hw03-02

Weijun Zhu

November 12, 2019

Contents


```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts ------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(cluster)
library(fpc)
library(rgl)
```

```
str(USArrests)
```

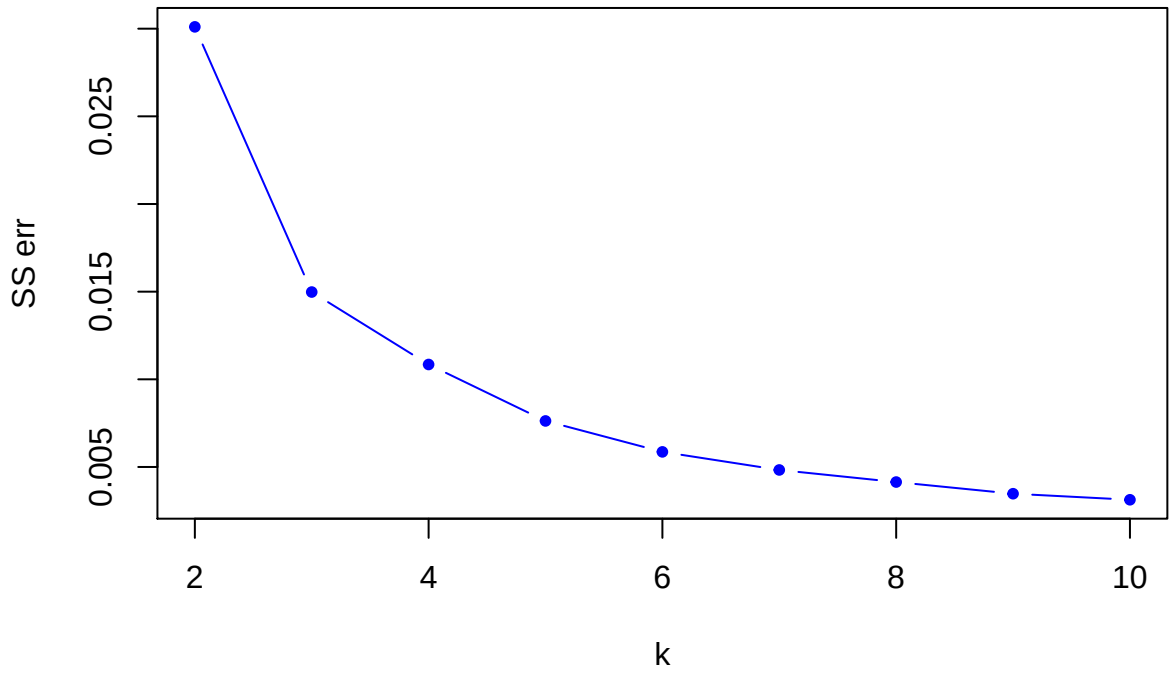
```
## 'data.frame':    50 obs. of  4 variables:
##  $ Murder  : num  13.2 10 8.1 8.8 9 7.9 3.3 5.9 15.4 17.4 ...
##  $ Assault : int  236 263 294 190 276 204 110 238 335 211 ...
##  $ UrbanPop: int  58 48 80 50 91 78 77 72 80 60 ...
##  $ Rape    : num  21.2 44.5 31 19.5 40.6 38.7 11.1 15.8 31.9 25.8 ...
```

2a).

For each value of k from 2 to 10, run the k-means algorithm. For each run set the nstart parameter to 20 so that the algorithm is run with 20 different random starting points. To make sure all results are reproducible before starting run the command set.seed(3). This will make all random numbers generated the same (this

