INSH5301 FINAL Solution

Yicheng Zhang

2022-12-04

```
library(psych)
library(leaps)
library(tidyverse)
library(ggplot2)
library(dplyr)
library(corrplot)
library(factoextra)
```

We use `set.seed(1)` at the begaining of each code chunk when necessary.

1.

```
set.seed(1)
# I will simulate 10000 rolls:
roll<-10000

prob_15to20 <- function(x) {
  # we first initiate a times variable
  # this variable will indicate the total times we have a result between15 and 20
  times<-0
  # sample(1:6, 5, replace = T) is the output of the showing dice result
  # we store them here, 6 indicates we have 6 sides and 5 means 5 dice
  # this is the sum of each dice each row
  # we store them here and as total sample pool
  total <- replicate(x, sum(sample(1:6, 5, replace = T)))
  # loop over the total sample pool
  for (i in 1:x) {
    if (total[i] >= 15 & total[i] <= 20) {
      # every time we have a result matchs then times should add one
      times<-times+1
    }
  }
  # calculate the probability
  return(times/roll)
}

# probability of get 15 to 20 in total after 10000 rolls
prob_15to20(roll)
```

```
## [1] 0.5525
```

The returned value above here as 0.5525, and it should be the approximate solutions of the probability of getting between 15 and 20 (inclusive) as the total amount of 10000 simulations.

2.

a.

H_o: The mean of the x and y is same

H_a: The true difference between x and y is not 0

Since we know that two-sample t-test is used when the data of two samples are statistically independent, and the paired t-test is used when we have data is in the form of matched pairs. Therefore , given the y is actually created based on x we might want to use paired t test for this question.

```
set.seed(1)

# initiate variables:
x <- rnorm(100)
epsilon <- rnorm(100)
y <- 0.1+2*x+epsilon

# t.test:
t.test(x,y,paired=T)
```

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

Since here we have a p value larger than 0.05, we fail to reject the null hypothesis that the mean of the x and y's observations is same.

b.

We first want to have the first 5 numbers of x and y:

```
meanX<- mean(x[1:5])
meanX
```

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

```
meany<- mean(y[1:5])
meany
```

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

Since we know

$$se_{diff} = \sqrt{se_x^2 + se_y^2}$$

and $se_x = sd_x/\sqrt{n_x}$, $se_y = sd_y/\sqrt{n_y}$. So we can calculate the standard error of x and y first:

```
seX<-sd(x[1:5])/sqrt(5)
sey<-sd(y[1:5])/sqrt(5)
seX
```

```
## [1] 0.4297899
```

```
sey
```

```
## [1] 1.035892
```

and we now can calculate the

$$se_{diff} = \sqrt{se_x^2 + se_y^2} = \sqrt{0.4297899^2 + 1.035892^2} = 1.121513$$

and then we calculate the test statistics with :

$$T = (x_{bar} - y_{bar})/se_{diff} = (0.1292699 - (-0.03860587))/1.121513 = 0.1496869$$

and the last one we need to calculate here is the df:

$$df = \frac{se_{diff}^4}{se_x^4/(n_x-1) + se_y^4/(n_y-1)} = 1.121513^4/(0.4297899^4/(5-1) + 1.035892^4/(5-1)) = 5.337491$$

Since we now know the df is 5.337491 and we can calculate the p value now:

```
qt(c(0.975,0.025),5.337491)
```

```
## [1]  2.522455 -2.522455
```

So we have a reject region as [2.522455,-2.522455] and the test statistics we got (0.1496869) actually lies within it, we fail to reject the null hypothesis that the mean of the x and y's first 5 observations is same.

c.

```
sdx<-sd(x[1:5])
sdy<-sd(y[1:5])
```

Since we know the z score of the 0.01 CI level is :

```
qnorm(0.995)
```

```
## [1] 2.575829
```

we also know the formula of 0.01 level CI here is and the mean is fixed:

$$CI = \bar{y} \pm z\frac{sd}{\sqrt{n_{new}}} = -0.03860587 \pm qnorm(1-0.01/2) \times \frac{2.316324}{\sqrt{n_{new}}}$$

and in the same time we know the se will also change as we are having a different n value, but the sd will be fixed:

$$se = sd_y/\sqrt{n_{new}} = 2.316324/\sqrt{n_{new}}$$

and we are asked to get the min total number (n_{new}) so that we can reject the null and claim the true mean of the population is different from 0. If we want to do so then the CI level should not include the 0 as if a confidence interval contains zero then we would say there is strong evidence that there is not a 'significant' difference between the population means. And this case, we certainly want to have no 0 included in the CI. By observing through the formula of the CI we provided the above, we know that lower bound of the CI will definitely less than 0 as $qnorm(0.995) \times \frac{2.316324}{\sqrt{n_{new}}}$ will be a positive figure and if we use a negative number -0.03860587 to minus it, the result will still less than 0. So we can only try to lower our 95% CI upper bound to less than 0 liek this:$-0.03860587 + qnorm(1-0.01/2) \times \frac{2.316324}{\sqrt{n_{new}}} < 0$ and we have a function as following:

```
min_total<-function(){
  n<-6 # start from 6
  while (n){
    # try to find the first value that low down upper band
    # to less than 0
    if(-0.03860587+qnorm(0.995)*(sdy/sqrt(n))<0){
      return(n)
    }
    else {
      n=n+1
    }
  }
}
min_total()
```

```
## [1] 23886
```

and let's try out the min total of 23886 to see if it works:

```
# se of the y
yse<-sdy / sqrt(23886)
pt((meany-0)/yse,23886-1)*2
```

```
## [1] 0.01000447
```

Well, this result is still larger than 0.01 and this is reasonable as we used the `qnorm(0.995)` in the function which is not precise enough but we should be very approaching to the true min total, so let's try add 1 to the result;

```
yse<-sdy / sqrt(23887)
pt((meany-0)/yse,23887-1)*2
```

```
## [1] 0.01000291
```

still not work, keep adding:

```
yse<-sdy / sqrt(23888)
pt((meany-0)/yse,23888-1)*2
```

```
## [1] 0.01000135
```

still not work, keep adding:

```
yse<-sdy / sqrt(23889)
pt((meany-0)/yse,23889-1)*2
```

```
## [1] 0.009999789
```

There we go, now we have a p value less than 0.01 and we can reject the null to claim that the true mean μ of the population is different from 0 at the p = 0.01. The minimum total number is 23889, and the minimum total number of additional observations we would need would be 23889-5 which is 23884.

3.

a.

```
set.seed(1)
y3 <- 0.1+0.2*x+epsilon
m1<-lm(y3~x)
summary(m1)
```

```
##
## Call:
## lm(formula = y3 ~ 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
```

```
# we use the coefficient of x and std error to get the 95% CI here:
0.1989396 + qt(0.975, 98) * 0.10773# 95%ci upper bound
```

```
## [1] 0.4127263
```

```
0.1989396 - qt(0.975, 98) * 0.10773# 95%ci lower bound
```

```
## [1] -0.01484708
```

From the result, we can tell that Rˆ2 value as 0.03363 which means the majority of the Y could not be explained by the x. The coefficient of the x indicates the slope of the line of the best fit and we actually can know that if we increase the x by 1 then the y will also increase by 0.19894.The standard error here can tell us the average distance of each point to the line of the best fit and we have a std error really approaching to 0.01 here. The 95% CI here is [-0.01484708,0.4127263].

b.

```
# use the test statistics from the previous question we got:
pt(coef(summary(m1))[[6]], lower.tail = FALSE, 98) * 2
```

```
## [1] 0.06780814
```

This result tells us that it corresponds to the regression results which demonstrates that at the p=0.05 threshold level, we may able to accept the null hypothesis as the coefficient is 0, implying that if we change the x, it will not effect on the value of y.

c.

```r
pf(3.41,1,98,lower.tail = F)
```

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

Since by referring to the text book we can know that the F test mainly used to measure the overall fitness of the model by checking if we have coefficients different from 0, and this result with a p value as 0.06782021 actually tells us that we need to accept the null hypothesis that we don't have coefficients significantly different from 0 and this also verified what we mentioned in the last question. In the same time, this p value also helps us to conclude that R-squared is nearly equal zero, and the correlation between the model and dependent variable is not statistically significant.

d.

```r
# we first move the mean of x and  to here and sd of x:
meanX
```

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

```r
meany3<-mean(y3[1:5])
meany3
```

```
## [1] -0.2712917
```

```r
sdx
```

```
## [1] 0.9610394
```

```r
# the first 5 obs of x and y3
x5<-x[1:5]
y3.5<-y3[1:5]

# first 5 x:
x5
```

```
## [1] -0.6264538  0.1836433 -0.8356286  1.5952808  0.3295078
```

```r
# first 5 y:
y3.5
```

```
## [1] -0.6456574  0.1788445 -0.9780474  0.5770849 -0.4886831
```

```r
# sd of the new y
sdy3.5<-sd(y3.5)

# correlation of x and new y:
cov<-cov(x5,y3.5)
sdy3.5
```

```
## [1] 0.6342866
```

```r
cov
```

```
## [1] 0.5473849
```

Since we know the coefficient formula is :

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

and we need to calculate the β_0 and β_1 in this case. For the β_1 the formula is :

$$r = \frac{\text{Cov}(x,y)}{s_x s_y} = 0.5473849/(0.9610394 \times 0.6342866) = \beta_1 \frac{s_x}{s_y} = 0.8979788$$

which means :

$$\beta_1 = \frac{\frac{r}{s_x}}{s_y} = \frac{0.8979788}{(\frac{0.9610394}{0.6342866})} = 0.5926666$$

Since we know that the slpe is

$$\beta_0 = \bar{y} - \beta_1 \bar{x} = -0.2712917 - 0.5926666 \times 0.1292699 = -0.3479057$$

Therefore we now have the coefficient formula as:

$$y = \beta_0 + \beta_1 x = -0.3479057 + 0.5926666x$$

Now let's calculate the Y with the first 5 value of the x:

$$y = \beta_0 + \beta_1 x = -0.3479057 + 0.5926666 \times -0.6264538 = -0.7191839$$
$$y = \beta_0 + \beta_1 x = -0.3479057 + 0.5926666 \times 0.1836433 = -0.2390664$$
$$y = \beta_0 + \beta_1 x = -0.3479057 + 0.5926666 \times -0.8356286 = -0.8431549$$
$$y = \beta_0 + \beta_1 x = -0.3479057 + 0.5926666 \times 1.5952808 = 0.5975639$$
$$y = \beta_0 + \beta_1 x = -0.3479057 + 0.5926666 \times 0.3295078 = -0.1526174$$

Since we know:

$$se_{\hat{y}} = \sqrt{\frac{\sum(y_i - \hat{y}_i)^2}{n-2}}$$

and the upper side of the $se_{\hat{y}}$ formula is actually SSE, so we can calculate the SSE result this way:

$SSE = \sum_i (y_i - \hat{y}_i)^2 = (-0.6456574 - (-0.7191839))^2 + (0.1788445 - (-0.2390664))^2 + (-0.9780474 - (-0.8431549))^2 + (0.5770849 - 0.5975639)^2 + (-0.4886831 - (-0.1526174))^2 = 0.3116112$

We can first get the TSS here:

$TSS = \sum_i (y_i - \bar{y})^2 = (-0.6456574 - (-0.2712917))^2 + (0.1788445 - (-0.2712917))^2 + (-0.9780474 - (-0.2712917))^2 + (0.5770849 - (-0.2712917))^2 + (-0.4886831 - (-0.2712917))^2 = 1.609278$

Since we know:

$$se_{\hat{y}} = \sqrt{\frac{\sum(y_i - \hat{y}_i)^2}{n-2}}$$

and the upper side of the $se_{\hat{y}}$ formula is actually SSE, so we can calculate the result this way:

$$se_{\hat{y}} = \sqrt{\frac{\sum(y_i - \hat{y}_i)^2}{n-2}} = \sqrt{\frac{SSE}{n-2}} = \sqrt{\frac{0.3116112}{5-2}} = 0.3222893$$

and we also have this formula for the formula of the coefficient:

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

$= \frac{0.3222893}{\sqrt{(-0.6264538-0.1292699)^2+(0.1836433-0.1292699)^2+(-0.8356286-0.1292699)^2+(1.5952808-0.1292699)^2+(0.3295078-0.1292699)^2}} =$ 0.1676775

By referring to the test book, we have:

$$R^2 = \frac{TSS - SSE}{TSS} = (1.609278 - 0.3116112)/1.609278 = 0.8063658$$

since the formula of the adjust R squared is here, and df_t=n-1, df_e=n-k-1 (where k is the number of variables and it is 1 (only x). So let's calculate:

$$\text{adjusted } R^2 = \frac{TSS/df_t - SSE/df_e}{TSS/df_t} = \frac{1.609278/(5-1) - 0.3116112/(5-1-1)}{1.609278/(5-1)}$$

Now we have a coefficient on x as 0.5926666, its standard error as 0.1676775, and the adjusted R2 as 0.7418211.

