Lab 4 - Vizualisation

Karthikeyan Devarajan (Karde799)

11/02/2021

Assignment 1

1) For further analysis, import data to R and keep only the columns with the following numbers: 1,2,5,6,7,9,10,16,17,18,19. Use the first column as labels in further analysis.

2) Plot a heatmap of the data without doing any reordering. Is it possible to see clusters, outliers?

The group formation is randomly distributed. Most number of outliers are in Vacation days for cities such as Bangkok, Berlin, Manila, and Sofia.

3) Compute distance matrices by a) using Euclidian distance and b) as one minus correlation. For both cases, compute orders that optimize Hamiltonian Path Length and use Hierarchical Clustering (HC) as the optimization algorithm. Plot two respective heatmaps and state which plot seems to be easier to analyse and why. Make a detailed analysis of the plot based on Euclidian distance. Use Euclidian Distance matrix in all coming steps.

3.1) Euclidean Distance

```
## [[1]]
## [[1]]$label
## [1] "Hours.Worked"
##
## [[1]]$values
## ~Hours.Worked
##
##
## [[2]]
## [[2]]$label
## [1] "Bread.kg.in.min."
##
## [[2]]$values
## ~Bread.kg.in.min.
##
##
## [[3]]
## [[3]]$label
## [1] "Rice.kg.in.min."
##
## [[3]]$values
## ~Rice.kg.in.min.
##
##
## [[4]]
## [[4]]$label
## [1] "iPhone.4S.hr."
##
## [[4]]$values
## ~iPhone.4S.hr.
##
##
## [[5]]
## [[5]]$label
## [1] "Big.Mac.min."
##
## [[5]]$values
## ~Big.Mac.min.
##
##
## [[6]]
## [[6]]$label
## [1] "Vacation.Days"
##
## [[6]]$values
## ~Vacation.Days
##
##
## [[7]]
## [[7]]$label
## [1] "Clothing.Index"
##
## [[7]]$values
## ~Clothing.Index
##
##
## [[8]]
## [[8]]$label
## [1] "Food.Costs..."
##
## [[8]]$values
## ~Food.Costs...
##
##
## [[9]]
## [[9]]$label
## [1] "Wage.Net"
##
## [[9]]$values
## ~Wage.Net
##
##
## [[10]]
## [[10]]$label
## [1] "Goods.and.Services..."
##
## [[10]]$values
## ~Goods.and.Services...
```

Euclidean distance is calculated as $\sum (x - x_i)^2$ and It works for scaled data.The clusters can be differentiated easily than the previous step. Minutes taken to buy Food (bread, big mac, rice) and hours to buy iphone 4s are inversely proportional to vacation days, clothing Index, Wage Net,and Food cost. The cities with high correlation between minutes to buy Food(bread, big mac, rice) and hours to buy iphone 4s have less correlation between vacation days, clothing Index, Wage Net,and Food cost. The outlier which is does not belong to flow is vacation days and hours required for buying iPhone 4s is Istanbul, Seoul, Hong Kong and Taipei.

The one minus correlation matrix is 1 minus the average between two values. It also have similar pattern to the heat map obtain through euclidean distance but the order of cities is different. The outliers are vaction days in Bangkok, Mexico city, Carasas.

4) Compute a permutation that optimizes Hamiltonian Path Length but uses Traveling Salesman Problem (TSP) as solver. Compare the heatmap given by this reordering with the heatmap produced by the HC solver in the previous step – which one seems to be better? Compare also objective function values such as Hamiltonian Path length and Gradient measure achieved by row permutations of TSP and HC solvers (Hint: use criterion() function)

The Hamiltonian Path Length optimization using Travelling Salesman Problem has similar heat map to that of heat map produced by one minus correlation matrix. Heat map produced through euclidean distance forms better cluster formation than one minus correlation distance calculation.
TSP and Gradient Measure criterion

```
## The Criterion value from Travelling Salesman Problem is 23140 120.8241
```

```
## The Criterion value from Gradient Measure is 41612 127.3152
```

Since the criterion selection is based on gradient measure and path length, the initialization and computation depends on the laptop performance. So, Set.seed is required to reduce randomization. The travelling salesman problem have lower criterion value than Gradient measure criteria.

5) Use Ploty to create parallel coordinate plots from unsorted data and try to permute the variables in the plot manually to achieve a better clustering picture. After you are ready with this, brush clusters by different colors and comment about the properties of the clusters: which variables are important to define these clusters and what values of these variables are specific to each cluster. Can these clusters be interpreted? Find the most prominent outlier and interpret it.

There are some clusters formed based on some variables.
i) Minutes for bread with 0 - 25 minutes.
ii) Minutes for big mac with 0 - 20 minutes.
iii) Good and Service tax above 2500.

Vacation days can used considered as significant parameter. The clustering based on vacation days more than 20 days plots a better view for analyze.

