install.packages("corrplot")

library(corrplot)

install.packages("dplyr")

library(dplyr)

install.packages("plm")

library(plm)

install.packages("ggplot2")

library(ggplot2)

install.packages("AER")

library(AER)

install.packages("factoextra")

library(factoextra)

install.packages("GGally")

library(GGally)

install.packages("FactoMineR")

library(FactoMineR)

install.packages("lmtest")

library(lmtest)

install.packages("car")

library(car)

install.packages("stargazer")

library(stargazer)

#####лес

install.packages("randomForest")

library(randomForest)

install.packages("rpart")

library(rpart)

install.packages("rpart.plot")

library(rpart.plot)

#логистическая регрессия

install.packages("AUC")

library(AUC)

install.packages("caret")

library(caret)

install.packages("klaR")

library(klaR)

install.packages("InformationValue")

library(InformationValue)

install.packages("erer")

library(erer)

#самоорганизующиеся карты

install.packages("kohonen")

library(kohonen)

install.packages("RColorBrewer")

library(RColorBrewer)

install.packages("viridis")

library(viridis)

install.packages("mlbench")

library(mlbench)

library(readxl)

data <- read\_excel("data.xlsx", col\_types = c("text",

"text", "numeric", "numeric",

"numeric", "numeric", "numeric", "numeric",

"numeric", "numeric", "numeric", "numeric","numeric"))

View(data)

dim(data)

data <- na.omit(data)

summary(data)

M <- as.data.frame(data)

stargazer(M, type="text", out="M.html")

##построим графики "ящик усов"

ggplot(data, aes(x=year, y = EVA, fill=year)) + geom\_boxplot() + xlab("") + ylab("EVA") + labs(fill = "Год")

ggplot(data, aes(x=year, y = ROA, fill=year)) + geom\_boxplot() + xlab("") + ylab("ROA") + labs(fill = "Год")

ggplot(data, aes(x=year, y = NPM, fill=year)) + geom\_boxplot() + xlab("") + ylab("NPM") + labs(fill = "Год")

ggplot(data, aes(x=year, y = size, fill=year)) + geom\_boxplot() + xlab("") + ylab("Размер компании") + labs(fill = "Год")

ggplot(data, aes(x=year, y = log(workers), fill=year)) + geom\_boxplot() + xlab("") + ylab("Количество работников") + labs(fill = "Год")

ggplot(data, aes(x=year, y = tangofassets, fill=year)) + geom\_boxplot() + xlab("") + ylab("Оборачиваемость активов") + labs(fill = "Год")

ggplot(data, aes(x=year, y = cashratio, fill=year)) + geom\_boxplot() + xlab("") + ylab("Коэффициент текущей ликвидности") + labs(fill = "Год")

ggplot(data, aes(x=year, y = debtratio, fill=year)) + geom\_boxplot() + xlab("") + ylab("Коэффициент финансовой зависимости") + labs(fill = "Год")

ggplot(data, aes(x=year, y = assetgrowth, fill=year)) + geom\_boxplot() + xlab("") + ylab("Прирост активов") + labs(fill = "Год")

ggplot(data, aes(x=year, y = leverage, fill=year)) + geom\_boxplot() + xlab("") + ylab("Финансовый рычаг") + labs(fill = "Год")

ggplot(data, aes(x=year, y = quickratio, fill=year)) + geom\_boxplot() + xlab("") + ylab("Коэффициент быстрой ликвидности") + labs(fill = "Год")

names(data)

##########корреляционная матрица#########

data\_cor <- dplyr::select(data,-company,-year)

corrplot(cor(data\_cor))

#построим матрицу со значениями

corrplot(cor(data\_cor), addCoef.col = TRUE, addgrid.col = TRUE)

#########################метод главных компонент##########################################

data2021 <- filter(data, year == 2021)

data\_2021 <- dplyr::select(data2021,-company,-year,)

rownames(data\_2021) <- data2021$company

View(data\_2021)

#нормируем данные

Data <- scale(data\_2021)

#применим метод главных компонент

mod2021 <- PCA(Data)

#получим собственные числа и собственные векторы при помощи спектрального разложения

mod\_eigen <- eigen(var(Data))