4.

a.

```r
set.seed(1)
y4 <-0.1+0.2*x-0.5*x^2+epsilon
m2 <- lm(y4 ~ x+I(x^2))
summary(m2)
```

```
##
## Call:
## lm(formula = y4 ~ 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
```

By referring to the P value we got here, since both of them less than 0.05, we could say that x and x^2 are statistically significant toward the dependent variable y, and the change of x or X^2 value will also strongly impact the y.

b.

Since we have been given the formula of the new Y as $y4 = 0.1 + 0.2 \times x - 0.5 \times x^2 + epsilon$, so we ca just apply x as 1 and 2 into this formula directly here:

```
# we ignored the epsilon since they both have it:
(0.1 + 0.2 * 2 - 0.5 * 2^2) - (0.1 + 0.2 * 1 - 0.5 * 1^2)
```

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

The exact effect on y of increasing x by 1 unit from 1 to 2 would be -1.3.

c.

Since we need to use what we had from 4a, so I will use the model I created in 4a `m2` here for this question:

```
# let's load the coefficient column in to the formula
m2$coefficients[1]+m2$coefficients[2]*(-0.7)+m2$coefficients[3]*((-0.7)^2)-
  (m2$coefficients[1]+m2$coefficients[2]*(-0.5)+m2$coefficients[3]*((-0.5)^2))
```

```
## (Intercept)
##  -0.1919733
```

Based on the coefficients estimated from 4(a), the effect on y of changing x from -0.5 to -0.7 is -0.1919733 .

5.

a.

```
set.seed(1)
x2 <- rnorm(100, mean=-1, sd=1)
y5 <- 0.1+0.2*x-0.5*x*x2+epsilon

# again, we ignored the epsilon since they both have it:
(0.1+0.2*mean(x)-0.5*mean(x)*1)-(0.1+0.2*mean(x)-0.5*mean(x)*0)
```

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

Based on the known coefficients, what is the exact effect of increasing x2 from 0 to 1 with x held at its mean is the y will be -0.05444368.

b.

```
# let's first create the regression model:
m3<- lm(y5 ~ x + x2 + I(x*x2))
summary(m3)
```

```
##
## Call:
## lm(formula = y5 ~ x + x2 + I(x * x2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9650 -0.6254 -0.1288  0.5803  2.2700
##
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.15672    0.11766   1.332    0.186
## x            0.09824    0.12901   0.761    0.448
## x2                 NA         NA      NA       NA
## I(x * x2)   -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:  0.4729, Adjusted R-squared:  0.462
## F-statistic: 43.51 on 2 and 97 DF,  p-value: 3.259e-14
```

```
# Use coefficient from m3 output:
m3$coefficients[1]+m3$coefficients[2]*(-0.7)+m2$coefficients[3]*(1^2)+m3$coefficients[4]*(-0.7*(1)^2)-
(m3$coefficients[1]+m3$coefficients[2]*(-0.5)+m2$coefficients[3]*(1^2)+m3$coefficients[4]*(-0.5*(1)^2))
```

```
## (Intercept)
##   0.1041363
```

Based on the regression-estimated coefficients, what is the effect on y of shifting x from -0.5 to -0.7 with $x2$ held at 1 is 0.1041363 .

c.

So for this F test we should have a hypothesis as following

```
m4 <- lm(y5~x)
summary(m4)
```

```
##
## Call:
## lm(formula = y5 ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.84054 -0.79225  0.01347  0.82817  2.66648
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.3346     0.1195  -2.800  0.00616 **
## x             0.6223     0.1328   4.688 8.94e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.186 on 98 degrees of freedom
## Multiple R-squared:  0.1832, Adjusted R-squared:  0.1748
## F-statistic: 21.97 on 1 and 98 DF,  p-value: 8.937e-06
```

Since we know

$$F = \frac{(R_u^2 - R_r^2)/(k-1)}{(1-R_u^2)/(n-k-1)}$$

The u here indicates the unrestricted complete model (`m3`) and r indicates restricted (reduced) model (`m4`), and since we have 2 additional variables in the unrestricted complete model (`m3`) so k=2 and df1=2-1=2. In the denominator, the n=100, k=2, so df2=100-2-1=97. now let's take out the R^2 value for both models:

```
summary(m3)
```

```
##
## Call:
## lm(formula = y5 ~ x + x2 + I(x * x2))
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9650 -0.6254 -0.1288  0.5803  2.2700
##
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.15672    0.11766   1.332    0.186
## x            0.09824    0.12901   0.761    0.448
## x2                NA         NA      NA       NA
## I(x * x2)   -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:  0.4729, Adjusted R-squared:  0.462
## F-statistic: 43.51 on 2 and 97 DF,  p-value: 3.259e-14
```

```
summary(m4)
```

```
##
## Call:
## lm(formula = y5 ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.84054 -0.79225  0.01347  0.82817  2.66648
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.3346     0.1195  -2.800  0.00616 **
## x             0.6223     0.1328   4.688 8.94e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.186 on 98 degrees of freedom
## Multiple R-squared:  0.1832, Adjusted R-squared:  0.1748
## F-statistic: 21.97 on 1 and 98 DF,  p-value: 8.937e-06
```

and we can now apply the R^2 of complete model 0.4729 and R^2 of reduced model 0.1832 into formula we had above:

$$F = \frac{(R_u^2 - R_r^2)/(k-1)}{(1-R_u^2)/(n-k-1)} = \frac{(0.4729 - 0.1832)/(2-1)}{(1-0.4729)/(100-2-1)} = 53.31227$$

and now we can calculate the p value of the F test:

```
pf(53.31227,1,97,lower.tail=F)
```

```
## [1] 7.926222e-11
```

Since we got a p value less than 0.05 and we can reject the null hypothesis to conclude that the complete model might be preferred by the F test rather than the reduced model since the complete model will include more confounding factors even though the x2 may not really meaningful in this model but the interaction term may still somehow improved the model. It actually makes sense as we note that the R^2 has gone up considerably between complete model and the reduced one. That is because there is a very strong correlation between y and interaction terms.

6.

a.

```
set.seed(1)
f <- c(rep("a", 100), rep("b", 100), rep("c", 100))

# in distinguishing with previous variables we call x1(x6.1), x2(x6.2) differently here
x6.1 <- c(rnorm(100, 1, 2), rnorm(100, 0, 1), rnorm(100, 1, 0.5))
x6.2 <- c(rnorm(100, 1, 2), rnorm(100, 1, 1), rnorm(100, 0, 0.5))

# we use cbind here so that we can attach the cluster column later easily
# meanwhile, I think the prompt actually says we need a dataset with 300 obs
# if we rbind then it will only be 2 obs
v <- as.data.frame(cbind(x6.1, x6.2, f))
v$f <- as.factor(v$f)
```

```
set.seed(1)
kout <- kmeans(v[,1:2], 3, 25)
centroids<-kout$centers
topvars_centroid1 <- centroids[1,order(centroids[1,])]
topvars_centroid2 <- centroids[2,order(centroids[2,])]
topvars_centroid3 <- centroids[3,order(centroids[2,])]

# check out the kout result with high scoring factors:
tail(topvars_centroid1)
```

```
##       x6.1       x6.2
## -0.6477379  0.3995778
```

```
tail(topvars_centroid2)
```

```
##     x6.1     x6.2
## 1.028320 2.777811
```

```
tail(topvars_centroid3)
```

```
##        x6.1        x6.2
##  1.42612561 -0.07850055
```

From this result we can tell that our dataset was been classified into 3 clusters. The first centroid has a cluster for numbers around 0 (-0.6477379 to 0.3995778) and it seems all data points are very concentrated in a small region; the second centroid has a cluster with a result of number from 1.028320 to 2.777811; and the third centroid has a cluster with numbers from -0.07850055 to 1.4261561. There are some overlaps between first&third cluster and second&third clusters.

```
# we apply the cluster into the dataframe:
v$cluster <- kout$cluster
table(v$f, v$cluster)
```

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

After we apply the cluster result into the dataframe we can see that the ‘a’ factor group a has very bad result since it has data points distributed evenly in two clusters and the correct hit is low; and ‘b’ is better with 60 correctly hitted data points, the ‘c’ has the most ideal result as its data points’ distribution is concentrated and it has 90 data points in the correct cluster. In order to compare what we had from the centriod with the true mean, we can use either scale or calculate the mean of each cluster by hand, I choosed to use the second method here. Now let’s calculate the true mean by filter out 3 dataset based on the f:

```
v$x6.1<-as.numeric(v$x6.1)
v$x6.2<-as.numeric(v$x6.2)

# create new datasets based on factor type
# true center calculated by using mean method:
a <- v %>%filter(f=="a")
c(mean(a$x6.1),mean(a$x6.2))
```

```
## [1] 1.217775 1.103204
```

```
b <- v %>%filter(f=="b")
c(mean(b$x6.1),mean(b$x6.2))
```

```
## [1] -0.03780808  0.96086576
```

```
c <- v %>%filter(f=="c")
c(mean(c$x6.1),mean(c$x6.2))
```

```
## [1]  1.01483677 -0.02225968
```

```
# centroids:
centroids
```

```
##         x6.1        x6.2
## 1 -0.6477379  0.39957780
## 2  1.0283202  2.77781098
## 3  1.4261256 -0.07850055
```

So here we can compare the centroids with the true center,though they are not very similar but there still some traces for us to recognize their features and pair them up. In both of them, we have a center with range around the 0 ([-0.03780808,0.96086576] vs [-0.6477379,0.39957780]), these two ranges match quite ideal; and then a center with range from negative to around 1 ([-0.07850055,1.4261256] vs [-0.03780808,0.96086576]) and these two center ranges also match well; and the last one we have a center with range bigger than 1 ([1.103204,1.217775] vs [1.0283202,2.77781098]), the centroid range of this pair is wider than the true center range here and we can say they don’t really match match well.

