Lab4

Priya(priku577) and Andreas(andch552)

October 3, 2018

Assignment1

1.Importing the data and filtering it

```
library(tidyverse)
```

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

```
## <U+221A> ggplot2 3.0.0     <U+221A> purrr   0.2.5
## <U+221A> tibble  1.4.2     <U+221A> dplyr   0.7.6
## <U+221A> tidyr   0.8.1     <U+221A> stringr 1.3.1
## <U+221A> readr   1.1.1     <U+221A> forcats 0.3.0
```

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

```
library(RColorBrewer)
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

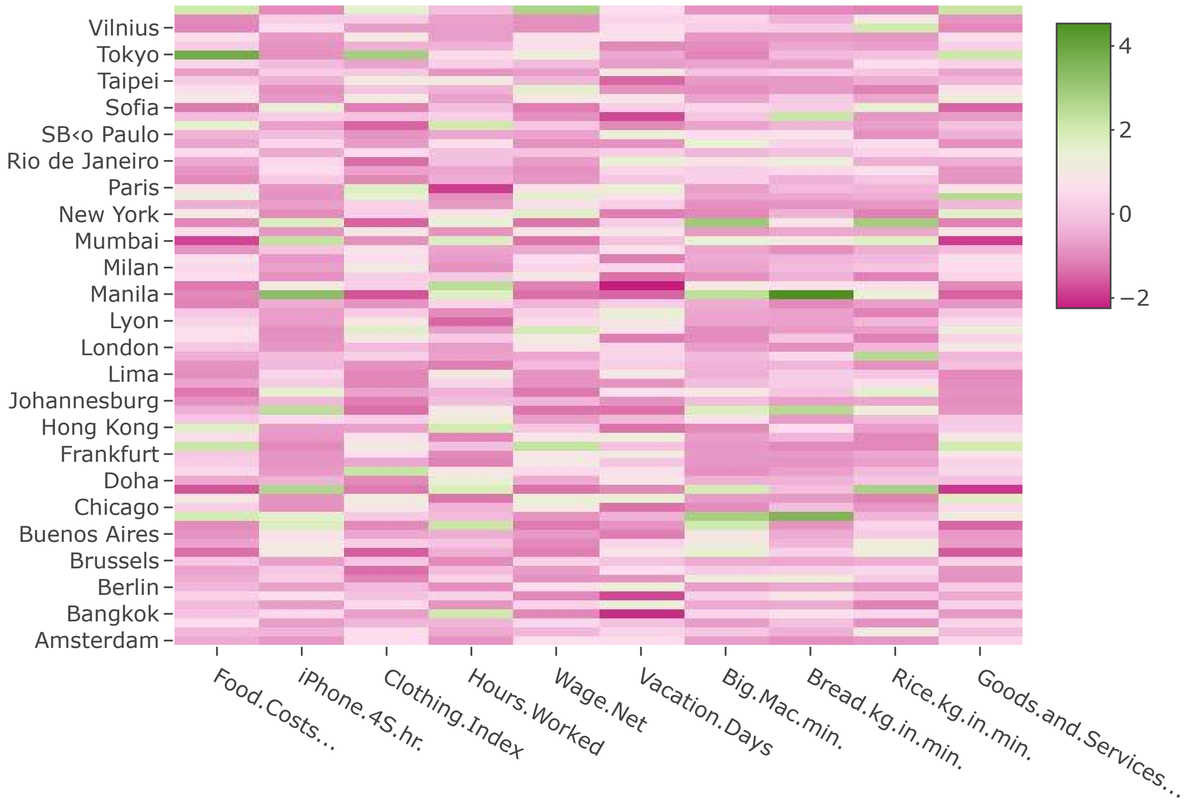
```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(seriation)
df<-read.csv("prices-and-earnings.csv")
adults<-df[c(1,2,5,6,7,9,10,16,17,18,19)]

rownames(adults)<-df[,1]
adults<-adults[,-c(1)]
```

2.Heat map of the data

```
coul = colorRampPalette(brewer.pal(8, "PiYG"))(25)
adscaled=scale(adults)
p2<-plot_ly(x=colnames(adscaled), y=rownames(adscaled), z=adscaled, type="heatmap", colors = cou
l)
p2
```

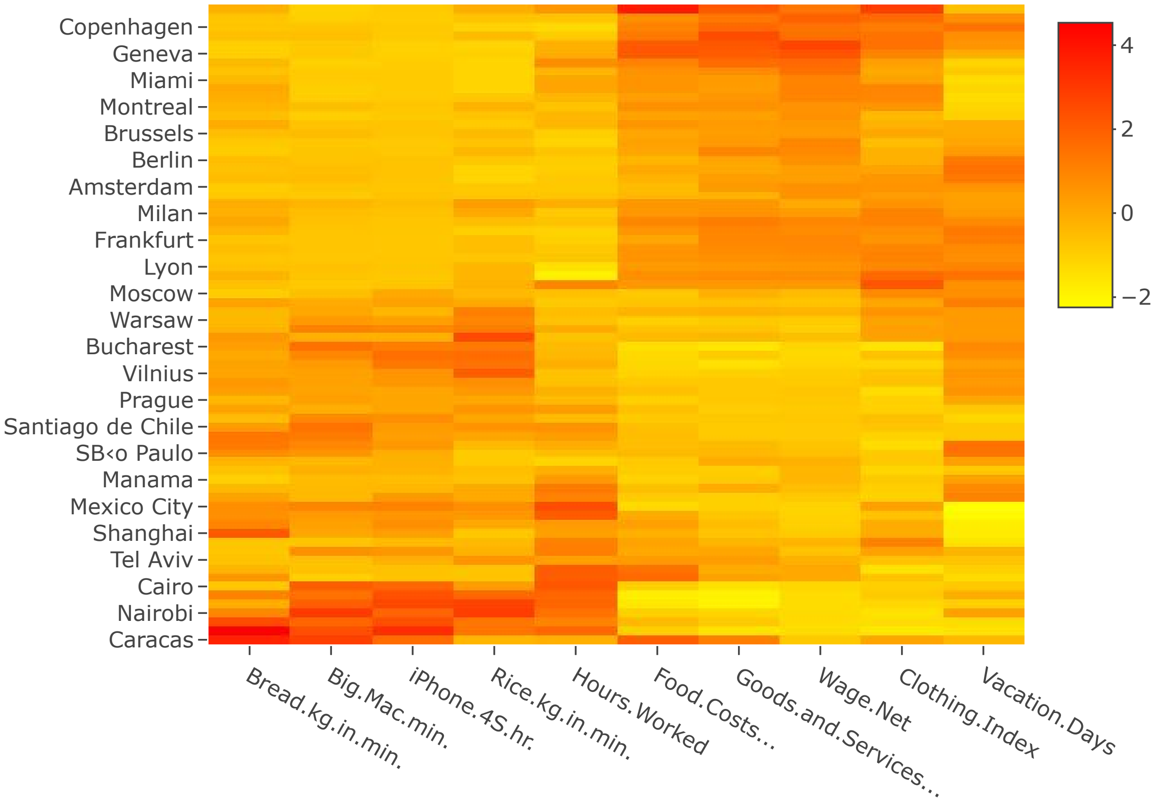


some outliers can be spotted.But it is very difficult to plot clusters since the data is unordered.

3.Heatmaps

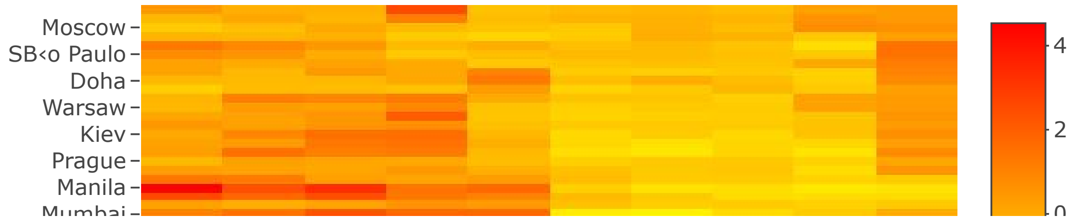
Heat map using eucledian distance