#вклады каждой переменной в главную компоненту

mod2021$var$contrib

#визуализаруем долю объясненной дисперсии через график каменистой осыпи

fviz\_eig(mod2021, addlabels = TRUE)

fviz\_pca\_ind(mod2021, pointsize = "cos2", pointshape = 21, fill = "blue", repel = TRUE)

mod\_clust <- HCPC(mod2021)

summary(mod\_clust)

str(mod\_clust$data.clust)

fviz\_pca\_ind(mod2021, pointshape = 21, pointsize = 2, fill.ind = mod\_clust$data.clust$clust,

repel = TRUE, addEllipses = TRUE)

#####качество репрезентации

corrplot(mod2021$var$cos2, is.corr = FALSE)

#вклад в 1

2 компоненту

fviz\_contrib(mod2021, choice = "var", axes = 2, top = 10)

#матрица нагрузок

corrplot(mod2021$var$cor)

mod2021$var$cor

mod\_pa <- HCPC(mod2021, graph = FALSE)

#номера кластеров

mod\_pa$data.clust

#интерпретация через средние координаты

mod\_pa$desc.var

#интерпретация через главные компоненты

mod\_pa$desc.axes

#типичные представители кластера

mod\_pa$desc.ind$para

#Визуализация дендрограммы:

fviz\_dend(mod\_pa, cex = 0.7, palette = "jco", rect = TRUE, rect\_fill = TRUE, rect\_border = "jco", labels\_track\_height = 0.8)

######просто модели#######

##############ROA###################################

modA1 <- lm(ROA~.-NPM-EVA-company-year-currentratio, data=data)

modA2 <- lm(ROA~.-NPM-EVA-company-year-quickratio, data=data)

AIC(modA1,modA2)

####лучше с quickratio

modA <- lm(ROA~.-NPM-EVA-company-year-currentratio-workers+log(workers), data=data)

summary(modA)

#Регрессиия без эффектов:

m.pooledA <- plm(data = data,ROA~size+tangofassets+quickratio+assetgrowth+debtratio+leverage+log(workers), model = "pooling",index = c("company", "year"))

#Регрессия со случайными эффектами:

m.reA <- plm(ROA~size+tangofassets+quickratio+assetgrowth+debtratio+leverage+log(workers) , data = data, model = "random", index = c("company", "year"))

summary(m.reA)

#Регрессия с фиксированными эффектами:

m.feA <- plm(ROA~size+tangofassets+quickratio+assetgrowth+debtratio+leverage+log(workers), data = data, model = "within", index = c("company", "year"))

summary(m.feA)

# FE против обычной.

pFtest(m.feA, m.pooledA)

#Нулевая гипотеза: pooled. Она отвергается(p-value =0)

#тест Хаусмана (RЕ против FЕ).

phtest(m.reA, m.feA)

#Нулевая гипотеза: RE. Она отвергается(p-value = 0)

#тест множителей Лагранжа (RЕ против обычной)

plmtest(m.reA, type = "bp")

#Нулевая гипотеза: pooled. Она отвергается(p-value = 0)

#Вывод:FE подойдет

data\_star <- as.data.frame(data)

stargazer(list(m.pooledA, m.reA, m.feA), column.labels = c("Pooling", "RE", "FE"), omit= c("leverage","size","workers"),type = "text",out = "dataA.html")

datayear <- mutate(data, year=as.factor(year), company=as.factor(company))

mROA <- lm(data=datayear, ROA~company+year+size+leverage+tangofassets+quickratio+assetgrowth+debtratio+log(workers))

summary(mROA)

resettest(mROA)

View(data)

View(d)

d <- dplyr::select(data, leverage, debtratio)

#H0: все хорошо, ничего добавлять не надо

crPlots(mROA)

plot(mROA, which=2)

# ошибки нормально распределены

plot(mROA, which=3)

#гетероскедастичности нет, если хаос

bptest(mROA)

vif(mROA)

##если коэффициенты не очень большие, то хорошо

#робастные ошибки

cse <- function(model) {

A <- sqrt(diag(vcovHC(model, type = "HC0")))

return(A)

}

cse(mROA)

V\_newA <- vcovHC(mROA, type="HC0")

coeftest(mROA, V\_newA)

stargazer(mROA, type = "text",df=FALSE, se = list(cse(mROA)), keep = c("assetgrowth","debtratio", "leverage"), out="ROA.html")

#####лаги

names(data)

datalag <- data %>% group\_by(company) %>% transmute(sizelag =dplyr:: lag(size),leveragelag =dplyr:: lag(leverage),tangofassetslag=dplyr:: lag(tangofassets),currentratiolag=dplyr:: lag(currentratio), quickratiolag=dplyr:: lag(quickratio), debtratiolag=dplyr:: lag(debtratio), assetgrowthlag=dplyr:: lag(assetgrowth), workerslag=dplyr:: lag(workers),ROA=ROA,EVA=EVA, year=year)