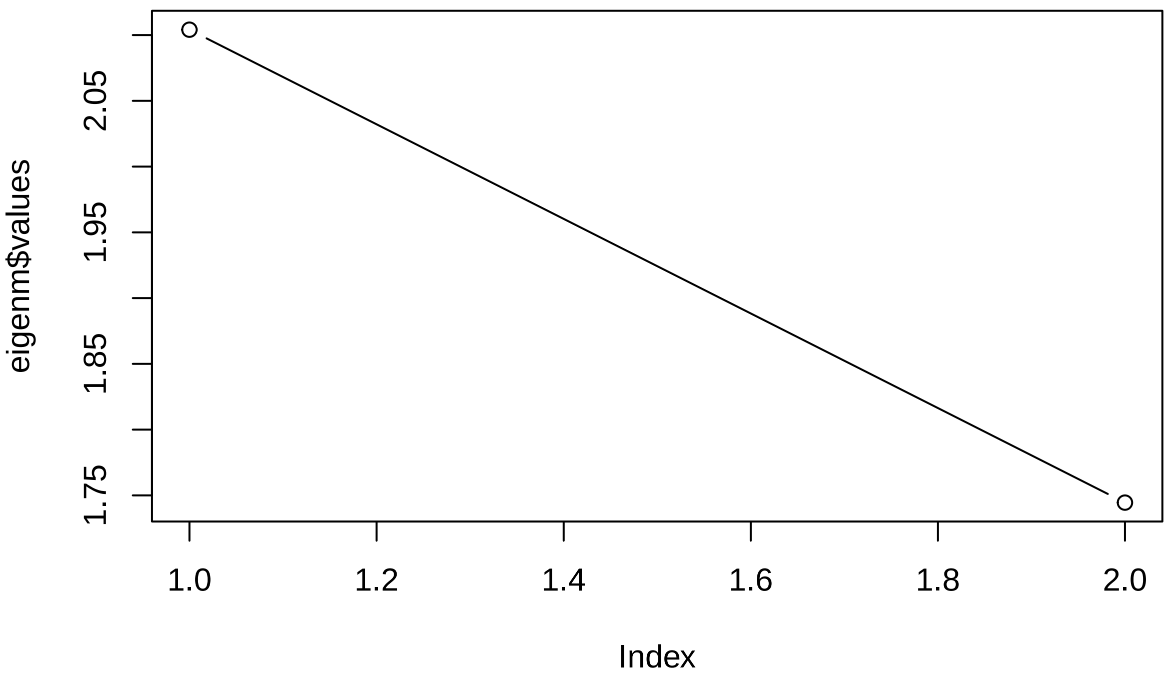
b.

```
v<-subset(v, select = -c(f,cluster) )
```

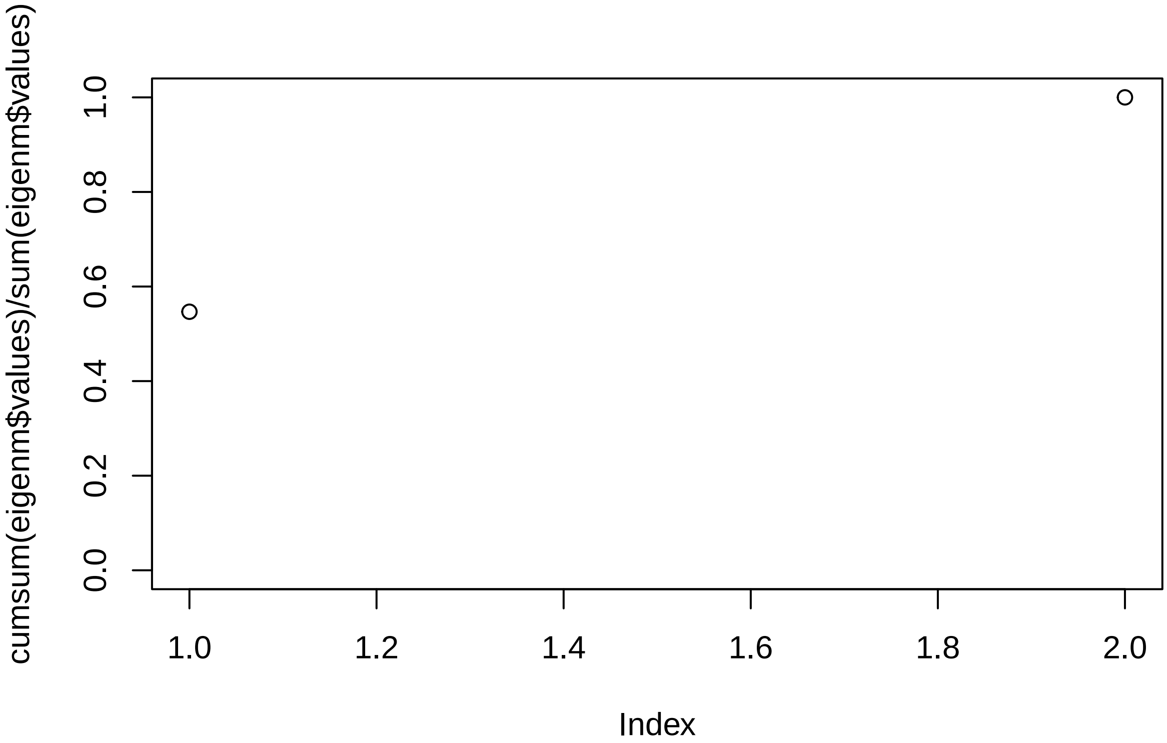
```
v$x6.1<-as.numeric(v$x6.1)
v$x6.2<-as.numeric(v$x6.2)
```

```
# calculate the principal components of the dataseT
pcaA <- prcomp(v)
pcaA1 <- pcaA$rotation[,1]

# calculate total variance explained by each principal component
var_explained <- pcaA$sdev^2 / sum(pcaA$sdev^2)
covm <- cov(v)
eigenm <- eigen(covm)
```

```r
plot(eigenm$values,type="b")
```

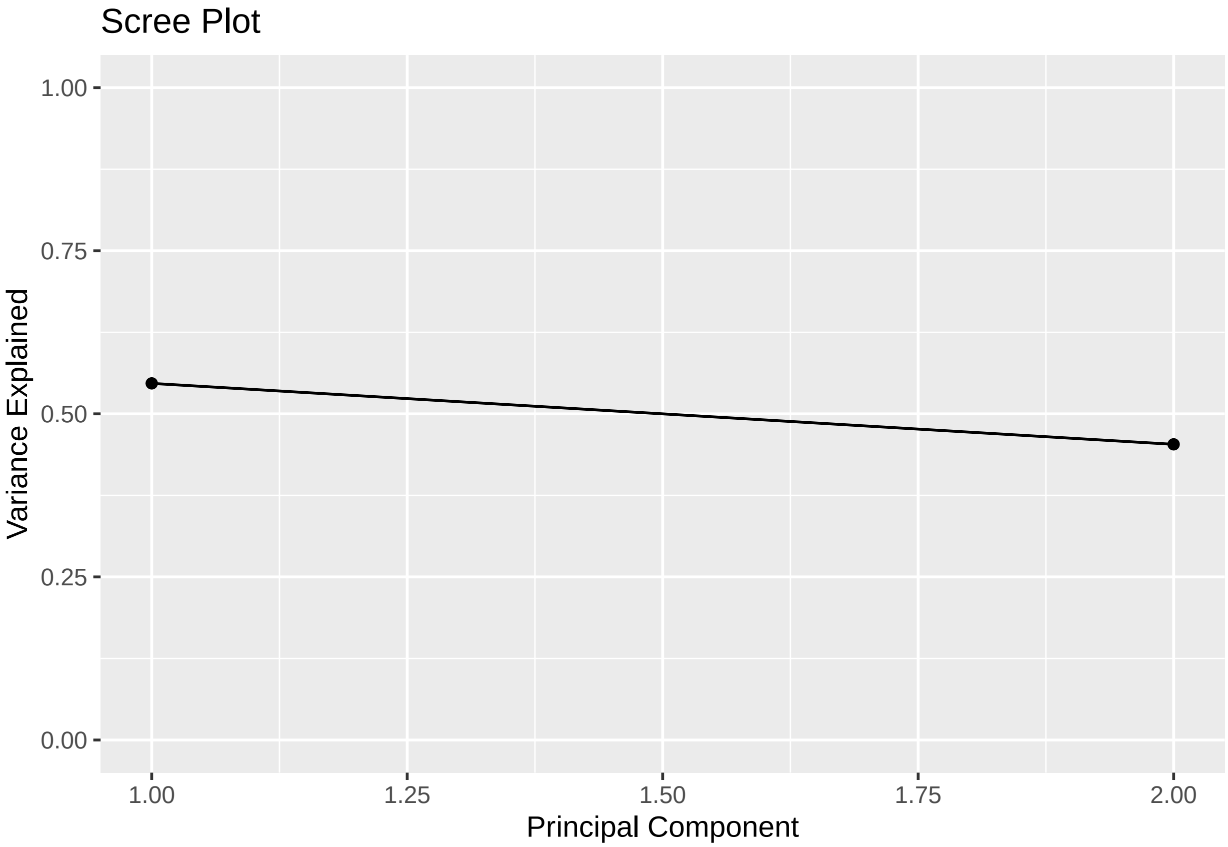


```r
plot(cumsum(eigenm$values)/sum(eigenm$values),ylim=c(0,1))
```



```r
# A better visualization of the scree plot
qplot(c(1:2), var_explained) +
  geom_line() +
  xlab("Principal Component") +
  ylab("Variance Explained") +
  ggtitle("Scree Plot") +
  ylim(0, 1)
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
```



From the result of the `cbind` dataset we can tell that the first column of factors can actually explained about 58% of the entire population . The main result why we have a such wired result is because that the x1 and x2 we created are actually related to each other as a group of their factor share the same mean and sd. The x2 can explain about 45% of the entire dataset which is also a big part of the entire population, so I would say, given the result of the PCA and similarity of x1 and x2 factors, we probably want to keep both x1 and x2 factors given they have a relatively similar percentage of description toward the entire data points, and it is really hard for us to just discard either one of them in this case. So what we found here is actually also related to the previous cluster result, since we know that the cluster results tells us that only the factor group c has a high hit into the correct cluster and the rest two factor group don't perform so well but they still partially hitted into the correct cluster. In this case, we can't really tell those correctly hitted data points belong to the x1 or x2, therefore we want to retain both of them so that we won't lose those partially correct data points from the x1 and x2.

7.

```
df<-read.csv("massachussets_crime_final.csv")
```

```
# we first see how many empty values included in this data:
colSums(is.na(df))
```

```
##                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
```

Ok, it seems only a few of them are missing values, so I want to just remove them:

```
df<-na.omit(df)
colSums(is.na(df))
```

```
##                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
```

by observing through the data, we also found out there are some ‘,’ exist in the numeric value which made it actually a character and we want to remove them. Also, we don’t need the city column here:

```
# remove ,
df$Population <- gsub(",", "", df$Population)
df$Violent.crime <- gsub(",", "", df$Violent.crime)
df$Robbery<- gsub(",", "", df$Robbery)
df$Aggravated.assault<- gsub(",", "", df$Aggravated.assault)
df$Property.crime<- gsub(",", "", df$Property.crime)
df$Burglary<- gsub(",", "", df$Burglary)
df$Larceny..theft<-gsub(",", "", df$Larceny..theft)
```

```
# a new dataset for later use
dfq9<-df

# remove city
df<-subset(df, select = -c(City) )

# as numeric:
df <- sapply(df,as.numeric)
df<-as.data.frame(df)
str(df)
```

```
## 'data.frame':    280 obs. of  11 variables:
##  $ Population         : num  16448 23780 10533 8028 28736 ...
##  $ Violent.crime      : num  23 32 12 26 82 25 99 8 2 34 ...
##  $ Murder_MANSLAUGHTER: num  4 0 0 0 0 0 0 0 0 0 ...
##  $ Rape               : num  5 6 5 10 13 3 28 6 0 5 ...
##  $ Robbery            : num  3 2 0 2 8 3 2 1 0 8 ...
##  $ Aggravated.assault : num  11 24 7 14 61 19 69 1 2 21 ...
##  $ Property.crime     : num  153 66 35 94 376 132 173 215 0 167 ...
##  $ Burglary           : num  23 13 14 34 133 18 55 28 0 16 ...
##  $ Larceny..theft     : num  122 50 19 59 228 107 103 180 0 139 ...
##  $ Motor_vehicle_theft: num  8 3 2 1 15 7 15 7 0 12 ...
##  $ Arson              : num  1 0 0 2 1 0 2 0 0 3 ...
```

now let’s see how many outliers exist in the data. Outliers increase the variability in the data, which decreases statistical power. Consequently, excluding outliers can cause our results to become statistically significant. For admission numbers here, we need to find out outliers. When we expect our data has a normal distribution, Z-scores may help us to identify how modd the real observation is.

