normal variable with mean 0 and standard deviation 1, and $= 0.1 + 2 * +$, where epsilon is also a random normal error with mean 0 and sd 1. (10pts)

Problem 2a: Perform a t test for whether the mean of Y equals the mean of X using R.

$$H0 : 1 = 2 (The\, Y\, and\, X\, means\, are\, equal)$$

$$H1 : 1 \neq 2 (The\, Y\, and\, X\, means\, are\, not\, equal)$$

The null hypothesis (H0) states that there is no significant difference between the means of the two groups. The alternative hypothesis (H1) states that there is a significant difference between the two population means, and that this difference is unlikely to be caused by sampling error or chance.

```
set.seed(1)
t.test(y, x, paired  = T)
```

```
##
##  Paired t-test
##
## data:  y and x
## t = 1.3035, df = 99, p-value = 0.1954
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.08934368  0.43150226
## sample estimates:
## mean of the differences
##               0.1710793
```

Here I have used a paired t-test (also known as a dependent or correlated t-test) is a statistical test that compares the averages/means and standard deviations of two related groups to determine if there is a significant difference between the two groups.

since, Y value is dependent on X, paired t-test is appropriate method, as there is a relationship between X and Y

The p-value is larger than significance level = 0.05, we cannot conclude that a significant difference exists.The results showed that the probability value is greater than 0.05. Higher the P-value, Based on this result, we shall reject the alternate hypothesis of no difference. It means that there is no significant difference between the means of the two groups. Therefore, The mean differences of Y and and X is 0.1710793. i.e, there is no significant difference between the means of the Y and mean of the X

Problem 2b: Now perform this test by hand using just the first 5 observations. Please write out all your steps carefully. To determine whether or not the mean of Y equals the mean of X, we will perform a paired samples t-test at significance level = 0.05

T-test for dependent variables is simple

```
set.seed(1)
# The first 5 observations
x <- x[1:5]
y <- y[1:5]
# mean and s.d of x
x_mean <- mean(x)
x_mean
```

```
## [1] 0.1292699
```

```
x_sd <- sd(x)
# mean and s.d of y
y_mean <- mean(y)
y_mean
```

```
## [1] -0.03860587
```

```r
y_sd <- sd(y)
# s.d of x and y
xy_sd <- sqrt(x_sd/length(x) + y_sd/length(y))
```

```r
### Step 1: Calculate the differences
xy_diff <- x-y
### mean of the difference
xy_mean <-mean(xy_diff)
xy_mean
```

```
## [1] 0.1678758
```

```r
## sd of the difference
xy_sd <- sd(xy_diff)
xy_sd
```

```
## [1] 1.3672
```

```r
### sample size
n <- length(x)
n
```

```
## [1] 5
```

```r
### Step 2: Define the hypotheses.
# We will perform the paired samples t-test with the following hypotheses:
# H0:  1 =  2 (the two population(x and y) means are equal)
# H1:  1   2 (the two population( x and y) means are not equal)
```

```r
### Step 3: Calculate the test statistic t
```

```r
# t-statistics
t_stats <- xy_mean/xy_sd/ sqrt(n)
t_stats
```

```
## [1] 0.05491245
```

$$Test\,Statistic(dependent\,sample) = \frac{x_{diff}}{sd_{diff}/\sqrt{n}}$$

$$Test\,Statistic(dependent\,sample) = \frac{0.1678758}{1.3672/\sqrt{5}} = 0.2745622$$

$$df = 5 - 1 = 4$$

```r
### Step 4: Calculate the p-value of the test statistic t.
#   = 0.05
df <- 4
qt(0.95, 4) # Critical t value for p > 0.05
```

```
## [1] 2.131847
```

```r
2*pt(0.054,4,lower.tail=F)
```

```
## [1] 0.9595246
```

```r
# 0.959 is greater than 0.05 therefore, we accept the null hypothesis of the equal means
```

$$Critical\, t\, value\, for\, p > 0.05$$

Step 5: Draw a conclusion.

Because the calculated t value (0.09273116) is less than our critical t value 2.13 (and our p-value is subsequently greater than 0.05), we reject the alternate hypothesis and conclude that the mean of X and Y are same.

Therefore, there is no significant difference between the means of the Y and mean of the X of first five observations

```r
min_obv <- function(x){
  # mean of first five obseravtions
  sample_mean<- mean(x)
  # sd of first five obseravtions
  sample_sd <- sd(x)
  #   = 0.01 confidence level
  given_alpha <- 0.01
  # let's assume true mean   of the population = 0
  # for loop that calculates
  # minimum total number of additional observations
  # you would need to be able to conclude that the true mean   of the population is different from 0

  for (i in 5:200000) {
    # calculating the standard error
    SE<-sample_sd/sqrt(i)
    # The main condition that the true mean   of the population is different from 0
    CI_level<-sample_mean + c(qt(given_alpha/2,i-1), qt(1-(given_alpha/2),i-1))*SE
    #This for loop ends with the main condition of CI is different from 0 (decrease)
    if (CI_level[2]< 0) # When the population mean differs from zero
    break
  }
  i
}
```

```r
x <-y[1:5]
min_obv(x)
```

Problem 2c: Assuming the mean and sd of the sample that you calculated from the first five observations would not change, what is the minimum total number of additional observations you would need to be able to conclude that the true mean   of the population is different from 0 at the   = 0.01 confidence level?

```
## [1] 23889
```

```
# The minimum total number of additional observations
# would need to be able to conclude that the true mean   of the population
# is different from 0 at the   = 0.01 confidence level is 23889
```

```
set.seed(1)
# create some simulated data
# x is a random normal variable
x <- rnorm(n = 23894, mean = 0, sd = 1)
# epsilon is also a random normal error with mean 0 and sd 1
e <- rnorm(n = 23894, mean = 0, sd = 1)
# The final equation
y <- 0.1 + 2 * x + e # vectorize
```

```
t.test(y, mu=0, conf.level = 0.01)
```

```
##
##  One Sample t-test
##
## data:  y
## t = 6.5542, df = 23893, p-value = 5.71e-11
## alternative hypothesis: true mean is not equal to 0
## 1 percent confidence interval:
##  0.09480745 0.09517075
## sample estimates:
## mean of x
## 0.0949891
```

```
# Therefore, p-value is less than 0.05.
# Hence, we accept the alternative hypothesis: true mean is not equal to 0.
```

```


set.seed(1)
# create some simulated data
# x is a random normal variable
x <- rnorm(n = 100, mean = 0, sd = 1)
# epsilon is also a random normal error with mean 0 and sd 1
e <- rnorm(n = 100, mean = 0, sd = 1)
# The final equation
y <- 0.1 + 0.2 * x + e # vectorize
```

Problem 3. Generate a new 100-observation dataset as before, except now $= 0.1 + 0.2 *$ $+$ (10pts)

Problem 3a. Regress y on x using R, and report the results. Discuss the coefficient on x and its standard error, and present its 95% CI.

```
set.seed(1)
# Regress y on x
y_on_x <- lm( y ~ x)
summary(y_on_x)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8768 -0.6138 -0.1395  0.5394  2.3462
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.06231    0.09699   0.642   0.5221
## x            0.19894    0.10773   1.847   0.0678 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9628 on 98 degrees of freedom
## Multiple R-squared:  0.03363,    Adjusted R-squared:  0.02377
## F-statistic:  3.41 on 1 and 98 DF,  p-value: 0.06781
```

RESULTS

Finally, our model equation can be written as follow:

$$y = 0.06231 + 0.19894 * X$$

```
# 95% CI
confint(y_on_x, level=0.95)
```

```
##                   2.5 %    97.5 %
## (Intercept) -0.13016074 0.2547755
## x           -0.01484117 0.4127204
```

(-0.01484117 0.4127204) conf_interval of x

Residuals : This model seems like a well-fitting model, the residuals should be normally distributed around 0. The residuals here look roughly symmetrical. Good..

Discuss the coefficient on x and its standard error, and present its 95% CI

Coefficients

1. The coefficient Estimate contains two rows; the first one is the intercept. Therefore, it takes an average coefficient of x to be 0.19894

2. The coefficient Standard Error measures the average amount that the coefficient estimates vary from the actual average value of our response variable. We'd ideally want a lower number relative to its coefficients. In this model the standard error of the estimate = 0.10773,
3. The t-value (which is just the estimate divided by its SE)= 1.847 The coefficient t-value is a measure of how many standard deviations our coefficient estimate is far away from 0. We want it to be far away from zero as this would indicate we could reject the null hypothesis - that is, we could declare a relationship between y and x. In our example, the t-statistic values are far away from zero and are large relative to the standard error, which could indicate a relationship exists. In general, t-values are also used to compute p-values.
4. The p-value associated with that t-value (the probability of getting a t-value more extreme than the observed t-value if the null hypothesis were true) = 0.0678

The Pr(>t) acronym found in the model output relates to the probability of observing any value equal or larger than t.A small p-value indicates that it is unlikely we will observe a relationship between the y and x variables due to chance. Typically, a p-value of 5% or less is a good cut-off point.

In our model example, the p-values is not significant. Note the 'signif. Codes' associated to each estimate. stars (or asterisks) represent a highly significant p-value. Finally, there is a no significance code for each coefficient using asterisks to indicate how small the p-value is. Consequently, a small p-value for the intercept and the slope indicates that we cannot reject the null hypothesis which allows us to conclude that there no significant relationship between y and x.

problem 3b. Use R to calculate the p-value on the coefficient on x from the t statistic for that coefficient as shown in the regression in 3a, and confirm that your p-value matches what is shown in 3a. What does this p-value represent (be very precise in your language here)?

```
2*pt(1.847,98, lower.tail = F) # p-value on the coefficient on x
```

```
## [1] 0.06776439
```

```
# Therefore the p-value 0.0677 matches the p-value that is shown in 3a on the coe icient on x
```

Coefficient - Pr(>t)

Therefore, The p value matches the p value in 3a. The coefficient is not significant, we can infer. The impact of x and y is inconsequential. Therefore, we must enhance this model.

problem 3c. Use R to calculate the p-value associated with the F statistic reported in your regression output. What does this test and its p-value indicate?

```
# Calculation of the p-value associated with F statistic
# Taking the F statistic reported the regression model =  3.41
#  3.41 on 1 is reported
pf( 3.41,1,(length(x)-1-1), lower.tail = F)
```

```
## [1] 0.06782021
```

```
# Therefore the calculated p-value 0.0169
# matches the p-value that is associated with the F statistic reported in the above regression output
```

F-statistic is a good indicator of whether there is a relationship between our predictor and the response variables. The further the F-statistic is from 1 the better it is. However, how much larger the F-statistic needs to be depends on both the number of data points and the number of predictors. Generally, when the number of data points is large, an F-statistic that is only a little bit larger than 1 is already sufficient to reject the null hypothesis (H0 : There is no relationship between y and x). The reverse is true as if the number of data points is small, a large F-statistic is required to be able to ascertain that there may be a relationship between predictor and response variables. In our example the F-statistic is 3.41 which is larger than 1 given the size of our data

The F value is used to calculate the P value - whether or not the F value is significant or not depends on the degrees of freedom. Whether or not it is above or below 0.05 does not directly indicate significance.

Here, the value - 0.06782021 is MORE than the alpha level 0.01, your results are NOT significant and we ACCEPT the null hypothesis, and the model is NOT FIT which means change in x does not have effect on y.

problem 3d. Using just the first five observations from your simulated dataset, calculate by hand the coefficient on x, its standard error, and the adjusted R2. Be sure to show your work, but you may use R for the simple math.

```
set.seed(1)
# create some simulated data
# x is a random normal variable
x <- rnorm(n = 100, mean = 0, sd = 1)
# epsilon is also a random normal error with mean 0 and sd 1
e <- rnorm(n = 100, mean = 0, sd = 1)
# The final equation
y <- 0.1 + 0.2 * x + e # vectorize
```

```
# The first 5 observations
x <- x[1:5]
y <- y[1:5]
```

```
summary(lm(y~x))
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##        1        2        3        4        5
##  0.07353  0.41791 -0.13489 -0.02048 -0.33607
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.3479     0.1458  -2.387   0.0970 .
## x             0.5927     0.1677   3.535   0.0385 *
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3223 on 3 degrees of freedom
## Multiple R-squared:  0.8064, Adjusted R-squared:  0.7418
## F-statistic: 12.49 on 1 and 3 DF,  p-value: 0.03851
```

```r
# mean and s.d of x
x_mean <- mean(x)
x_sd <- sd(x)
# mean and s.d of y
y_mean <- mean(y)
# co-variance of x and y
cov_xy <- cov(x,y)
paste0("The is covariance of  x and y  is:", cov_xy)
```

```
## [1] "The is covariance of  x and y  is:0.547384853408087"
```

```r
# variance of x
var_x <-var(x)
paste0("The is variance of  x is:", var_x)
```

```
## [1] "The is variance of  x is:0.923596776496731"
```

```r
# coeff of x
beta_1 <- cov_xy/var_x
paste0("The is coe icient on x is:", beta_1)
```

```
## [1] "The is coe icient on x is:0.592666483185831"
```

$$y = \beta_0 + \beta_1 x$$

$$\beta_1 = \frac{\text{Cov}(x, y)}{\text{Var}(x)} = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}$$

Thus,

$$\beta_1 = \frac{0.5473}{0.9235} = 0.59266$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x} = (-0.271 - 0.129 * 0.592) = -0.347$$

Best fit line equation can be given as:

$$y = \beta_0 + \beta_1 x$$

That is,

$$y = -0.347 + 0.592x$$

$$TSS = \sum_i (y_i - \bar{y})^2$$

```r
y<- y[1:5]
mean_y <- mean(y)
y_meany <- y-mean_y
TSS <- sum((y_meany)^2)
TSS
```

```
## [1] 1.609278
```

$$TSS = (-0.3743658)^2 + (0.4501362)^2 + (-0.7067557)^2 + (0.8483766)^2 + (-0.2173914)^2 = 1.609$$

So, now we compute SSE using the predicted y values.

$$y = -0.347 + 0.592 * (-0.6264538) = -0.7178606$$
$$y = -0.347 + 0.592 * 0.1836433 = -0.2382832$$
$$y = -0.347 + 0.592 * -0.8356286 = -0.8416918$$
$$y = -0.347 + 0.592 * 1.5952808 = 0.5974062$$
$$y = -0.347 + 0.592 * 0.329507 = -0.1519319$$

```r
y_pred <- c(-0.7178606, -0.2382832, -0.8416921, 0.5974062, -0.1519314)
y <- y[1:5]
y_y_pred <- y-y_pred
SSE<- sum((y_y_pred)^2)
SSE
```

```
## [1] 0.3116163
```

$$SSE = \sum_i (y_i - \hat{y}_i)^2 = 0.3116163$$

Now,

$$R^2 = \frac{TSS - SSE}{TSS} = \frac{1.609 - 0.32}{1.609} = 0.8063$$

```r
r_sq <- (TSS - SSE)/ TSS
r_sq
```

```
## [1] 0.8063627
```

R-squared is the square of the correlation. It ranges from values (0,1) unlike correlation which ranges between (-1,+1) . The R-squared value of 0.806 indicates that 80.6% of the variation is captured by the model which is good.

