`Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Інститут комп'ютерних наук та інформаційних технологій

Кафедра систем штучного інтелекту



**Лабораторна робота №4**

*З дисципліни:*

**“** **Видуботок великих даних”**

**Виконала:**

Студентка групи КН-318

Яворська М.О.

**Перевірила:**

професор Шаховська Н.Б.

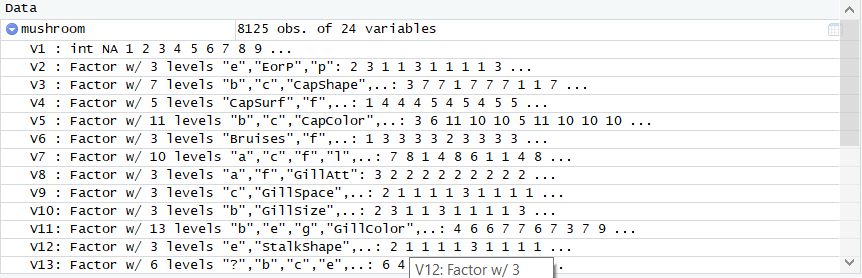
Львів-2019

**Хід роботи**

• Load the data found at the above URL.

setwd("D:/R")

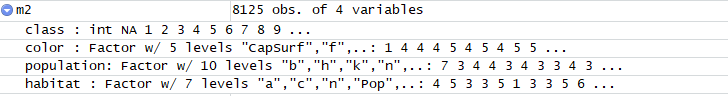
mushroom <- read.csv("mushroom.csv", header= FALSE, sep=",")



• Create a subset of data and rename the empty column with appropriate fieldnames.

m2 <- m1[,c("V1","V4", "V22", "V23")]

colnames(m2) <- c("class", "color", "population","habitat")



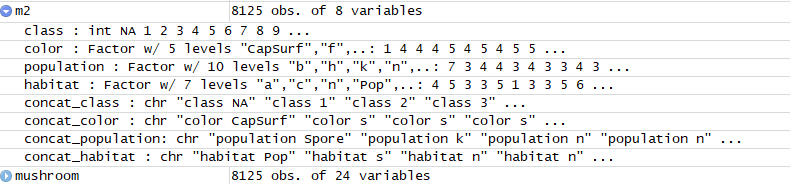
• There are a few directions we can take with the data at this point

m2$concat\_class <- paste("class",m2$class)

m2$concat\_color <- paste("color",m2$color)

m2$concat\_population <- paste("population",m2$population)

m2$concat\_habitat <- paste("habitat",m2$habitat)



• Once the field have been processed using concatentation, we can then merge these values, in a fashion similar to a sql join. Run head(m2) for further clarification of this step.Or use parameter residuals.

head (m2)

lu = setNames(lu,c('column','code','change', 'concat\_class'))

m3=merge(m2,lu,by="concat\_class")

lu = setNames(lu,c('column','code','change', 'concat\_color'))

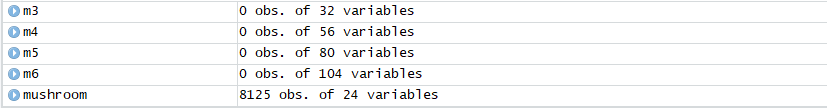
m4=merge(m3,lu,by="concat\_color")

lu = setNames(lu,c('column','code','change', 'concat\_population'))

m5=merge(m4,lu,by="concat\_population")

lu = setNames(lu,c('column','code','change', 'concat\_habitat'))

m6=merge(m5,lu,by="concat\_habitat")



• The best way for estimating is the distribution of data using bar graphs, but we have just eleven points, so we can use a point schedule.

m6=setNames(m6,c('n','n','n','n','n','n','n','n','n','n','class','n','n','color','n','n','population','n','n','habitat'))

m7<-subset(m6, select=c(class, color, population, habitat ))