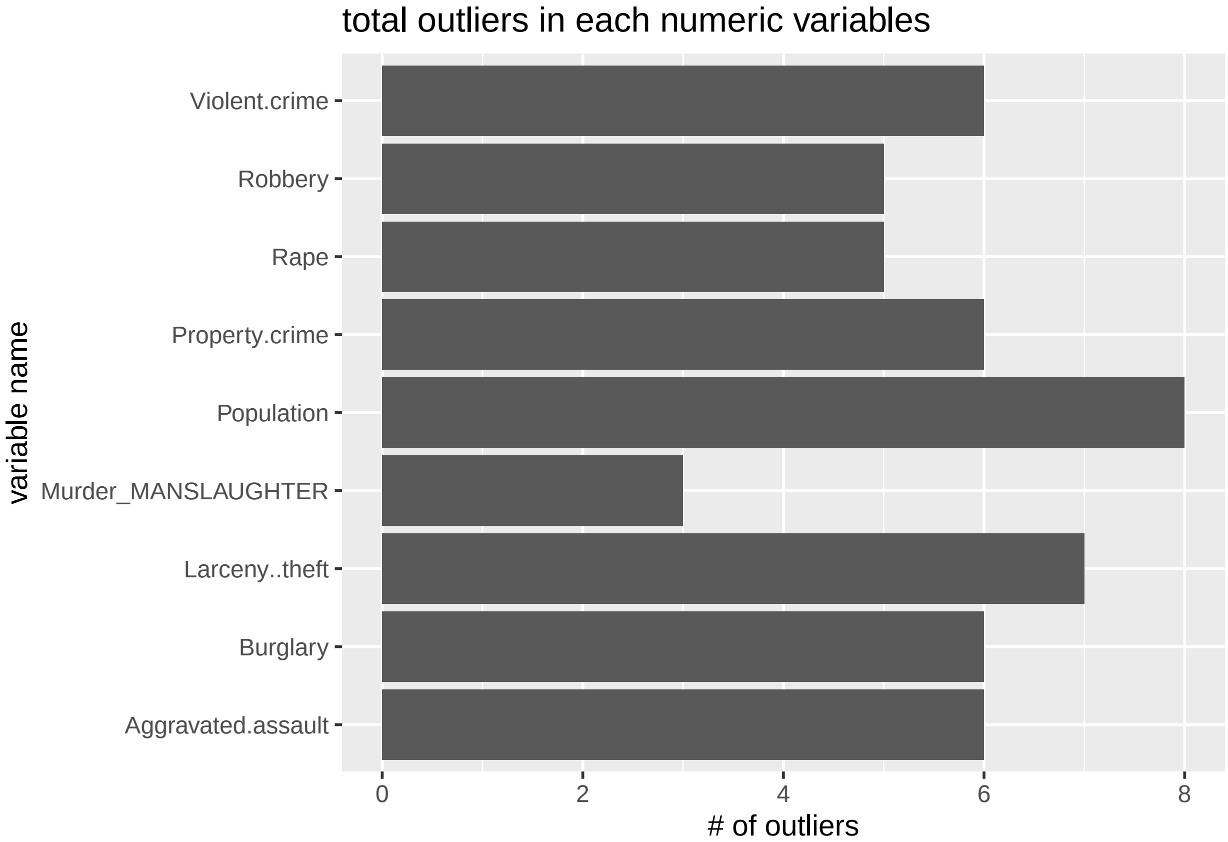
The degree of standard deviations around the mean decrease for each value is represented by the Z-score. The Z score equals to zero refers to the mean value, and in order to calculate it, we need to take the original admissions, calculate the mean and the use it to minus origin admission value, then divide by the SD to get the Z-score of the admission.

The formula of Z score in this case is:

$$Z = \frac{x - \bar{x}}{sd_x}$$

A huge the distance between an variable's Z-score and zero can indicate they are actually extravagant. Z ± 3 is a widely accepted cut-off point for identifying outliers, and we will apply the same logistics in our visualization here.Number of outliers in each column is shown in the figure below::

```
outliers <- function(x){
  # use z score for outliers
  sum(abs((x[!is.na(x)] - mean(x, na.rm = TRUE)) / sd(x , na.rm = TRUE)) > 3)
}

enframe(sapply(df[, 1:9], outliers)) %>%
  ggplot() +
  geom_bar( mapping = aes(x=name, y = value ), stat='identity') +
  coord_flip() +
  labs( x = "variable name", y = "# of outliers", title = "total outliers in each numeric variables")
```



Well the outliers aren't that much in the dataset, and given the fact our dataset is not very big so that we don't want to lose more observations by removing outliers. We can just leave outliers here.

The violent.crime is the sum of the Murder_MANSLAUGHTER, Rape,Robbery, Aggravated.assault; the property crime is the sum of the "Burglary and Larceny..theft and Motor_vehicle_theft. So let's split the data set before we dive in:

```
# this df only for crime types
types<-df%>%select(Population,Violent.crime,Property.crime,Arson)

# this is for crimes
crimes<-df%>%select(Population,Murder_MANSLAUGHTER,Rape,Robbery,Aggravated.assault,Burglary,
                    Larceny..theft,Motor_vehicle_theft,Arson)
```

```
# structure of the new datasets:
str(crimes)
```

```
## 'data.frame':    280 obs. of  9 variables:
##  $ Population         : num  16448 23780 10533 8028 28736 ...
##  $ Murder_MANSLAUGHTER: num  4 0 0 0 0 0 0 0 0 0 ...
##  $ Rape               : num  5 6 5 10 13 3 28 6 0 5 ...
##  $ Robbery            : num  3 2 0 2 8 3 2 1 0 8 ...
##  $ Aggravated.assault : num  11 24 7 14 61 19 69 1 2 21 ...
##  $ Burglary           : num  23 13 14 34 133 18 55 28 0 16 ...
##  $ Larceny..theft     : num  122 50 19 59 228 107 103 180 0 139 ...
##  $ Motor_vehicle_theft: num  8 3 2 1 15 7 15 7 0 12 ...
##  $ Arson              : num  1 0 0 2 1 0 2 0 0 3 ...
```

```
str(types)
```

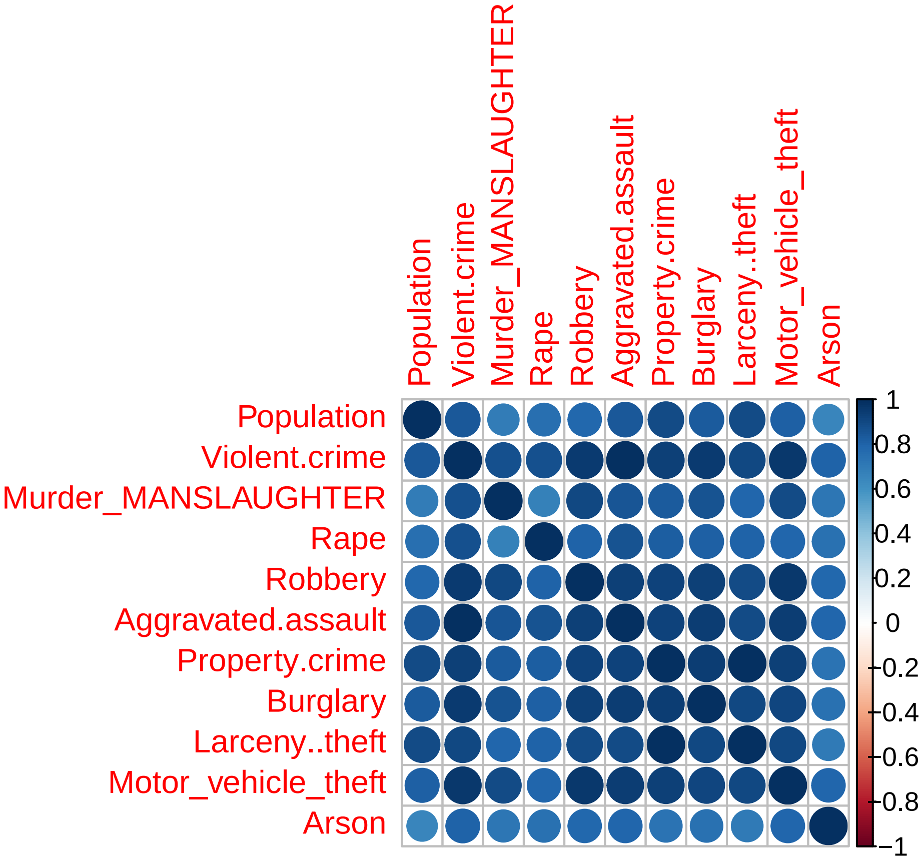
```
## 'data.frame':    280 obs. of  4 variables:
##  $ Population    : num  16448 23780 10533 8028 28736 ...
##  $ Violent.crime : num  23 32 12 26 82 25 99 8 2 34 ...
##  $ Property.crime: num  153 66 35 94 376 132 173 215 0 167 ...
##  $ Arson         : num  1 0 0 2 1 0 2 0 0 3 ...
```

So, given the fact, regression result can only explain correlation(association) but not the causation, we only propose the association relationship in our hypothesis. For this dataset, my hypothesis is that I think the population might be associated with crime cases numbers in the table. Since we know population may somehow vary due to the unsteadily community with crimes, but on the other hand more population might cause more “potential criminals” which raise crime cases. We first want to see the regression result:

```
# correlation matrix result:
cor(df, use='complete.obs')
```

```
##                     Population Violent.crime Murder_MANSLAUGHTER      Rape
## Population           1.0000000     0.8463782           0.6924567 0.7591273
## Violent.crime        0.8463782     1.0000000           0.8740058 0.8785449
## Murder_MANSLAUGHTER  0.6924567     0.8740058           1.0000000 0.6743528
## Rape                 0.7591273     0.8785449           0.6743528 1.0000000
## Robbery              0.7895348     0.9572792           0.9064118 0.8039970
## Aggravated.assault   0.8483604     0.9960693           0.8555946 0.8646799
## Property.crime       0.8964237     0.9395640           0.8382685 0.8257456
## Burglary             0.8378683     0.9519248           0.8614980 0.8174382
## Larceny..theft       0.8971656     0.9057407           0.7993429 0.8086405
## Motor_vehicle_theft  0.8191517     0.9625588           0.8967231 0.7940624
## Arson                0.6554161     0.8050913           0.7299702 0.7434084
##                       Robbery Aggravated.assault Property.crime  Burglary
## Population          0.7895348          0.8483604      0.8964237 0.8378683
## Violent.crime       0.9572792          0.9960693      0.9395640 0.9519248
## Murder_MANSLAUGHTER 0.9064118          0.8555946      0.8382685 0.8614980
## Rape                0.8039970          0.8646799      0.8257456 0.8174382
## Robbery              1.0000000          0.9313499      0.9281737 0.9342456
## Aggravated.assault   0.9313499          1.0000000      0.9270530 0.9427992
## Property.crime       0.9281737          0.9270530      1.0000000 0.9471130
## Burglary             0.9342456          0.9427992      0.9471130 1.0000000
## Larceny..theft       0.8941026          0.8927475      0.9939976 0.9099879
## Motor_vehicle_theft  0.9664991          0.9497198      0.9381592 0.9179922
## Arson                0.7872696          0.7929931      0.7389608 0.7469434
##                      Larceny..theft Motor_vehicle_theft     Arson
## Population                0.8971656           0.8191517 0.6554161
## Violent.crime             0.9057407           0.9625588 0.8050913
## Murder_MANSLAUGHTER       0.7993429           0.8967231 0.7299702
## Rape                      0.8086405           0.7940624 0.7434084
## Robbery                   0.8941026           0.9664991 0.7872696
## Aggravated.assault        0.8927475           0.9497198 0.7929931
## Property.crime            0.9939976           0.9381592 0.7389608
## Burglary                  0.9099879           0.9179922 0.7469434
## Larceny..theft            1.0000000           0.9072865 0.7074577
## Motor_vehicle_theft       0.9072865           1.0000000 0.7938008
## Arson                     0.7074577           0.7938008 1.0000000
```

```
# visualization:
corrplot(cor(df, use='complete.obs'), method="circle")
```