makes it easier to grade). Create an empty vector ssw. For each k from 2 to 10, compute ssw=(total within ss)/(total ss). This is the ratio of within-cluster sum-of-square distances to the total sum-of-square of all distances. By looking at the plot decide what would be a good choice of k.

```
n1 = 2
n = 10
km=kmTotss=kmBetweenss=kmWithinss=c()
for (i in n1:n){
  set.seed(3)
  km1<-kmeans(USArrests,i,iter.max=1000, nstart=20)
  kmTotss=c(kmTotss,km1$totss)
  kmWithinss=c(kmWithinss,sum(km1$withinss))
  kmBetweenss=c(kmBetweenss,km1$betweenss)
  km<-c(km,km1)
}
# plot(n1:n, kmBetweenss/sum(kmBetweenss), "b", col="blue",pch=20,xlab="k", ylab="SS err")
# lines(n1:n, kmWithinss/sum(kmWithinss), "b", col="orange",pch=20)
# lines(n1:n, kmTotss/sum(kmTotss), "b", col="red",pch=20)
plot(n1:n, kmWithinss/sum(kmTotss), "b", col="blue",pch=20,xlab="k", ylab="SS err")
```

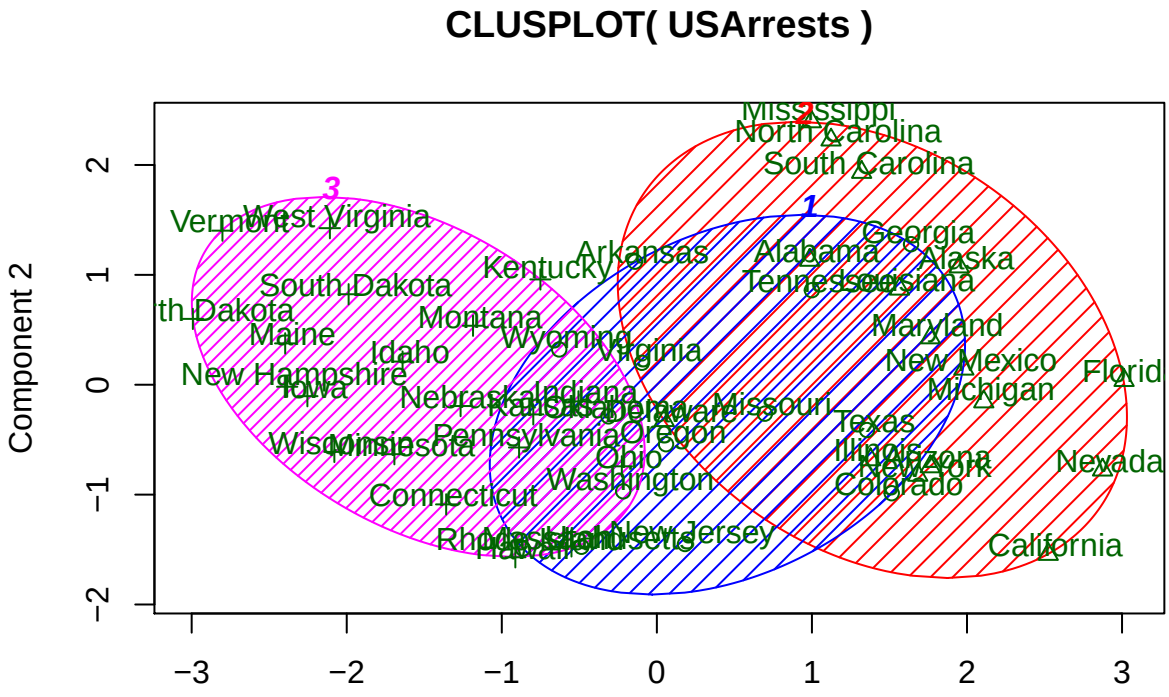


Conclusion: 5 or 6 are good chance to be the number of clusters.

2b).

Set k, the number of clusters to 3. Run the k-means algorithm for k = 3 and with nstart=20. Print the cluster of each state based on the result.

