library(car)

library(lmtest)

library(ggplot2)

library(gridExtra)

library(quantmod)

library(dplyr)

library(xts)

library(tidyr)

library(reshape)

library(readxl)

#загрузим данные

full\_data <- read\_excel("Уни/8 семестр/Диплом/full\_data.xlsx")

View(full\_data)

#отберем переменные, необходимые для постоения модели

data <- full\_data[,c(1,2,3,4,5,6,7,8,10,13,14,15,16,34,47,58,66)]

View(data)

summary(data)

names(data)

data$Industry <- as.factor(data$Industry)

data$`Company Name` <- as.factor(data$`Company Name`)

data$`Identifier (RIC)` <- as.factor(data$`Identifier (RIC)`)

data$Date <- as.factor(data$Date)

#скачаем цены акций

?getSymbols

sym <- data$`Identifier (RIC)`

s <- unique(sym)

#построим цикл

D <- new.env()

sDate <- as.Date("2012-10-30")

eDate <- as.Date("2021-10-30")

ticker <- s[-c(61,71,90)]

num\_ticker <- length(ticker)

Temp\_D <- list()

counter <- 1L

for(i in ticker) {

getSymbols(

i,

env = D,

reload.Symbols = FALSE,

from = sDate,

to = eDate,

verbose = FALSE,

warnings = TRUE,

src = "yahoo",

symbol.lookup = TRUE)

print(counter)

Temp\_D[[i]] <- Cl(D[[i]])

if (counter == length(ticker))

{

#Merge all the objects of the list into one object.

Daily\_Price <- do.call(merge, Temp\_D)

Monthly\_Price <- Daily\_Price[endpoints(Daily\_Price,'months')]

}

else

{

counter <- counter + 1

}

}

View(Monthly\_Price)

#обработам полученные данные для того, чтобы было удобно с ними дальше работать

Price <- Monthly\_Price[c(1,13,25,37,49,61,73,85,97,109),]

View(Price)

Price <- as.data.frame(Price)

library("writexl")

write\_xlsx(Price, "Price.xlsx")

library(readxl)

P <- read\_excel("C:/Users/valer/OneDrive/Desktop/Price.xlsx")

View(P)

Pr <- gather(P, key = "Ticker")

Pr <- Pr[-c(1:10),]

View(Pr)

#Соединим данные по котировкам с финансовыми данными

View(data)

data <- data[-c(601:610,701:710,891:900),]

data$Price <- Pr$value

#выведем описательные статистики

#data <- na.omit(data)

summary(data)

#почистим сразу от выбросов

low\_b\_ESG <- quantile(data$ESG\_score,0.005)

low\_b\_ESG

up\_b\_ESG <- quantile(data$ESG\_score,0.995)

up\_b\_ESG

ind\_ESG <- which(data$ESG\_score < low\_b\_ESG | data$ESG\_score > up\_b\_ESG)

ind\_ESG

low\_b\_S <- quantile(data$Social\_score,0.005)

low\_b\_S

up\_b\_S <- quantile(data$Social\_score,0.995)

up\_b\_S

ind\_S <- which(data$Social\_score < low\_b\_S | data$Social\_score > up\_b\_S)

ind\_S

low\_b\_G <- quantile(data$Gov\_score,0.005)

low\_b\_G

up\_b\_G <- quantile(data$Gov\_score,0.995)

up\_b\_G

ind\_G <- which(data$Gov\_score < low\_b\_G | data$Gov\_score > up\_b\_G)

ind\_G

low\_b\_E <- quantile(data$Env\_score,0.005)

low\_b\_E

up\_b\_E <- quantile(data$Env\_score,0.995)

up\_b\_E

ind\_E <- which(data$Env\_score < low\_b\_E | data$Env\_score > up\_b\_E)

ind\_E

low\_b\_BVPS <- quantile(data$BVPS,0.005)

low\_b\_BVPS

up\_b\_BVPS <- quantile(data$BVPS,0.995)

up\_b\_BVPS

ind\_BVPS <- which(data$BVPS < low\_b\_BVPS | data$BVPS > up\_b\_BVPS)

ind\_BVPS

low\_b\_TA <- quantile(data$Total\_assets,0.005)

low\_b\_TA

up\_b\_TA <- quantile(data$Total\_assets,0.995)

up\_b\_TA

ind\_TA <- which(data$Total\_assets <= low\_b\_TA | data$Total\_assets >= up\_b\_TA)

ind\_TA

low\_b\_NI <- quantile(data$Net\_income,0.005)

low\_b\_NI

up\_b\_NI <- quantile(data$Net\_income,0.995)

up\_b\_NI

ind\_NI <- which(data$Net\_income < low\_b\_NI | data$Net\_income > up\_b\_NI)

ind\_NI

low\_b\_Sh <- quantile(data$Shares,0.005)