The correlation result tells us that the population has strong correlations with basically every variables in the original data set, but as we separated the dataset into two new datasets, we can first try to use population regress all variables in each datasets to see what we can get from them:

```
lm1<-lm(Population~Violent.crime+Property.crime , data = types)
summary(lm1)
```

```
##
## Call:
## lm(formula = Population ~ Violent.crime + Property.crime, data = types)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -50217  -6219  -2440   4327  54093
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    10603.342    722.058  14.685   <2e-16 ***
## Violent.crime      5.677     12.526   0.453    0.651
## Property.crime    46.439      4.182  11.104   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10650 on 277 degrees of freedom
## Multiple R-squared:  0.8037, Adjusted R-squared:  0.8023
## F-statistic: 567.1 on 2 and 277 DF,  p-value: < 2.2e-16
```

Well, this result tells us that the Population is highly associated with property crime since it has a p value less than 0.05 and we want to double check our result by using another table:

```
lm2<-lm(Population~. , data = crimes)
summary(lm2)
```

```
##
## Call:
## lm(formula = Population ~ ., data = crimes)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -47068  -5870  -1864   3793  51937
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          8718.639    784.408  11.115  < 2e-16 ***
## Murder_MANSLAUGHTER -1969.538    902.641  -2.182   0.0300 *
## Rape                 -135.815    124.637  -1.090   0.2768
## Robbery              -218.006     87.938  -2.479   0.0138 *
## Aggravated.assault    114.883     25.339   4.534  8.7e-06 ***
## Burglary               20.890     26.617   0.785   0.4332
## Larceny..theft         59.187      5.217  11.346  < 2e-16 ***
## Motor_vehicle_theft   -15.094     66.911  -0.226   0.8217
## Arson                 208.927    372.728   0.561   0.5756
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9831 on 271 degrees of freedom
## Multiple R-squared:  0.8363, Adjusted R-squared:  0.8315
## F-statistic: 173.1 on 8 and 271 DF,  p-value: < 2.2e-16
```

Now, this regression result just provide us more details that Murder_MANSLAUGHTER,Robbery, Aggravated.assault,Larceny..theft are highly associated with population of a certain county in MA. And the Larceny..theft has the smallest p value here which makes sense as this is actually one of the comm-est crime in the US which might cause people want to move away from their communities. So, for now, our null hypothesis will be rejected and conclude that the county population is associated with crime cases, and population may vary due to the unsteadily community with certain crimes exist, or more population may cause higher chance to have criminals within it. However, we might want to refine our model by using a back ward stepwise feature selection, and we mainly focus on the BIC value and R^2 to decide which model is the best for us:

```
model <- regsubsets(Population~ ., data = crimes, method="backward")
sumresult <- summary(model)
model.subsets <- cbind(sumresult$which, sumresult$bic, sumresult$rsq, sumresult$adjr2)
model.subsets <- as.data.frame(model.subsets)

# attach the BIC and r square to the table
colnames(model.subsets)[10:12] <- c("BIC","R^2","adj R^2")

# check result:
model.subsets
```

```
##   (Intercept) Murder_MANSLAUGHTER Rape Robbery Aggravated.assault Burglary
## 1           1                   0    0       0                  0        0
## 2           1                   0    0       0                  1        0
## 3           1                   0    0       1                  1        0
## 4           1                   1    0       1                  1        0
## 5           1                   1    1       1                  1        0
## 6           1                   1    1       1                  1        1
## 7           1                   1    1       1                  1        1
## 8           1                   1    1       1                  1        1
##   Larceny..theft Motor_vehicle_theft Arson       BIC       R^2   adj R^2
## 1              1                   0     0 -446.3273 0.8049062 0.8042044
## 2              1                   0     0 -457.0581 0.8159823 0.8146536
## 3              1                   0     0 -478.3622 0.8328617 0.8310450
## 4              1                   0     0 -476.4816 0.8350877 0.8326889
## 5              1                   0     0 -471.8433 0.8356735 0.8326748
## 6              1                   0     0 -466.9686 0.8361190 0.8325172
## 7              1                   0     1 -461.6273 0.8362907 0.8320776
## 8              1                   1     1 -456.0451 0.8363214 0.8314896
```

Given the R^2 result are all pretty much similar we will mainly focus on BIC to find the best model for us. The model 3 has the smallest BIC here and we want to try out `Population~Robbery+Aggravated.assault+Larceny..theft`:

```
lm3<-lm(Population~Robbery+Aggravated.assault+Larceny..theft , data = crimes)
summary(lm3)
```

```
##
## Call:
## lm(formula = Population ~ Robbery + Aggravated.assault + Larceny..theft,
##     data = crimes)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -51367  -5573  -1902   3476  53747
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        8584.079    737.512  11.639  < 2e-16 ***
## Robbery            -289.760     54.884  -5.280 2.63e-07 ***
## Aggravated.assault  108.327     16.175   6.697 1.19e-10 ***
## Larceny..theft       59.983      4.388  13.669  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9844 on 276 degrees of freedom
## Multiple R-squared:  0.8329, Adjusted R-squared:  0.831
## F-statistic: 458.4 on 3 and 276 DF,  p-value: < 2.2e-16
```

So all of the independent variables has a significant impact toward the dependent variable, though we are not required to use better model for prediction but I still want to say this model worth a try if we want to predict the population of certain counties.

My conclusion from these regressions is that our null hypothesis will be rejected and we say that the county population is associated with crimes. High crime cases may cause the lose of population since people want to live safely, but high crimes may also due to high population at the same time. A place exist crimes such as those with p value less than 0.05 above can also indicate the population level there.

8.

This time I will intentionally to create a wrong regression here:

```
lm3<-lm(df$Population ~df$Property.crime)
summary(lm3)
```

```
##
## Call:
## lm(formula = df$Population ~ df$Property.crime)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -49370  -6214  -2477   4277  53511
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       10554.54     712.97   14.80   <2e-16 ***
## df$Property.crime    48.22       1.43   33.72   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10630 on 278 degrees of freedom
## Multiple R-squared:  0.8036, Adjusted R-squared:  0.8029
## F-statistic:  1137 on 1 and 278 DF,  p-value: < 2.2e-16
```

It seems the property.crime is significant to population, and let's try to regress the population with larceny..theft+property.crime:

```
lm3.1<-lm(df$Population ~df$Larceny..theft+df$Property.crime)
summary(lm3.1)
```

```
##
## Call:
## lm(formula = df$Population ~ df$Larceny..theft + df$Property.crime)
##
```

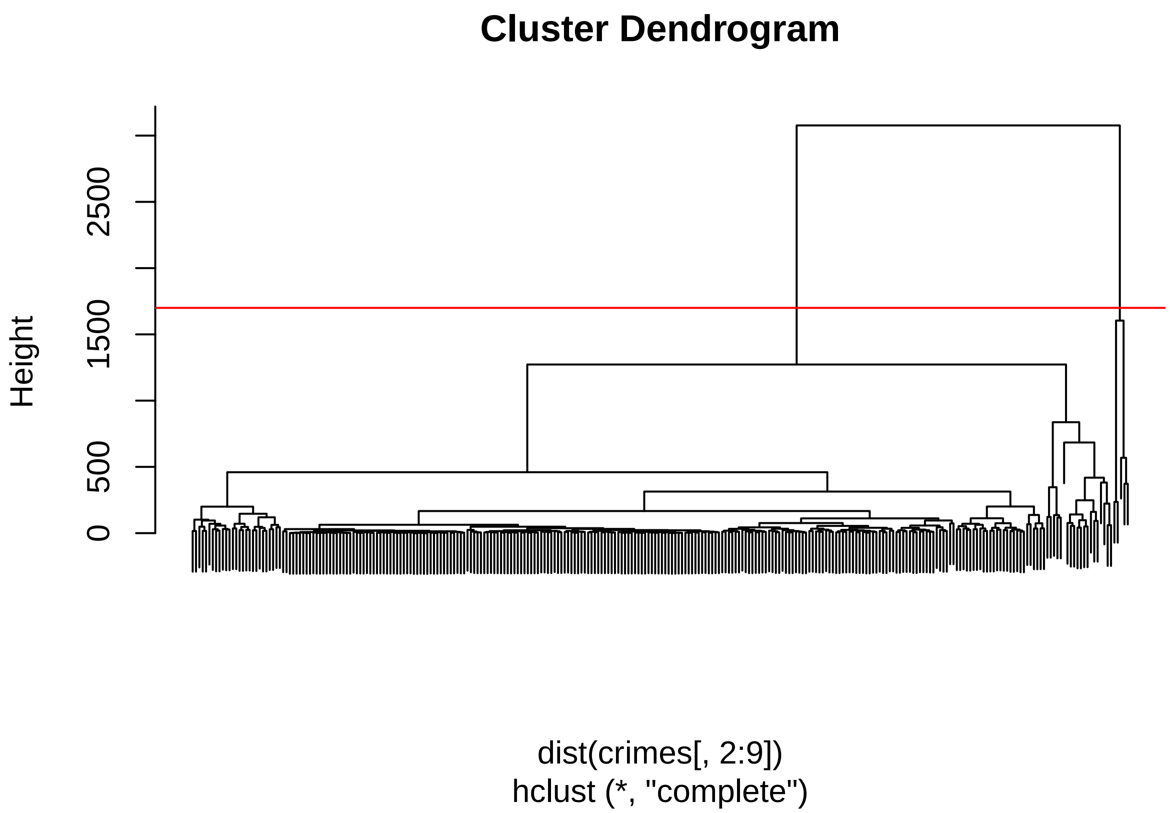
```
## Residuals:
##    Min     1Q Median     3Q    Max
## -46161  -5911  -2522   4062  51723
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       10186.14     729.56  13.962   <2e-16 ***
## df$Larceny..theft    37.99      17.93   2.119    0.035 *
## df$Property.crime    20.87      12.99   1.607    0.109
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10570 on 277 degrees of freedom
## Multiple R-squared:  0.8067, Adjusted R-squared:  0.8053
## F-statistic:   578 on 2 and 277 DF,  p-value: < 2.2e-16
```

Here we can referring to the text book to claim that we have a chained causation error in these variables.crime and in the second regression we can tell that Property.crime actually has no effect on Population at all; and it shows the danger of trusting in bivariate regressions. Either chained causation or spurious association error might exist here and we may need more investigation about their sociology or political science knowledge to interrupt which of these two error causes Property.crime no longer impact the population after we regress it with Larceny..theft. My guess here is that this is more like a spurious association issue, Larceny..theft -> population and Larceny..theft -> Property.crime by using the provided example in the text book chapter 9.1.

9.

