Homework 6

Hwasoo Shin

2019 10 5

Problem 3

```
parta<-function(x){apply(x,2,mean)}
#Check the ratio of success

set.seed(12345)
P4<-matrix(rbinom(10,1,prob=(30:40)/100),nrow=10,ncol=10,byrow=FALSE) #Make a 10x10 matrix
apply(P4,1,mean) #Check the mean of each row
```

```
##  [1] 1 1 1 1 0 0 0 0 1 1
```

```
apply(P4,2,mean) #Check the mean of each column
```

```
##  [1] 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6
```

We can see that every row and column are the same. It looks like we are not getting the matrix that we wanted

```
problem3<-function(p){rbinom(10,1,p)} #Function for bernoulli generator
sapply((30:40)/100,problem3) #Use the function for probability from0.3 to 0.4.
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
##  [1,]    0    0    1    1    1    1    1    1    1     0     1
##  [2,]    0    0    0    0    1    0    0    0    1     1     0
##  [3,]    1    1    0    1    0    1    0    0    0     1     1
##  [4,]    0    1    1    1    0    1    0    0    0     1     0
##  [5,]    0    0    0    0    1    1    0    0    1     0     1
##  [6,]    0    0    0    0    0    0    0    0    0     1     1
##  [7,]    0    1    1    0    1    1    1    1    1     1     1
##  [8,]    0    0    1    0    0    0    1    0    1     1     0
##  [9,]    0    0    0    0    0    0    0    1    0     0     0
## [10,]    1    0    0    0    0    1    0    0    0     0     0
```

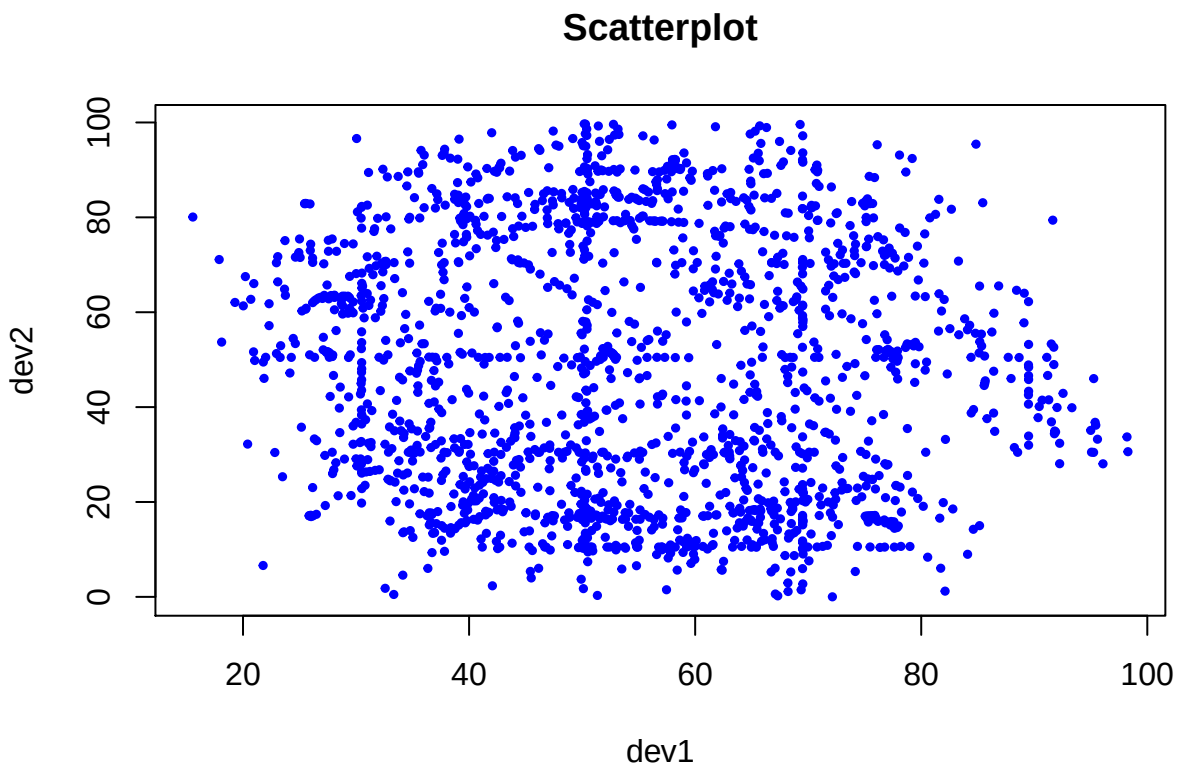
```
#Each column reperesents values on specific probability.
P<-sapply((30:40)/100,problem3)
parta(P) #Check the ratio of success by columns.
```

```
##  [1] 0.4 0.5 0.4 0.5 0.5 0.7 0.3 0.4 0.4 0.3 0.3
```

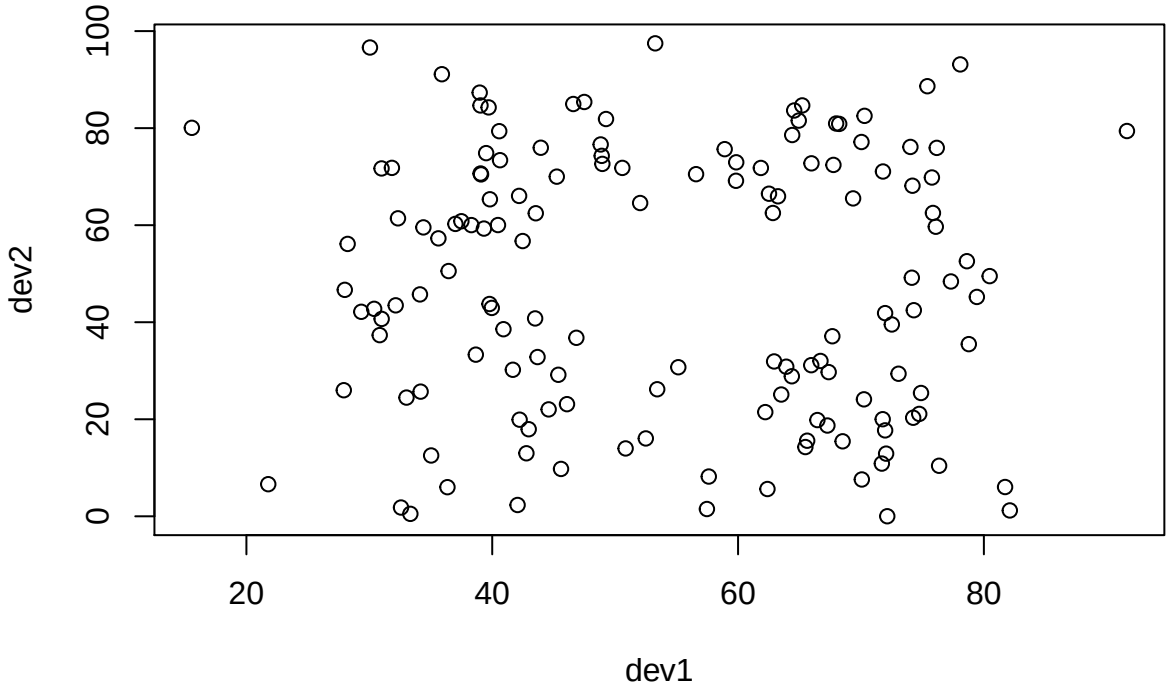
We can now see that the probabilites of each column has different value. This is the function and matrices that we were looking for.