Adjusted R- square:

$$adjustedR2 = \frac{TSS/df_t - SSE/df_e}{TSS/df_t}$$

$$df_t = n - 1 = 4$$
$$df_e = n - k - 1 = 5 - 1 - 1 = 3$$

```r
# Calculate adjusted R
df_t <- 4
df_e <- 3
(TSS/df_t - SSE/df_e) / (TSS/df_t)
```

```
## [1] 0.7418169
```

$$adjustedR2 = \frac{TSS/df_t - SSE/df_e}{TSS/df_t} = 0.7418169$$

Standard Error

$$\beta_1 = se_{\hat{y}} \frac{1}{\sqrt{\sum(x_i - \bar{x})^2}}$$

$$se_{\hat{y}} = \sqrt{\frac{SSE}{n-2}} = 0.33$$

```r
x <- x[1:5]
mean_x <- mean(x)
x_meanx <- x-mean_x
sum((x_meanx)^2)
```

```
## [1] 3.694387
```

```r
0.33 * 1/sqrt( 3.69)
```

```
## [1] 0.1717911
```

$$se_{\beta_1} = 0.33 * \frac{1}{\sqrt{3.69}} = 0.17$$

probelm 4: Now generate = 0.1 + 0.2 ∗ − 0.5 ∗ 2 + with 100 observations(10pts)

```r
set.seed(1)
# create some simulated data
# x is a random normal variable
x <- rnorm(n = 100, mean = 0, sd = 1)
# epsilon is also a random normal error with mean 0 and sd 1
e <- rnorm(n = 100, mean = 0, sd = 1)
# The final equation
y <- 0.1 + 0.2 * x - 0.5* x^2 + e # vectorize
```

probelm 4a : Regress y on x and 2 and report the results. If x or 2 are not statistically significant, suggest why

```r
set.seed(1)
# Regress y on x and x^2
y_on_x2 <- lm(y ~ x + I(x^2))
# Report the results
summary(y_on_x2)
```

```
##
## Call:
## lm(formula = y ~ x + I(x^2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9650 -0.6254 -0.1288  0.5803  2.2700
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.15672    0.11766   1.332   0.1860
## x            0.21716    0.10798   2.011   0.0471 *
## I(x^2)      -0.61892    0.08477  -7.302 7.93e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.958 on 97 degrees of freedom
## Multiple R-squared:  0.3602, Adjusted R-squared:  0.347
## F-statistic: 27.31 on 2 and 97 DF,  p-value: 3.912e-10
```

Both x and x^2 are significant because the p value is less than 0.05 i.e, 0.0471 and 7.93e-11

x^2 is really significant because the p value is 7.93e-11

The Pr(>t) acronym found in the model output relates to the probability of observing any value equal or larger than t. A small p-value indicates that it is unlikely we will observe a relationship between the predictor and response variables due to chance. Typically, a p-value of 5% or less is a good cut-off point. In our model example, the p-values are close to 0.01. Note the 'signif. Codes' associated to each estimate. The one star (or asterisks) represent significant p-value. Consequently, a small p-value for the intercept and the slope indicates that we can reject the null hypothesis which allows us to conclude that there is a relationship between Y and x and Y and x^2

```r
y <- 0.1 + 0.2 * x - 0.5* x^2  # main equation
y1 <- 0.1 + 0.2 * 1 - 0.5* 1^2  # y when x is 1
y2 <- 0.1 + 0.2 * 2 - 0.5* 2^2  # y when x is 2
y2-y1
```

probelm 4 b: Based on the known coefficients that we used to create y, what is the exact effect on y of increasing x by 1 unit from 1 to 2?

```
## [1] -1.3
```

```
# Therefore, increasing x by 1 unit from 1 to 2 results in decrease in value of y by -1.3
# Negative relationship
```

```
y <-  0.15672 + 0.21716 * x -0.61892* x^2  # main equation by coe icients estimated from 4(a)
y1 <-  0.15672 + 0.21716 * (-0.5) -0.61892 * ((-0.5))^2 # y when x is -0.5
y2 <-  0.15672 + 0.21716 * (-0.7) -0.61892 * ((-0.7))^2 # y when x is -0.7
y2-y1
```

probelm 4c : Based on the coefficients estimated from 4(a), what is the effect on y of changing x from -0.5 to -0.7?

```
## [1] -0.1919728
```

```
# Therefore, decreasing x by from -0.5 to -0.7 results in decrease in value of y by -0.19
```

```
set.seed(1)
# create some simulated data
# x is a random normal variable
x <- rnorm(n = 100, mean = 0, sd = 1)
#  2 as a random normal variable with a mean of -1 and an sd of 1
x2 <- rnorm(n = 100, mean = -1, sd = 1)
# epsilon is also a random normal error with mean 0 and sd 1
e <- rnorm(n = 100, mean = 0, sd = 1)
# The final equation
y <- 0.1 + 0.2 * x - 0.5* x *x2 + e # vectorize
```

Problem 5 : now generate 2 as a random normal variable with a mean of -1 and an sd of 1. create a new dataset where $= 0.1 + 0.2 * \quad - 0.5 * \quad * \ 2 +$ and answer the following items. (20 pts)

```
x_mean <- mean(x)
y1 <- 0.1 + 0.2 * x_mean - 0.5* x_mean *0 + e # when x2 is 0
y2 <- 0.1 + 0.2 * x_mean - 0.5* x_mean *1 + e # when x2 is 1
y2[1] - y1[1]
```

Problem 5a: Based on the known coefficients, what is the exact effect of increasing x2 from 0 to 1 with x held at its mean?

```
## [1] -0.05444368
```

```
# Therefore, increasing x2 by from 0 to 1,
# while keeping x at its mean results in decrease in value of y by -0.054
```

```r
set.seed(1)
# Regress y on x and x^2
y_on_xx2 <- lm(y ~ x + x2 + x * x2)
# Report the results
summary(y_on_xx2)
```

Problem 5b : Regress y on x, 2, and their interaction. Based on the regression-estimated coefficients, what is the effect on y of shifting x from -0.5 to -0.7 with 2 held at 1?

```
##
## Call:
## lm(formula = y ~ x + x2 + x * x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.92554 -0.43139  0.00249  0.65651  2.60188
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.10285    0.15470   0.665    0.508
## x           -0.07321    0.21598  -0.339    0.735
## x2          -0.02822    0.10970  -0.257    0.798
## x:x2        -0.73968    0.14847  -4.982 2.78e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.035 on 96 degrees of freedom
## Multiple R-squared:  0.4476, Adjusted R-squared:  0.4304
## F-statistic: 25.93 on 3 and 96 DF,  p-value: 2.262e-12
```

```r
# The final equation
y <- 0.102 -0.073 * x - 0.739 * x *x2 # vectorize
#The effect on y of shifting x from -0.5 to -0.7 with  2 held at 1?
y1 <- 0.102 -0.073 * (-0.5) - 0.739 * (-0.5) *1  # when x2 is 1, x is -0.5
y2 <- 0.102 -0.073 * (-0.7) - 0.739 * (-0.7) *1  # when x2 is 1, x is -0.7
y2[1] - y1[1]
```

```
## [1] 0.1624
```

```r
# Therefore, decreasing x by from -0.5 to -0.7,
# while keeping x2 constant 1 results in increase in value of y by  0.1624
```

```r
set.seed(1)
# create some simulated data
# x is a random normal variable
x <- rnorm(n = 100, mean = 0, sd = 1)
#  2 as a random normal variable with a mean of -1 and an sd of 1
x2 <- rnorm(n = 100, mean = -1, sd = 1)
```

```r
# epsilon is also a random normal error with mean 0 and sd 1
e <- rnorm(n = 100, mean = 0, sd = 1)
# The final equation
y <- 0.1 + 0.2 * x - 0.5* x *x2 + e # vectorize
```

```r
# Regress y on x alone
set.seed(1)
# Regress y on x
y_only_x <- lm(y ~ x)
# Report the results
summary(y_only_x)
```

Problem 5c :Regress y on x alone. Using the R2 from this regression and the R2 from 5(b), perform by hand an F test of the complete model (5b) against the reduced, bivariate model. What does this test tell you?

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9227 -0.7076  0.0501  0.6996  3.3161
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.1174     0.1162   1.011    0.315
## x             0.8352     0.1291   6.470 3.87e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.154 on 98 degrees of freedom
## Multiple R-squared:  0.2993, Adjusted R-squared:  0.2922
## F-statistic: 41.86 on 1 and 98 DF,  p-value: 3.873e-09
```

$$F = \frac{(R_c^2 - R_r^2)/df_1}{(1 - R_c^2)/df2} = \frac{(0.4476 - 0.2993)/2}{1 - 0.4476/100 - 3 - 1} = 12.89$$

```r
# p-value
pf(12.89,3-1,(100-3-1), lower.tail = F)
```

```
## [1] 1.099949e-05
```

There is a significant difference since the p-value is less than 0.05. We have enough evidence to reject the null hypothesis and conclude that complelete model is better than reduced, bivariate model and has significant difference between both the models

```r
# three variables: f, x1, and x2
# f should be a factor with three levels,
# where level 1 corresponds to observations 1-100,
# level 2 to 101-200, and level 3 to 201-300.
# (Eg, f can be "a" for the first 100 observations,
# "b" for the second 100, and "c" for the third 100.)
f <- as.factor(c(rep("a",100),rep("b",100),rep("c",100)))
# Create x1 such that the first 100 observations have a mean of 1 and sd of 2;
# The second 100 have a mean of0 and sd of 1
# The third 100 have a mean of 1 and sd of 0.5
x1 <- c(rnorm(100, 1, 2), rnorm(100, 0, 1), rnorm(100, 1, 0.5))
# Create x2 such that the first 100 observations have a mean of 1 and sd of 2;
# The second 100 have a mean of 1 and sd of 1;
# and the third 100 have a mean of 0 and sd of 0.5.
x2 <- c(rnorm(100, 1, 2), rnorm(100, 1, 1), rnorm(100, 0, 0.5))
```

```r
# create three 100-observation datasets first, and then stack them with rbind()
df_3var <- data.frame(cbind(x1,x2,f))
head(df_3var, 3)
```

Problem 6 :Generate a dataset with 300 observations and three variables: f, x1, and x2. f should be a factor with three levels, where level 1 corresponds to observations 1-100, level 2 to 101-200, and level 3 to 201-300. (Eg, f can be "a" for the first 100 observations, "b" for the second 100, and "c" for the third 100.) Create x1 such that the first 100 observations have a mean of 1 and sd of 2; the second 100 have a mean of0 and sd of 1; andt he third 100 have a mean of 1 and sd of 0.5. Create x2 such that the first 100 observations have a mean of 1 and sd of 2; the second 100 have a mean of 1 and sd of 1; and the third 100 have a mean of 0 and sd of 0.5. (Hint: It is probably easiest to create three 100-observation datasets first, and then stack them with rbind(). And make sure to convert f to a factor before proceeding.) (20pts)

```
##           x1        x2 f
## 1 -0.2529076  2.787347 1
## 2  1.3672866 -1.094596 1
## 3 -0.6712572  4.942675 1
```

```r
# The k-means algorithm, perform a cluster analysis of these data using a k of 3
# use only x1 and x2 in your calculations
subset_dfvar <- df_3var[,1:2]
kmean_x1x2<- kmeans(subset_dfvar, centers=3, nstart=25)
```

```r
#  checking the centers
kmean_x1x2$centers
```

Problem 6a : Using the k-means algorithm, perform a cluster analysis of these data using a k of 3 (use only x1 and x2 in your calculations; use f only to verify your results). Comparing

your clusters with f, how many datapoints are correctly classified into the correct cluster? How similar are the centroids from your analysis to the true centers?