6) Use the data obtained by using the HC solver and create a radar chart diagram with juxtaposed radars. Identify two smaller clusters in your data (choose yourself which ones) and the most distinct outlier.

nPlot = 72

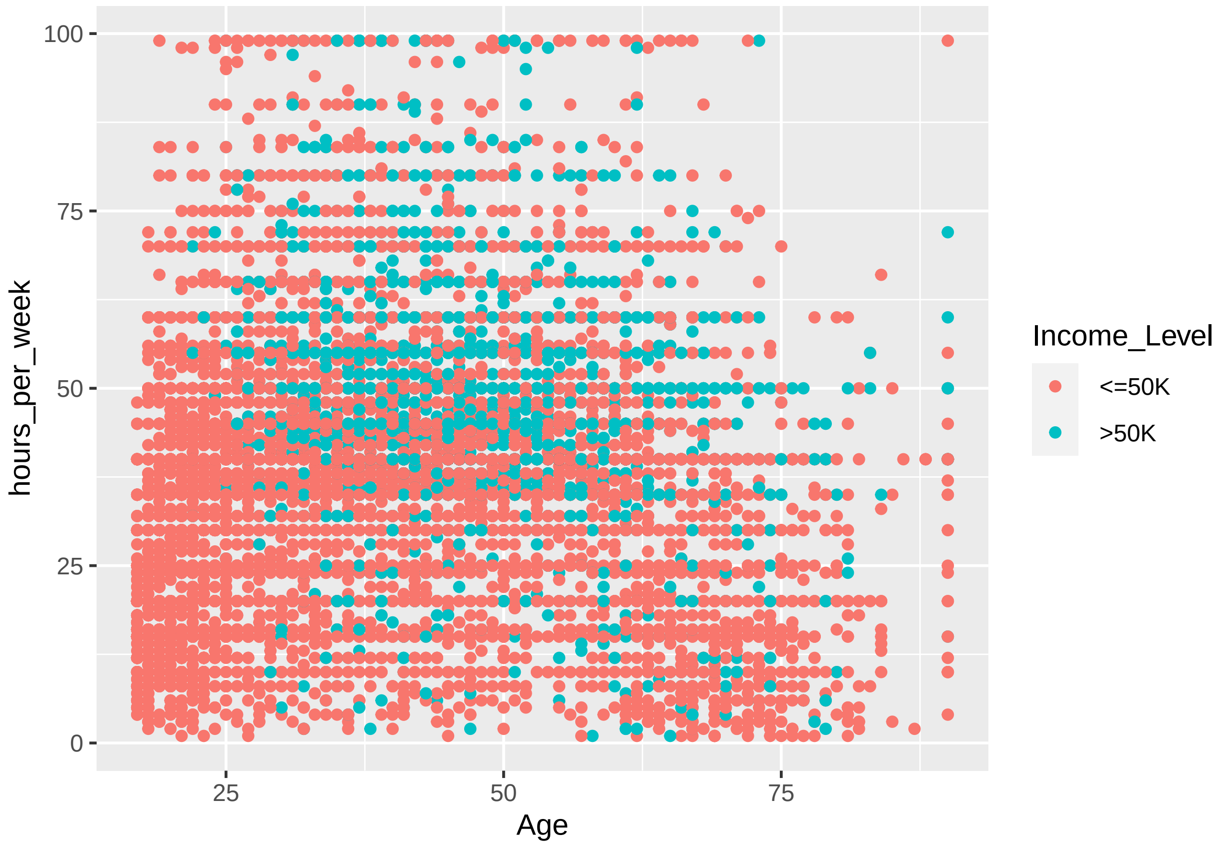
The cluster can be plotted based on similarity between pattern.
Cluster 1: Bucharest and Sofia.
Cluster 2: Kiev and Budapest.
Johannesburg seems to have different pattern compared to other cities. So it could be considered as outlier.

7) Which of the tools you have used in this assignment (heatmaps, parallel coordinates or radar charts) was best in analyzing these data? From which perspective? (e.g. efficiency, simplicity, etc.)