```
rowdist_e<-dist(adscaled,method = "minkowski", p=2)
coldist_e<-dist(t(adscaled),method = "minkowski", p=2)
order1_e<-seriate(rowdist_e, "GW")
order2_e<-seriate(coldist_e, "GW")
ord1_e<-get_order(order1_e)
ord2_e<-get_order(order2_e)
reordmatr_euc<-adscaled[rev(ord1_e),ord2_e]
plot_ly(x=colnames(reordmatr_euc), y=rownames(reordmatr_euc),
        z=as.matrix(reordmatr_euc), type="heatmap", colors =colorRamp(c("yellow", "red")))
```



Heat Map with 1-Correlation HC

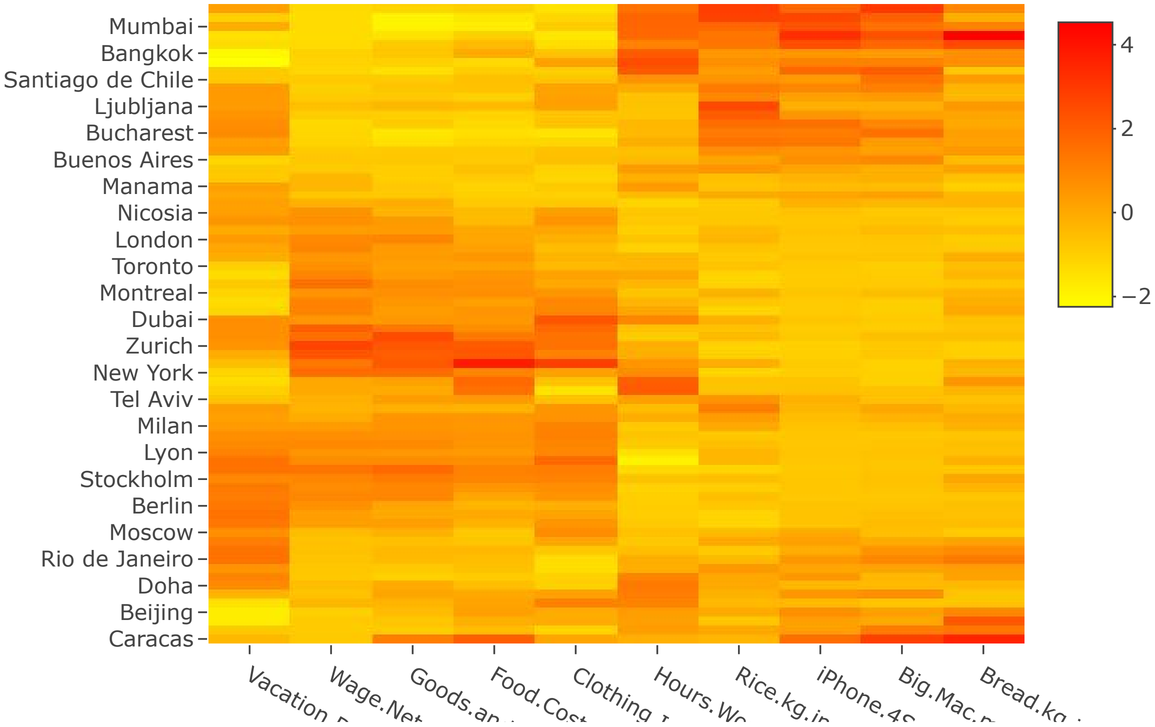
```
coldist_c<-as.dist(1-cor(adscaled))
rowdist_c<-as.dist(1-cor(t(adscaled)))
order1_c<-seriate(rowdist_c, "GW")
order2_c<-seriate(coldist_c, "GW")
ord1_c<-get_order(order1_c)
ord2_c<-get_order(order2_c)
reordmatr_c<-adscaled[rev(ord1_c),ord2_c]
plot_ly(x=colnames(reordmatr_c), y=rownames(reordmatr_c),
        z=as.matrix(reordmatr_c), type="heatmap", colors =colorRamp(c("yellow", "red")))
```



THe two plots arrive at same conclusions as both of the plots are related.

4.Heat Map with Euclidean Distance TSP

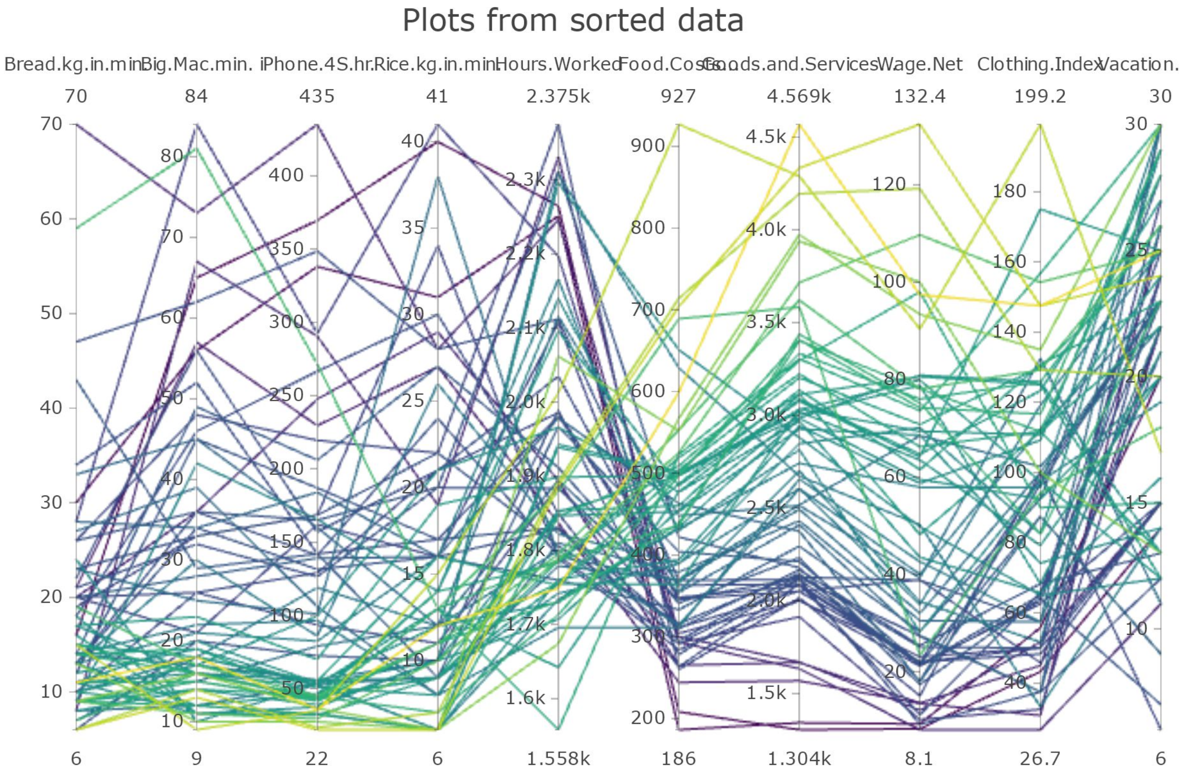
```
rowdist_tsp<-dist(adscaled,method = "minkowski", p=2)
coldist_tsp<-dist(t(adscaled),method = "minkowski", p=2)
order1_tsp<-seriate(rowdist_tsp, "TSP")
order2_tsp<-seriate(coldist_tsp, "TSP")
ord1_tsp<-get_order(order1_tsp)
ord2_tsp<-get_order(order2_tsp)
reordmatr_tsp<-adscaled[rev(ord1_tsp),ord2_tsp]
plot_ly(x=colnames(reordmatr_tsp), y=rownames(reordmatr_tsp),
        z=as.matrix(reordmatr_tsp), type="heatmap", colors =colorRamp(c("yellow", "red")))
```



TSP during each run produce a different result due to the randomness.

5. Parallel Coordinate Plot for Unsorted data