```
##           x1          x2
## 1  1.4261256 -0.07850055
## 2  1.0283202  2.77781098
## 3 -0.6477379  0.39957780
```

```
# cluster
df_3var$cluster <- as.vector(kmean_x1x2$cluster)
```

```
# f only to verify your results
df_3var$f <- f
table(df_3var[c("f","cluster")])
```

```
##    cluster
## f    1  2  3
##   a 41 41 18
##   b 16 24 60
##   c 90  0 10
```

```
centroids<-aggregate(df_3var[,1:2], by=list(cat=df_3var$f), FUN = mean)
print(centroids) # checking the centroids
```

```
##   cat          x1          x2
## 1   a  1.21777473  1.10320372
## 2   b -0.03780808  0.96086576
## 3   c  1.01483677 -0.02225968
```

```
# accuracy
accuracy <- (41+60+90)/300
paste0("The Model shows an accuracy of :", percent(accuracy, accuracy = 1))
```

```
## [1] "The Model shows an accuracy of :64%"
```

Comparing your clusters with f, how many datapoints are correctly classified into the correct cluster? There are a total of 300 data points:
191 data points are correctly classified : 1. 41 correctly classifications for factor a, 2. 60 correct classifications for factor b, 3. and 90 correctly classifications for factor c

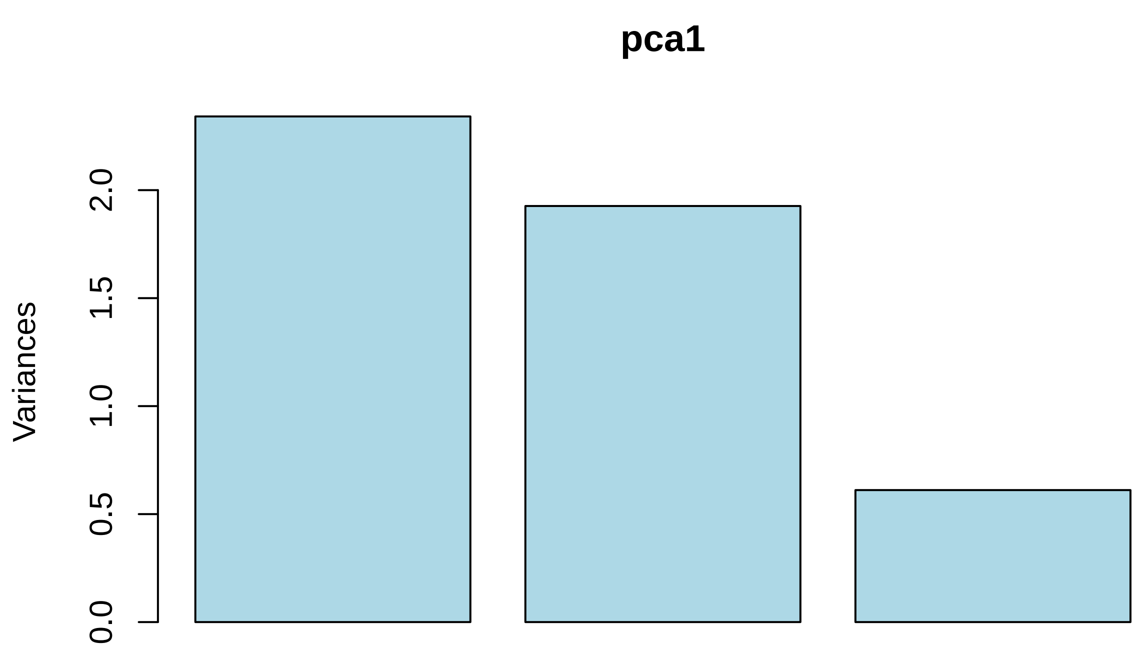
109 datapoints are in-correctly classified

How similar are the centroids from your analysis to the true centers ? Also, The centroids of the first two clusters and the real centers from cluster f are very different from one another. For the third cluster, where the x1 and x2 values were comparable to those for cluster f, we discovered a similar trend. For f, the values are x1=1.015 and x2=-0.022; for our study, they are x1=1.426 and x2=-0.079.

```r
# three variables: f, x1, and x2
# f should be a factor with three levels,
# where level 1 corresponds to observations 1-100,
# level 2 to 101-200, and level 3 to 201-300.
# (Eg, f can be "a" for the first 100 observations,
# "b" for the second 100, and "c" for the third 100.)
f <- as.factor(c(rep("a",100),rep("b",100),rep("c",100)))
# Create x1 such that the first 100 observations have a mean of 1 and sd of 2;
# The second 100 have a mean of0 and sd of 1
# The third 100 have a mean of 1 and sd of 0.5
x1 <- c(rnorm(100, 1, 2), rnorm(100, 0, 1), rnorm(100, 1, 0.5))
# Create x2 such that the first 100 observations have a mean of 1 and sd of 2;
# The second 100 have a mean of 1 and sd of 1;
# and the third 100 have a mean of 0 and sd of 0.5.
x2 <- c(rnorm(100, 1, 2), rnorm(100, 1, 1), rnorm(100, 0, 0.5))
```

```r
df_var <- data.frame(cbind(x1,x2,f))
pca1<-prcomp(df_var)
screeplot(pca1, col= "light blue")
```

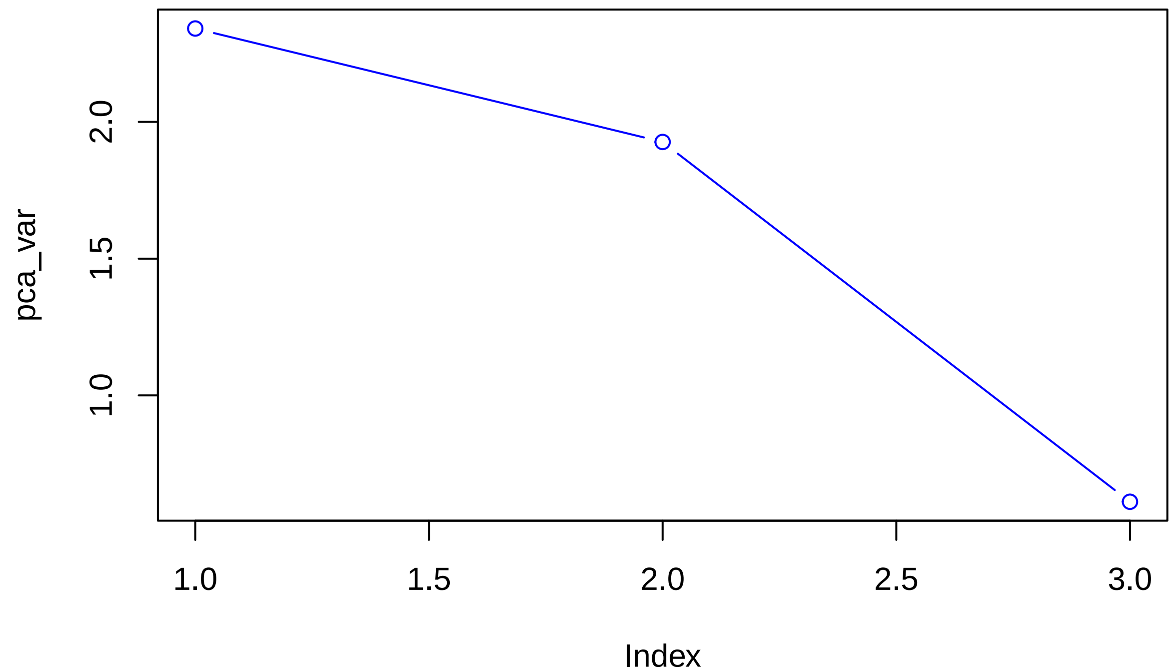
Problem 6b :Perform a factor analysis of this data using your preferred function. Using a scree plot and/or cumulative variance plot, how many factors do you think you should include? Speculate about how these results relate to those you got with the cluster analysis.



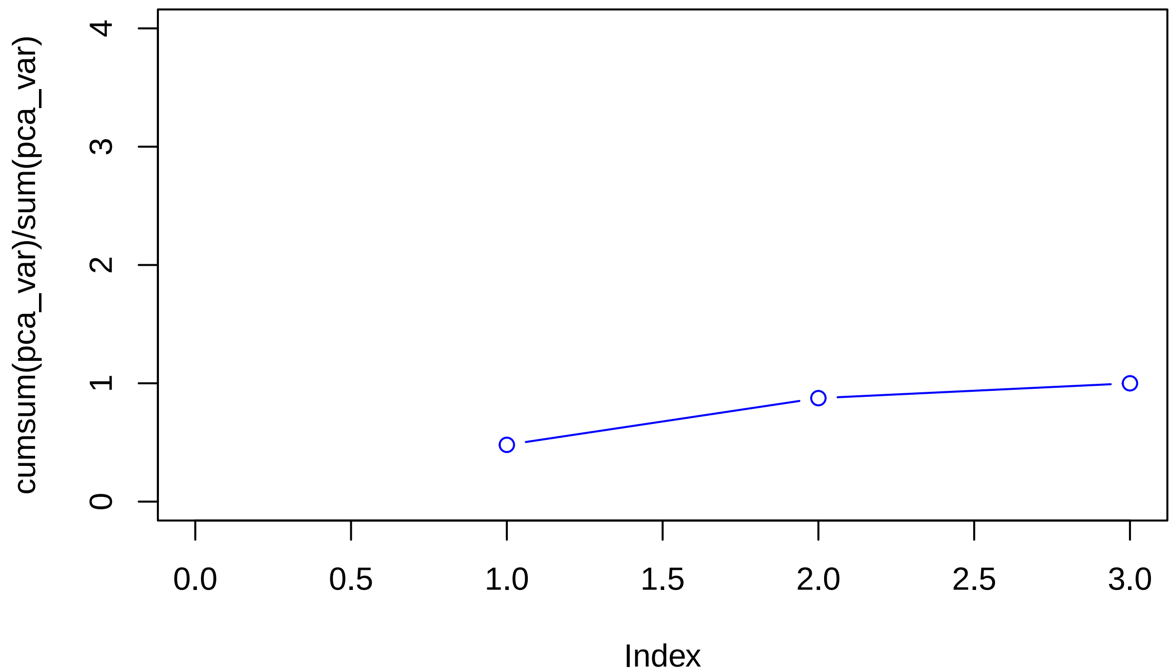
```r
summary(pca1)
```

```
## Importance of components:
##                           PC1    PC2    PC3
## Standard deviation     1.5301 1.3880 0.7818
## Proportion of Variance 0.4799 0.3949 0.1253
## Cumulative Proportion  0.4799 0.8747 1.0000
```

```r
pca_var<-pca1$sdev^2
plot(pca_var, type = "b", col="blue")
```




```r
plot(cumsum(pca_var)/sum(pca_var),type = "b",
     xlim =c(0,3), ylim=c(0,4), col="blue")
```



Therefore, I Speculate that only 3 factors should be included with the cluster analysis

About 57% of the data is made up of the first principal component, and 43% of the data of the second. We should include x1 and x2 because they are equivalent and neither one dominates the other. Additionally, it is consistent with how we generated the data as x1 and x2 are supposed to be unrelated.

```
# loading the dataset
df <-read.csv("/Users/sanjanagorlla/Desktop/massachussets_crime_final.csv",
              header = TRUE, stringsAsFactors = FALSE)
```

For the next questions use the Modified Massachussets Crimes dataset of 2019 available on the final canvas page ("mass_crimes_final.csv"). Modifying the dataframe object in R to perform your analysis correctly might be a part of the evaluation

```
dim(df)
```

Exploring the dataset

```
## [1] 281  12
```

```r
# The Modified Massachussets Crimes dataset
# of 2019 has 281 obseravtions with only 12 columns
```

```r
head(df, 6) # to print out the first 6 rows of every column in the dataset
```

```
##       City Population Violent.crime Murder_MANSLAUGHTER Rape Robbery
## 1 Abington     16,448            23                   4    5       3
## 2    Acton     23,780            32                   0    6       2
## 3 Acushnet     10,533            12                   0    5       0
## 4    Adams      8,028            26                   0   10       2
## 5   Agawam     28,736            82                   0   13       8
## 6 Amesbury     17,595            25                   0    3       3
##   Aggravated.assault Property.crime Burglary Larceny..theft Motor_vehicle_theft
## 1                 11            153       23            122                   8
## 2                 24             66       13             50                   3
## 3                  7             35       14             19                   2
## 4                 14             94       34             59                   1
## 5                 61            376      133            228                  15
## 6                 19            132       18            107                   7
##   Arson
## 1     1
## 2     0
## 3     0
## 4     2
## 5     1
## 6     0
```

```r
colnames(df) # To print out every column name.
```

```
##  [1] "City"                "Population"          "Violent.crime"
##  [4] "Murder_MANSLAUGHTER" "Rape"                "Robbery"
##  [7] "Aggravated.assault"  "Property.crime"      "Burglary"
## [10] "Larceny..theft"      "Motor_vehicle_theft" "Arson"
```