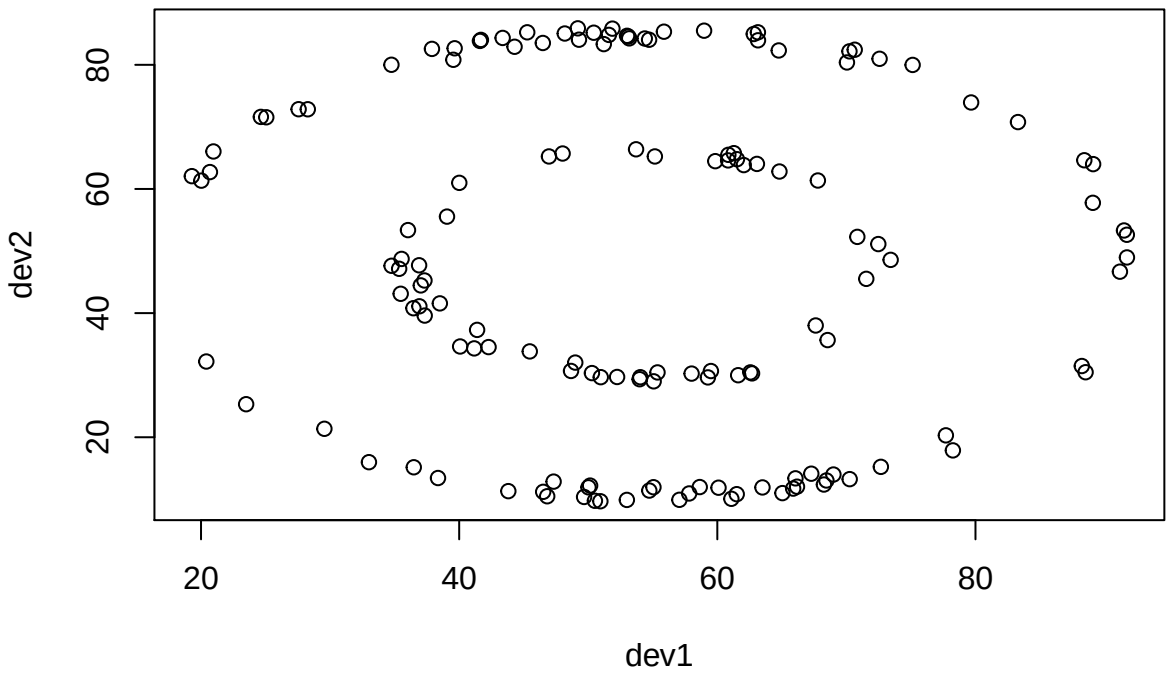
Problem 4

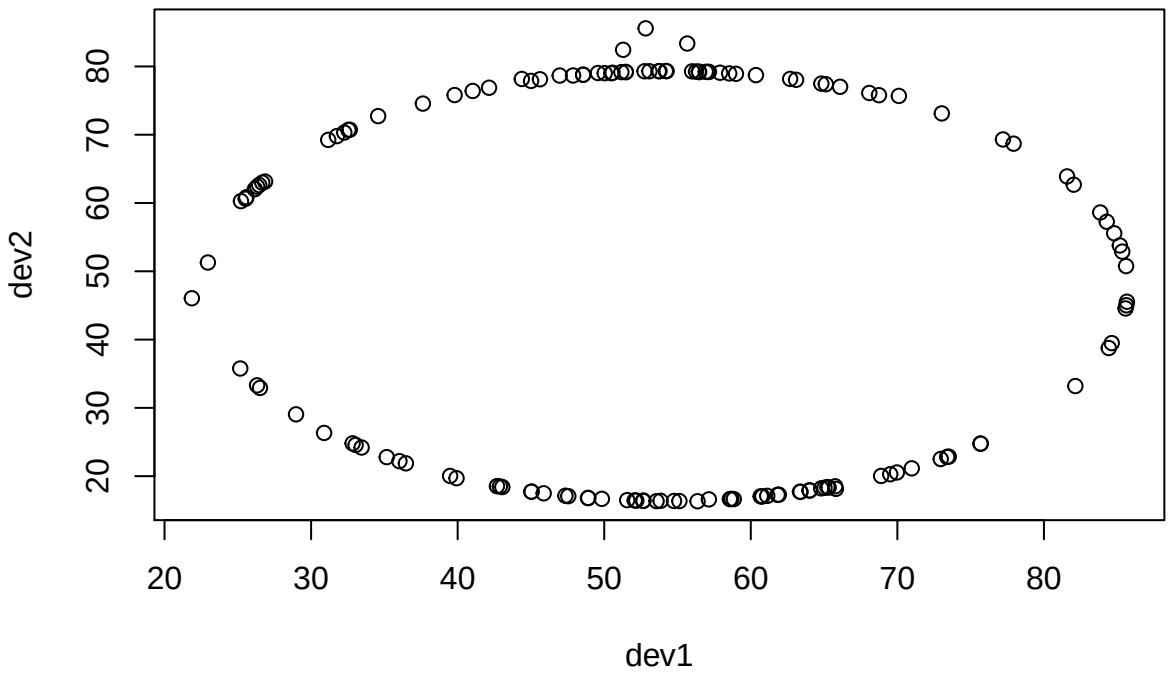
```
problem4<-function(x=NULL){
  RDS<-readRDS("C:/Users/pc/Desktop/HWASOO/STUDY/StatPackage/Homework6/HW6_data.rds")
  #Read the data file
    if(is.null(x)){ #if argument not given, plot all the data in the same graph
  plot(RDS$dev1,RDS$dev2,pch=19,col='blue',main='Scatterplot',xlab='dev1',ylab="dev2",cex=0.5)}

  else{ #if argument given, plot data in each graph
    RDSsplit<-split(RDS[-1],RDS$Observer)
    lapply(RDSsplit,plot) #Apply by elements in a list
  }
}

problem4() #Make the general plot
```