```
v<-1:11
ord=sample(v)
#ord=ord2

dims0=list()
for( i in 1:ncol(adults)){
  dims0[[i]]=list( label=colnames(adults)[ord2_e[i]],
                   values=as.formula(paste("~",colnames(adults)[ord2_e[i]])))
}
p <-adults %>%
  plot_ly(type = 'parcoords',
          line = list(color = ~as.numeric(Goods.and.Services...)),
          dimensions = dims0
  )%>%layout(title = "Plots from sorted data")

p
```



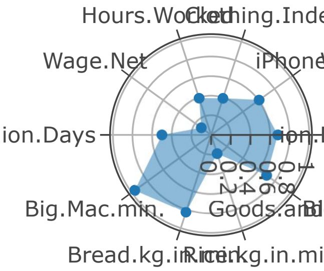
6. Radar plot

```
library(scales)
Ps=list()
nPlot=72

ad<-as.data.frame(adscaled)
ad[rev(ord1_e),]%>%
  add_rownames( var = "group" ) %>%
  mutate_each(funs(rescale), -group) -> pe_radar
```

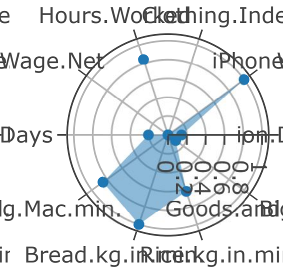
```
## Warning: Deprecated, use tibble::rownames_to_column() instead.
```

```
for (i in 1:nPlot){
  Ps[[i]] <- htmltools::tags$div(
    plot_ly(type = 'scatterpolar',
            r=as.numeric(pe_radar[i,-1]),
            theta= colnames(pe_radar)[-1],
            fill="toself")%>%
      layout(title=pe_radar$group[i]), style="width: 20%;")  # 4 plots per row
}

h <-htmltools::tags$div(style = "display: flex; flex-wrap: wrap", Ps)
p <- htmltools::browsable(h)
p
```

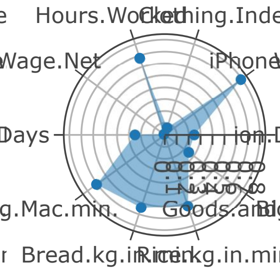
Caracas



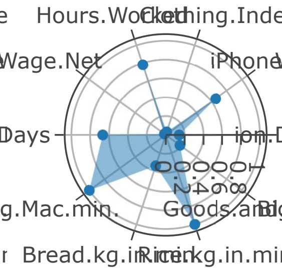
Manila



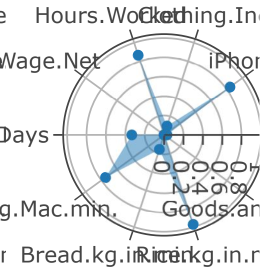
Jakarta



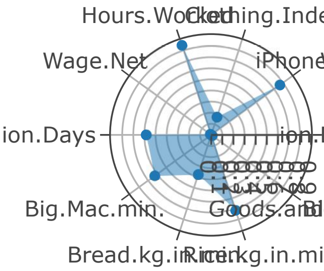
Nairobi



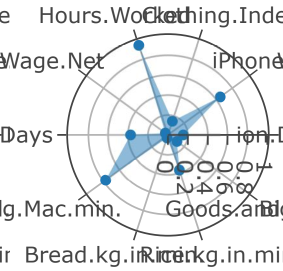
Delhi



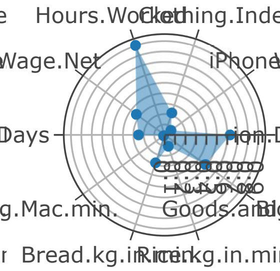
Mumbai



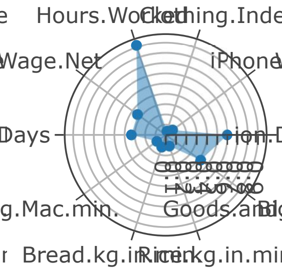
Cairo



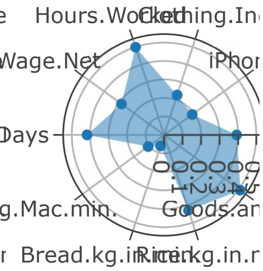
Hong Kong



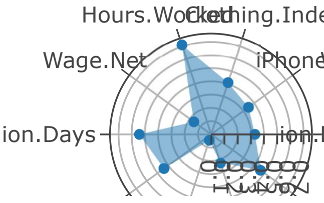
Seoul



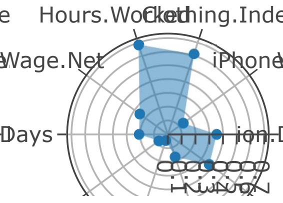
Tel Aviv



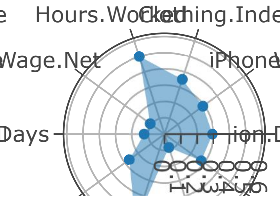
Istanbul



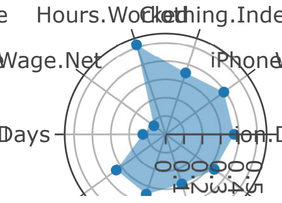
Taipei



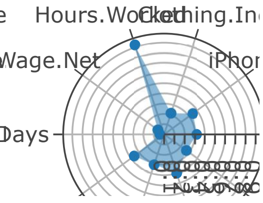
Shanghai



Beijing



Bangkok