```r
# summary statistics on the numeric columns of the dataset
summary(df)
```

```
##      City            Population        Violent.crime      Murder_MANSLAUGHTER
##  Length:281         Length:281         Length:281         Min.   : 0.0000
##  Class :character   Class :character   Class :character   1st Qu.: 0.0000
##  Mode  :character   Mode  :character   Mode  :character   Median : 0.0000
##                                                           Mean   : 0.5302
##                                                           3rd Qu.: 0.0000
##                                                           Max.   :42.0000
##
##       Rape           Robbery          Aggravated.assault Property.crime
##  Min.   :  0.000   Length:281         Length:281         Length:281
##  1st Qu.:  1.000   Class :character   Class :character   Class :character
##  Median :  3.000   Mode  :character   Mode  :character   Mode  :character
##  Mean   :  7.413
##  3rd Qu.:  7.000
```

```
##  Max.   :231.000
##
##    Burglary         Larceny..theft     Motor_vehicle_theft     Arson
##  Length:281         Length:281         Min.   :  0.00      Min.   : 0.000
##  Class :character   Class :character   1st Qu.:  1.00      1st Qu.: 0.000
##  Mode  :character   Mode  :character   Median :  5.00      Median : 0.000
##                                        Mean   : 17.03      Mean   : 1.032
##                                        3rd Qu.: 12.00      3rd Qu.: 1.000
##                                        Max.   :493.00      Max.   :31.000
##                                        NA's   :1           NA's   :1
```

There are NA values in the dataset

```
# structure of the dataset
str(df)
```

```
## 'data.frame':    281 obs. of  12 variables:
##  $ City               : chr  "Abington" "Acton" "Acushnet" "Adams" ...
##  $ Population         : chr  "16,448" "23,780" "10,533" "8,028" ...
##  $ Violent.crime      : chr  "23" "32" "12" "26" ...
##  $ Murder_MANSLAUGHTER: int  4 0 0 0 0 0 0 0 0 0 ...
##  $ Rape               : int  5 6 5 10 13 3 28 6 0 5 ...
##  $ Robbery            : chr  "3" "2" "0" "2" ...
##  $ Aggravated.assault : chr  "11" "24" "7" "14" ...
##  $ Property.crime     : chr  "153" "66" "35" "94" ...
##  $ Burglary           : chr  "23" "13" "14" "34" ...
##  $ Larceny..theft     : chr  "122" "50" "19" "59" ...
##  $ Motor_vehicle_theft: int  8 3 2 1 15 7 15 7 0 12 ...
##  $ Arson              : int  1 0 0 2 1 0 2 0 0 3 ...
```

```
# this shows that we have to change the type of few variables
```

```
#skimming the dataset
skim(df)
```

Table 1: Data summary

Name	df
Number of rows	281
Number of columns	12
Column type frequency:	
character	8
numeric	4
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
City	0	1	3	21	0	281	0
Population	0	1	3	7	0	280	0
Violent.crime	0	1	1	5	0	102	0
Robbery	0	1	1	5	0	38	0
Aggravated.assault	0	1	1	5	0	96	0
Property.crime	0	1	0	5	1	190	0
Burglary	0	1	1	5	0	85	0
Larceny..theft	0	1	1	6	0	164	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Murder_MANSLAUGHTER	0	1	0.53	2.99	0	0	0	0	42	
Rape	0	1	7.41	17.17	0	1	3	7	231	
Motor_vehicle_theft	1	1	17.03	47.03	0	1	5	12	493	
Arson	1	1	1.03	2.79	0	0	0	1	31	

data pre-processing

1. NA values

```
# checking the NA's
sapply(df, function(x) sum(is.na(x)))
```

```
##                City          Population       Violent.crime Murder_MANSLAUGHTER
##                   0                   0                   0                   0
##                Rape             Robbery  Aggravated.assault      Property.crime
##                   0                   0                   0                   0
##            Burglary      Larceny..theft Motor_vehicle_theft               Arson
##                   0                   0                   1                   1
```

```
# Remove NAs from the data.
clean_df <- na.omit(df)
```

```
sapply(clean_df, function(x) sum(is.na(x)))
```

```
##                City          Population       Violent.crime Murder_MANSLAUGHTER
##                   0                   0                   0                   0
##                Rape             Robbery  Aggravated.assault      Property.crime
##                   0                   0                   0                   0
##            Burglary      Larceny..theft Motor_vehicle_theft               Arson
##                   0                   0                   0                   0
```

Here, I have obsereved zero values in few columns. Hence, removing them for further analysis

```
df_no_zero <- filter_if(clean_df, is.numeric, all_vars((.) != 0))
dim(clean_df)
```

```
## [1] 280  12
```

```
dim(df_no_zero)
```

```
## [1] 28 12
```

Here, I have obsereved many numeric columns are represented as charecter. It would be difficult to work with them. Hence, converting them to numeric. if we use as.numeric() warning message "NAs introduced by coercion" would occur. This is because, some of the input values are not formatted properly, because they contain commas (i.e. ,) between the numbers. We can remove these commas by using the gsub function:

```
clean_df$Population <- as.integer(gsub(",", "", clean_df$Population))
clean_df$Violent.crime <- as.integer(gsub(",", "", clean_df$Violent.crime))
clean_df$Robbery <- as.integer(gsub(",", "", clean_df$Robbery))
clean_df$Aggravated.assault <- as.integer(gsub(",", "",
                                              clean_df$Aggravated.assault))
clean_df$Property.crime <- as.integer(gsub(",", "", clean_df$Property.crime))
clean_df$Burglary <- as.integer(gsub(",", "", clean_df$Burglary))
clean_df$Larceny..theft<- as.integer(gsub(",", "", clean_df$Larceny..theft))
```

```
# changing the column names to maintain consistency with the column names
colnames(clean_df)[colnames(clean_df) == "Violent.crime"] <- "Violent_crime"
colnames(clean_df)[colnames(clean_df) == "Murder_MANSLAUGHTER"] <- "Murder_manslaughter"
colnames(clean_df)[colnames(clean_df) == "Aggravated.assault"] <- "Aggravated_assault"
colnames(clean_df)[colnames(clean_df) == "Property.crime"] <- "Property_crime"
colnames(clean_df)[colnames(clean_df) == "Larceny..theft"] <- "Larceny_theft"
```

```
# inspecting data structure
glimpse(clean_df)
```

```
## Rows: 280
## Columns: 12
## $ City                <chr> "Abington", "Acton", "Acushnet", "Adams", "Agawam"~
## $ Population          <int> 16448, 23780, 10533, 8028, 28736, 17595, 39603, 36~
## $ Violent_crime       <int> 23, 32, 12, 26, 82, 25, 99, 8, 2, 34, 8, 21, 47, 1~
## $ Murder_manslaughter <int> 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,~
## $ Rape                <int> 5, 6, 5, 10, 13, 3, 28, 6, 0, 5, 2, 4, 7, 27, 0, 1~
## $ Robbery             <int> 3, 2, 0, 2, 8, 3, 2, 1, 0, 8, 0, 1, 2, 14, 1, 2, 1~
## $ Aggravated_assault  <int> 11, 24, 7, 14, 61, 19, 69, 1, 2, 21, 6, 16, 37, 82~
## $ Property_crime      <int> 153, 66, 35, 94, 376, 132, 173, 215, 0, 167, 28, 7~
## $ Burglary            <int> 23, 13, 14, 34, 133, 18, 55, 28, 0, 16, 7, 9, 9, 6~
## $ Larceny_theft       <int> 122, 50, 19, 59, 228, 107, 103, 180, 0, 139, 18, 6~
## $ Motor_vehicle_theft <int> 8, 3, 2, 1, 15, 7, 15, 7, 0, 12, 3, 8, 4, 26, 13, ~
## $ Arson               <int> 1, 0, 0, 2, 1, 0, 2, 0, 0, 3, 0, 0, 1, 1, 3, 0, 0,~
```

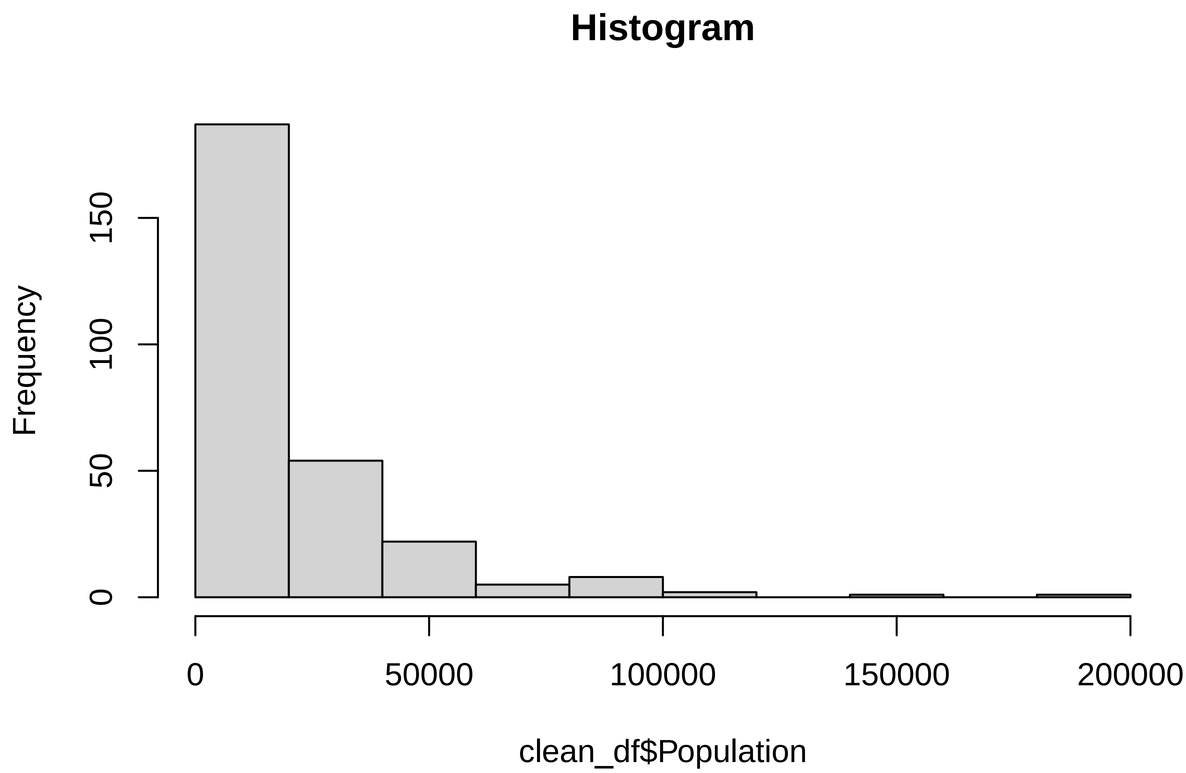
```
# Final check for missing value
anyNA(clean_df)
```

```
## [1] FALSE
```

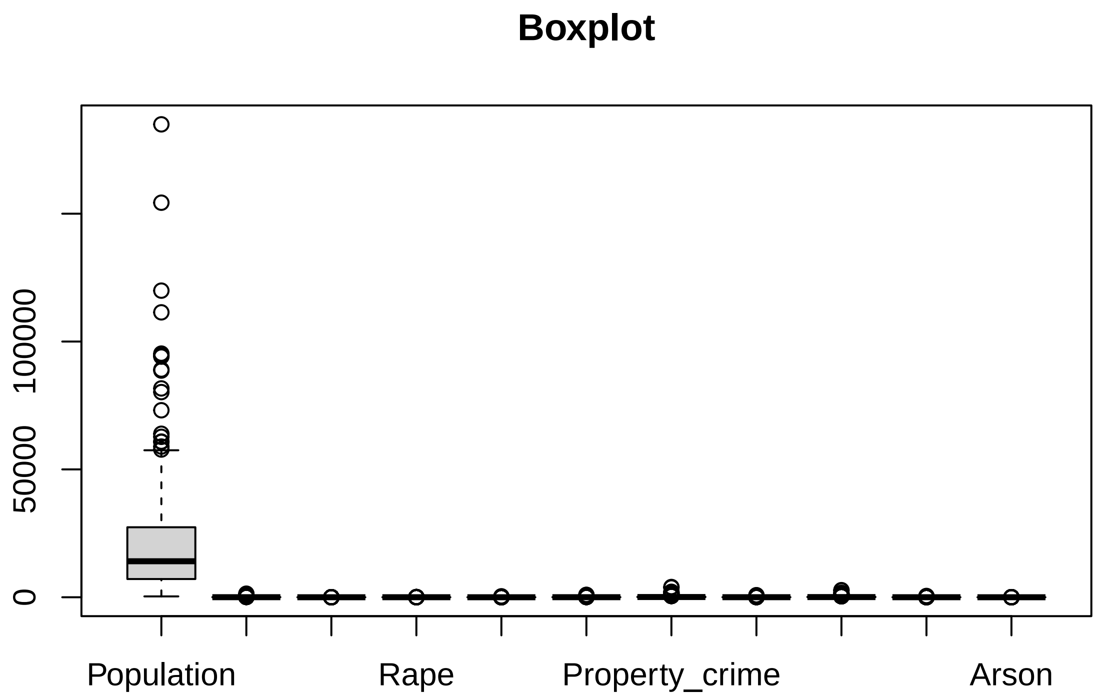
```r
### checking the distribution
summary(clean_df)
```