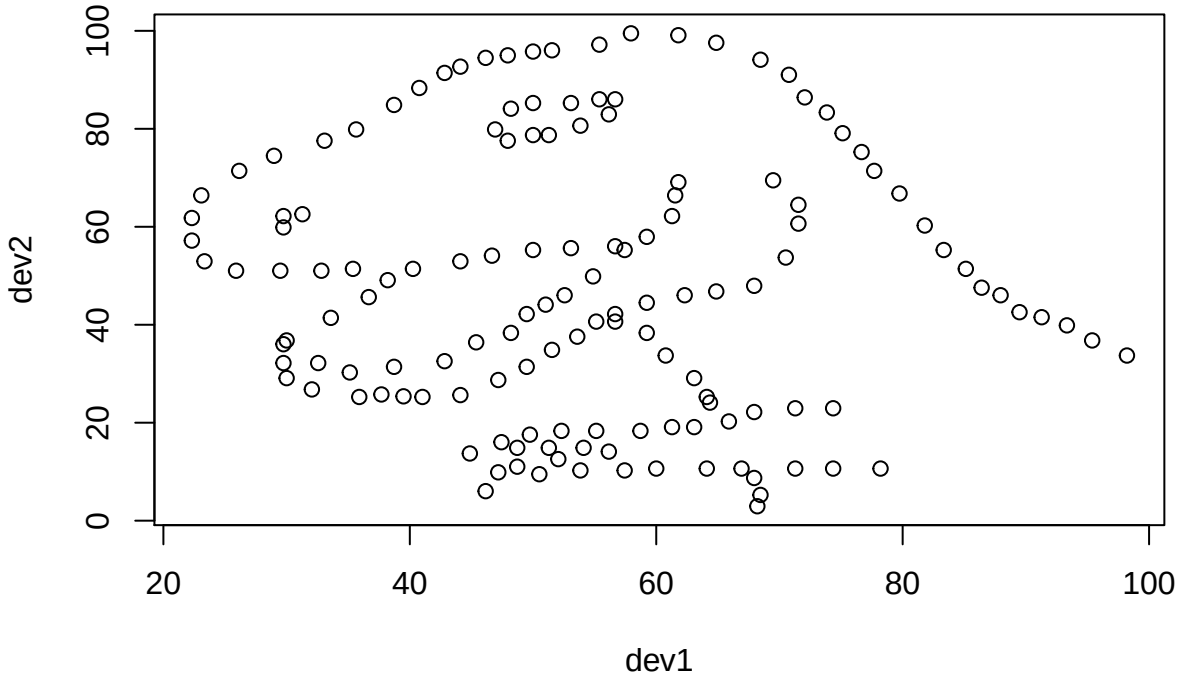


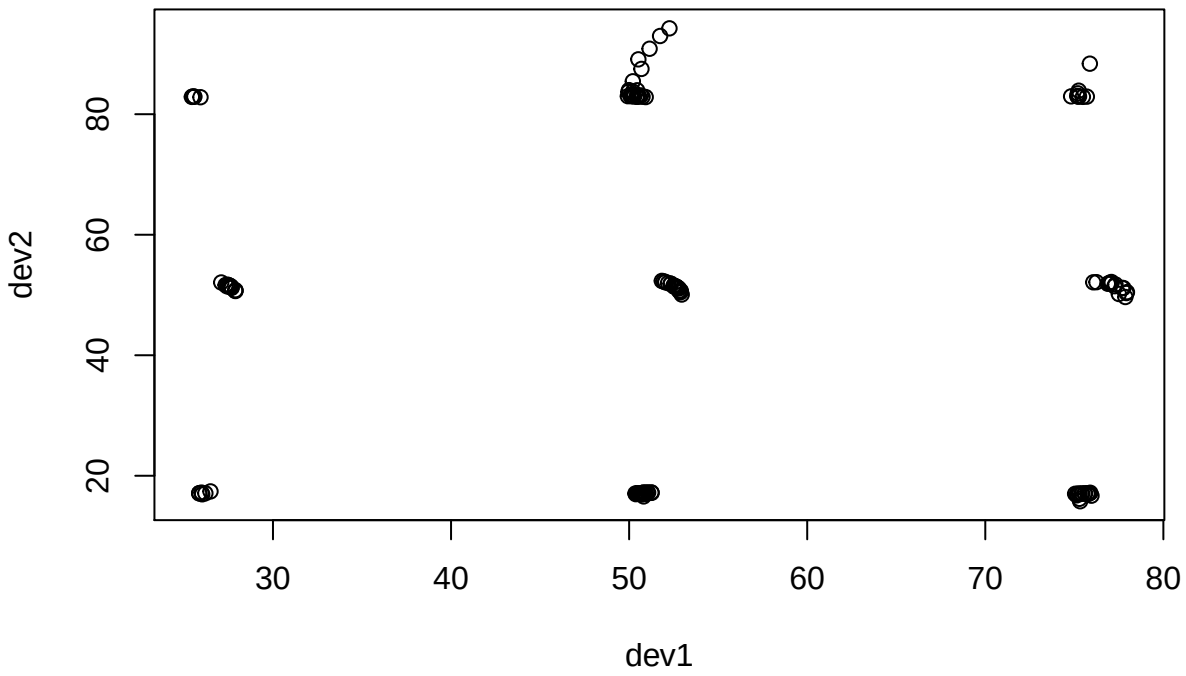

```
problem4(1) #Make plots by observations
```