```
#Using Juxtaposed and set scale to True
# stars(adults[rev(ord1_e),],
#       key.loc=c(23,2),
#       col.stars =rep("lightblue", nrow(df)),
#       cex = 0.5, lwd = 0.25, lty = par("lty"),
#       scale = TRUE)
```

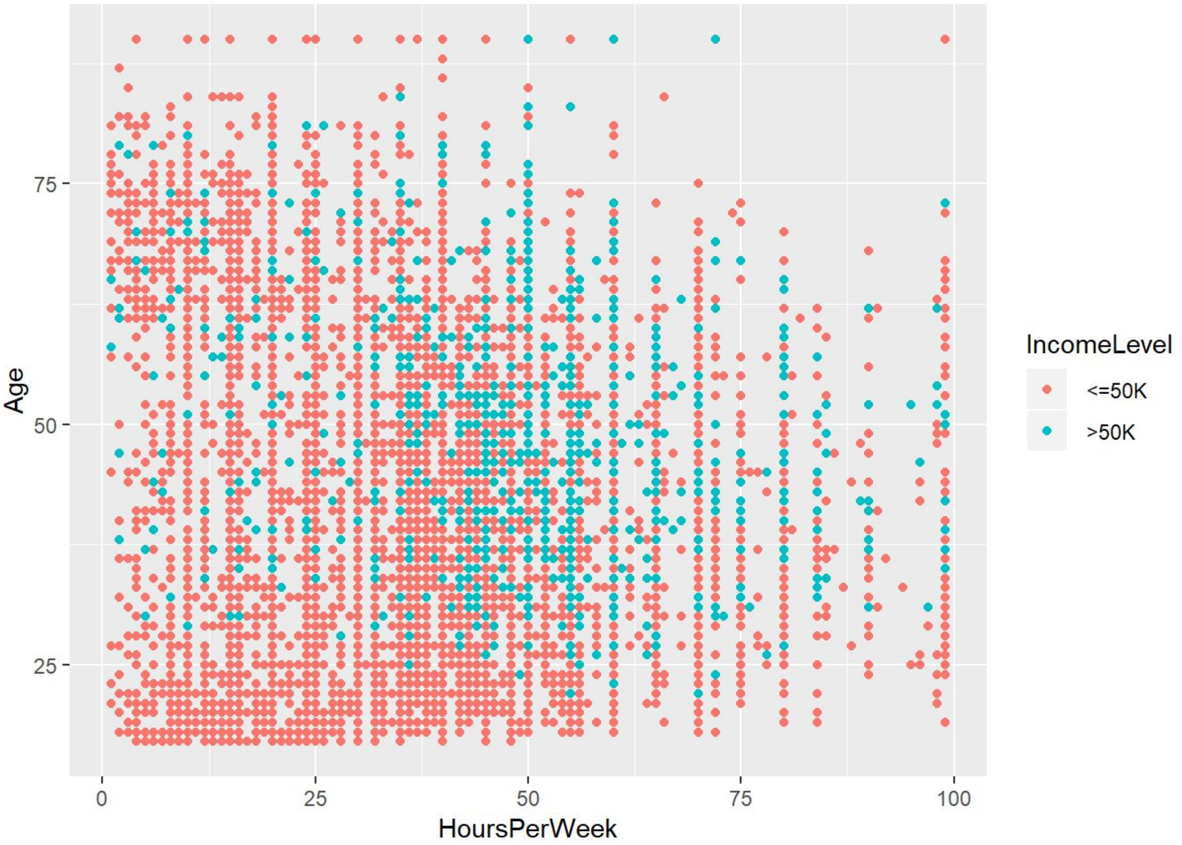
7.Best one

We think radarchart is good.Radar charts are best when we need to compare small amount of observations.Plotting one over the other we can come to conclusions.BUt if the data is dissimilar then it is hard to compare.

Assignment 2

1.Scatterplot

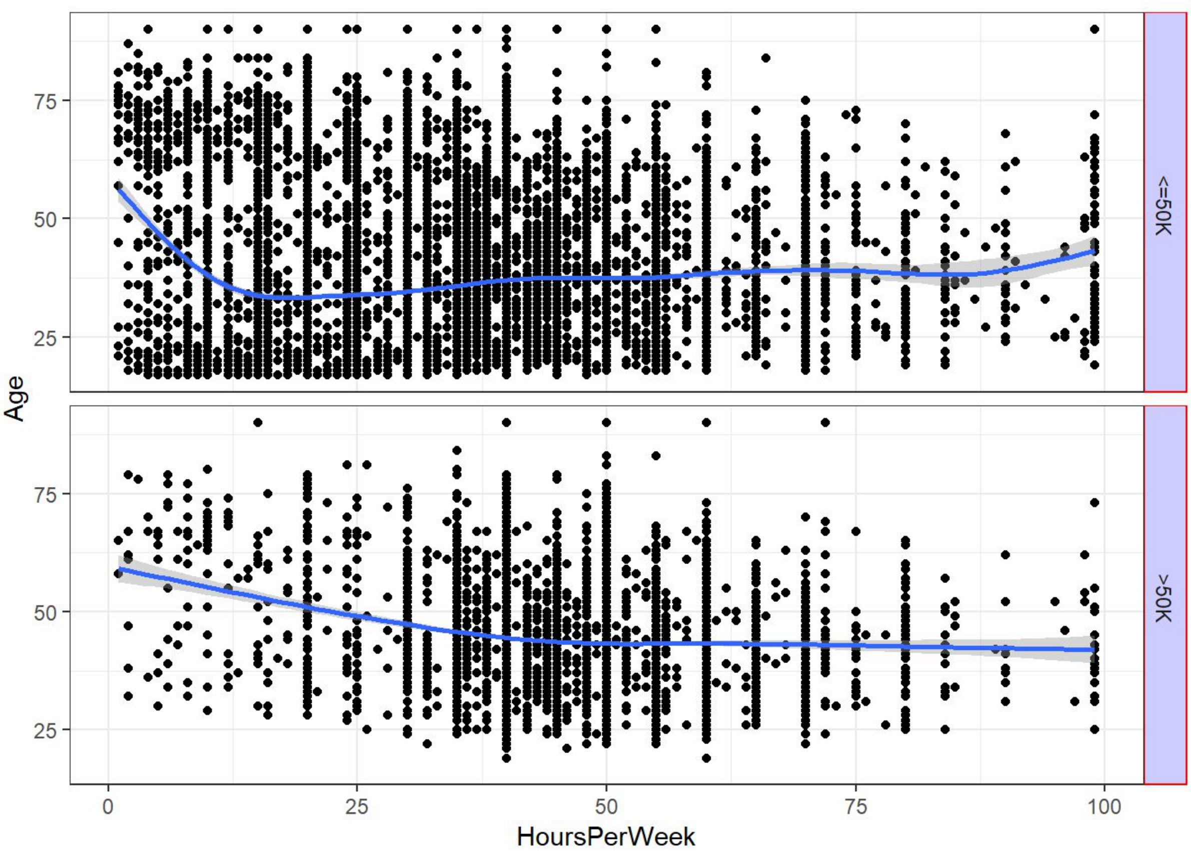
```
library(tidyverse)
library(RColorBrewer)
library(plotly)
library(ggplot2)


adults<-read.csv("adult.csv",sep=",")
names(adults)<-c("Age","Workclass","PolpulationIndex","Education","EducationNum","MaritalStatus"
,"Occupation","Relationship",
                 "Race","Sex","CapitalGain","CapitalLoss","HoursPerWeek","NativeCountry","Income
Level")


sc1<-ggplot(adults,aes(x=HoursPerWeek,y=Age,col=IncomeLevel))+geom_point()
sc1
```



trillis plot