```
##      City             Population     Violent_crime     Murder_manslaughter
##  Length:280         Min.   :   328   Min.   :   0.00   Min.   : 0.0000    
##  Class :character   1st Qu.:  7176   1st Qu.:   6.00   1st Qu.: 0.0000    
##  Mode  :character   Median : 14080   Median :  19.00   Median : 0.0000    
##                     Mean   : 21458   Mean   :  62.34   Mean   : 0.3821    
##                     3rd Qu.: 27320   3rd Qu.:  47.00   3rd Qu.: 0.0000    
##                     Max.   :184945   Max.   :1397.00   Max.   :20.0000    
##       Rape           Robbery         Aggravated_assault Property_crime   
##  Min.   : 0.000   Min.   :  0.000   Min.   :  0.00     Min.   :   0.00  
##  1st Qu.: 1.000   1st Qu.:  0.000   1st Qu.:  4.00     1st Qu.:  29.75  
##  Median : 3.000   Median :  1.000   Median : 14.00     Median :  82.00  
##  Mean   : 6.614   Mean   :  8.714   Mean   : 46.63     Mean   : 226.11  
##  3rd Qu.: 7.000   3rd Qu.:  4.000   3rd Qu.: 37.00     3rd Qu.: 208.50  
##  Max.   :81.000   Max.   :358.000   Max.   :938.00     Max.   :4005.00  
##     Burglary      Larceny_theft   Motor_vehicle_theft     Arson       
##  Min.   :  0.00   Min.   :   0.0   Min.   :  0.00      Min.   : 0.000  
##  1st Qu.:  5.00   1st Qu.:  23.0   1st Qu.:  1.00      1st Qu.: 0.000  
##  Median : 12.50   Median :  63.0   Median :  5.00      Median : 0.000  
##  Mean   : 36.58   Mean   : 172.5   Mean   : 17.03      Mean   : 1.032  
##  3rd Qu.: 28.00   3rd Qu.: 164.8   3rd Qu.: 12.00      3rd Qu.: 1.000  
##  Max.   :786.00   Max.   :2766.0   Max.   :493.00      Max.   :31.000
```

```r
# check for outliers in the dataset
hist(clean_df$Population, main = "Histogram")
```




```
boxplot(clean_df[,-1], main = "Boxplot")
```

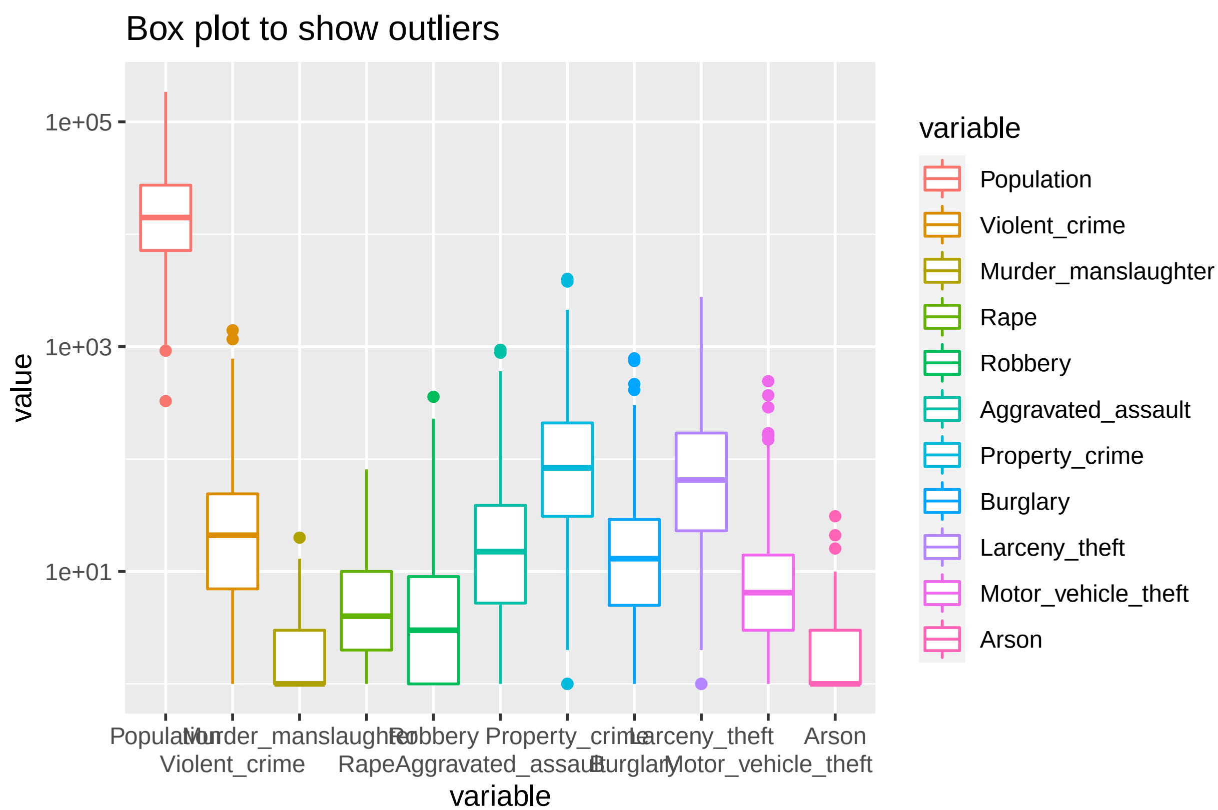



```
df.cols <- names(clean_df[,-1])
data.boxplot <- melt(clean_df[,-1], measure.vars=df.cols)

ggplot(data.boxplot)+
  geom_boxplot(aes(x =variable, y= value, color = variable))+
  labs(title = "Box plot to show outliers", caption = "crime Data")+
  scale_x_discrete(guide = guide_axis(n.dodge = 2))+
  scale_y_log10()
```

```
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 676 rows containing non-finite values (stat_boxplot).
```



crime Data

There are few outliers but these doesnot effect the analysis. Hence not removing the outliers

```
# city with highest population
filter(clean_df, Population == max(Population))
```

Population

```
##        City Population Violent_crime Murder_manslaughter Rape Robbery
## 1 Worcester     184945          1165                  13   40     229
##   Aggravated_assault Property_crime Burglary Larceny_theft Motor_vehicle_theft
## 1                883           3792      786          2637                 369
##   Arson
## 1     6
```

```
# city with least population
filter(clean_df, Population == min(Population))
```

```
##       City Population Violent_crime Murder_manslaughter Rape Robbery
## 1 Aquinnah        328             2                   0    0       0
##   Aggravated_assault Property_crime Burglary Larceny_theft Motor_vehicle_theft
## 1                  2              0        0             0                   0
##   Arson
## 1     0
```

```r
# which city is the biggest:
clean_df$City[which.max(clean_df$Population)]
```

```
## [1] "Worcester"
```

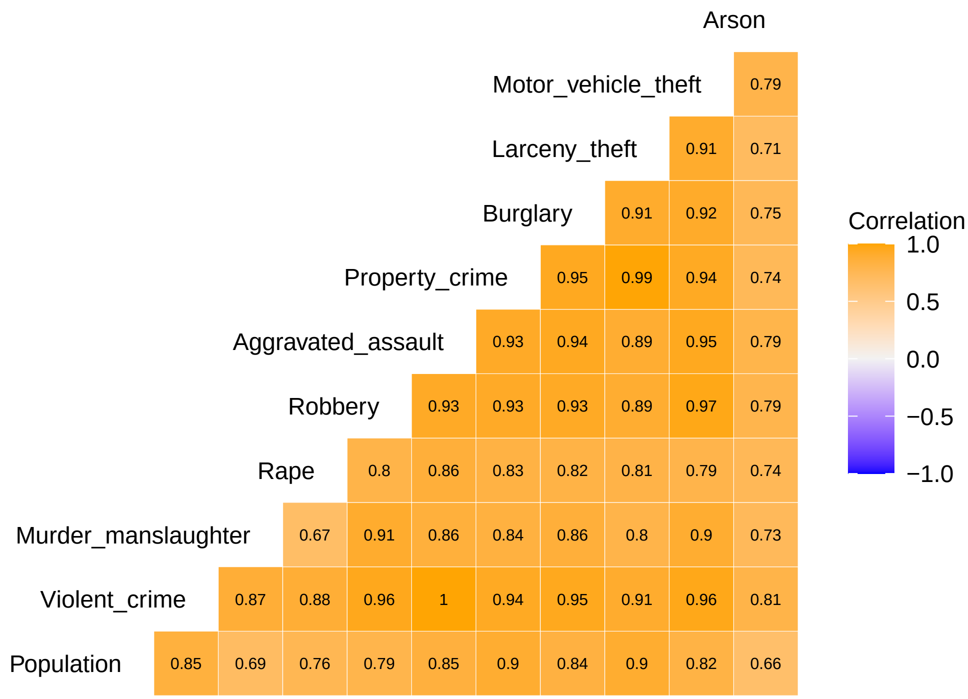
```r
# How many people:
max(clean_df$Population)
```

```
## [1] 184945
```

Building the correlation matrix

```r
data_num <- clean_df %>%
                              select_if(is.numeric)

ggcorr(data_num,
       label = T,
       label_size = 2,
       label_round = 2,
       hjust = 1,
       size = 3,
       color = "black",
       layout.exp = 5,
       low = "blue",
       mid = "gray95",
       high = "orange",
       name = "Correlation")
```

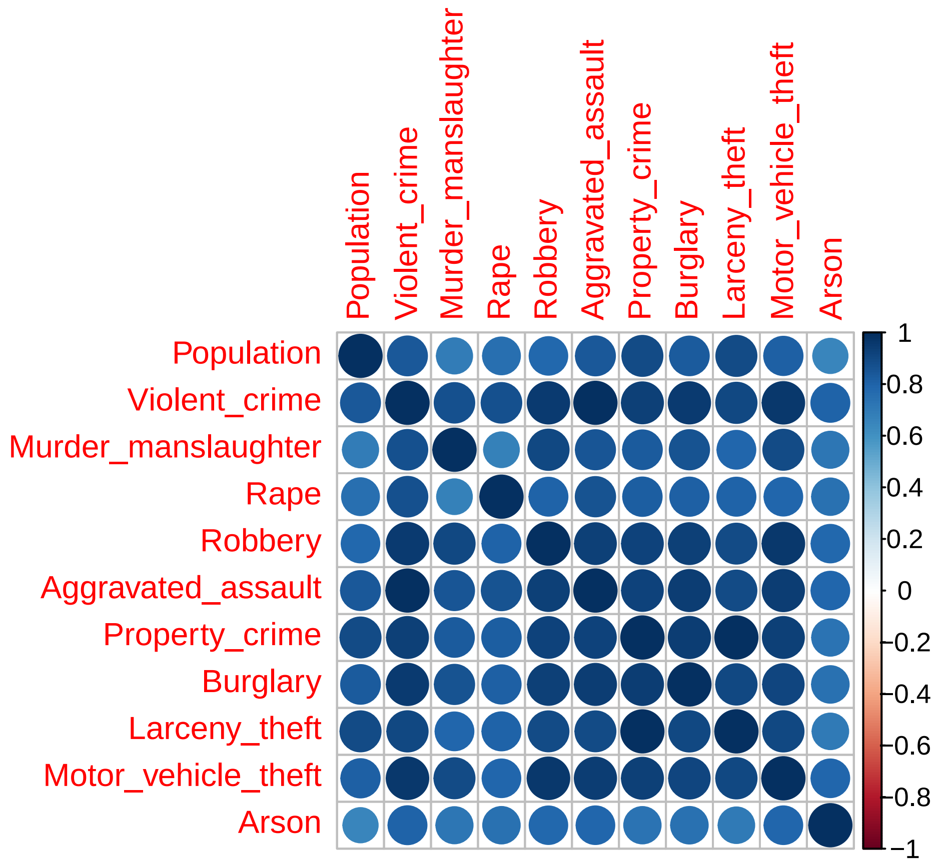


The Population as dependent variable has somewhat strong positive correlation with Property_crime, Larceny_theft, Violent_crime, Aggravated_assault, Burglary, Motor_vehicle_theft.

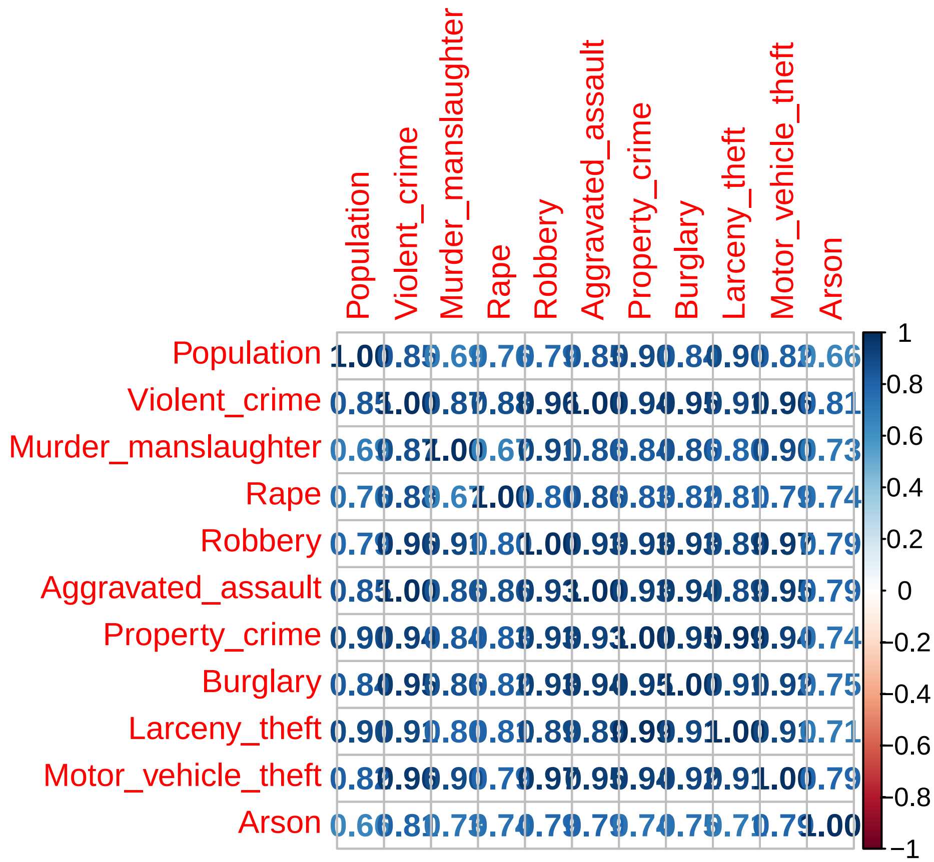
And this is a valid finding, because high Population leads to crimes like Property_crime, Larceny_theft, Violent_crime, Aggravated_assault, Burglary, Motor_vehicle_theft.

And based on the Corr Matrix, we can see there is very strong correlation between them. This strong correlation indicates multicollinearity among them.