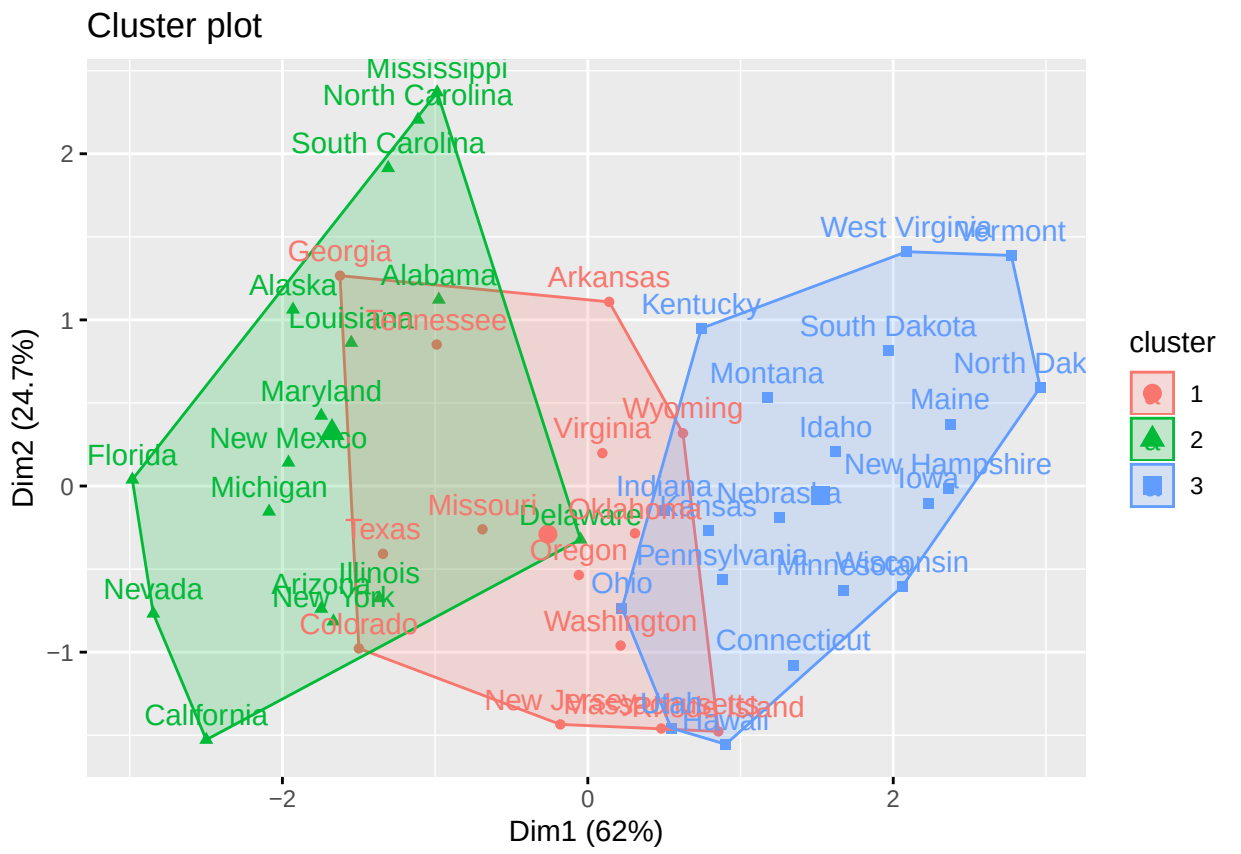
```
model <- kmeans(USArrests,3,iter.max=1000, nstart=20)
clusplot(USArrests, model$cluster, color=TRUE, shade=TRUE,
         labels=2, lines=0)
```




```
library(factoextra)
```

```
## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ
```

```
fviz_cluster(model, data = USArrests)
```