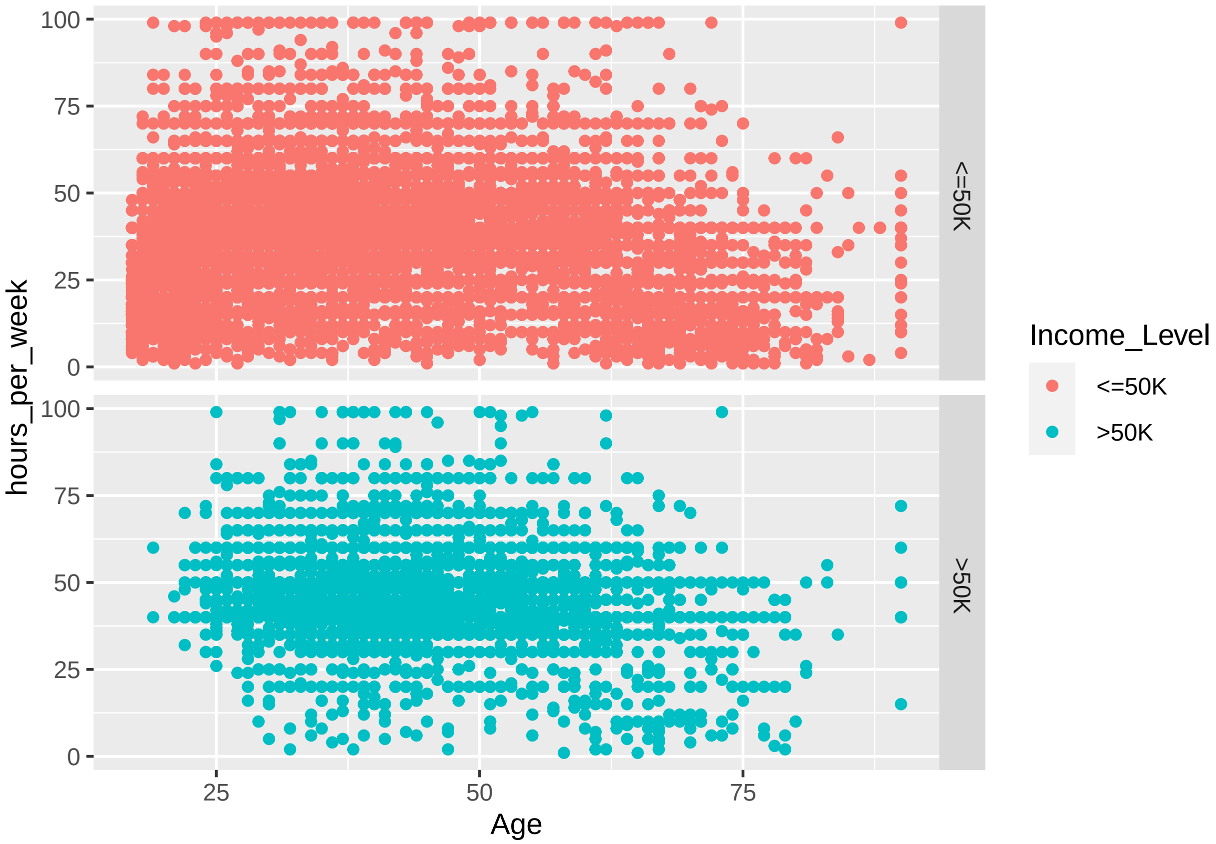
Parallel Coordinate - If a data has many observation then there will many lines will be coinciding and difficult to analyze.Even in this data set, It was difficult analyze because of many lines coinciding.

Radar charts - If number of parameter increases, each chart will compressed. This is plotted after scaling, So the scale between parameter can be same. The number of observation in this data is higher, so it is visually difficult to find similarity between observation.

Heat Map - In terms of simplicity, Heat map is better than other charts. We could easily find cluster in the heat map.
# Assignment 2
1) Use ggplot2 to make a scatter plot of Hours per Week versus age where observations are colored by Income level. Why it is problematic to analyze this plot? Make a trellis plot of the same kind where you condition on Income Level. What new conclusions can you make here?



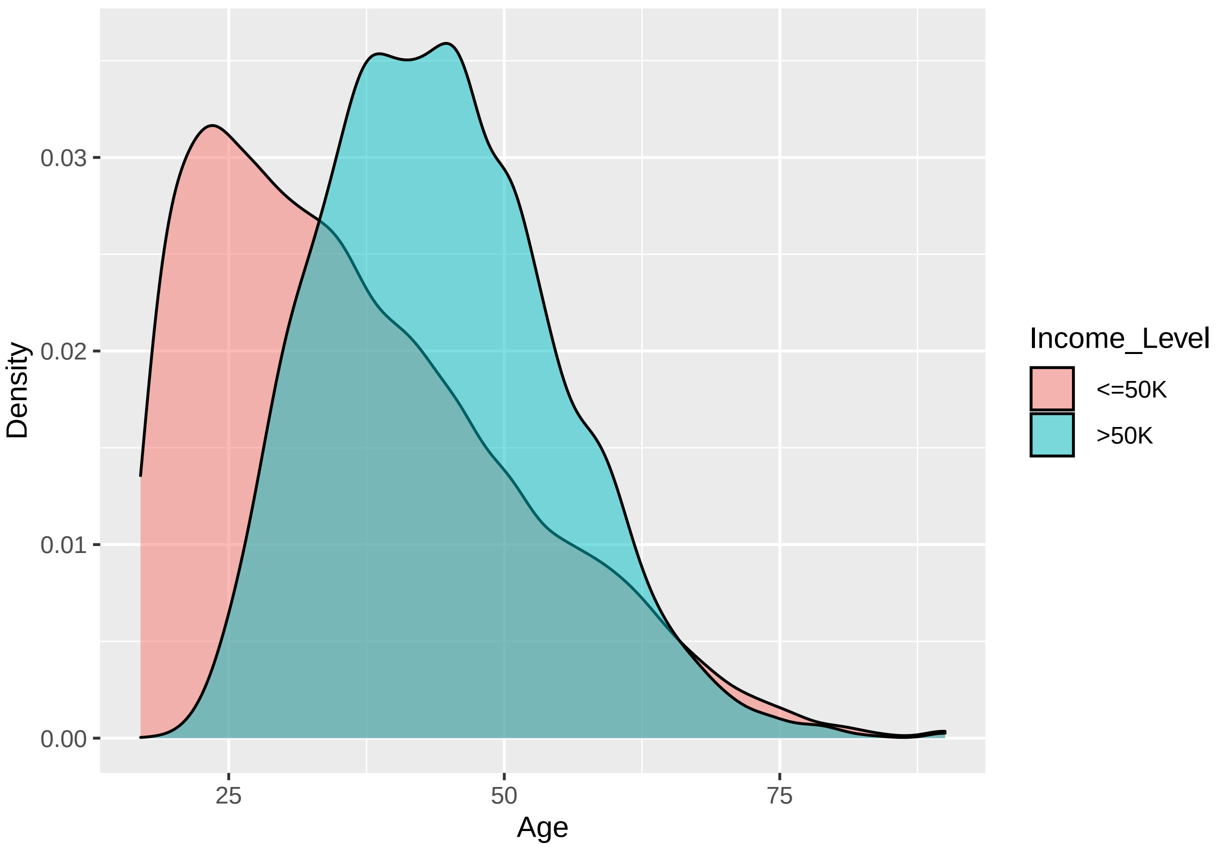
In this plot, the Income level more than 50K and less than equal to 50K coincide in a large area, which is a very difficult to view properly.