```
tr1<-ggplot(adults,aes(x=HoursPerWeek,y=Age))+geom_point()+facet_grid(IncomeLevel~.)+theme_bw()+
theme(strip.background = element_rect(colour="red", fill="#CCCCFF"))+geom_smooth()
tr1
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

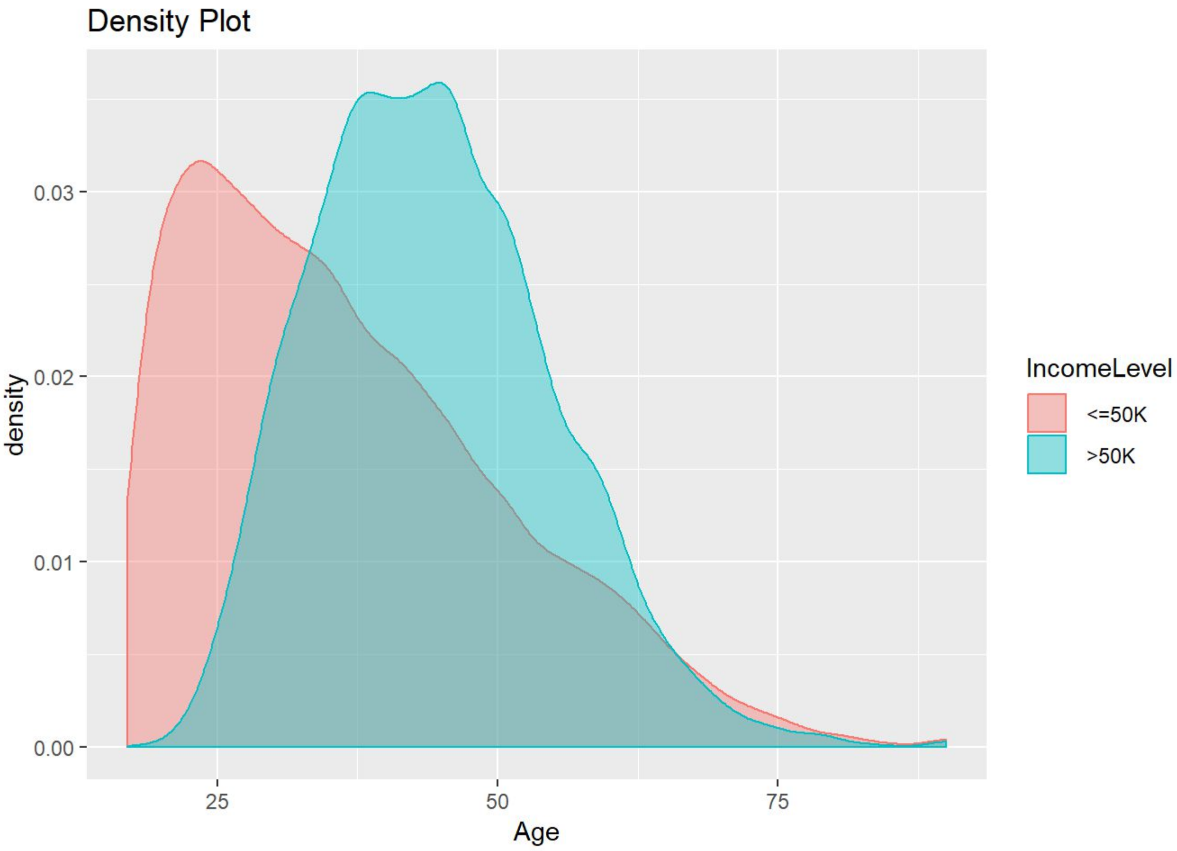


THe first plot is clear case of overplotting and hence difficult to make conclusions.People having less income work more hours per week.The people having less income also seems to work for a longer age.We can make this conclusion easily using trillis plot.

2.

Use ggplot2 to create a density plot of age grouped by the Income level.

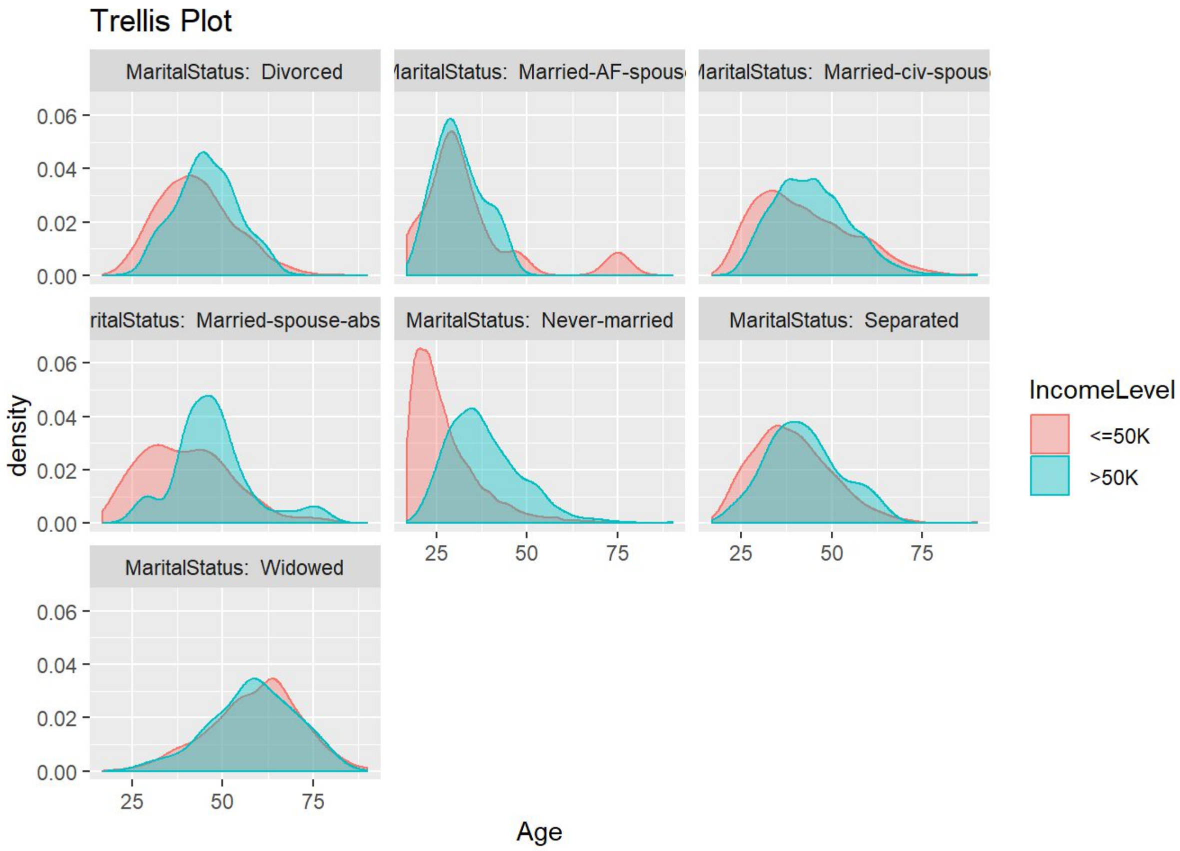
```
d1<-ggplot(adults,aes(x=Age))+geom_density(aes(fill=IncomeLevel,color=IncomeLevel),alpha=0.4)+gg
title("Density Plot")
d1
```