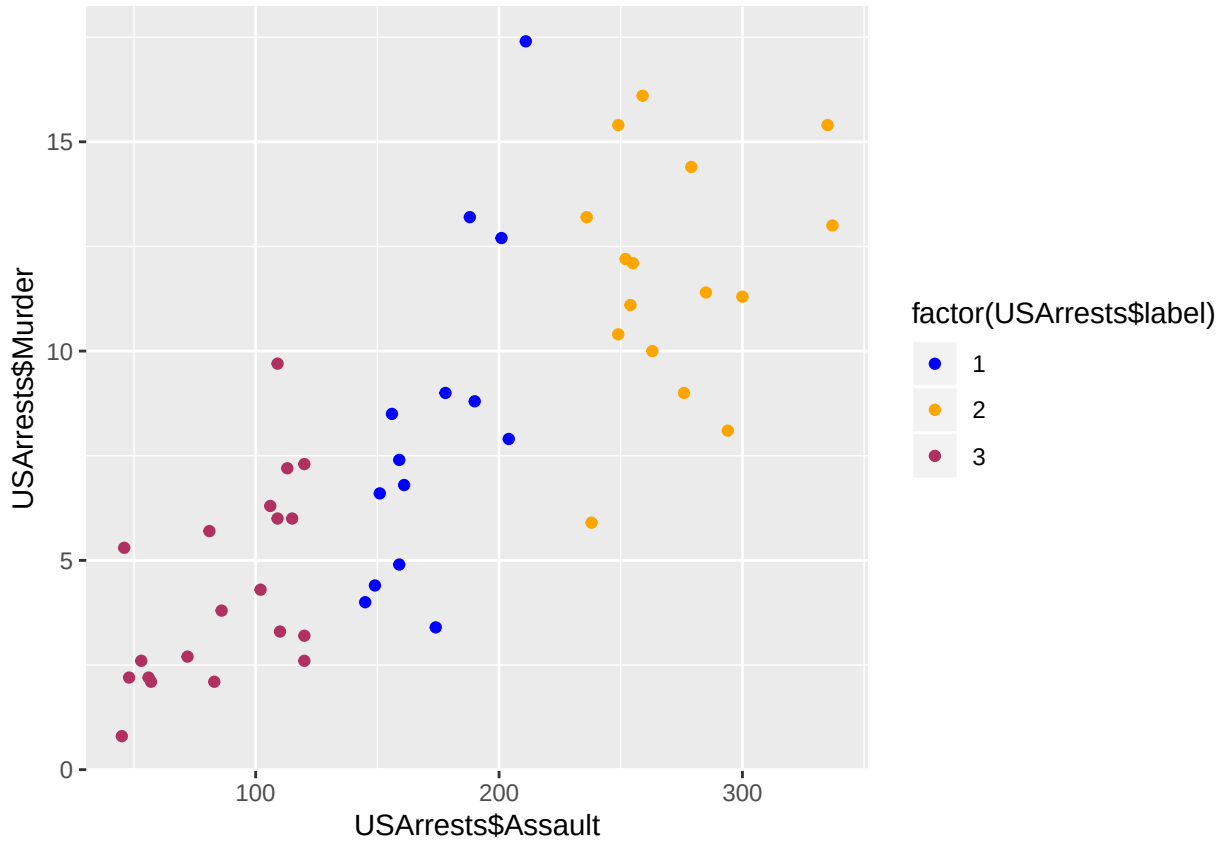


2c).

Create a scatter plot with the X axis assault rate, and the Y axis as the murder rate. Color those states in cluster 1 as blue, those in cluster 2 as orange, and those in cluster 3 as maroon. Also, in addition to "dots" on the scatter plot, print the name of the state as well. For this check documentation for the text function of R.

```
USArrests['label'] <- model$cluster
```