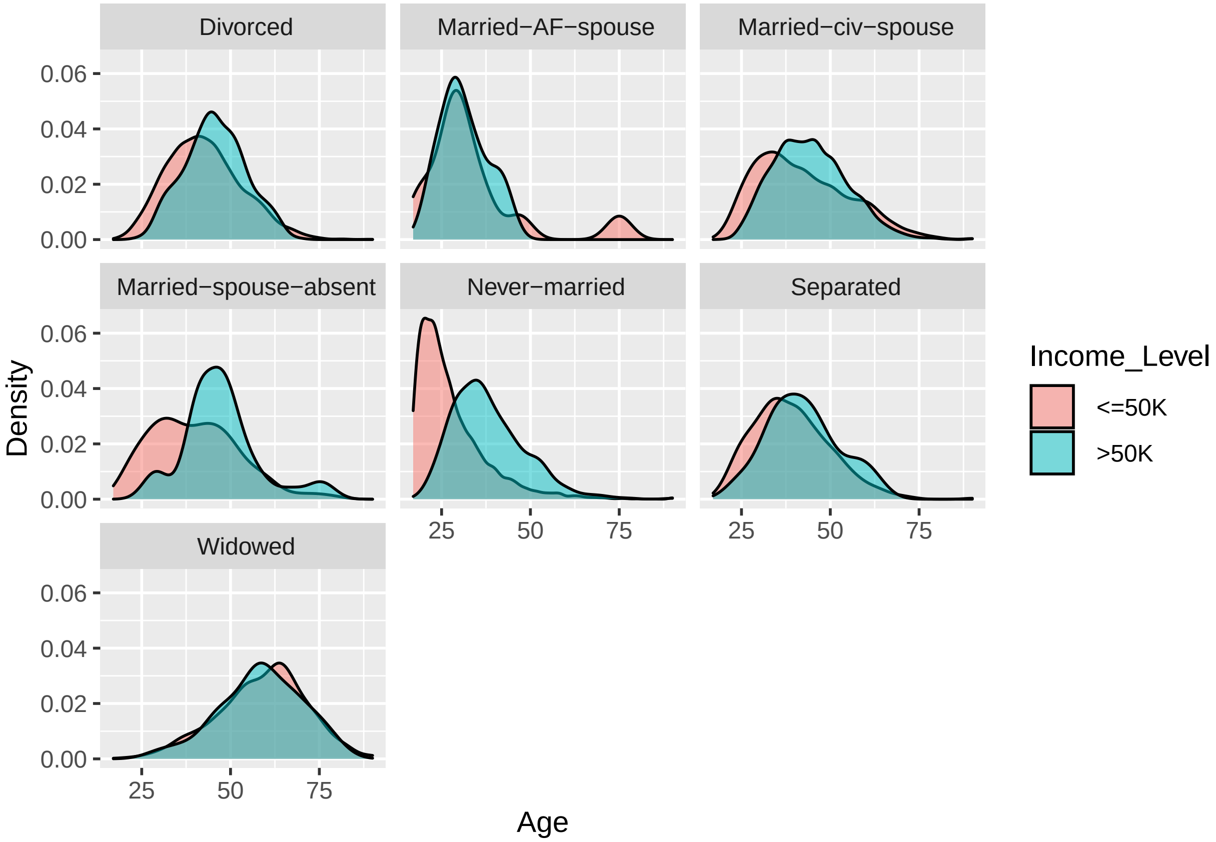


After trellis plot, they are plotted in separate plots and easy to visualize. More people with age 0 to 50, have income level less than 50K. Most people who earn more than 50K lie between age 50 to 75.