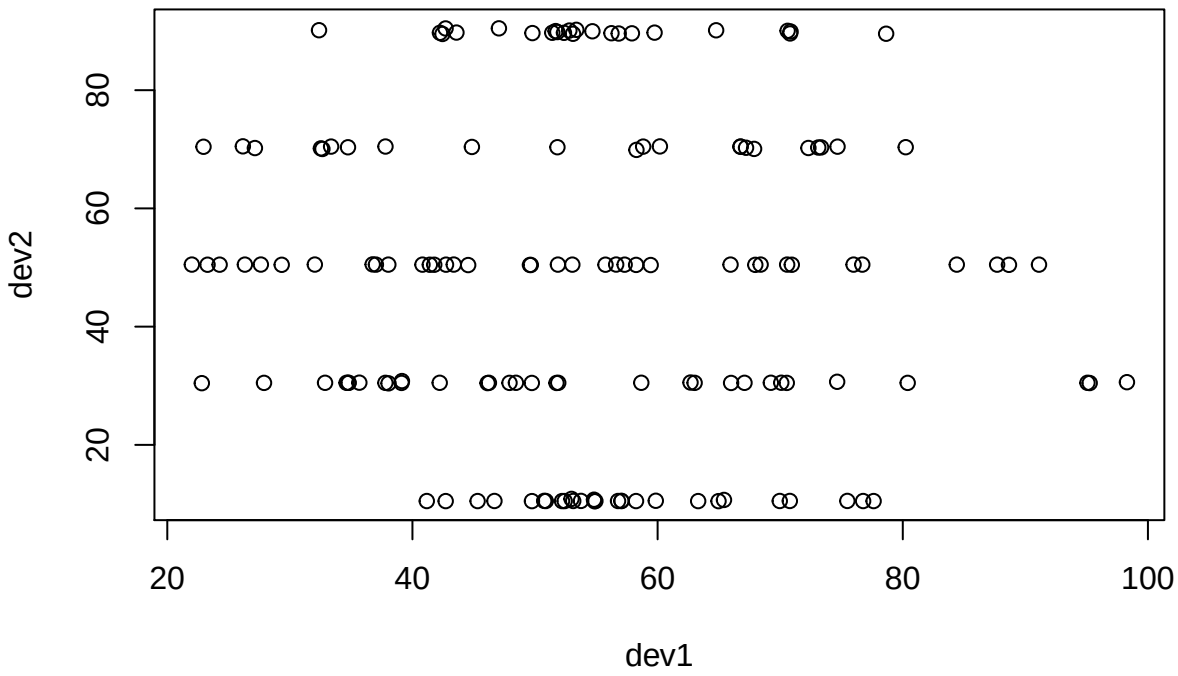

dev2
0
20
40
60
80
100
20
40
60
80
dev1

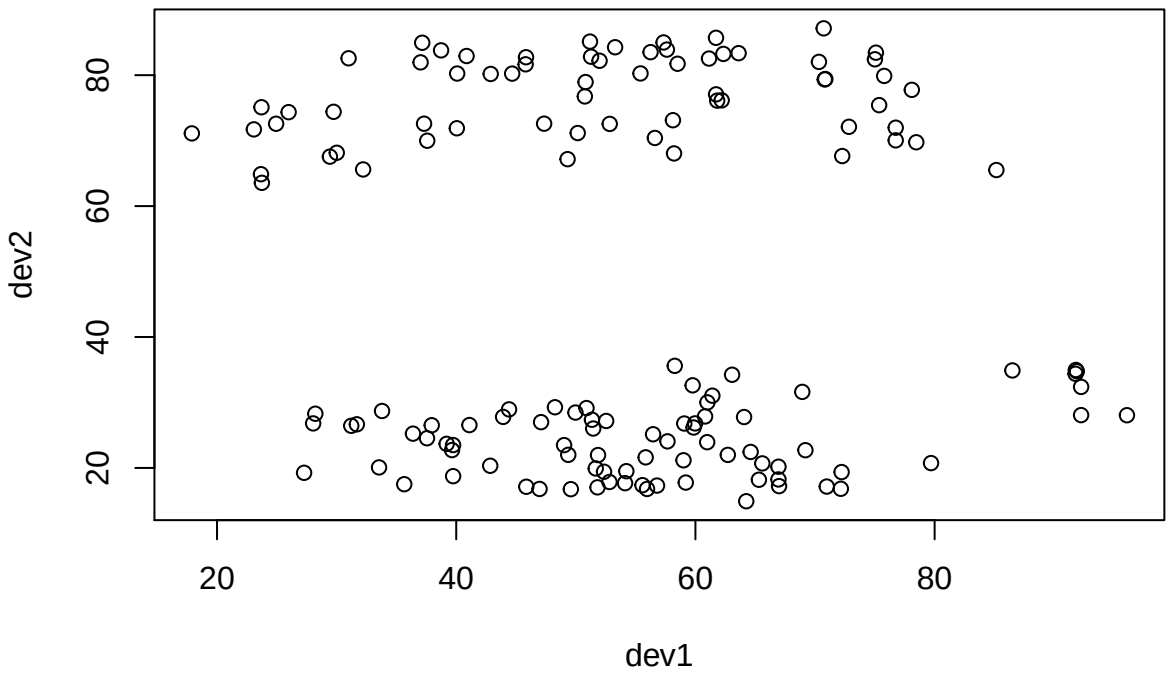

dev2
20
40
60
80
dev1
20
40
60
80

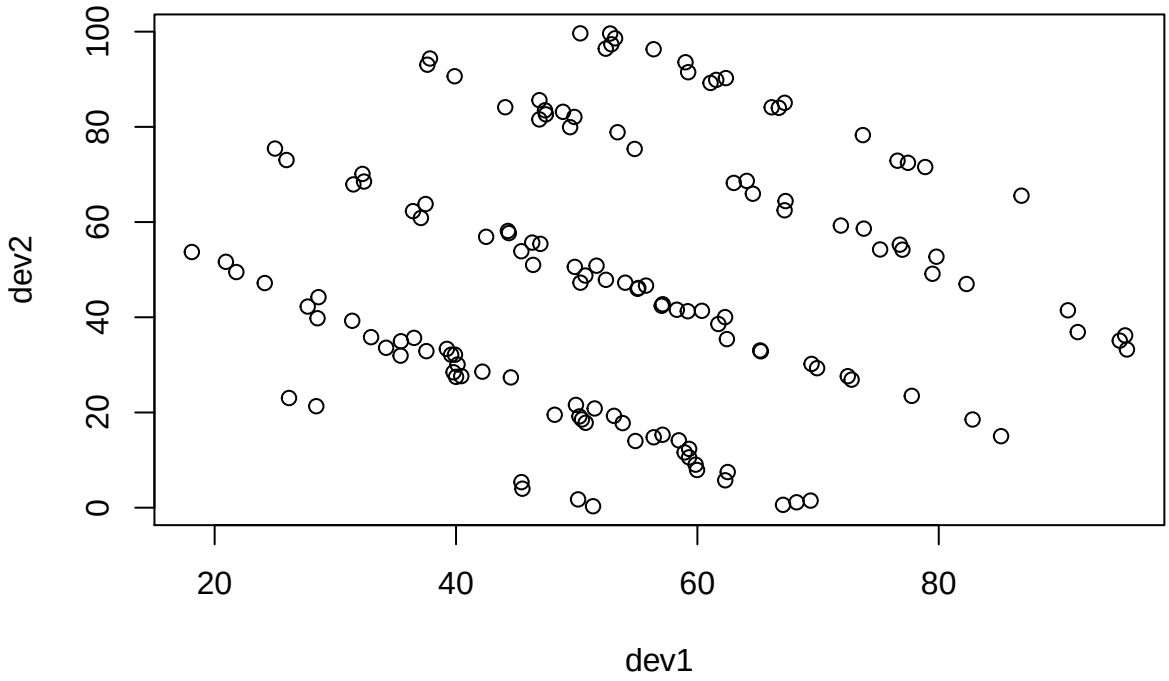

dev2
20
30
40
50
60
70
80
dev1
20
30
40
50
60
70
80

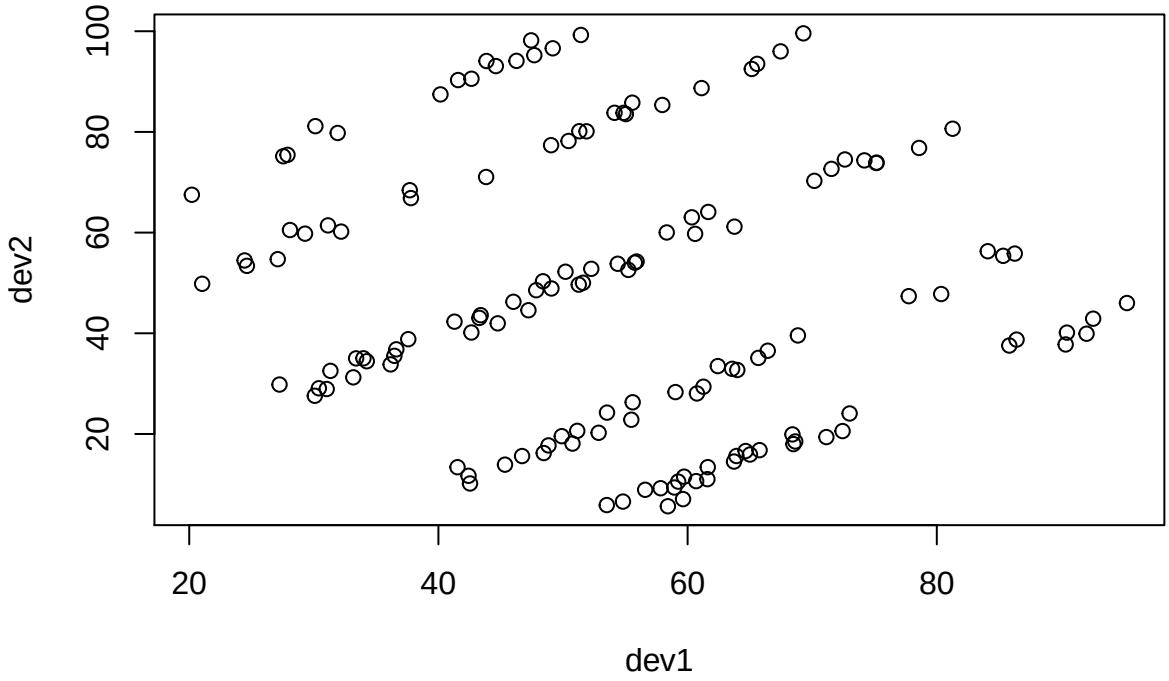

dev2
0
20
40
60
80
100
20
40
60
80
100
dev1

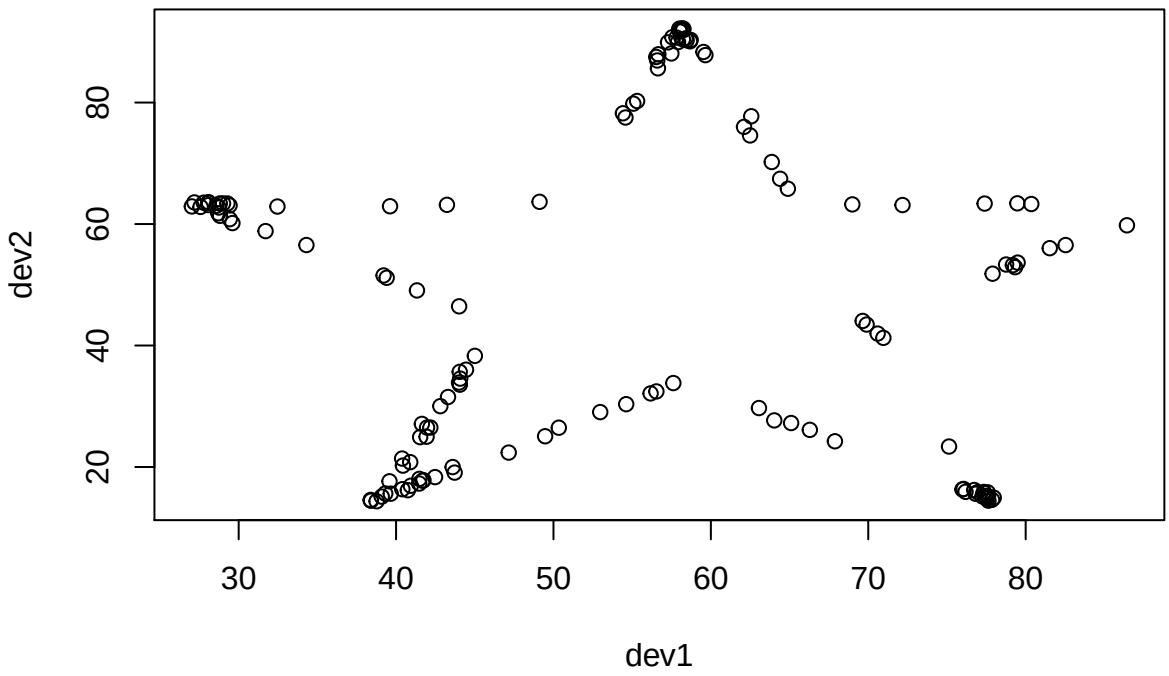

dev2
20
40
60
80
30
40
50
60
70
80
dev1

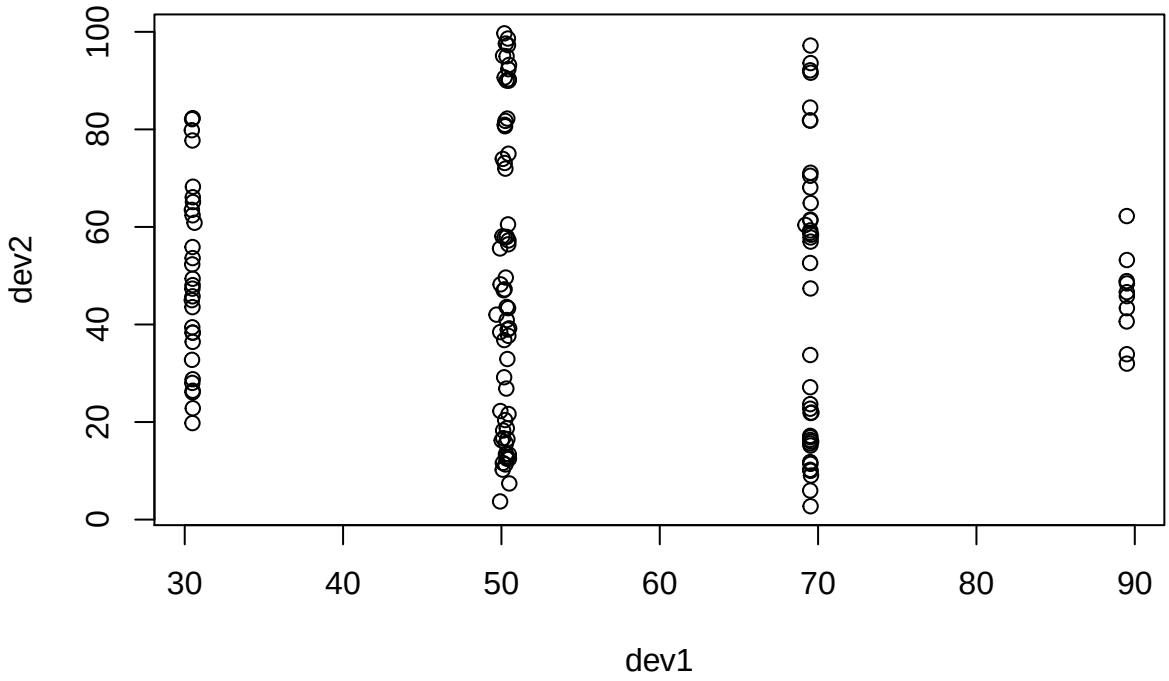

dev2
20
40
60
80
dev1
20
40
60
80
100

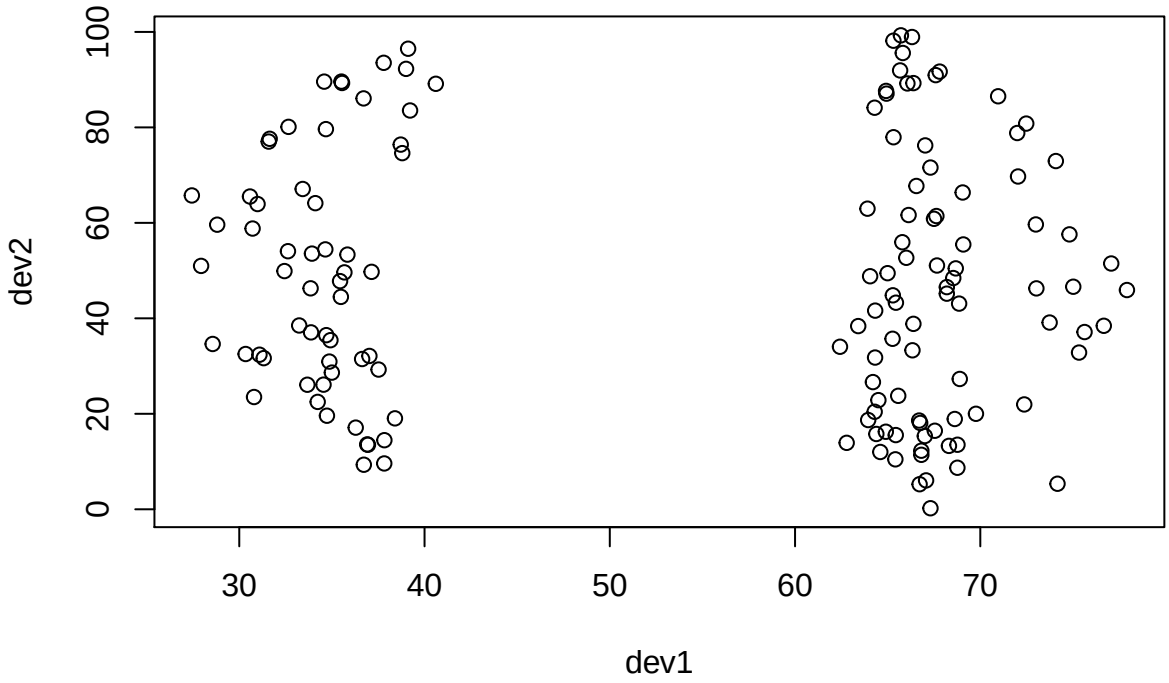

dev2
20
40
60
80
dev1
20
40
60
80