```r
M <- cor(clean_df[,2:12])
corrplot(M,method = "circle")
```



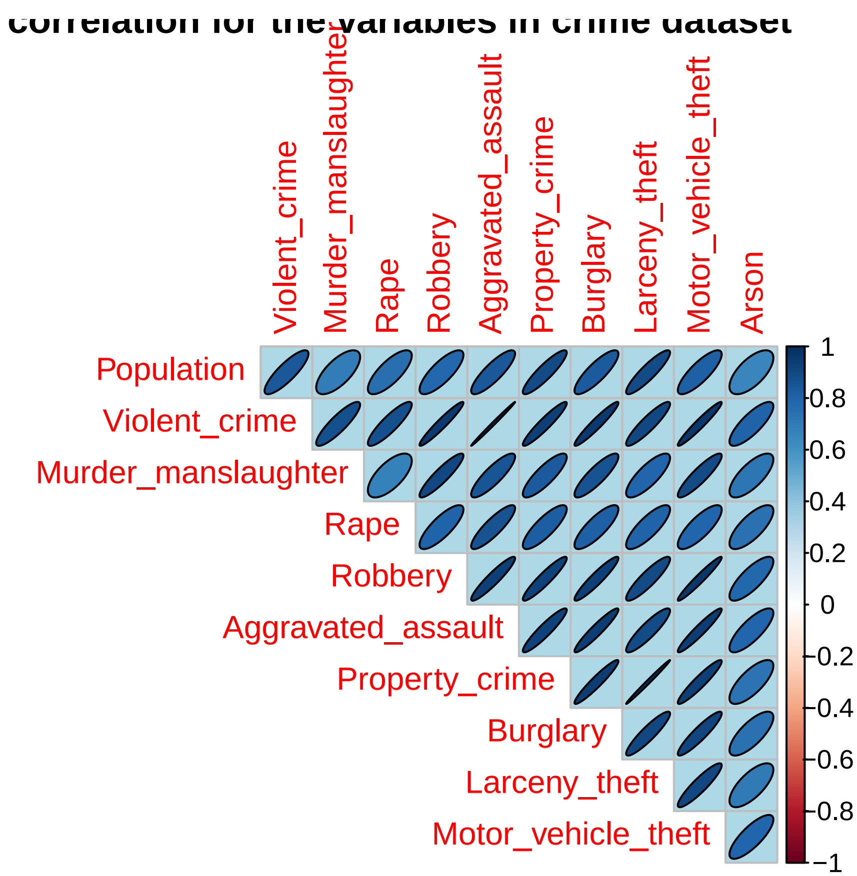

```
corrplot(M,method = "number")
```



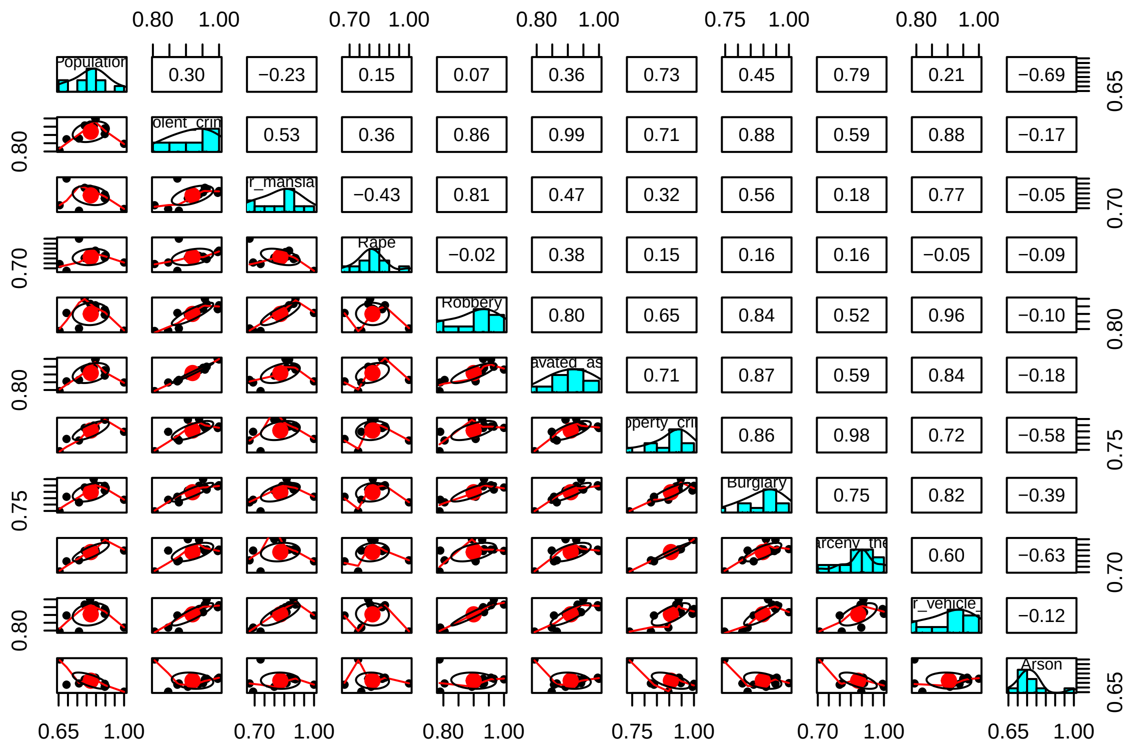

```r
corrmatrix = cor(clean_df[,-1])
kable(t(corrmatrix))
```

	Population	Violent_crime	Murder_manslaughter	Rape	Robbery	Aggravated_assault	Property_crime	Burglary	Larceny_theft	Motor_vehicle_theft	Arson
Population	1.0000000	0.8463782	0.6924567	0.7591273	0.7895348	0.8483604	0.8964237	0.8378683	0.8971656	0.8191517	0.6554161
Violent_crime	0.8463782	1.0000000	0.8740058	0.8785449	0.9572792	0.9960693	0.9395640	0.9519248	0.9057407	0.9625588	0.8050913
Murder_manslaughter	0.6924567	0.8740058	1.0000000	0.6743528	0.9064118	0.8555946	0.8382685	0.8614980	0.7993429	0.8967231	0.7299702
Rape	0.7591273	0.8785449	0.6743528	1.0000000	0.8039970	0.8646799	0.8257456	0.8174382	0.8086405	0.7940624	0.7434084
Robbery	0.7895348	0.9572792	0.9064118	0.8039970	1.0000000	0.9313499	0.9281737	0.9342456	0.8941026	0.9664991	0.7872696
Aggravated_assault	0.8483604	0.9960693	0.8555946	0.8646799	0.9313499	1.0000000	0.9270530	0.9427992	0.8927475	0.9497198	0.7929931
Property_crime	0.8964237	0.9395640	0.8382685	0.8257456	0.9281737	0.9270530	1.0000000	0.9471130	0.9939976	0.9381592	0.7389608
Burglary	0.8378683	0.9519248	0.8614980	0.8174382	0.9342456	0.9427992	0.9471130	1.0000000	0.9099879	0.9179922	0.7469434
Larceny_theft	0.8971656	0.9057407	0.7993429	0.8086405	0.8941026	0.8927475	0.9939976	0.9099879	1.0000000	0.9072865	0.7074577
Motor_vehicle_theft	0.8191517	0.9625588	0.8967231	0.7940624	0.9664991	0.9497198	0.9381592	0.9179922	0.9072865	1.0000000	0.7938008
Arson	0.6554161	0.8050913	0.7299702	0.7434084	0.7872696	0.7929931	0.7389608	0.7469434	0.7074577	0.7938008	1.0000000

```r
corrplot (cor(clean_df[,-1]),
          method="ellipse",
          bg = " light blue", type = "upper",
          title= " correlation for the variables in crime dataset",
          diag = F,
          outline = T,
          insig = "pch",
          pch= 3)
```

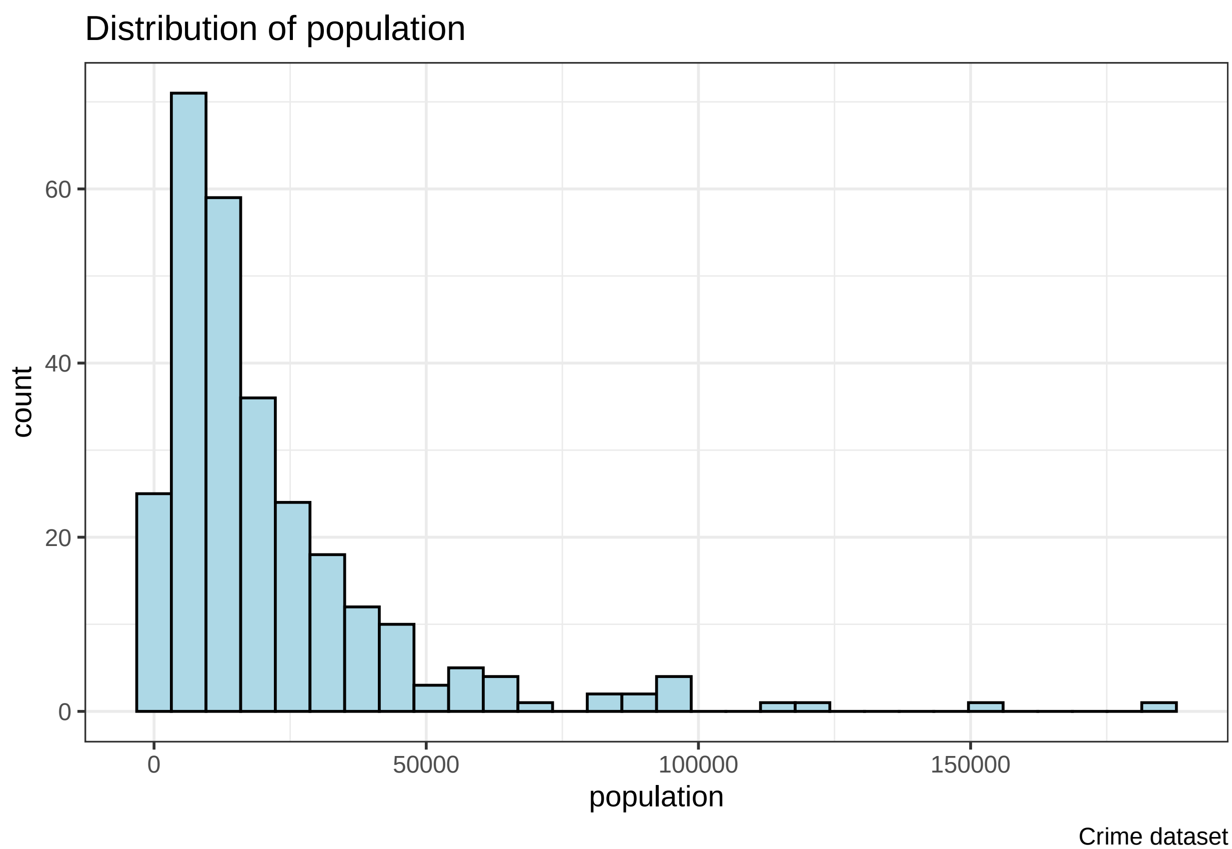



```
pairs.panels(cor(clean_df[,-1]))
```




```r
ggplot(clean_df, aes(clean_df$Population))+ geom_histogram(color="black",
                                                          fill="light blue")+
  labs(title = "Distribution of population", x= "population", y= "count",
       caption = "Crime dataset")+
  theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