People earning less than 50k are younger age group mostly between 20 and 35 and those with income level greater than 50k are in range of age group 30-50

Create a trellis plot of the same kind where you condition on Marital Status

```
d2<-ggplot(adults,aes(x=Age))+
  geom_density(aes(col=IncomeLevel,fill=IncomeLevel),alpha=0.4)+facet_wrap(~MaritalStatus, labeller = "label_both")+ggtitle("Trellis Plot ")
d2
```

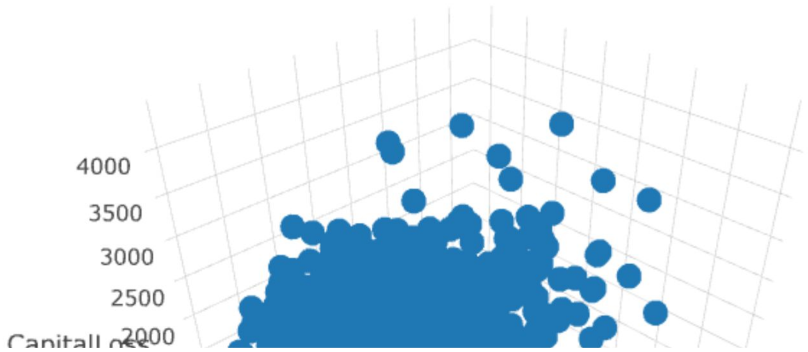


There are 7 different plots for different married status.Never married and Married AF spouse have the same pattern.Rest five plots follow another same pattern of income distribution for different age group and also these plots are similar to the normal density plot of income vs age group.

2.3

3 d scatter plot

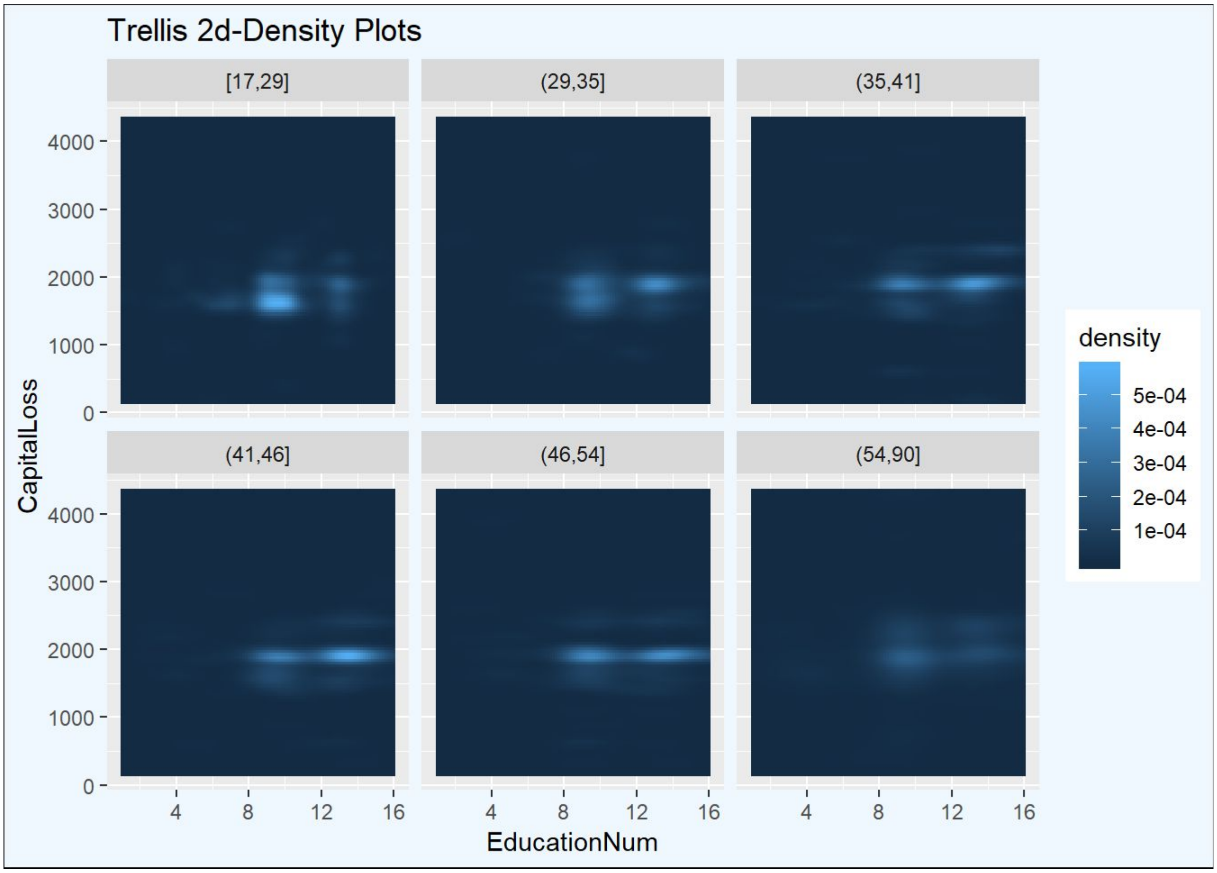
```
pp <- adults%>%filter(CapitalLoss!=0)%>%plot_ly( x = ~EducationNum, y = ~Age, z = ~CapitalLoss)
%>%
  add_markers() %>%
  layout(scene = list(xaxis = list(title = 'EducationNum'),
                      yaxis = list(title = 'Age'),
                      zaxis = list(title = 'CapitalLoss')))
pp
```



The above plot is a clear example of overplotting.Hence we cannot get a clear understanding of the data.