We did both PCA and clustering analysis for this question. Assume we have no idea about this dataset at all and we want to first have a general idea of the crime population by using a cluster dendrogram plot:

```
hout <- hclust(dist(crimes[,2:9]),method="complete")
plot(hout,labels=FALSE)
abline(a=1700,b=0,col="red")
```



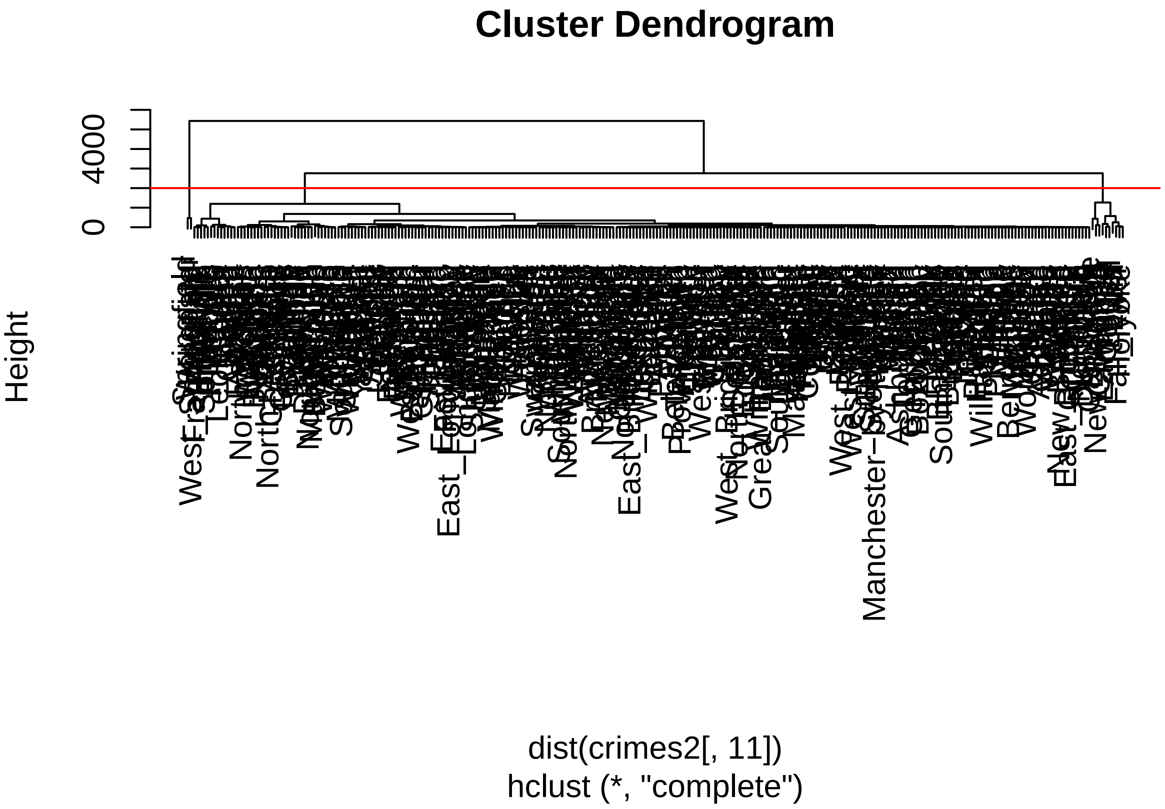
From the cluster dendrogram result we can tell that we might can have two clusters in general for this dataset. But this result is not so ideal as the nodes in either cluster are complicated so that we can't draw a conclusion easily. We want to improve the cluster result in this case.

Let's be creative, I want to create another cluster result and this time I would like to focus on the total crimes cases number based on each county so that we can dive deeper with the datasets:

```
# this is for crimes2
# we have both populaiton and city this time
crimes2<-dfq9%>%select(City,Population,Murder_MANSLAUGHTER,Rape,Robbery,
                       Aggravated.assault,Burglary,Larceny..theft,Motor_vehicle_theft,Arson)

# as.numeric
crimes2$Population<-as.numeric(crimes2$Population)
crimes2$Robbery<-as.numeric(crimes2$Robbery)
crimes2$Aggravated.assault<-as.numeric(crimes2$Aggravated.assault)
crimes2$Burglary<-as.numeric(crimes2$Burglary)
crimes2$Larceny..theft<-as.numeric(crimes2$Larceny..theft)

# create a new column called the crimes_total which is the total crimes number of each city
crimes2$crimes_total<-crimes2$Murder_MANSLAUGHTER+crimes2$Rape+crimes2$Robbery+
  crimes2$Aggravated.assault+crimes2$Burglary+crimes2$Larceny..theft+crimes2$Motor_vehicle_theft+
  crimes2$Arson

# check what we have here:
str(crimes2)
```

```
## 'data.frame':    280 obs. of  11 variables:
##  $ City               : chr  "Abington" "Acton" "Acushnet" "Adams" ...
##  $ Population         : num  16448 23780 10533 8028 28736 ...
##  $ 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            : num  3 2 0 2 8 3 2 1 0 8 ...
##  $ Aggravated.assault : num  11 24 7 14 61 19 69 1 2 21 ...
##  $ Burglary           : num  23 13 14 34 133 18 55 28 0 16 ...
##  $ Larceny..theft     : num  122 50 19 59 228 107 103 180 0 139 ...
##  $ 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 ...
##  $ crimes_total       : num  177 98 47 122 459 157 274 223 2 204 ...
##  - attr(*, "na.action")= 'omit' Named int 32
##   ..- attr(*, "names")= chr "32"
```

```
# new cluster basd on only the total crime cases and we label them with city names
hout2 <- hclust(dist(crimes2[,11]),method="complete")
plot(hout2,labels=crimes2$City)
abline(a=2000,b=0,col="red")
```



(Sorry for the terrible visualization here. The visualization here is kind of mess as all city names been squeezed into a very narrow space, you might want to test run my code and perform a same visualization in your terminal in order to have a clearer output.)

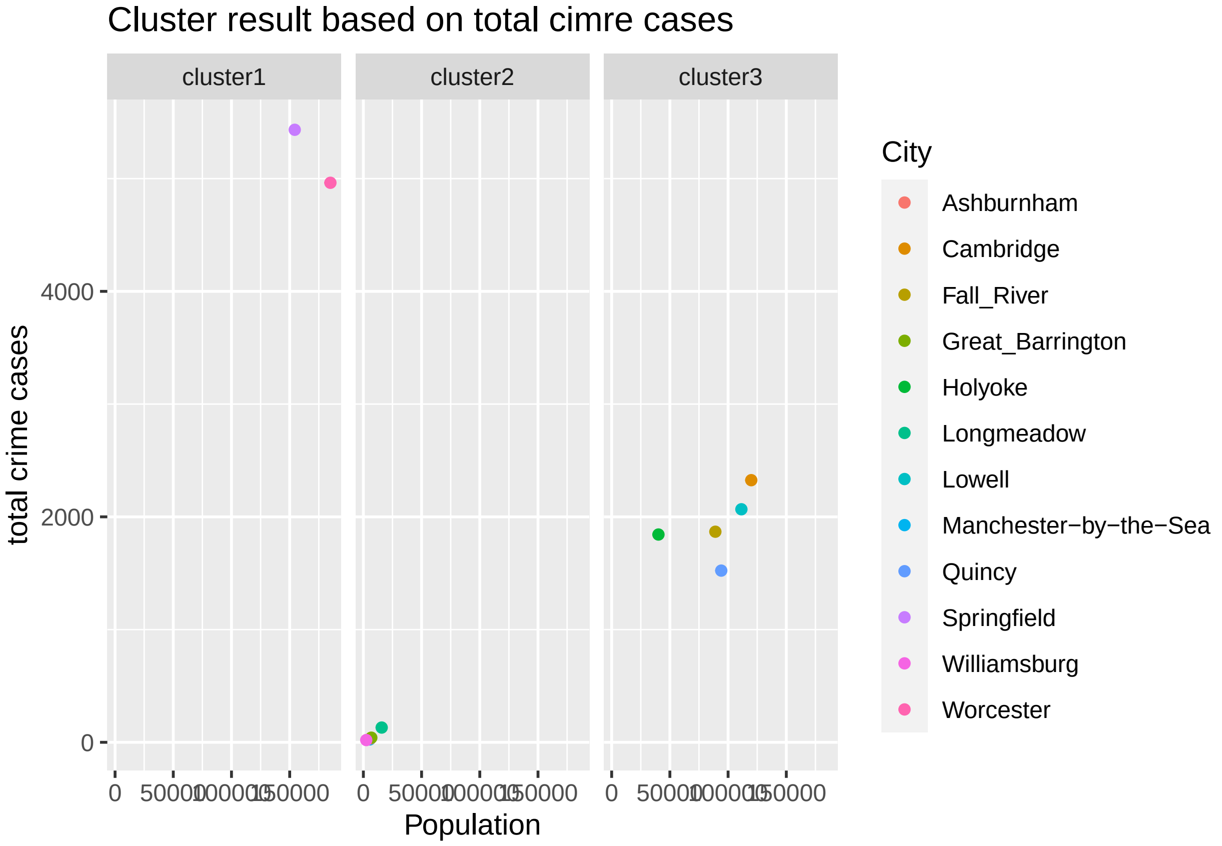
Well, it probably looks similar with what we had before but we can have 3 clusters easily this time, and it means we can divide the entire dataset into 3 clusters in general if we based on the total crimes cases. Let's call the cluster on the left as cluster 1, the one in the middle as cluster 2 and the one on the right as cluster 3. Since our assumption at the begaining is regarding to the association between Population and crimes. So, based on what we have in the cluster dendrogram, we want to dive deeper now by using different methods to see if we can have some results that related to what we got from early regression result. I want to pick 5 cities from each of these 3 clusters with population number and see what we can find:

```r
# create a new table with only 5 cities from each cluster:
# since we only have two cities in clkuster1: springfiled and worcester:
c1<-crimes2%>%filter(City=="Springfield")
c1.1<-crimes2%>%filter(City=="Worcester")

# we pick Longmeadow,Manchester-by-the-Sea,Ashburnham,Great_BarringtonN,Williamsburg
c2<-crimes2%>%filter(City=="Longmeadow")
c2.2<-crimes2%>%filter(City=="Manchester-by-the-Sea")
c2.3<-crimes2%>%filter(City=="Ashburnham")
c2.4<-crimes2%>%filter(City=="Great_Barrington")
c2.5<-crimes2%>%filter(City=="Williamsburg")

# Cambridge,Fall_River,Holyoke,Quincy,Lowell from cluster 3
c3<-crimes2%>%filter(City=="Cambridge")
c3.2<-crimes2%>%filter(City=="Fall_River")
c3.3<-crimes2%>%filter(City=="Holyoke")
c3.4<-crimes2%>%filter(City=="Quincy")
c3.5<-crimes2%>%filter(City=="Lowell")

# rbind:
samples<-rbind(c1,c1.1,c2,c2.2,c2.3,c2.4,c2.5,c3,c3.2,c3.3,c3.4,c3.5)
```

```r
# cluster column:
class<-c(rep("cluster1", 2), rep("cluster2", 5), rep("cluster3", 5))

# rbind to attach their respective cluster name:
samples<-cbind(samples,class)


samples
```

```
##                      City Population Murder_MANSLAUGHTER Rape Robbery
## 1            Springfield     154306                  20   81     358
## 2              Worcester     184945                  13   40     229
## 3             Longmeadow      15737                   0    3       0
## 4  Manchester-by-the-Sea       5423                   0    0       0
## 5             Ashburnham       6330                   0    2       0
## 6       Great_Barrington       6822                   0    2       1
## 7           Williamsburg       2489                   0    1       0
## 8              Cambridge     119908                   1   26      67
## 9             Fall_River      89066                   5   51     113
## 10               Holyoke      40178                   4   36      53
## 11                Quincy      94113                   1   31      61
## 12                Lowell     111423                   4   20     100
##    Aggravated.assault Burglary Larceny..theft Motor_vehicle_theft Arson
## 1                 938      746           2766                 493    31
## 2                 883      786           2637                 369     6
## 3                   7       28             82                  10     1
## 4                   2        4             19                   1     0
## 5                   6        7             18                   3     0
## 6                   8        5             23                   0     2
## 7                   1        4             14                   0     0
## 8                 240      161           1724                  98     8
## 9                 604      412            499                 163    21
## 10                252      213           1184                  97     4
## 11                282      216            875                  55     2
## 12                281      292           1190                 170    10
##    crimes_total    class
## 1          5433 cluster1
## 2          4963 cluster1
## 3           131 cluster2
## 4            26 cluster2
## 5            36 cluster2
## 6            41 cluster2
## 7            20 cluster2
## 8          2325 cluster3
## 9          1868 cluster3
## 10         1843 cluster3
## 11         1523 cluster3
## 12         2067 cluster3
```