View(datalag)

mROAlag <- lm(data=datalag, ROA~company+year+tangofassetslag+quickratiolag+assetgrowthlag+debtratiolag+log(workerslag))

summary(mROAlag)

cse(mROAlag)

stargazer(mROAlag, type = "text",df=FALSE, se = list(cse(mROAlag)), keep = c("tangofassets","quickratio", "leverage"), out="ROAlag.html")

#построим дерево

modtreeA <- rpart(ROA~.-NPM-EVA-company-year-cashratio-workers+log(workers), data = data2021, control = rpart.control(minsplit=3,minbucket=3,maxdebth=10,cp=0.01))

rpart.plot(modtreeA)

#построим случайный лес

mod\_treeA <- randomForest(ROA~.-NPM-EVA-company-year-currentratio-workers+log(workers)-leverage, data = data2021)

#влияние переменных

varImpPlot(mod\_treeA)

######################NPM#########################

modNPM1 <- lm(NPM~.-ROA-EVA-company-year-currentratio, data=data)

modNPM2 <- lm(NPM~.-ROA-EVA-company-year-quickratio, data=data)

AIC(modNPM1,modNPM2)

####лучше с currentratio

modNPM <- lm(NPM~.-ROA-EVA-company-year-quickratio-workers+log(workers),

data=data)

summary(modNPM)

#Регрессиия без эффектов:

m.pooledNPM <- plm(data = data,NPM~size+leverage+tangofassets+currentratio+assetgrowth+debtratio, model = "pooling",index = c("company", "year"))

#Регрессия со случайными эффектами:

m.reNPM <- plm(NPM~size+leverage+tangofassets+currentratio+assetgrowth+debtratio , data = data, model = "random", index = c("company", "year"))

summary(m.reNPM)

#Регрессия с фиксированными эффектами:

m.feNPM <- plm(NPM~size+leverage+tangofassets+currentratio+assetgrowth+debtratio, data = data, model = "within", index = c("company", "year"))

summary(m.feNPM)

# FE против обычной.

pFtest(m.feNPM, m.pooledNPM)

#Нулевая гипотеза: pooled. Она отвергается(p-value =0)

#тест Хаусмана (RЕ против FЕ).

phtest(m.reNPM, m.feNPM)

#Нулевая гипотеза: RE. Она отвергается(p-value = 0)

#тест множителей Лагранжа (RЕ против обычной)

plmtest(m.reNPM, type = "bp")

#Нулевая гипотеза: pooled. Она принимается(p-value = 0.02)

#Вывод:FE подойдет

stargazer(list(m.pooledNPM, m.reNPM, m.feNPM), column.labels = c("Pooling", "RE", "FE"), type = "text",out = "dataNPM1.html")

mNPM <- lm(data=datayear, NPM~company+year+size+leverage+tangofassets+currentratio+assetgrowth+debtratio+log(workers))

summary(mNPM)

resettest(mNPM)

#H0: все хорошо, ничего добавлять не надо

crPlots(mNPM)

plot(mNPM, which=2)

# ошибки нормально распределены

plot(mNPM, which=3)

#гетероскедастичности нет, если хаос

bptest(mNPM)

vif(mNPM)

##если коэффициенты не очень большие, то хорошо

#робастные ошибки

cse(mNPM)

V\_newNPM <- vcovHC(mNPM, type="HC0")

coeftest(mNPM, V\_newNPM)

stargazer(mNPM, type = "text",df=FALSE, se = list(cse(mNPM)), keep = c("assetgrowth","debtratio", "leverage"), out="NPM.html")

#############разные периоды

datan1 <- filter(data, year>2019)

View(datan1)

datayear1 <- mutate(datan1, year=as.factor(year), company=as.factor(company))

mNPM1 <- lm(data=datayear1, NPM~company+year+size+leverage+tangofassets+currentratio+assetgrowth+debtratio+log(workers))

datan2 <- filter(data, year<2020)

View(datan2)

datayear2 <- mutate(datan2, year=as.factor(year), company=as.factor(company))

mNPM2 <- lm(data=datayear2, NPM~company+year+size+leverage+tangofassets+currentratio+assetgrowth+debtratio+log(workers))

summary(mNPM1)

stargazer(list(mNPM2, mNPM1), column.labels = c("Before COVID-19", "After COVID-19"), keep = c("assetgrowth","debtratio", "leverage"), type = "text",out = "dataNPM2.html")