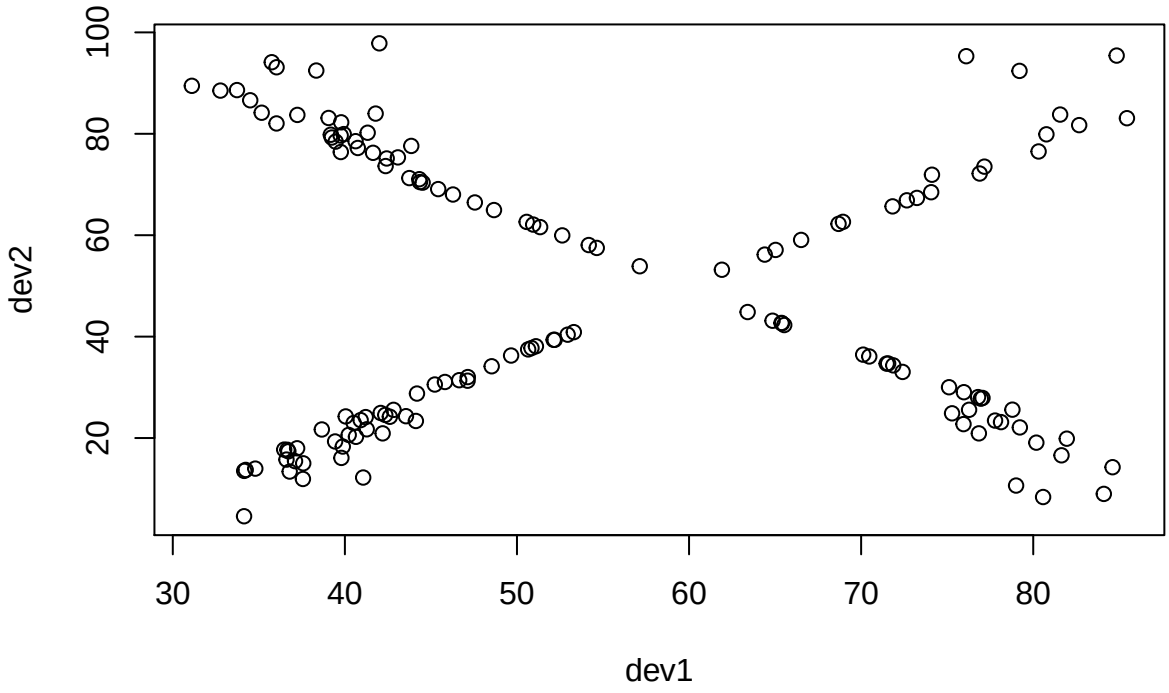

dev2
0
20
40
60
80
100
20
40
60
80
dev1


dev2
100
80
60
40
20
20
40
60
80
dev1


dev2
20
40
60
80
30
40
50
60
70
80
dev1


dev2
0
20
40
60
80
100
30
40
50
60
70
80
90
dev1


dev2
0
20
40
60
80
100
30
40
50
60
70
dev1




```
## $`1`
## NULL
##
## $`2`
## NULL
##
## $`3`
## NULL
##
## $`4`
## NULL
##
## $`5`
## NULL
##
## $`6`
## NULL
##
## $`7`
## NULL
##
## $`8`
## NULL
##
## $`9`
## NULL
```

```
##
## $`10`
## NULL
##
## $`11`
## NULL
##
## $`12`
## NULL
##
## $`13`
## NULL
```

Problem 5

```
setwd("C:/Users/pc/Desktop/HWASOO/STUDY/StatPackage/Homework6")