Create a trellis plot with 6 panels in ggplot2 in which each panel shows a raster-type 2d-density plot of Capital Loss versus Education-num conditioned on values of Age (use cut_number())

```
tr2<-adults[which(adults$CapitalLoss!=0),]%>%ggplot(aes(y=CapitalLoss,x=EducationNum))+stat_density2d(aes(fill = stat(density)), geom = "raster",contour = FALSE)+facet_wrap(~cut_number(Age,n=6))
tr2+ggtitle("Trellis 2d-Density Plots")+theme_gray()%+replace%theme(plot.background = element_rect(fill = "aliceblue"))
```



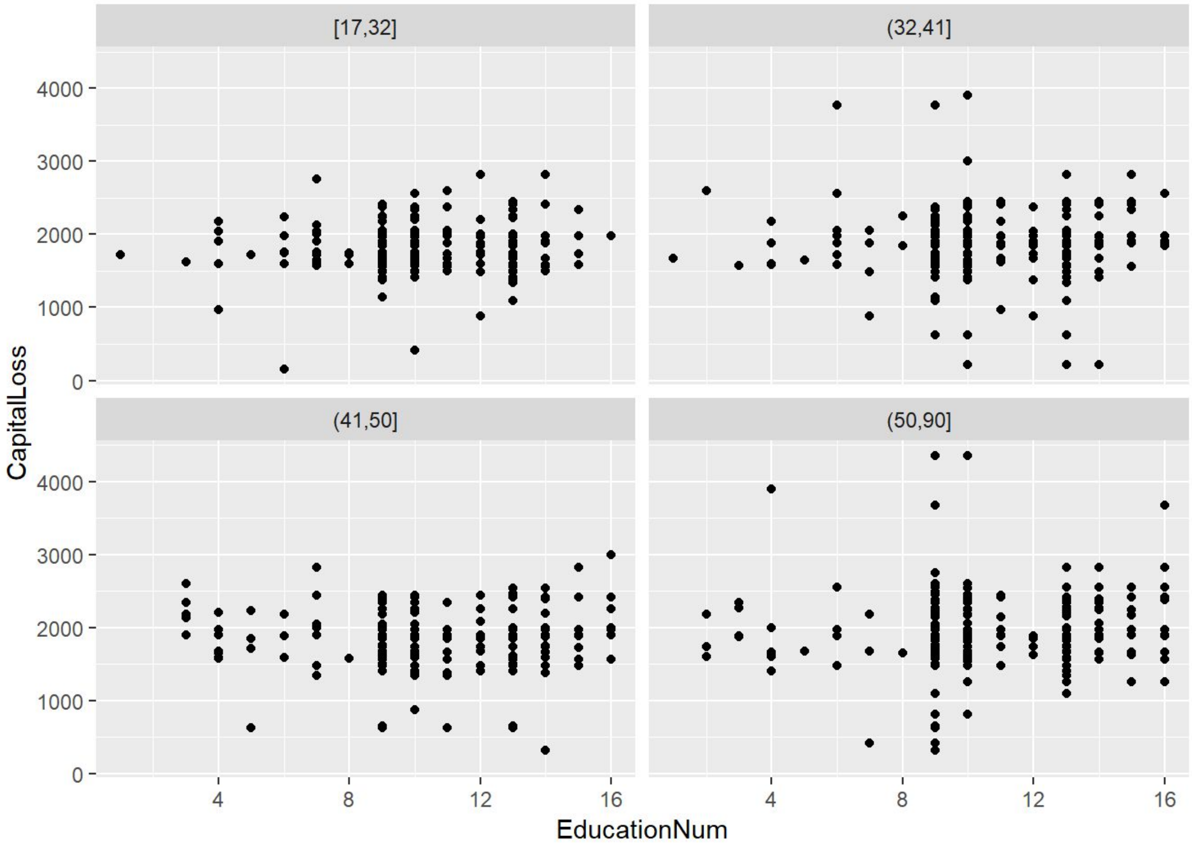
The data is better understandable in this plot when compared to the 3 d plot.We can say there is high density of capital loss around 2000 for age groups [17,29],(46,54].

2.4

Make a trellis plot containing 4 panels where each panel should show a scatter plot of Capital Loss versus Education-num conditioned on the values of Age by

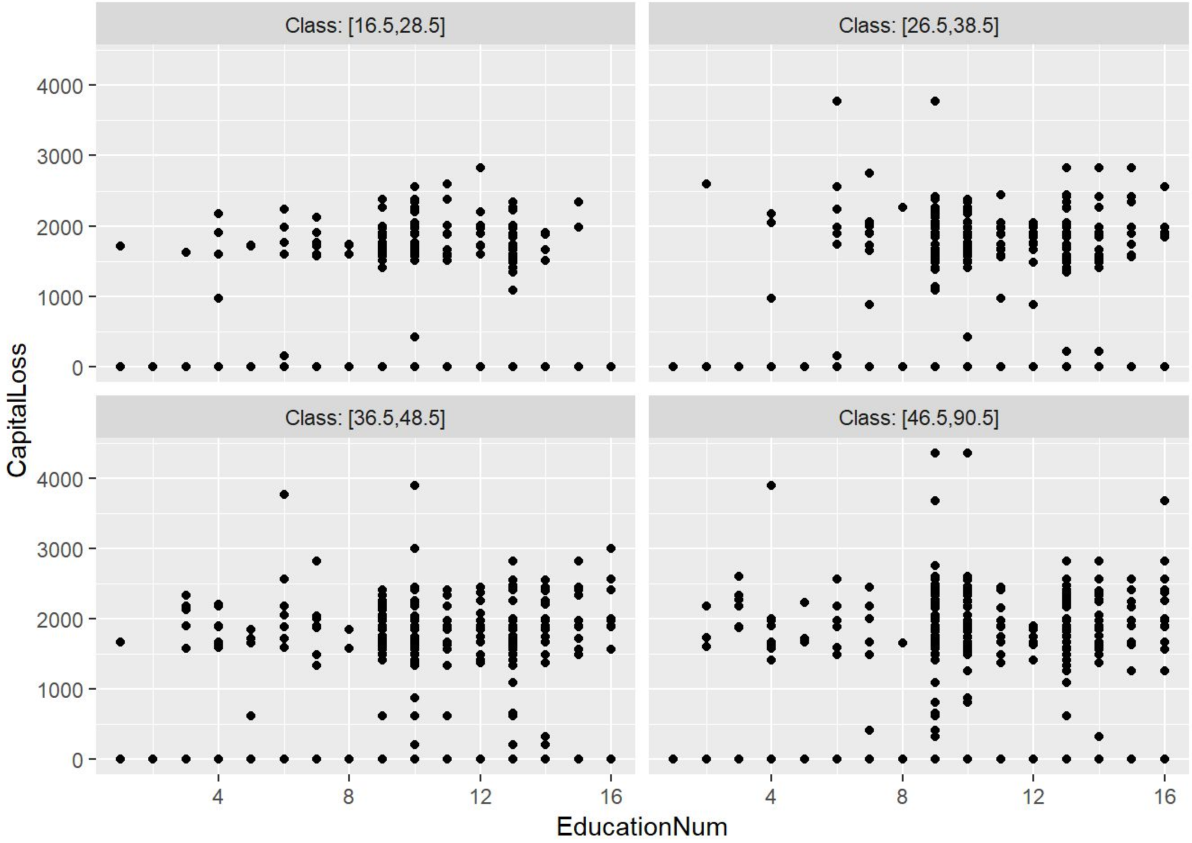
a. using cut_number()

```
adults%>%filter(CapitalLoss!=0)%>%ggplot(aes(x=EducationNum,y=CapitalLoss))+geom_point()+facet_w
rap(~cut_number(Age,n=4))
```



b) using Shingles with 10% overlap.

```
Agerange<-lattice::equal.count(adults$Age, number=4, overlap=0.10) #overlap is 10%
L<-matrix(unlist(levels(Agerange)), ncol=2, byrow = T)
L1<-data.frame(Lower=L[,1],Upper=L[,2], Interval=factor(1:nrow(L)))
index=c()
Class=c()
for(i in 1:nrow(L)){
  Cl=paste("[", L1$Lower[i], ",", L1$Upper[i], "]", sep="")
  ind=which(adults$Age>=L1$Lower[i] & adults$Age<=L1$Upper[i])
  index=c(index,ind)
  Class=c(Class, rep(Cl, length(ind))) }

df4<-adults[index,]
df4$Class<-as.factor(Class)
h1<-ggplot(df4, aes(x=EducationNum,y=CapitalLoss))+ geom_point()+ facet_wrap(~Class, labeller =
"label_both")
h1
```



Advantage of the shingles is that sudden jumps in data would not be missed if it occured in the transition period .One disadvantage is that amount of coding is high and the next is that sometimes the data will be overalpped and hence be present in both plots and can lead to misleading calculations.