#построим дерево

modtreeNPM <- rpart(NPM~.-ROA-EVA-company-year-quickratio-workers+log(workers), data = data2021,control = rpart.control(minsplit=3,minbucket=3,maxdebth=10,cp=0.01))

rpart.plot(modtreeNPM)

#построим лес

mod\_treeNPM <- randomForest(NPM~.-ROA-EVA-company-year-quickratio-workers+log(workers), data = data2021)

#влияние переменных

varImpPlot(mod\_treeNPM)

#####################EVA###########################

modEVA1 <- lm(EVA~.-ROA-NPM-company-year-currentratio, data=data)

modEVA2 <- lm(EVA~.-ROA-NPM-company-year-quickratio, data=data)

AIC(modEVA1,modEVA2)

####лучше с cashratio

modEVA <- lm(EVA~.-ROA-NPM-company-year-quickratio-workers+log(workers), data=data)

summary(modEVA)

#Регрессиия без эффектов:

m.pooledEVA <- plm(data = data,EVA~size+leverage+tangofassets+currentratio+assetgrowth+debtratio+log(workers), model = "pooling",index = c("company", "year"))

#Регрессия со случайными эффектами:

m.reEVA <- plm(EVA~size+leverage+tangofassets+currentratio+assetgrowth+debtratio+log(workers) , data = data, model = "random", index = c("company", "year"))

summary(m.reEVA)

#Регрессия с фиксированными эффектами:

m.feEVA <- plm(EVA~size+leverage+tangofassets+currentratio+assetgrowth+debtratio+log(workers), data = data, model = "within", index = c("company", "year"))

summary(m.feEVA)

# FE против обычной.

pFtest(m.feEVA, m.pooledEVA)

#Нулевая гипотеза: pooled. Она отвергается(p-value =0)

#тест Хаусмана (RЕ против FЕ).

phtest(m.reEVA, m.feEVA)

#Нулевая гипотеза: RE. Она отвергается(p-value = 0)

#тест множителей Лагранжа (RЕ против обычной)

plmtest(m.reEVA, type = "bp")

#Нулевая гипотеза: pooled. Она отвергается(p-value = 0)

#Вывод:FE подойдет

stargazer(list(m.pooledEVA, m.reEVA, m.feEVA), column.labels = c("Pooling", "RE", "FE"), type = "text",out = "dataEVA.html")

mEVA <-

lm(data=datayear, EVA~company+year+leverage+tangofassets+quickratio+assetgrowth+debtratio+workers)

summary(mEVA)

resettest(mEVA)

#H0: все хорошо, ничего добавлять не надо

crPlots(mEVA)

plot(mEVA, which=2)

# ошибки нормально распределены

plot(mEVA, which=3)

#гетероскедастичности нет, если хаос

bptest(mEVA)

vif(mEVA)

##если коэффициенты не очень большие, то хорошо

#робастные ошибки

cse(mEVA)

V\_newEVA <- vcovHC(mEVA, type="HC0")

coeftest(mEVA, V\_newEVA)

stargazer(mEVA, type = "text",df=FALSE, se = list(cse(mEVA)), keep = c("assetgrowth","debtratio", "workers","size","tangofassets","leverage"), out="EVA.html")

mEVAlag <- lm(data=datalag, EVA~company+year+assetgrowthlag+debtratiolag+log(workerslag))

summary(mEVAlag)

cse(mEVAlag)

V\_newEVAl <- vcovHC(mEVAlag, type="HC0")

coeftest(mEVAlag, V\_newEVAl)

stargazer(mEVAlag, type = "text",df=FALSE, se = list(cse(mEVAlag)), keep = c("workers","debtratio"), out="EVAlag.html")

#построим дерево

modtreeEVA <- rpart(EVA~.-NPM-ROA-company-year-quickratio-workers+log(workers), data = data2021, control = rpart.control(minsplit=3,minbucket=3,maxdebth=10,cp=0.01))

rpart.plot(modtreeEVA)

#построим лес

mod\_treeEVA <- randomForest(EVA~.-NPM-ROA-company-year-quickratio-workers+log(workers), data = data2021)

#влияние переменных

varImpPlot(mod\_treeEVA)