library(downloader)
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

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

```
library(mapproj)
```

```
## Loading required package: maps
```

```
##
## Attaching package: 'maps'
```

```
## The following object is masked from 'package:purrr':
##
##     map
```

```
download("http://www.farinspace.com/wp-content/uploads/us_cities_and_states.zip",dest="us_cities_states
#download the data
unzip("us_cities_states.zip")
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

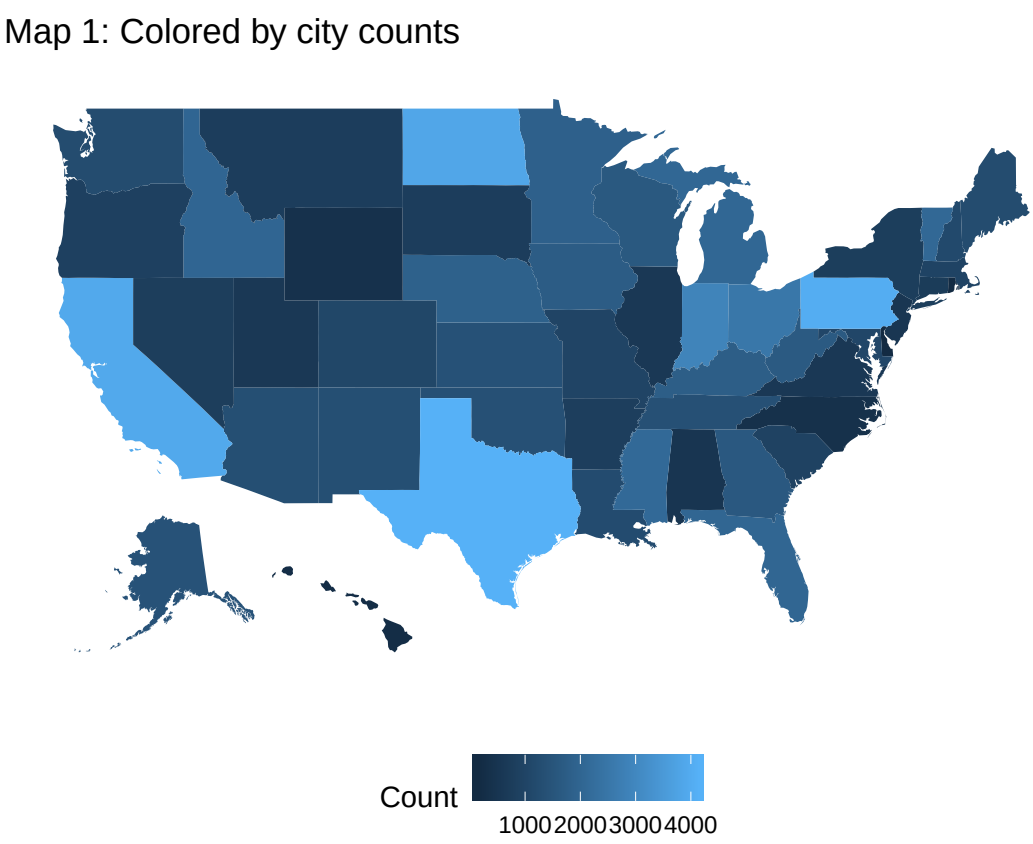
```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last

## The following object is masked from 'package:purrr':
##
##     transpose
```

```
states <- fread(input = "./us_cities_and_states/states.sql",skip = 23,sep = "'", sep2 = ",", header = F
cities <- fread(input = "./us_cities_and_states/cities.sql",skip = 23,sep = "'", sep2 = ",", header = F
citiesExtend <- fread(input = "./us_cities_and_states/cities_extended.sql",skip = 23,sep = "'", sep2 =
#Read all the data needed
city<-rbind(cities,citiesExtend) #Merge cities and citiesExtend datasets
table(city[,2]) #Table of number of cities
```