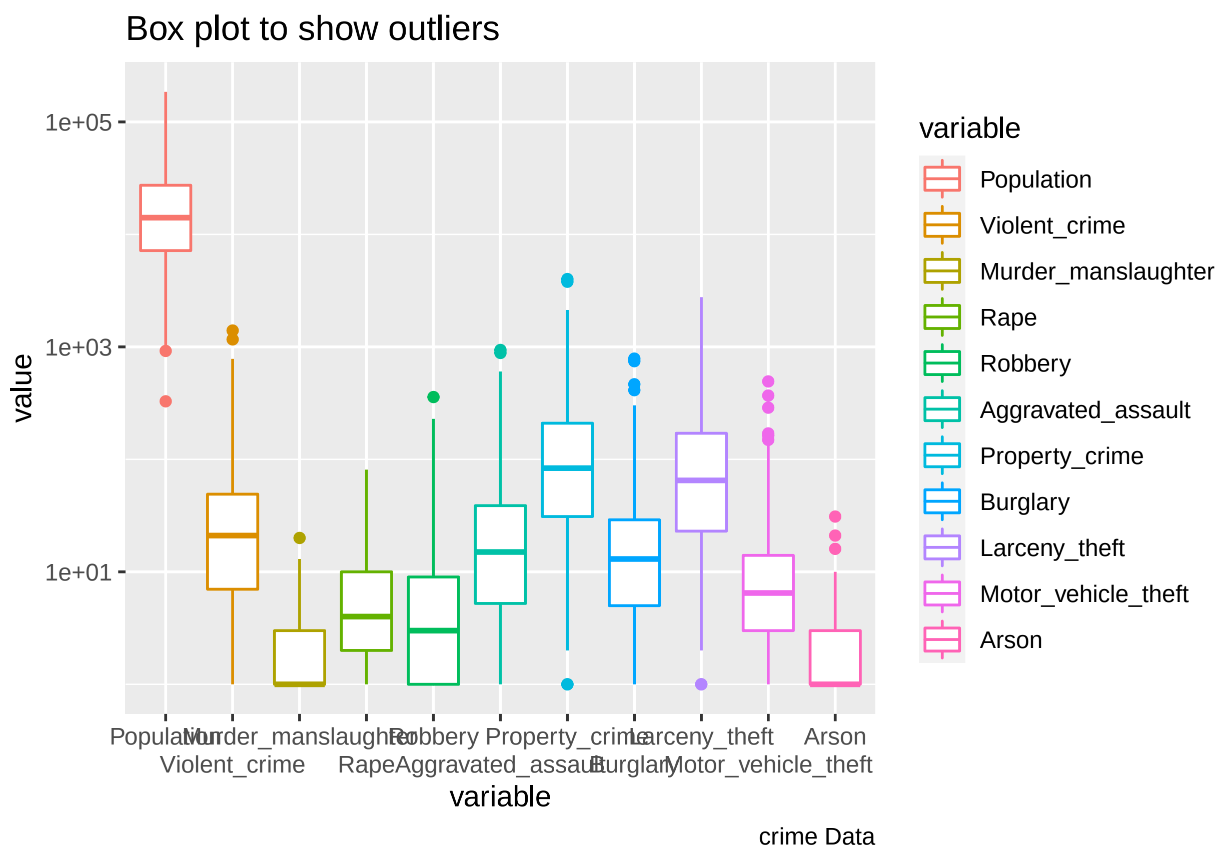



```
df.cols_no <- names(clean_df[,-1])
data.boxplot_no <- melt(clean_df[,-1], measure.vars=df.cols_no)

ggplot(data.boxplot_no)+
  geom_boxplot(aes(x =variable, y= value, color = variable))+
  labs(title = "Box plot to show outliers", caption = "crime Data")+
  scale_x_discrete(guide = guide_axis(n.dodge = 2))+
  scale_y_log10()
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Removed 676 rows containing non-finite values (stat_boxplot).
```



7. Raise some hypothesis about the dataset, motivate it, filter the rows and columns (if needed), so that it can be tested using multiple regression. State all steps clearly and document your conclusions. (5pts). *#### First hypothesis : I hypothize that there might be an effect on population with the crimes like Property_crime, Robbery, Burglary, Violent_crime. And these types of crime might be prevalent in the cities with high population*

Therefore, filtering the 50 cities with highest population

```
# rearranging the dataset in descending order to filter
# top 50 cities with the highest population
filter_df<- clean_df %>%
  arrange(desc(Population))
# Top  cities with highest population
new_filter <- head(filter_df, 50)
```

```
# deslecting city
# Now, In Total, there are 12 Variables,
# 11 of them are Numerical, and 1 of them are chr.
# We will need to deselect some variables:
# Also, Murder_manslaughter has zero values, Theferore, not including them in further analysis
final_data <- new_filter %>%
  dplyr::select(-c(City, Murder_manslaughter))
```

```
Model_1 <- lm(Population ~ Property_crime +Robbery +Burglary +
                Violent_crime, data = final_data)
```

```r
summary(Model_1)
```

```
##
## Call:
## lm(formula = Population ~ Property_crime + Robbery + Burglary +
##     Violent_crime, data = final_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -45895  -9587    173   9571  40196
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    30252.180   3947.240   7.664 1.07e-09 ***
## Property_crime    33.826      9.082   3.725 0.000543 ***
## Robbery         -122.720    128.416  -0.956 0.344357
## Burglary          11.429     45.453   0.251 0.802618
## Violent_crime     27.114     29.595   0.916 0.364466
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14710 on 45 degrees of freedom
## Multiple R-squared:  0.8006, Adjusted R-squared:  0.7829
## F-statistic: 45.17 on 4 and 45 DF,  p-value: 3.333e-15
```

```r
summary(Model_1)$adj.r.squared
```

```
## [1] 0.7828767
```

The model_1 has the 0.7828767 value of Adjusted R-Squared The Model_1 has the largest parameter estimate that is property_crime which is 12.53. The property_crime will affect the Population the most in a positive direction. The p-value of property_crime is much lower than 0.05, thus indicating they are very significant predictors for Population. This model has R-squared value 0.7828767, which indicates the Model can describe its predictors condition by 74%. Hence we can conclude that Robbery + Burglary + Violent_crime are not significant in the multiple regression model. As these variables is not significant, it is possible to remove it from the model

second hypothesis : I hypothize that the criminals who commit Robbery, Larceny_theft and violent crime, might also commit burglary. Therefore, burgulary can be predicted by the number of Robbery, Larceny_theft crimes and voilent_crime

```r
Model_2 <- summary(lm(Burglary~ Robbery  + Larceny_theft+ Violent_crime,  data = final_data))
Model_2
```

```
##
## Call:
## lm(formula = Burglary ~ Robbery + Larceny_theft + Violent_crime,
##     data = final_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -161.320  -13.752    1.589   18.350  120.915
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -9.18470   13.79914  -0.666 0.508991
## Robbery        0.31558    0.43635   0.723 0.473194
## Larceny_theft  0.07529    0.03084   2.441 0.018554 *
## Violent_crime  0.34865    0.09001   3.873 0.000338 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 51.28 on 46 degrees of freedom
## Multiple R-squared:  0.9094, Adjusted R-squared:  0.9035
## F-statistic: 153.9 on 3 and 46 DF,  p-value: < 2.2e-16
```

The model_2 has the 0.9035 value of Adjusted R-Squared The Model_2 has the largest parameter estimate that is Violent_crime which is 0.34865. The Violent_crime, Robbery and Larceny_theft will affect the Burglary the most in a positive direction. The p-value of Larceny_theft and Violent_crime is much lower than 0.05, thus indicating they are very significant predictors for Burglary. This model has R-squared value 0.9094, which indicates the Model can describe its predictors condition by 90%. Hence we can conclude that robbery is not significant in the multiple regression model.As this variable is not significant, it is possible to remove it from the model

```r
Model_3 <- summary(lm(Burglary~ Larceny_theft+ Violent_crime,  data = final_data))
Model_3
```

```
##
## Call:
## lm(formula = Burglary ~ Larceny_theft + Violent_crime, data = final_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -163.402  -13.443    3.678   19.414  115.896
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -15.27416   10.87714  -1.404  0.16682
## Larceny_theft   0.08584    0.02703   3.176  0.00264 **
## Violent_crime   0.40105    0.05316   7.544 1.23e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 51.02 on 47 degrees of freedom
## Multiple R-squared:  0.9083, Adjusted R-squared:  0.9044
## F-statistic: 232.9 on 2 and 47 DF,  p-value: < 2.2e-16
```

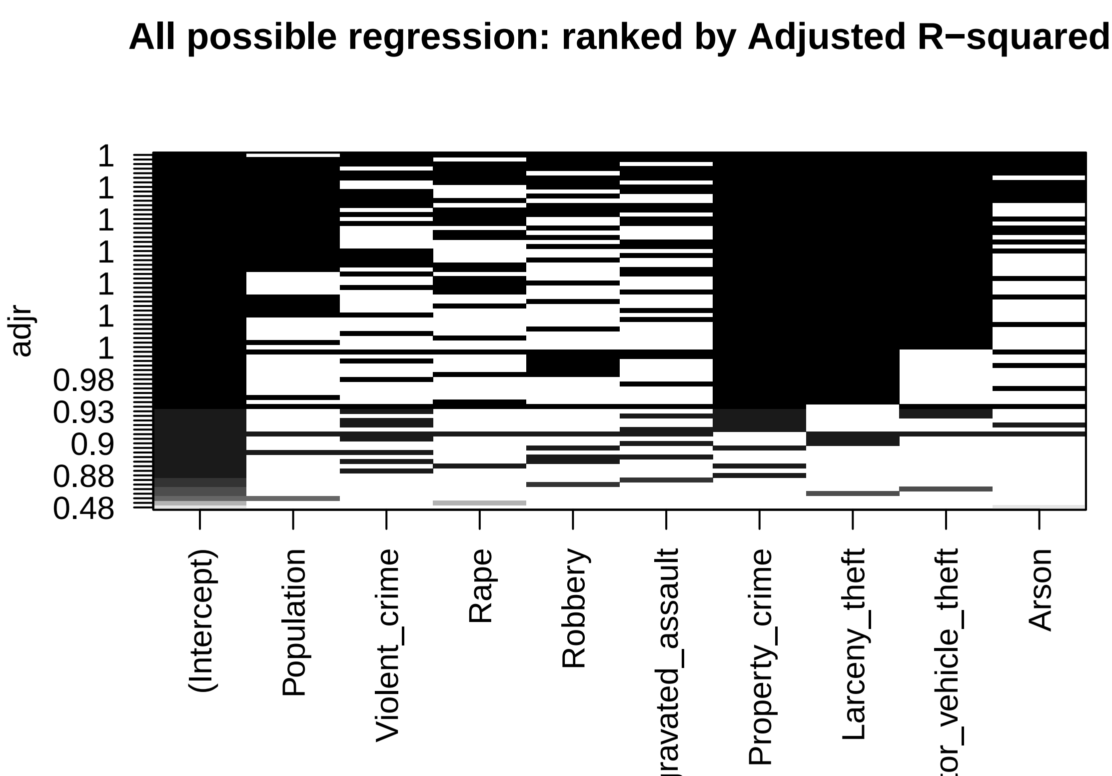
After removing the robbery the model seems to be improved and Larceny_theft and Violent_crime have significant effect on burglary

Third hypothesis : I hypothezie if double the voilent crime rate does the burglary is effected

```r
Model_3 <- summary(lm(Burglary~ I(Violent_crime^2) + Larceny_theft, data = final_data))
Model_3
```

```
##
## Call:
## lm(formula = Burglary ~ I(Violent_crime^2) + Larceny_theft, data = final_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -124.33  -24.00  -14.12   12.58  180.93
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.806e+01  1.549e+01   1.166     0.25
## I(Violent_crime^2) 2.414e-04  4.738e-05   5.095 6.10e-06 ***
## Larceny_theft      1.378e-01  2.933e-02   4.698 2.32e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 60.89 on 47 degrees of freedom
## Multiple R-squared:  0.8694, Adjusted R-squared:  0.8639
## F-statistic: 156.5 on 2 and 47 DF,  p-value: < 2.2e-16
```

```r
regs <- regsubsets(Burglary~., data = final_data, nbest=10)
plot(regs,
     scale="adjr",
     main="All possible regression: ranked by Adjusted R-squared")
```



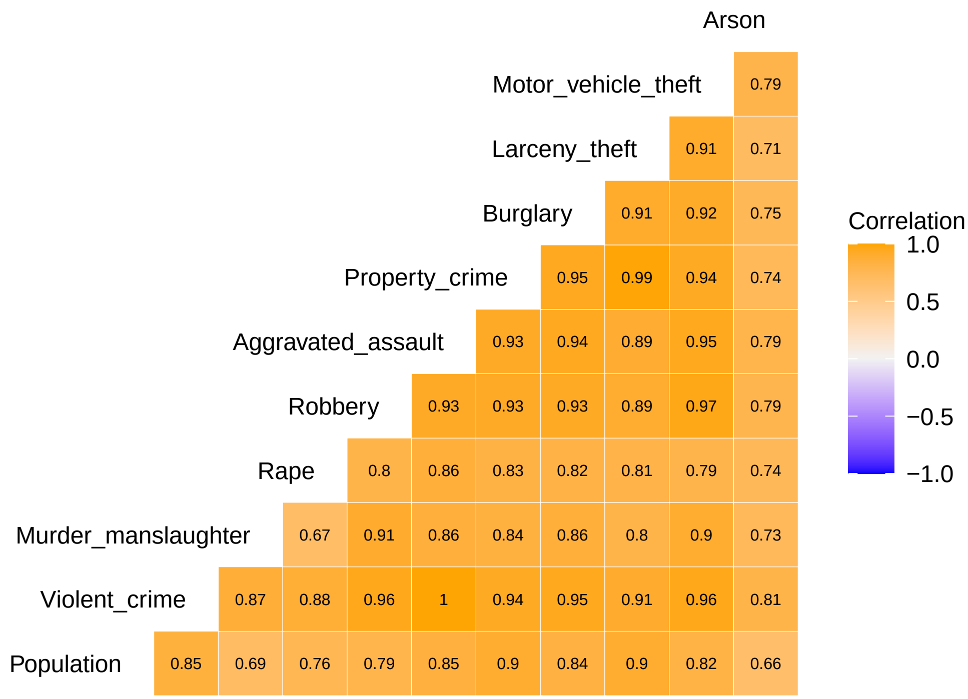
Based on given Plot, we can determine the most significant Variables based on Largest Adj. R-Squared: By adjusted R^2, the best model includes "Violent_crime","Rape","Robbery","Aggravated_assault","Property_crime" (variables that have black boxes at the higest Y-axis value).