Appendix

```r
library(tidyverse)
library(RColorBrewer)
library(plotly)
library(seriation)

df<-read.csv("prices-and-earnings.csv")
adults<-df[c(1,2,5,6,7,9,10,16,17,18,19)+1]

rownames(adults)<-df[,1]
coul = colorRampPalette(brewer.pal(8, "PiYG"))(25)
adscaled=scale(as.matrix(adults))
p2<-plot_ly(x=colnames(adscaled), y=rownames(adscaled), z=adscaled, type="heatmap", colors = cou
l)
p2


rowdist<-dist(adscaled,method = "minkowski", p=2)
coldist<-dist(t(adscaled),method = "minkowski", p=2)
order1<-seriate(rowdist, "HC")
order2<-seriate(coldist, "HC")
ord1<-get_order(order1)
ord2<-get_order(order2)
reordmatr_euc<-adscaled[rev(ord1),ord2]
plot_ly(x=colnames(reordmatr_euc), y=rownames(reordmatr_euc),
        z=as.matrix(reordmatr_euc), type="heatmap", colors =colorRamp(c("yellow", "red")))

v<-1:11
ord=sample(v)
#ord=ord2
dims0=list()
for( i in 1:ncol(adults)){
  dims0[[i]]=list( label=colnames(adults)[ord[i]],
                   values=as.formula(paste("~",colnames(adults)[ord[i]])))
}
p <-adults%>%
  plot_ly(type = 'parcoords',
          line = list(color = ~as.numeric(Hours.Worked)),dimensions = dims0)
p


radar_value <- reordmatr_euc%>%
  as.tibble(rownames = "name")%>%
  tidyr::gather(variable, value, -name, factor_key=T)%>%
  mutate(name = factor(name,levels = rownames(reordmatr_euc)))

radar_value %>%
  ggplot(aes(x=variable, y=value, group=name)) +
  coord_polar() + theme_bw() + facet_wrap(~name, ncol=13) +
  theme(axis.text.x = element_text(size = rel(.5)))
Ps<-list()
for (i in 1:12){
  Ps[[i]] <- htmltools::tags$div(
    plot_ly(type = 'scatterpolar',
            r=as.numeric(reordmatr_euc[i,-1]),
            theta= colnames(reordmatr_euc)[-1],
            fill="toself")%>%
      layout(title=rownames(reordmatr_euc)[i]), style="width: 25%;")
}

h <-htmltools::tags$div(style = "display: flex; flex-wrap: wrap", Ps)

htmltools::browsable(h)

library(tidyverse)
library(RColorBrewer)
library(plotly)


adults<-read.csv("adult.csv",sep=",")
names(adults)<-c("Age","Workclass","PolpulationIndex","Education","EducationNum","MaritalStatus"
,"Occupation","Relationship",
                 "Race","Sex","CapitalGain","CapitalLoss","HoursPerWeek","NativeCountry","Income
Level")

### scatter plot of Hours per Week versus age where observations are colored by Income level.
# Make a trellis plot of the same kind where you condition on Income Level

sc1<-ggplot(adults,aes(x=HoursPerWeek,y=Age,col=IncomeLevel))+geom_point()
sc1

tr1<-ggplot(adults,aes(x=HoursPerWeek,y=Age))+geom_point()+facet_grid(IncomeLevel~.)+theme_bw()+
theme(strip.background = element_rect(colour="red", fill="#CCCCFF"))+geom_smooth()
tr1

##Use ggplot2 to create a density plot of age grouped by the Income level.
##Create a trellis plot of the same kind where you condition on Marital Status


d1<-ggplot(adults,aes(x=Age))+geom_density(aes(fill=IncomeLevel,color=IncomeLevel),alpha=0.4)+gg
title("Density Plot")
d1


d2<-ggplot(adults,aes(x=Age))+
  geom_density(aes(col=IncomeLevel,fill=IncomeLevel),alpha=0.4)+facet_wrap(~MaritalStatus, label
ler = "label_both")+ggtitle("Trellis Plot ")
d2


##Filter out all observations having Capital loss equal to zero. For the remaining data,
##use Plotly to create a 3D-scatter plot of Education-num vs Age vs Captial Loss

pp <- adults%>%filter(CapitalLoss!=0)%>%plot_ly( x = ~EducationNum, y = ~Age, z = ~CapitalLoss,c
olor=~EducationNum) %>%
  add_markers() %>%
  layout(scene = list(xaxis = list(title = 'EducationNum'),
                      yaxis = list(title = 'Age'),
                      zaxis = list(title = 'CapitalLoss')))
pp

###Create a trellis plot with 6 panels in ggplot2 in which each panel shows a raster-type 2d-den
sity plot of Capital Loss
```

```
####versus Education-num conditioned on values of Age (use cut_number())


tr2<-adults[which(adults$CapitalLoss!=0),]%>%ggplot(aes(x=CapitalLoss,y=EducationNum))+geom_density_2d()+facet_wrap(~cut_number(Age,n=6))
tr2+ggtitle("Trellis 2d-Density Plots")+theme_gray()%+replace%theme(plot.background = element_rect(fill = "aliceblue"))






###Make a trellis plot containing 4 panels where each panel should show a scatter plot of Capital Loss versus Education-num
###conditioned on the values of Age by a) using cut_number() b) using Shingles with 10% overlap.


#a)
adults%>%filter(CapitalLoss!=0)%>%ggplot(aes(x=CapitalLoss,y=EducationNum))+geom_point()+facet_wrap(~cut_number(Age,n=4))


#b)

#cap_adults<-adults[which(adults$CapitalLoss!=0),]
Agerange<-lattice::equal.count(adults$Age, number=4, overlap=0.10) #overlap is 10%
L<-matrix(unlist(levels(Agerange)), ncol=2, byrow = T)

index=c()
Class=c()
for(i in 1:nrow(L)){
  Cl=paste("[", L1$Lower[i], ",", L1$Upper[i], "]", sep="")
  ind=which(df3$age>=L1$Lower[i] &df3$age<=L1$Upper[i])
  index=c(index,ind)
  Class=c(Class, rep(Cl, length(ind))) }


df4<-adults[index,]
df4$Class<-as.factor(Class)
h1<-ggplot(df4, aes(x=CapitalLoss,y=EducationNum))+ geom_point()+ facet_wrap(~Class, labeller = "label_both")
h1
```