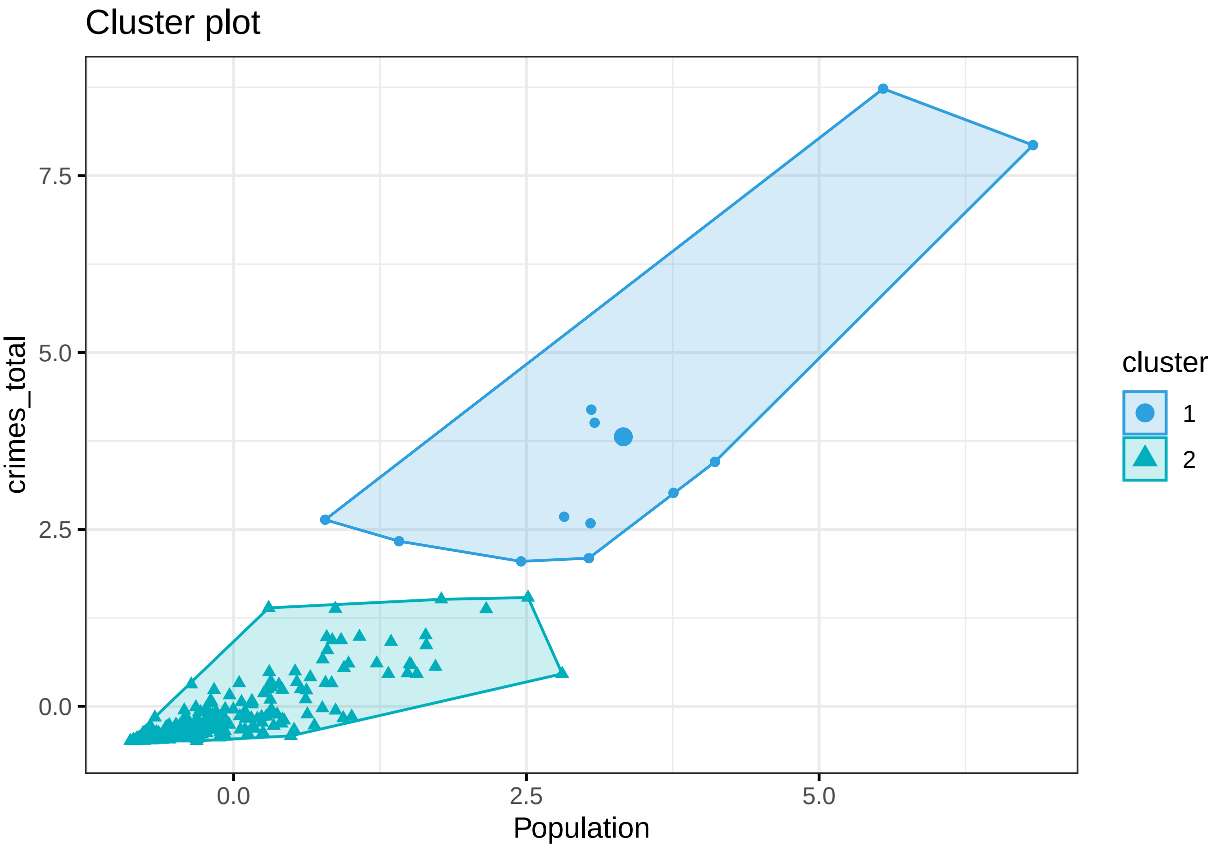
```
# visualizaiton, and put cities from each cluster into one plot:
ggplot(samples, aes(x = Population, y = crimes_total, label=City,fill = City, color = City )) +
  geom_point() +
  ggtitle("Cluster result based on total cimre cases") +
  ylab("total crime cases")+
  facet_wrap(. ~ class)
```



Great, there is a very clear pattern from each cluster plot. For the tw cities from cluster 1, they all have very high populaiton and crime cases; for cities from cluster 2, they all have low population and crimes;for cities from cluster 3, they have medium crime cases and population. An interesting thing is taht we can find the crime cases seems correlated to the population, just like if a city has high population then the crime cases will also pretty high and vice versa. Therefore, what we find here with the cluster analysis again proved the previous regression result that crimes and population are associated. Again,we can present this conclusion into a more obvious cluster result visualization:

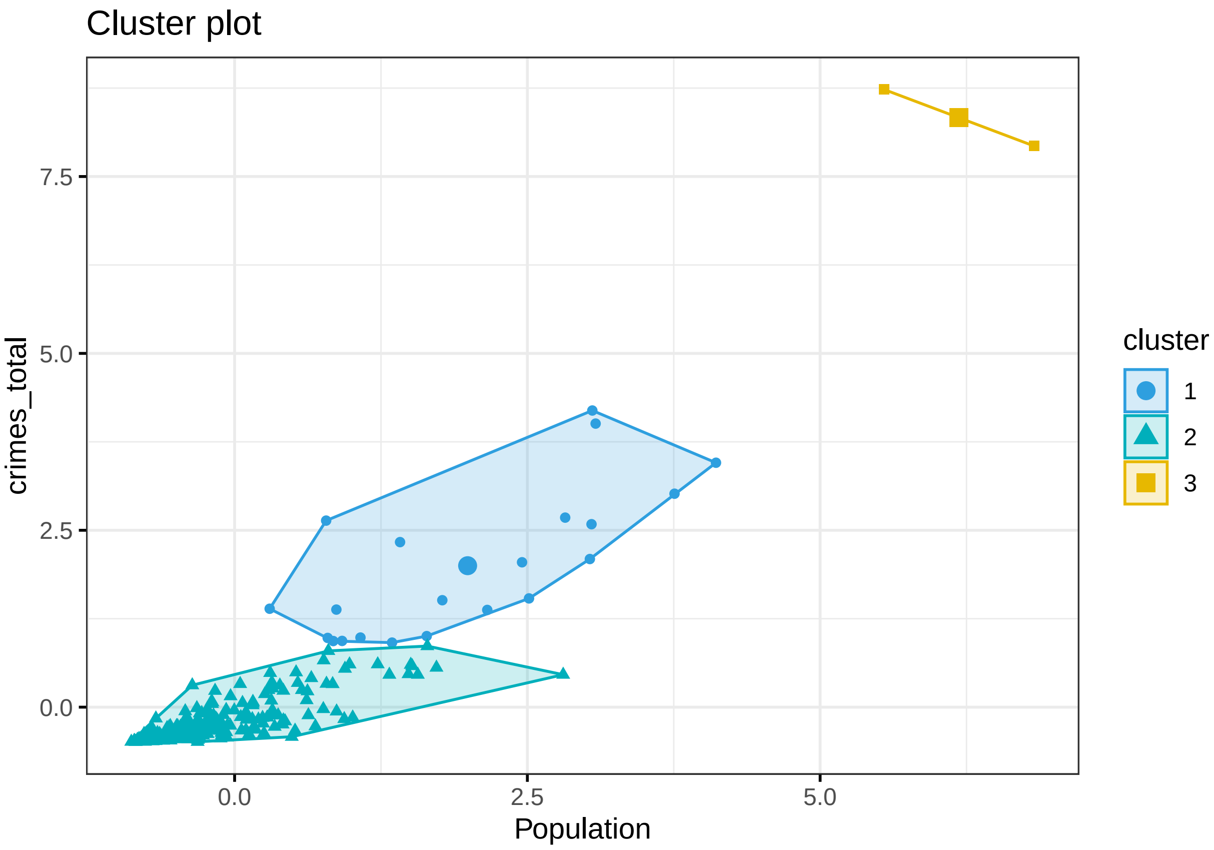
```
# select out only the populaiton and crimes_total
pop_crimes<-crimes2%>%select(Population, crimes_total)
```

```
res.km <- kmeans(scale(crimes2$crimes_total), 2, nstart = 25)
fviz_cluster(res.km, data = pop_crimes,
             palette = c("#2E9FDF", "#00AFBB", "#E7B800"),
             geom = "point",
             ellipse.type = "convex",
             ggtheme = theme_bw())
```



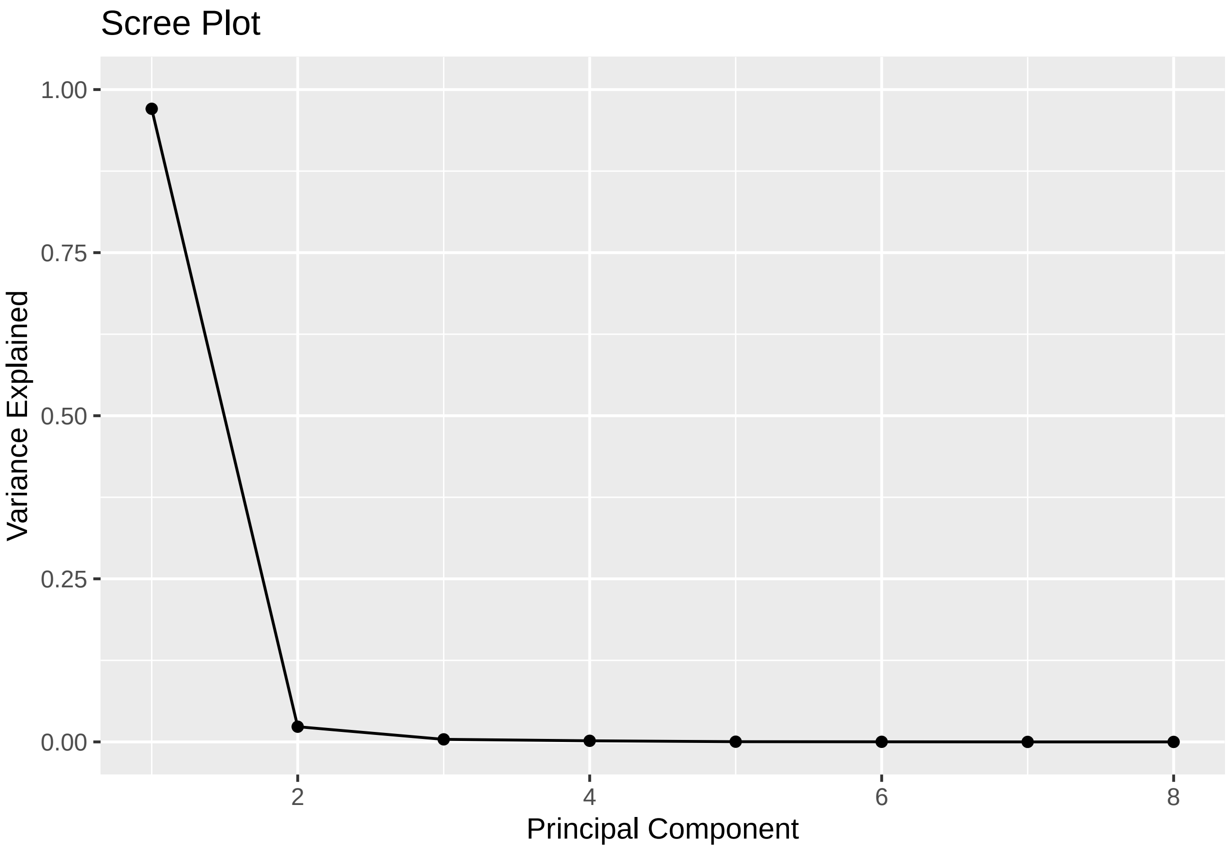
K = 2 doesn’t provide an idea result. Try k=3 as indicated in the cluster plot:

```
# as we find 3 clusters above so we use 3 here as well
# this time we use kmeans:
res.km <- kmeans(scale(crimes2$crimes_total), 3, nstart = 25)
fviz_cluster(res.km, data = pop_crimes,
             palette = c("#2E9FDF", "#00AFBB", "#E7B800"),
             geom = "point",
             ellipse.type = "convex",
             ggtheme = theme_bw()
 )
```