8. Perform a WRONG regression using the dataset. A wrong regression is one that uses either inappropriate variables or other substantial errors, but that still results in a table and coefficients. Explain the results obtained and why they're not a proper application of the methods we learnt this semester? (5pts) A wrong regression is when we chose two variables are highly correlated, they are basically measuring the same phenomenon. When one enters into the regression equation, it tends to explain same thing

checking the corr plot again to select the variables that are highly correlated

```
data_num <- clean_df %>%
                             select_if(is.numeric)

ggcorr(data_num,
       label = T,
       label_size = 2,
       label_round = 2,
       hjust = 1,
       size = 3,
       color = "black",
       layout.exp = 5,
       low = "blue",
       mid = "gray95",
       high = "orange",
       name = "Correlation")
```



we can see property_crime is highly correlated with larency_theft is 0.99 burglary 0.95 and motor_vehicle theft 0.94

```
Model_4 <- summary(lm(Property_crime ~  Motor_vehicle_theft + Burglary +
                        Larceny_theft, data = final_data))
```

```
## Warning in summary.lm(lm(Property_crime ~ Motor_vehicle_theft + Burglary + :
## essentially perfect fit: summary may be unreliable
```

```
Model_4
```

```
##
## Call:
## lm(formula = Property_crime ~ Motor_vehicle_theft + Burglary +
##     Larceny_theft, data = final_data)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -2.000e-12 -8.311e-14 -3.113e-14  3.900e-14  1.017e-12
##
## Coefficients:
##                       Estimate Std. Error    t value Pr(>|t|)
## (Intercept)         -2.010e-13  8.389e-14 -2.396e+00   0.0207 *
## Motor_vehicle_theft  1.000e+00  1.455e-15  6.872e+14   <2e-16 ***
## Burglary             1.000e+00  8.269e-16  1.209e+15   <2e-16 ***
```

```
## Larceny_theft        1.000e+00  2.402e-16  4.163e+15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.741e-13 on 46 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 7.385e+31 on 3 and 46 DF,  p-value: < 2.2e-16
```

There is a warning that the summary may be unreliable due to the essentially perfect fit.

This means we have overfitted model with only 3 perfectly fit-able data points.

Multicollinearity happens when independent variables in the regression model are highly correlated to each other. It makes it hard to interpret of model and also creates an overfitting problem.

it is recommended to avoid having correlated features in your dataset. Indeed, a group of highly correlated features will not bring additional information (or just very few), but will increase the complexity of the algorithm, thus increasing the risk of errors.

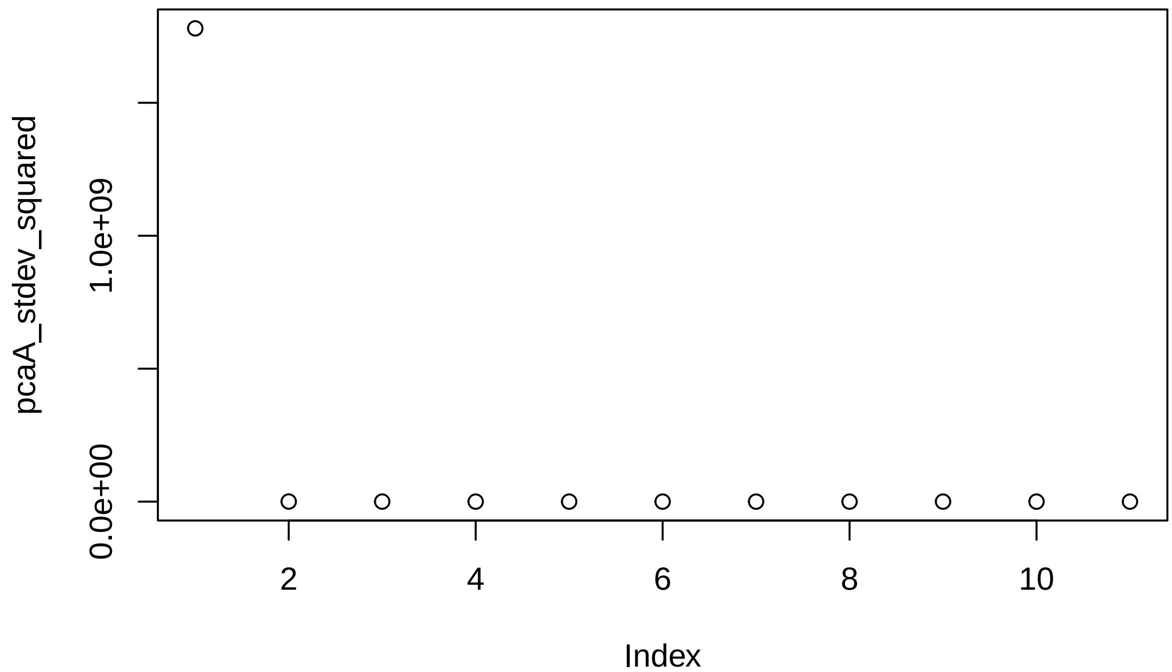
```
Model_5<- summary(lm(Population~ .,
                data = clean_df))
```

Including the city in the multi regression is also wrong is a blunder mistake. Before using a chararecter varaible we need to factorize it. city cannot be factorize

```
new_df <- clean_df[,-1]
# removing zeroes before pca
row_sub = apply(new_df, 1, function(row) all(row !=0 ))
##Subset as usual
new_df_wozero <- new_df[row_sub,]
```

```
# Factor eigenvalues or variances
# (or the sdev or standard deviations as reported by prcomp or princomp)
pcaA<- prcomp(new_df_wozero) #prcomp()
pcaA1 <- pcaA$rotation[,1]
# Extracting standard deviations
pcaA_stdev <- pcaA$sdev
# Squaring to get variances
pcaA_stdev_squared<- pcaA_stdev^2
pcaA1 <- pcaA$rotation[,1]
# Extracting standard deviations
pcaA_stdev <- pcaA$sdev
# Squaring to get variances
pcaA_stdev_squared<- pcaA_stdev^2
#Plot these in a scree plot and use the "elbow" test to guess how many factors one should retain
plot(pcaA_stdev_squared)
```

9. Perform either PCA or Clustering on the dataset. Present your results and conclusions into one or more paragraphs. (10pts)



The common criteria used for choosing the number of factors is based on an examination of these values. First, we look for the "elbow" in the curve – where it goes from the steep decline, then the flat area, where the presumption is the flat are is all the factors that are just noise.

In the scree plot, from the 2nd number, the line becomes flat. So we would include 2 factors and the rest is noise

```
new_df_wozero$Population <- new_df_wozero$Population/100
summary(lm(Property_crime ~  Population + Motor_vehicle_theft
           + Burglary + Larceny_theft, data = new_df_wozero))
```

10. Repeat the procedure chosen on question 8 but now transform the data to a new dataframe so that population is taken into account (normally crime data is presented in X offenses by 100.000 habitants). Show your results and compare them to the ones of question 9 with another paragraph (10pts Bonus)

```
## Warning in summary.lm(lm(Property_crime ~ Population + Motor_vehicle_theft + :
## essentially perfect fit: summary may be unreliable
```

```
##
## Call:
## lm(formula = Property_crime ~ Population + Motor_vehicle_theft +
##     Burglary + Larceny_theft, data = new_df_wozero)
```

```
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.058e-12 -2.355e-14  1.489e-14  4.013e-14  1.199e-12
##
## Coefficients:
##                       Estimate Std. Error    t value Pr(>|t|)
## (Intercept)         -1.243e-13  1.450e-13 -8.570e-01     0.40
## Population           1.850e-16  4.024e-16  4.600e-01     0.65
## Motor_vehicle_theft  1.000e+00  1.459e-15  6.855e+14   <2e-16 ***
## Burglary             1.000e+00  8.550e-16  1.170e+15   <2e-16 ***
## Larceny_theft        1.000e+00  2.778e-16  3.600e+15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.529e-13 on 22 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 5.122e+31 on 4 and 22 DF,  p-value: < 2.2e-16
```

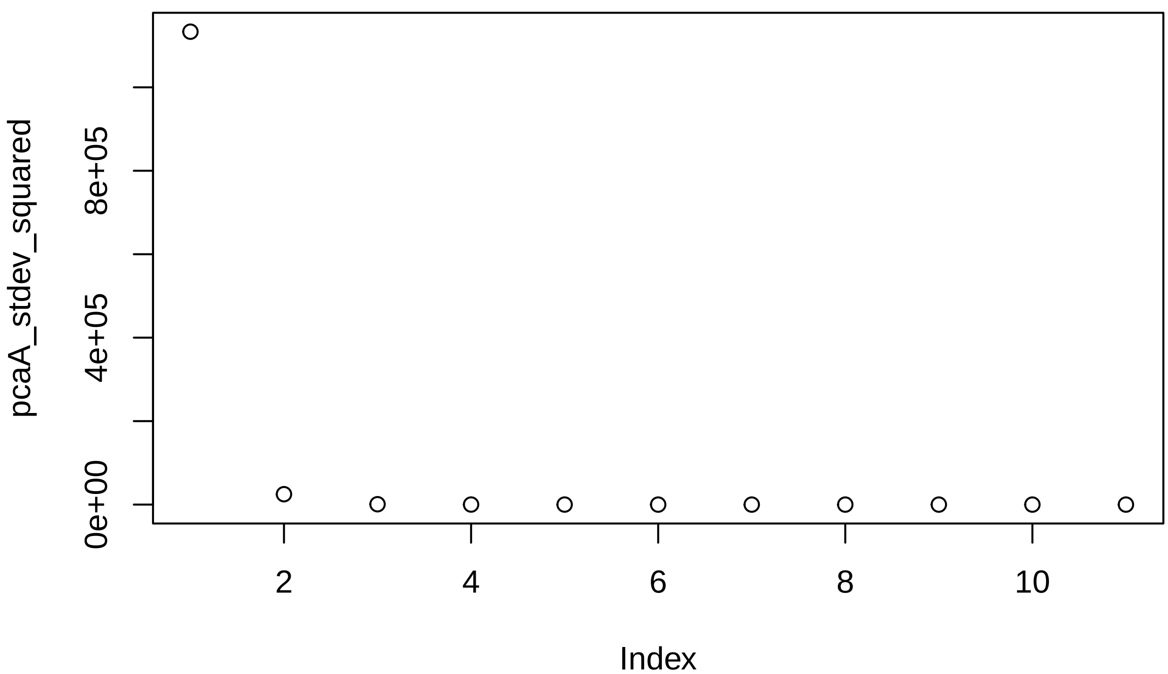
Population is noted to be less significant

```
new_df_wozero$Population <- new_df_wozero$Population/10000
new_df_wozero$Motor_vehicle_theft <- new_df_wozero$Motor_vehicle_theft /10000
new_df_wozero$Burglary <-new_df_wozero$Burglary/10000
new_df_wozero$Larceny_theft <- new_df_wozero$Larceny_theft/10000
summary(lm(Property_crime ~  Population + Motor_vehicle_theft
           + Burglary + Larceny_theft, data = new_df_wozero))
```

```
## Warning in summary.lm(lm(Property_crime ~ Population + Motor_vehicle_theft + :
## essentially perfect fit: summary may be unreliable
```

```
##
## Call:
## lm(formula = Property_crime ~ Population + Motor_vehicle_theft +
##     Burglary + Larceny_theft, data = new_df_wozero)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -2.757e-13 -8.035e-14 -4.714e-14  3.254e-14  1.123e-12
##
## Coefficients:
##                       Estimate Std. Error    t value Pr(>|t|)
## (Intercept)          1.078e-14  1.138e-13  9.500e-02    0.925
## Population          -2.115e-12  3.158e-12 -6.700e-01    0.510
## Motor_vehicle_theft  1.000e+04  1.145e-11  8.734e+14   <2e-16 ***
## Burglary             1.000e+04  6.710e-12  1.490e+15   <2e-16 ***
## Larceny_theft        1.000e+04  2.180e-12  4.587e+15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.77e-13 on 22 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 8.315e+31 on 4 and 22 DF,  p-value: < 2.2e-16
```

```r
# Factor eigenvalues or variances
# (or the sdev or standard deviations as reported by prcomp or princomp)
pcaA<- prcomp(new_df_wozero) #prcomp()
pcaA1 <- pcaA$rotation[,1]
# Extracting standard deviations
pcaA_stdev <- pcaA$sdev
# Squaring to get variances
pcaA_stdev_squared<- pcaA_stdev^2
pcaA1 <- pcaA$rotation[,1]
# Extracting standard deviations
pcaA_stdev <- pcaA$sdev
# Squaring to get variances
pcaA_stdev_squared<- pcaA_stdev^2
#Plot these in a scree plot and use the "elbow" test to guess how many factors one should retain
plot(pcaA_stdev_squared)
```



There is no difference in 8th and 9th questions even after taking population into account