2) 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. Analyze these two plots and make conclusions.

```
p2 <- ggplot(data=adults, aes(x=Age, group=Income_Level,fill=Income_Level))
p2 <- p2 + geom_density(alpha=0.5) + ylab("Density")
p2
```

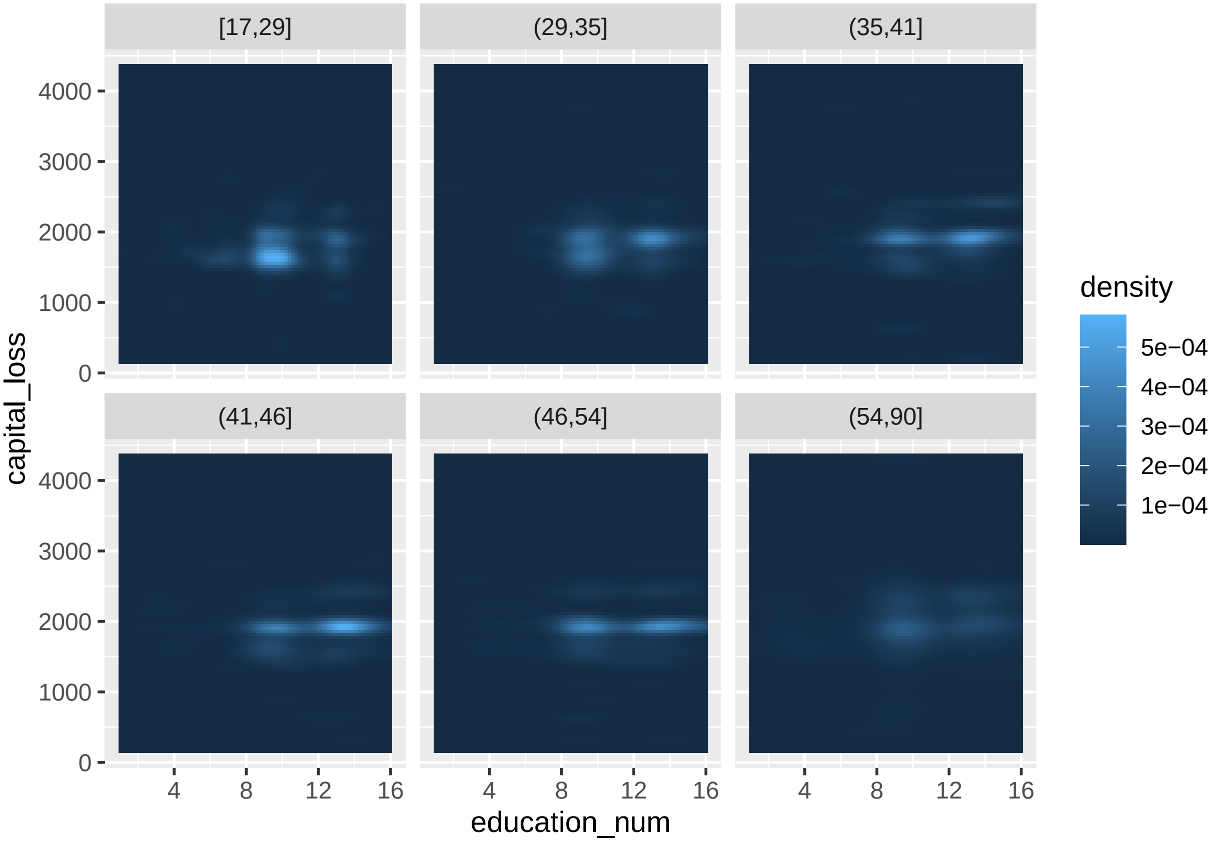

Density
0.03
0.02
0.01
0.00
25
50
75
Age
Income_Level
<=50K
>50K



Most of married spouse absent people have more income than 50K. More people who had never married have mostly income less than 50K.

3) 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. Why is it difficult to analyze this plot? 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()) . Analyze this plot.

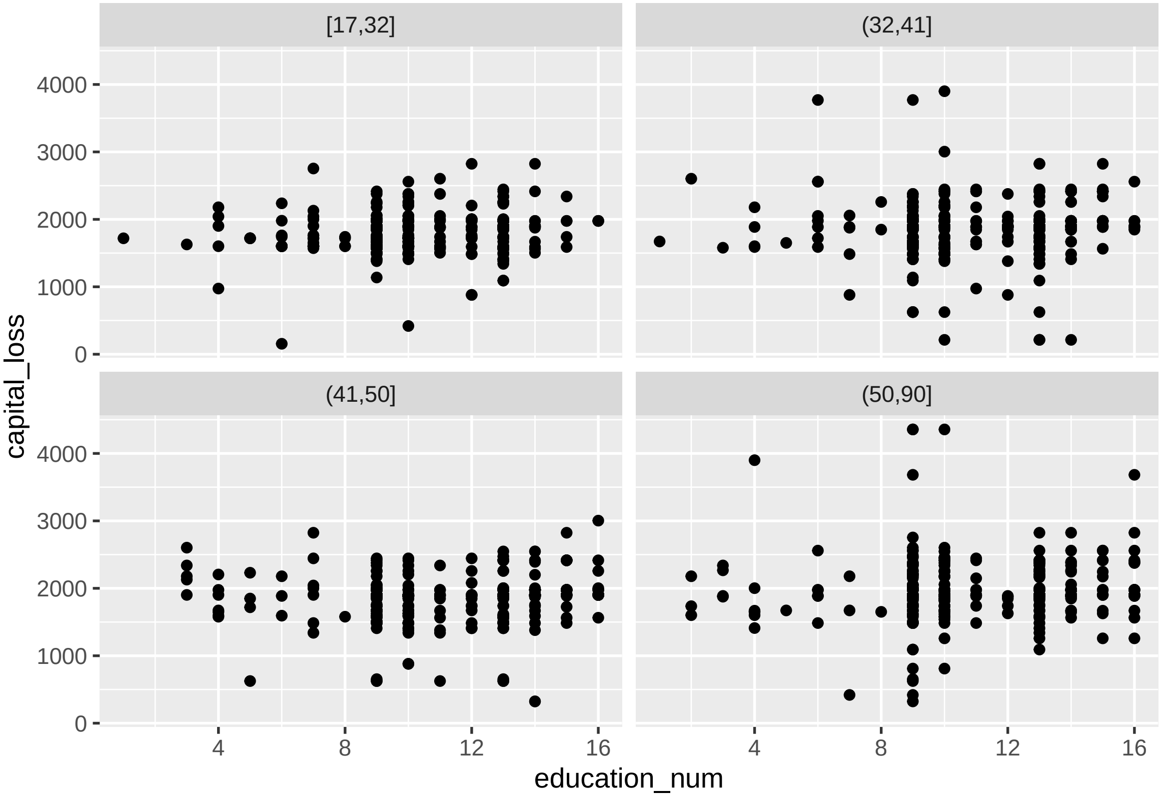
The 3-d plot between the parameters are very difficult to because all the data points are accumulated in the middle and it is very difficult to analyze a pattern in this 3 dimension pattern.