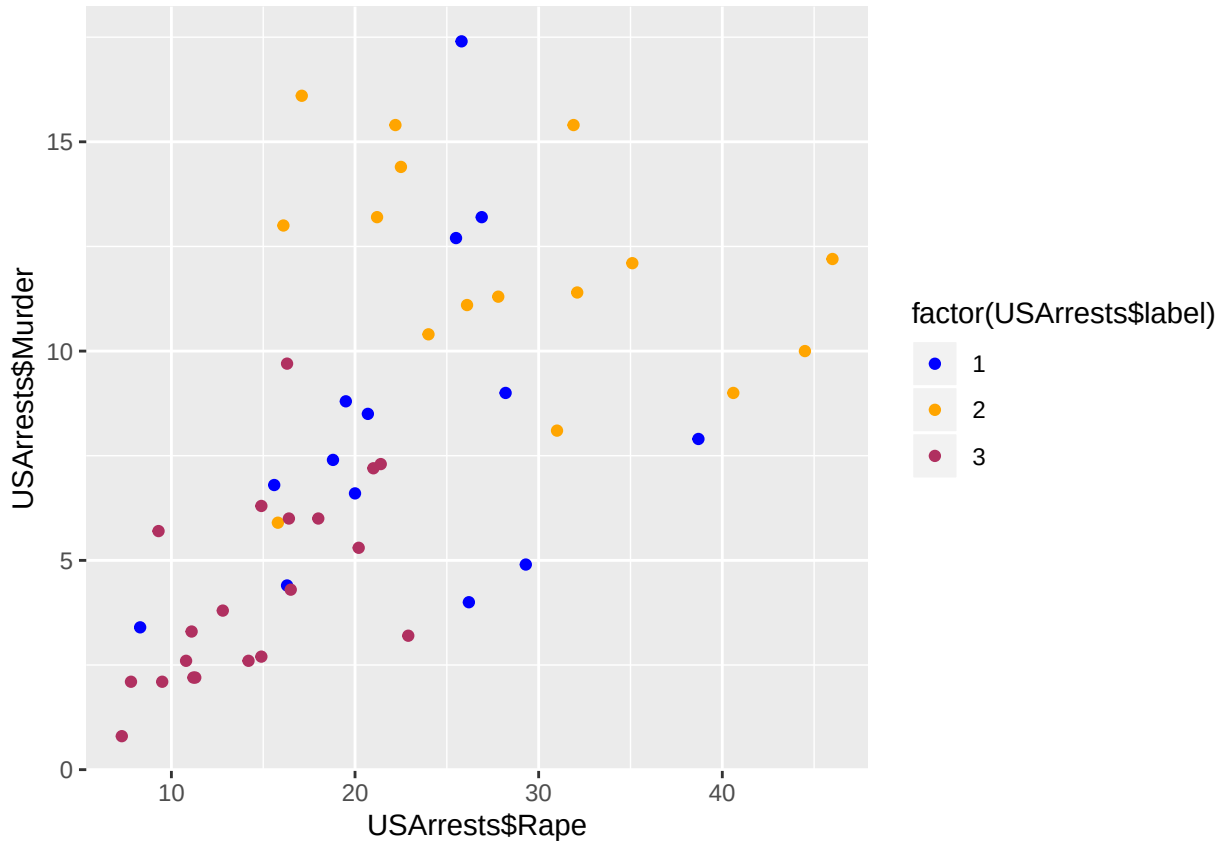
```
ggplot(USArrests, aes(x = USArrests$Assault, y = USArrests$Murder)) +
  geom_point(aes(color = factor(USArrests$label))) +
  scale_color_manual(breaks = c('1','2','3'), values = c('blue','orange','maroon'))
```



2d).

Repeat 2c) but using Rape on the X axis and Murder on the Y axis.