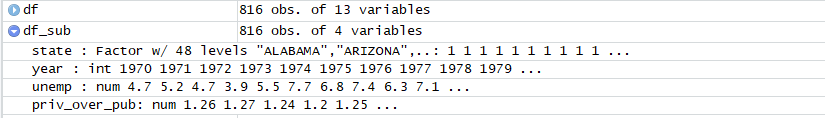
• Load the data and create a private versus public spending ratio for each year for each state.

df<read.csv("http://vincentarelbundock.github.io/Rdatasets/csv/plm/Produc.csv", header= TRUE, sep=",")

options(warn=-1)

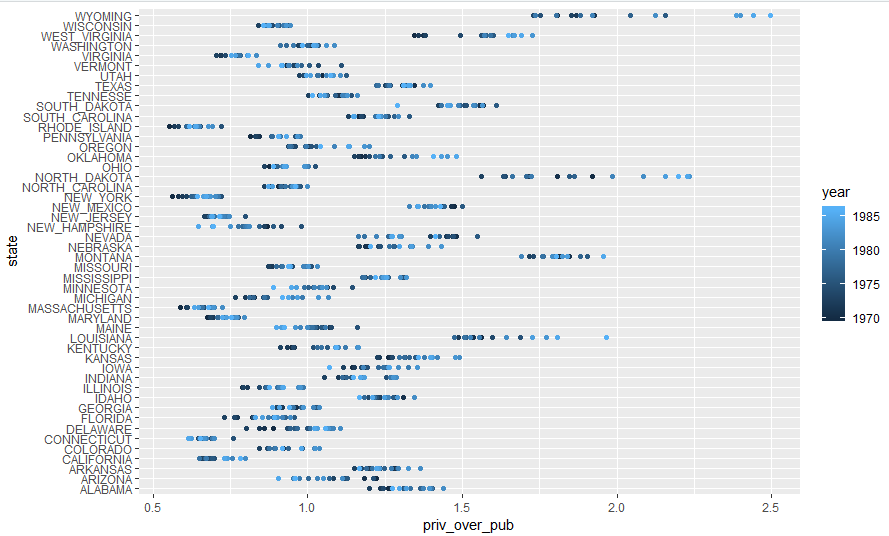
df$priv\_over\_pub <- with(df, pc / gsp)

df\_sub<-subset(df, select=c(state, year, unemp,priv\_over\_pub))



• A boxplot of the data per year reveals some further insight into the trends and behavior of spending per state over this time period.

ggplot(data=df\_sub, aes(x=priv\_over\_pub, y=state, col=year)) + geom\_point()



• In order to reduce the data set, decide to calculate the ranges between the lowest and highest ratio values per state for each year.

min\_vals <-aggregate(priv\_over\_pub ~ state, df\_sub, function(x) min(x))

max\_vals <-aggregate(priv\_over\_pub ~ state, df\_sub, function(x) max(x))

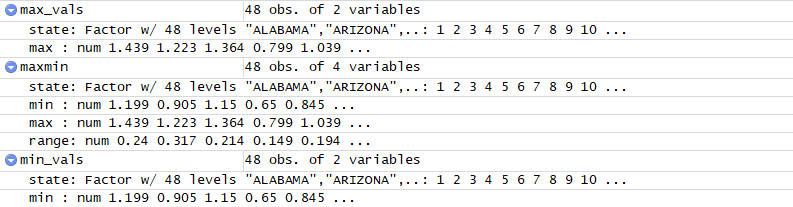
names(min\_vals)[1:2]<-c("state","min")