New cluster visualization result tell us we should have about three clusters for different crimes in general. And if we use k=2, one of the cluster will be able to include much bigger content of the entire population, and there is actually another small cluster exist in the bigger cluster if we look at the clustering dendrogram so we can even divide the entire population into 3 clusters. Since the crime cases and population in Springfiled and Worcester are too extreme and we can only produce a linear cluster (and this is the previous defined cluster1). The yellow area here has medium population and crimes(and this is the previous defined cluster3), the shallow blue area which is cluster with both low population and crime cases. What we find in the previous regession analysis has again been proved in the custer result that number of crime cases is associated with population. Now let's do PCA and focus on the frequency of different crimes:

```
pca1<-prcomp(crimes[,2:9])

# calculate total variance explained by each principal component
var_explained <- pca1$sdev^2 / sum(pca1$sdev^2)
qplot(c(1:8), var_explained) +
  geom_line() +
  xlab("Principal Component") +
  ylab("Variance Explained") +
  ggtitle("Scree Plot") +
  ylim(0, 1)
```



From what we found above we can tell that the first factor can explain about 90% of the total population and the elbow part occurs at the index 2-3 which are the part we mainly want to focus on, after that, all the rest population in a very smooth trend and this result can testify what we found in the previous cluster result that we have majority of data in the middle of the cluster tree, and the rest two cluster will only include very limited data. Now let's use the `prcomp` PCA method to check out the scoring of different crimes and see how is there frequency related to each other:

```
# Prcomp method using eigen of cov
pcaB <- prcomp(crimes[,2:9],scale. = T)

# first factor group:
pcaB1 <- pcaB$rotation[,1][order(pcaB$rotation[,1])]
pcaB1
```

```
## Motor_vehicle_theft  Aggravated.assault             Robbery            Burglary
##          -0.3698307          -0.3698208          -0.3696865          -0.3650220
##      Larceny..theft Murder_MANSLAUGHTER                Rape               Arson
##          -0.3541566          -0.3440245          -0.3317579          -0.3204772
```

```
# second factor group:
pcaB2 <- pcaB$rotation[,2][order(pcaB$rotation[,2])]
pcaB2
```

```
## Murder_MANSLAUGHTER             Robbery Motor_vehicle_theft            Burglary
##         -0.46747072         -0.18863844         -0.17779638         -0.14602096
##      Larceny..theft  Aggravated.assault               Arson                Rape
##         -0.11238332          0.02467729          0.57043034          0.59525057
```

This result shows that on the one side we have rare type of crimes which include murder_manslaughter, rape and arson, but on the other side we have more common crimes which includes Larceny..theft, Aggravated.assault. The general logic in this case are based on the frequency of different crimes. And the cases of the murder_manslaughter, rape and arson is much lower than the other crimes in the, we may call them uncommon crimes, for larceny theft and burglary we can call them common crimes due to high cases number.

10.

```
# create a new table called df10
df10<-df
```

```
# convert all data into X offenses by 100000 habitats
df10$Violent.crime<-df10$Violent.crime/df10$Population*100000
df10$Murder_MANSLAUGHTER<-df10$Murder_MANSLAUGHTER/df$Population*100000
df10$Rape<-df10$Rape/df$Population*100000
df10$Robbery<-df10$Robbery/df$Population*100000
df10$Aggravated.assault<-df10$Aggravated.assault/df10$Population*100000
df10$Property.crime<-df10$Property.crime/df10$Population*100000
df10$Burglary<-df10$Burglary/df10$Population*100000
df10$Larceny..theft<-df10$Larceny..theft/df$Population*100000
df10$Motor_vehicle_theft<-df10$Motor_vehicle_theft/df$Population*100000
df10$Arson<-df10$Arson/df10$Population*100000
```

Redo what we prtformed in Q8

```
# same regression as we did in q8
lm3.2<-lm(Population ~Property.crime,data=df10)
summary(lm3.2)
```

```
##
## Call:
## lm(formula = Population ~ Property.crime, data = df10)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -57278 -10680  -4609   6017 141699
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    7826.112   2219.943   3.525 0.000494 ***
## Property.crime   17.275      2.276   7.590 4.85e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21840 on 278 degrees of freedom
## Multiple R-squared:  0.1717, Adjusted R-squared:  0.1687
## F-statistic: 57.61 on 1 and 278 DF,  p-value: 4.85e-13
```

```
lm3.3<-lm(Population ~Larceny..theft+Property.crime,data=df10)
summary(lm3.3)
```

```
##
## Call:
## lm(formula = Population ~ Larceny..theft + Property.crime, data = df10)
##
## Residuals:
```

```
##    Min     1Q Median     3Q    Max
## -57251 -11351  -4278   7190 130163
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      6383.74    2192.54   2.912  0.00389 **
## Larceny..theft    -56.60      14.18  -3.993 8.36e-05 ***
## Property.crime     62.97      11.66   5.402 1.42e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21270 on 277 degrees of freedom
## Multiple R-squared:  0.2167, Adjusted R-squared:  0.2111
## F-statistic: 38.33 on 2 and 277 DF,  p-value: 2.02e-15
```

and if we go back to see what we have in Q9:

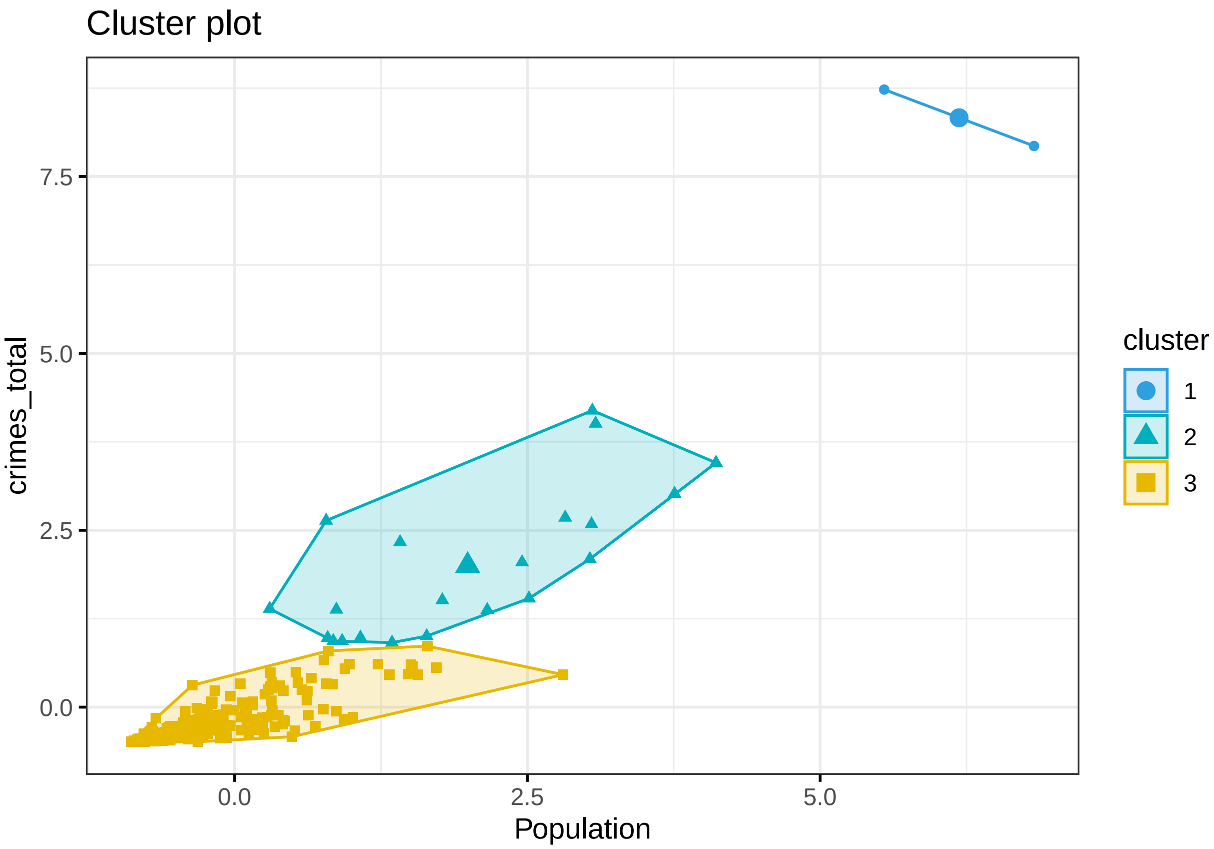
```
# first factor group:
pcaB1
```

```
## Motor_vehicle_theft  Aggravated.assault             Robbery            Burglary
##          -0.3698307          -0.3698208          -0.3696865          -0.3650220
##      Larceny..theft Murder_MANSLAUGHTER                Rape               Arson
##          -0.3541566          -0.3440245          -0.3317579          -0.3204772
```

```
# second factor group:
pcaB2
```

```
## Murder_MANSLAUGHTER             Robbery Motor_vehicle_theft            Burglary
##         -0.46747072         -0.18863844         -0.17779638         -0.14602096
##      Larceny..theft  Aggravated.assault               Arson                Rape
##         -0.11238332          0.02467729          0.57043034          0.59525057
```

```
res.km <- kmeans(scale(crimes2$crimes_total), 3, nstart = 25)
fviz_cluster(res.km, data = pop_crimes,
              palette = c("#2E9FDF", "#00AFBB", "#E7B800"),
              geom = "point",
              ellipse.type = "convex",
              ggtheme = theme_bw()
 )
```



Well, this time we can have regression result that very different from previously question 8 output because both of the larceny theft and the property crime are still having strong influence toward the population at this time. But if we compare with question 9 result we can say this result is also reasonable because that the larceny theft (or the property crime in general) as one of the commonest crimes on the opposite side of the rare (uncommon) crimes and it will certainly has statistically significant impact toward the population as we transfer them into more precise data. Though, we know there might be some casual pathways issue in this regression result. Meanwhile, I would say if we combine it with the cluster/PCA result we got in the previous questions, as these two items in the `lm3.3` will took majority of the `cimres_total` that we used to plot the against `populaiotn` for the cluster plot above. Even though we haven't transform the the data into X offenses by 100000 habitants at that time, but property crime as a majority part of the `crimes_total` certainly still has strong impact to the population as we can see that both of the`hclust` and kmeans algorithm can help us to divide the dataset into 3 clusters perfectly. Thus, we may say this result also makes sense if we ignore the fact there are potential casual pathways issue within.