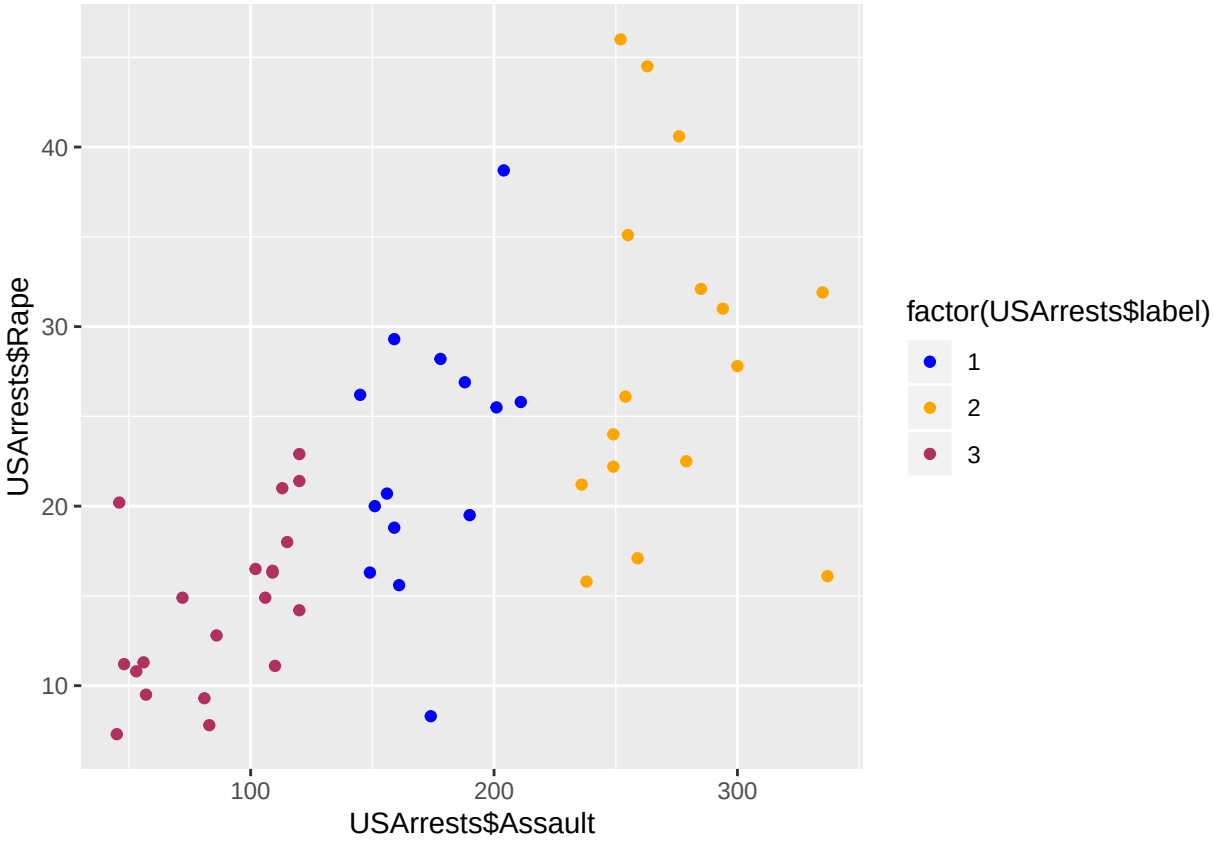
```
ggplot(USArrests, aes(x = USArrests$Rape, y = USArrests$Murder)) +
  geom_point(aes(color = factor(USArrests$label))) +
  scale_color_manual(breaks = c('1','2','3'), values = c('blue','orange','maroon'))
```



2e).

Repeat 2c) but using Assault on the X axis and Rape on the Y axis.