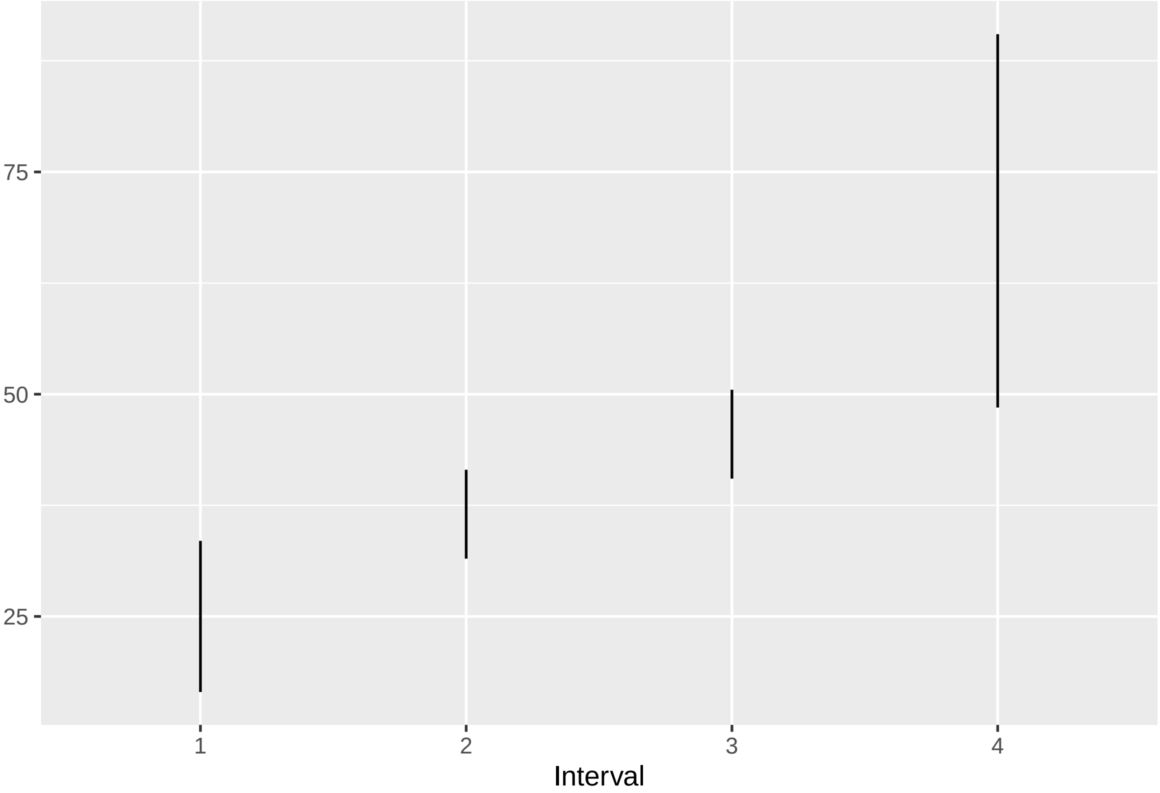
When the age increases, the capital loss is wide spread. In the age group (35 - 41), (41-46) and (46-54) have density around 2000.