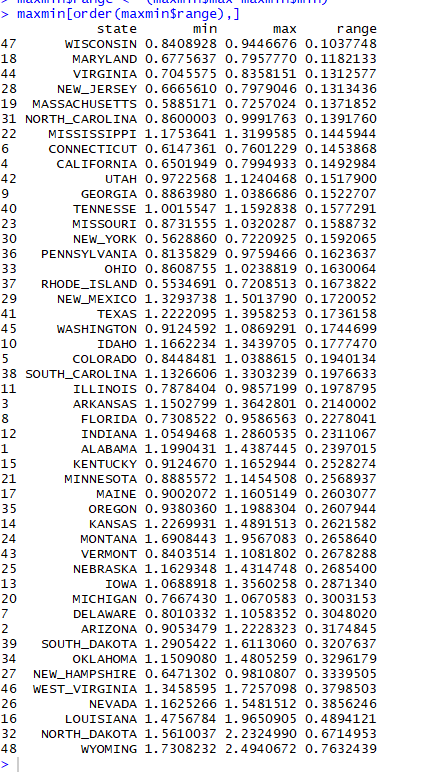
names(max\_vals)[1:2]<-c("state","max")

maxmin=merge(min\_vals,max\_vals,by='state')

maxmin$range <- (maxmin$max-maxmin$min)

maxmin[order(maxmin$range),]





• We can then subset the two dataframes into the 5 greatest, and 5 lowest ranges per state per ratio.

po=maxmin

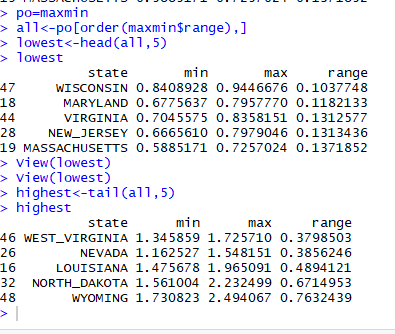
all<-po[order(maxmin$range),]

lowest<-head(all,5)

lowest

highest<-tail(all,5)

highest



• A plot of our lowest or most “consistent” ranges of subset [Wisconsin, Maryland, Virginia, New Jersey, Massachusetts] shows a range in unemployment.

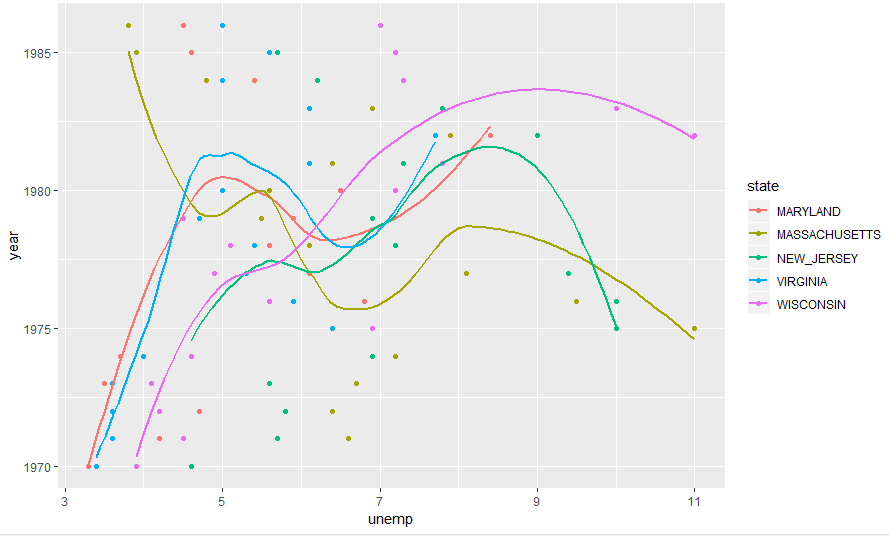
ggplot(low, aes(x=unemp, y=year, col=state)) + geom\_point()+ geom\_smooth( se=FALSE)

low=merge(lowest,df\_sub,by='state')

low

highe=merge(highest,df,by='state')

highe



• A plot of our lowest or most “inconsistent” ranges of subset [West Virginia, Nevada, Louisiana, North Dakota, Wyoming] shows a range in unemployment and trend that differs from the earlier graph.

ggplot(highe, aes(x=unemp, y=year, col=state)) + geom\_point()+ geom\_smooth(se=FALSE)