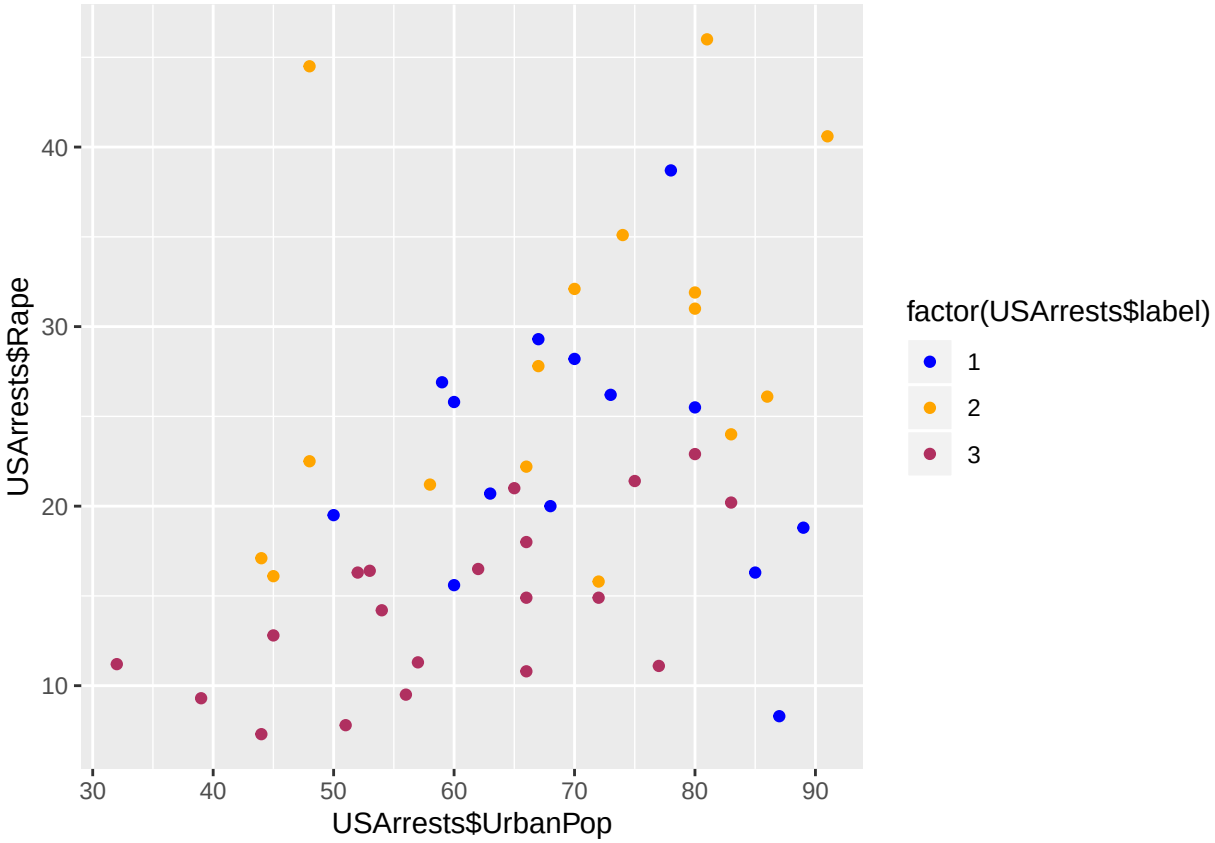
```
ggplot(USArrests, aes(x = USArrests$Assault, y = USArrests$Rape)) +
  geom_point(aes(color = factor(USArrests$label))) +
  scale_color_manual(breaks = c('1','2','3'), values = c('blue','orange','maroon'))
```



2f).

Repeat 2c) but using UrbanPop on the X axis and Murder on the Y axis.

```
ggplot(USArrests, aes(x = USArrests$UrbanPop, y = USArrests$Rape)) +
  geom_point(aes(color = factor(USArrests$label))) +
  scale_color_manual(breaks = c('1','2','3'), values = c('blue','orange','maroon'))
```



2g).

Now load the rgl library. Plot a 3D graph with Murder on the X axis, Rape on the Y axis and Assault on the Z axis. Use the same color code as the 2D plots for each cluster. Also add the name of each state on each data point using text3d function (check the documentation).

```
# library(rgl)
attach(USArrests)
plot3d(Murder,Rape,Assault,size=9,col=c('blue','orange','maroon')[USArrests$label])
text3d(USArrests$Murder,USArrests$Rape,USArrests$Assault,texts=row.names(USArrests))
detach(USArrests)
```