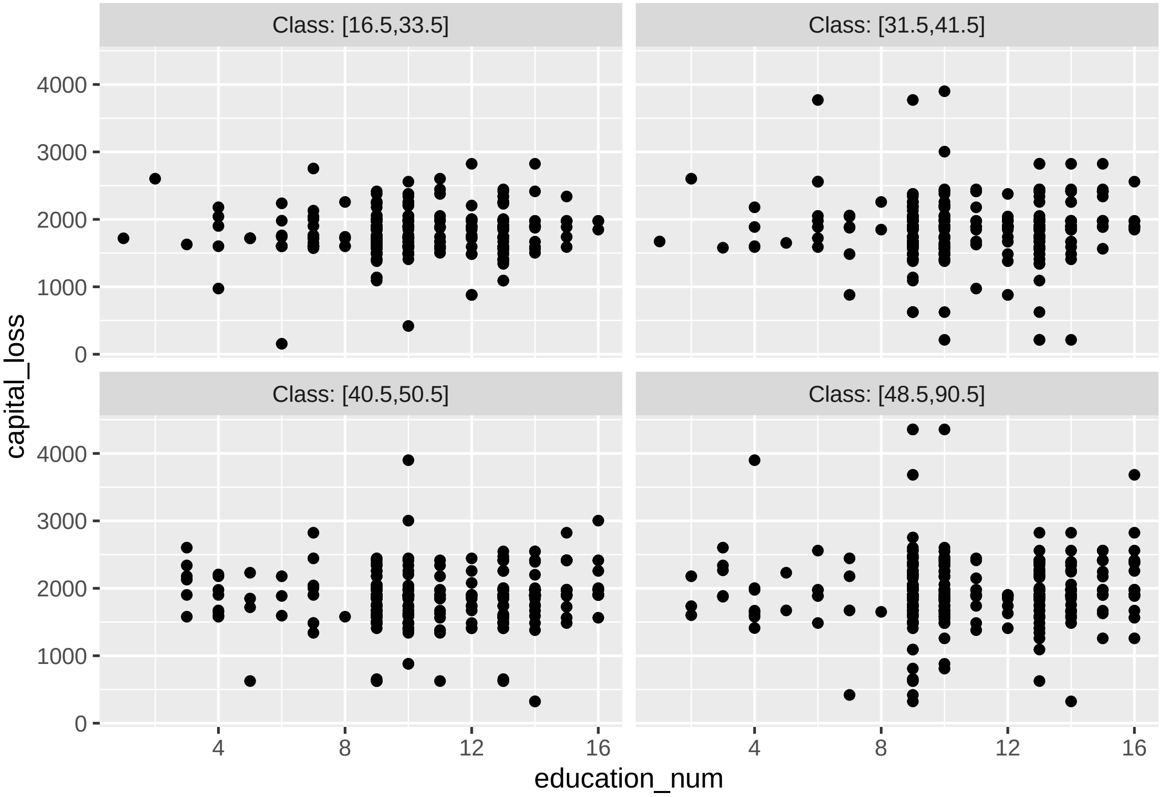
4) Make a trellis plot containing 4 panels where each panel should show a scatter plot of Capital Loss versus Education-num conditioned on the values of Age by a) using cut_number() b) using Shingles with 10% overlap. Which advantages and disadvantages you see in using Shingles?

i) Trellis Plot using cut_number(4)