#логистическая регрессия

summary(data)

datalog <- mutate(data, A = ifelse(EVA > 4.4288, 1, 0))

maBina(mod\_logit)

mod\_logit <- glm(A ~ .-ROA-NPM-EVA-company-year-workers+log(workers), datalog, x=TRUE, family = "binomial")

summary(mod\_logit)

stargazer(mod\_logit, type="text",df=FALSE, se = list(mod\_logit), keep = c("size","debtratio","leverage"), out="logit.html")

#качество прогноза

mod <- stepAIC(mod\_logit)

summary(mod)

Pr <- predict(mod\_logit)

#мера качества модели - площадь под ROC кривой

Y <- ifelse(data$EVA < 4.4288, 0, 1)

Y

plotROC(actuals = Y, predictedScores = Pr)

#минимальное значение - это 0.5

#максимальное значение - 1

##модель с инструментом для EVA

#перейдем к лагам

data <- pdata.frame(data, index = c("company","year"))

data$debtratio\_l <- lag(data$debtratio, 1)

data3 <- na.omit(data)

#сделаем инструмент

mod <- lm(data3$debtratio ~ data3$debtratio\_l)

Inst <- fitted(mod)

#модель с лагом

names(data)

mod4 <- plm(EVA ~ size + Inst + tangofassets, data3, model = "within", effect = "twoways")

summary(mod4)

#выгрузим

cse <- function(model) {

A <- sqrt(diag(vcovHC(model)))

return(A)

}

cse(mod4)

stargazer(mod4, type = "html", column.labels =

c("Instrument"), df = FALSE,

se = list(cse(mod4)), out = "Instrument.html")

#########самоорганизующиеся карты#####################################

#построим пустую карту

Datah <- dplyr::select(data2021, - company, - year)

View(data2021)

#стандартизируем данные

Data\_scale <- as.matrix(scale(Datah))

View(Data\_scale)

#сделаем разметку

set.seed(259)

som\_grid <- somgrid(xdim = 4, ydim = 4, topo = "hexagonal")

#обучим модель

som\_model <- som(Data\_scale, grid = som\_grid, rlen = 100, keep.data = TRUE)

#rlen - сколько раз прогнояем всю выборку для обучения модели

#keep.data = TRUE - сохраняет информацию о наборе данных

#сколько наблюдений попало в каждый кластер?

plot(som\_model, type = "counts")

plot(som\_model, type = "mapping")

#достаточно, не надо курпнять кластеры

#как выглядят кластеры?

plot(som\_model, type = "code")

#достанем информацию о каждом кластере

Som\_codes <- getCodes(som\_model)

#зададим цветовую палитру

palette\_new <- function(n) {

colorRampPalette(brewer.pal(8, "RdYlBu"))(n)

}

names(Data\_scale)

par(mfrow = c(1,2))

plot(som\_model, type = "property", property = Som\_codes[,7], main = colnames(Som\_codes)[7] ,palette = palette\_new)

plot(som\_model, type = "property", property = Som\_codes[,8], main = colnames(Som\_codes)[8] ,palette = palette\_new)

plot(som\_model, type = "property", property = Som\_codes[,9], main = colnames(Som\_codes)[9] ,palette = palette\_new)

plot(som\_model, type = "property", property = Som\_codes[,11], main = colnames(Som\_codes)[11] ,palette = palette\_new)

plot(som\_model, type = "property", property = Som\_codes[,10], main = colnames(Som\_codes)[10] ,palette = palette\_new)

plot(som\_model, type = "property", property = Som\_codes[,5], main = colnames(Som\_codes)[5] ,palette = palette\_new)

plot(som\_model, type = "property", property = Som\_codes[,4], main = colnames(Som\_codes)[4] ,palette = palette\_new)

plot(som\_model, type = "property", property = Som\_codes[,6], main = colnames(Som\_codes)[6] ,palette = palette\_new)

#добавим иерархическую кластеризацию

mydata <- as.matrix(Som\_codes)

#используем иерархическую кластеризацию

som\_cluster <- cutree(hclust(dist(mydata)), 3)

# Определяем палитру цветов

palette\_new\_2 <- function(n) {

viridis(n)

}

par(mfrow = c(1,1))

# Показываем разными цветами кластеры узлов и переменные

plot(som\_model, type = "codes",

bgcol = palette\_new\_2(3)[som\_cluster], main = NA)

add.cluster.boundaries(som\_model, som\_cluster)