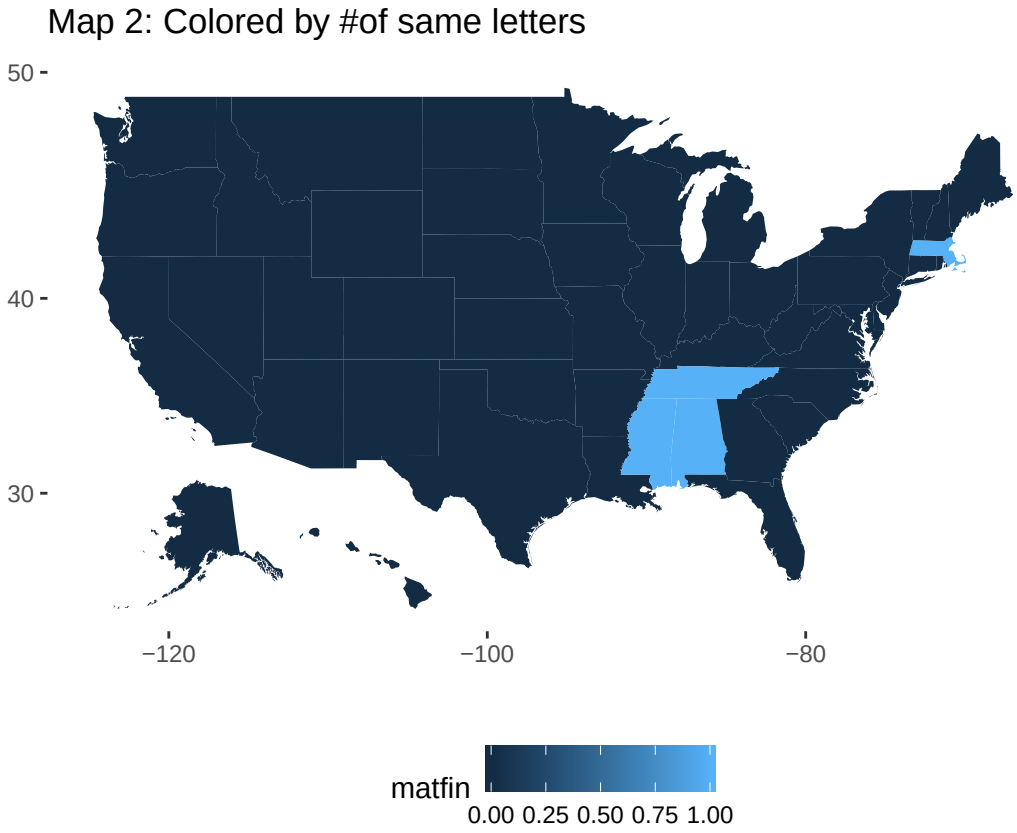
```
##
##   AK   AL   AR   AZ   CA   CO   CT   DC   DE   FL   GA   HI   IA   ID   IL
##  502 1417 1314  796 3890 1059  707  287  155 2011 1601  231 1997  591 2874
##   IN   KS   KY   LA   MA   MD   ME   MI   MN   MO   MS   MT   NC   ND   NE
## 1727 1390 1764 1204 1214 1049  950 2055 1841 2112  973  765 1852  780 1148
##   NH   NJ   NM   NV   NY   OH   OK   OR   PA   PR   RI   SC   SD   TN   TX
##  539 1312  772  352 3819 2515 1359  863 4010  275  161  916  758 1343 4116
##   UT   VA   VT   WA   WI   WV   WY
##  594 2077  597 1225 1651 1612  371
```

```
problem5<-function(x){ #Function that will have inputs as "states"
  one<-str_split(x,"") #Split the word by every letter
  two<-unlist(one) #Unlist the data
  three<-tolower(two) #Make all letters in lower case
  return(three) #Return each letter
}
lett<-function(x){ #Function that will have inputs as "letters"
  x==letters #Check if the letter is same as any alphabet
}
wrapit<-function(x){ #Function that will count how many letters are in the word
  apply(sapply(problem5(x),lett),1,sum) #Use apply function to apply to every alphabet letter
}

states2<-states[order(states[,1]),] #Order states in alphabetical order
mat<-matrix(0,51,26) #Make a matrix for every state name
for(i in 1:51){
  mat[i,]<-wrapit(states2[i,1])
}
mat<-mat[-9,] #Remove "DC" (Considered not as a state)

matfin<-apply(mat,1,max) #Get the maximum of each row
matfin<-matfin>3 #Get the bool type for variable
matfin<-as.numeric(matfin) #Get numeric values for it
matfin<-data.frame(state=tolower(rownames(USArrests)),matfin)

tabcity<-table(city[,2])[-c(8,40)] #Remove "DC" and "PR"data
tabcity<-data.frame(tabcity) #Make into a data frame
colnames(tabcity)<-c("StateSt","Count") #Change the variable names
```

```
tabcity<-data.frame(state=tolower(rownames(USArrests)),tabcity)


load("C:/Users/pc/Desktop/HWASOO/STUDY/StatPackage/Homework6/fifty_states.rda")
crimes<-data.frame(state=tolower(rownames(USArrests)),USArrests)
p1<-ggplot(tabcity,aes(map_id=state))+
geom_map(aes(fill=Count),map=fifty_states) +
  expand_limits(x=fifty_states$long,y=fifty_states$lat) +
  coord_map() +
  scale_x_continuous(breaks=NULL) +
  scale_y_continuous(breaks=NULL) +
  labs(x ="", y = "",title="Map 1: Colored by city counts") +
  theme(legend.position="bottom",panel.background=element_blank())
p1 #Plot for the first map; color it by number of cities
```




```
p2<-ggplot(matfin,aes(map_id=state))+
geom_map(aes(fill=matfin),map=fifty_states) +
  expand_limits(x=fifty_states$long,y=fifty_states$lat) +
  coord_map() +
  labs(x ="", y = "",title="Map 2: Colored by #of same letters") +
  theme(legend.position="bottom",panel.background=element_blank())
p2 #Plot for the second map; color it only when more than same three letters appeared
```


Map 2: Colored by #of same letters
50
40
30
−120
−100
−80
matfin
0.00 0.25 0.50 0.75 1.00