ii) Using Shingles with 10% overlap





The graph with shingles and without shingles looks similar. While using shingles, the interval is different depending on the percentage of overlap defined. Since, Overlap will include a data point on many segments. This will make the plot difficult to interpet the data.

```
knitr::opts_chunk$set(echo = TRUE)
library(ggplot2)
library(plotly)
library(seriation)
library(dplyr)
library(scales)
price <- read.delim2("price-and-housing.txt")
columns_required <- c(1,2,5,6,7,9,10,16,17,18,19)
price <- price[,columns_required]
row.names(price) <- price$City
price[,c(2:11)] <- matrix(sapply(price[,c(2:11)],as.numeric))
scaleData <- scale(price[,c(2:11)])
p <- plot_ly(x=colnames(scaleData), y=rownames(scaleData), z = scaleData, type = "heatmap")
p

rowdist_euc <- dist(scaleData)
coldist_euc <- dist(t(scaleData))

order_row <- seriate(rowdist_euc, "HC")
order_col <- seriate(coldist_euc, "HC")
order_row <- get_order(order_row)
order_col <- get_order(order_col)
```

```
reordermatrix <- scaleData[rev(order_row),order_col]

dims=list()
for( i in 1:ncol(reordermatrix)){
  dims[[i]]=list(label=colnames(reordermatrix)[i],
    values=as.formula(paste("~",colnames(reordermatrix)[i])))
}

dims
p1 <- as.data.frame(reordermatrix) %>%
  plot_ly(x=colnames(reordermatrix), y=rownames(reordermatrix), z = reordermatrix, type = "heatmap")
p1
rowdist_corr <- 1 - as.dist(cor(t(scaleData)))
coldist_corr <- 1 - as.dist(cor(scaleData))

order_row <-seriate(rowdist_corr, "HC")
order_col <-seriate(coldist_corr, "HC")
order_row <-get_order(order_row)
order_col <-get_order(order_col)

reordermatrix <- scaleData[rev(order_row),order_col]

dims=list()
for( i in 1:ncol(reordermatrix)){
  dims[[i]]=list(label=colnames(reordermatrix)[i],
    values=as.formula(paste("~",colnames(reordermatrix)[i])))
}

p2 <- as.data.frame(reordermatrix) %>%
  plot_ly(x=colnames(reordermatrix), y=rownames(reordermatrix), z = reordermatrix, type = "heatmap")

p2
order_row <-seriate(rowdist_euc, "TSP")
order_col <-seriate(coldist_euc, "TSP")
order_row <-get_order(order_row)
order_col <-get_order(order_col)

reordermatrix <- scaleData[rev(order_row),order_col]

dims=list()
for( i in 1:ncol(reordermatrix)){
  dims[[i]]=list(label=colnames(reordermatrix)[i],
    values=as.formula(paste("~",colnames(reordermatrix)[i])))
}

p3 <- as.data.frame(reordermatrix) %>%
  plot_ly(x=colnames(reordermatrix), y=rownames(reordermatrix), z = reordermatrix, type = "heatmap")

p3
set.seed(12345)
#TSP
order_row <-seriate(rowdist_euc, "TSP")
TSP_criteria <- criterion(rowdist_euc,order =order_row,method = c("Gradient_raw","Path_length"))
```

```
cat("The Criterion value from Travelling Salesman Problem is",TSP_criteria,"\n")

# Gradient Measure
order_row <-seriate(rowdist_euc, "GW")
GM_criteria <- criterion(rowdist_euc,order =order_row,method = c("Gradient_raw","Path_length"))
cat("The Criterion value from Gradient Measure is",GM_criteria,"\n")
p4 <- price %>%
  plot_ly(type = 'parcoords',
          dimensions = list(
            list(label = 'Food Costs', values = ~Food.Costs...),
            list(label = 'Hours iPhone 4S', values = ~iPhone.4S.hr.),
            list(label = 'Clothing Index', values = ~Clothing.Index),
            list(label = 'Total Hours', values = ~Hours.Worked),
            list(label = 'Net Wage', values = ~Wage.Net),
            list(label = 'Vacation Days', values = ~Vacation.Days),
            list(label = 'Min for Bread', values = ~Bread.kg.in.min.),
            list(label = 'Min for Rice', values = ~Rice.kg.in.min.),
            list(label = 'Min for Mac', values = ~Big.Mac.min.),
            list(label = 'Goods&Ser', values = ~Goods.and.Services...)
          )
  )

p4
p2 <- price %>%
  mutate(price = as.integer(Vacation.Days > 20)) %>%
  plot_ly(type = 'parcoords',
            dimensions = list(
              list(label = "Food.Costs...", values =~Food.Costs...),
              list(label = "Clothing.Index", values =~Clothing.Index),
              list(label = "iPhone.4S.hr.", values =~iPhone.4S.hr.),
              list(label = "Hours.Worked", values =~Hours.Worked),
              list(label = "Wage.Net", values =~Wage.Net),
              list(label = "Vacation.Days", values =~Vacation.Days),
              list(label = "Big.Mac.min.", values =~Big.Mac.min.),
              list(label = "Bread.kg.in.min.", values =~Bread.kg.in.min.),
              list(label = "Rice.kg.in.min.", values =~Rice.kg.in.min.),
              list(label = "Goods.and.Services...", values =~Goods.and.Services...)
          ),
            line = list(color = ~as.numeric(price))
  )
p2
Ps=list()
nPlot=72
as.data.frame(reordermatrix) %>%
  add_rownames( var = "group" ) %>%
  mutate_each(funs(rescale), -group) -> reorder_euc_radar
for (i in 1:nPlot){
  Ps[[i]] <- htmltools::tags$div(
    plot_ly(type = 'scatterpolar',
            r=as.numeric(reorder_euc_radar[i,-1]),
            theta= colnames(reorder_euc_radar)[-1],
            fill="toself")%>%
      layout(title=reorder_euc_radar$group[i]), style="width: 25%;")
```

```
}
h <-htmltools::tags$div(style = "display: flex; flex-wrap: wrap", Ps)
htmltools::browsable(h)
adults <- read.csv("adults.csv")
names(adults) <- c("Age", "work_Class", "population_Index", "education",
                  "education_num", "marital_status", "occupation", "relationship",
                  "race", "sex", "capital_gain", "capital_loss", "hours_per_week", "native_country", "In
p1 <- ggplot(adults, aes(x=Age, y=hours_per_week, colour= Income_Level)) + geom_point()
p1
p1 <- p1 + facet_grid(Income_Level~.)
p1
p2 <- ggplot(data=adults, aes(x=Age, group=Income_Level,fill=Income_Level))
p2 <- p2 + geom_density(alpha=0.5) + ylab("Density")
p2
p2 <- p2 + facet_wrap(marital_status~.)
p2
filtered_adults <- adults %>% filter(capital_loss > 0)
p3 <- plot_ly(filtered_adults, x = ~education_num, y = ~Age, z = ~capital_loss) %>%
  add_markers()
p3
p4 <-  filtered_adults %>% ggplot(aes(x = education_num, y = capital_loss)) +
    stat_density2d(aes(fill = ..density..),geom = "raster", contour = FALSE) +
    facet_wrap(cut_number(Age, 6)~.)
p4
p5 <- filtered_adults %>% ggplot(aes(x = education_num, y = capital_loss)) +
    geom_point() +
    facet_wrap(cut_number(Age, 4)~.)
p5
Agerange<-lattice::equal.count(filtered_adults$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)))
ggplot(L1)+geom_linerange(aes(ymin = Lower, ymax = Upper, x=Interval))

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

df4 <- filtered_adults[index,]
df4$Class <- as.factor(Class)

ggplot(df4, aes(x=education_num, y=capital_loss))+
  geom_point()+
  facet_wrap(~Class, labeller = "label_both")
```