low\_b\_Sh

up\_b\_Sh <- quantile(data$Shares,0.995)

up\_b\_Sh

ind\_Sh <- which(data$Shares <= low\_b\_Sh | data$Shares >= up\_b\_Sh)

ind\_Sh

low\_b\_FL <- quantile(data$FNCL\_LVRG,0.005, na.rm = TRUE)

low\_b\_FL

up\_b\_FL <- quantile(data$FNCL\_LVRG,0.995, na.rm = TRUE)

up\_b\_FL

ind\_FL <- which(data$FNCL\_LVRG < low\_b\_FL | data$FNCL\_LVRG > up\_b\_FL)

ind\_FL

low\_b\_P <- quantile(data$Price,0.005, na.rm = TRUE)

low\_b\_P

up\_b\_P <- quantile(data$Price,0.995, na.rm = TRUE)

up\_b\_P

ind\_P <- which(data$Price < low\_b\_P | data$Price > up\_b\_P)

ind\_P

low\_roe <- quantile(data$ROE,0.005, na.rm = TRUE)

low\_roe

up\_roe <- quantile(data$ROE,0.995, na.rm = TRUE)

up\_roe

ind\_roe <- which(data$ROE < low\_roe | data$ROE > up\_roe)

ind\_roe

low\_roa <- quantile(data$RETURN\_ON\_ASSET,0.005, na.rm = TRUE)

low\_roa

up\_roa <- quantile(data$RETURN\_ON\_ASSET,0.995, na.rm = TRUE)

up\_roa

ind\_roa <- which(data$RETURN\_ON\_ASSET < low\_roa | data$RETURN\_ON\_ASSET > up\_roa)

ind\_roa

low\_ag <- quantile(data$ASSET\_GROWTH,0.005, na.rm = TRUE)

low\_ag

up\_ag <- quantile(data$ASSET\_GROWTH,0.995, na.rm = TRUE)

up\_ag

ind\_ag <- which(data$ASSET\_GROWTH < low\_ag | data$RETURN\_ON\_ASSET > up\_ag)

ind\_ag

low\_qr <- quantile(data$QUICK\_RATIO,0.005, na.rm = TRUE)

low\_qr

up\_qr <- quantile(data$QUICK\_RATIO,0.995, na.rm = TRUE)

up\_qr

ind\_qr <- which(data$QUICK\_RATIO < low\_qr | data$QUICK\_RATIO > up\_qr)

ind\_qr

outlier <- c(ind\_BVPS,ind\_E,ind\_ESG,ind\_FL,ind\_G,ind\_NI,ind\_P,ind\_S,ind\_Sh,ind\_TA, ind\_roe, ind\_roa, ind\_qr, ind\_ag)

outlier2 <- unique(outlier)

data\_out <- data[-c(outlier2),]

data\_out$NIPS <- data\_out$Net\_income/data\_out$Shares

data\_out <- data\_out[,-c(10,11)]

#outlier <- c(ind\_BVPS,ind\_E,ind\_ESG,ind\_FL,ind\_G,ind\_NI,ind\_P,ind\_S,ind\_Sh,ind\_TA)

#outlier1 <- unique(outlier)

#data\_out <- data[-c(outlier1),]

View(data\_out)

summary(data\_out)

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

unique(data\_out$`Identifier (RIC)`)

summary(data\_out$Date)

min(summary(data\_out$Industry))

stargazer(data\_out, type = "text", column.labels=names(data\_out[,-c(1,2,3,10)]),

df=FALSE, digits=3, out = "summary1.doc")

####Графики####

View(data\_out)

ggplot(data\_out, aes(x = Industry)) +geom\_bar()

install.packages("RColorBrewer")

library("RColorBrewer")

display.brewer.all(colorblindFriendly = TRUE)

ggplot(data\_out, aes(x=factor(Date), y=ESG\_score, fill=Date)) +

geom\_boxplot(outlier.size=1.5, outlier.shape=21)+ stat\_summary(fun.y="mean", geom="point", shape=23, size=3, fill="black")+

xlab("")+ylab("ESG\_score")+labs(fill="Год")+scale\_fill\_brewer(palette="PiYG")

ggplot(data\_out, aes(x=factor(Date), y=Gov\_score, fill=Date)) +

geom\_boxplot(outlier.size=1.5, outlier.shape=21)+ stat\_summary(fun.y="mean", geom="point", shape=23, size=3, fill="black")+

xlab("")+ylab("Gov\_score")+scale\_fill\_brewer(palette="PiYG")

ggplot(data\_out, aes(x=factor(Date), y=Social\_score, fill=Date)) +

geom\_boxplot(outlier.size=1.5, outlier.shape=21)+ stat\_summary(fun.y="mean", geom="point", shape=23, size=3, fill="black")+

xlab("")+ylab("Social\_score")+scale\_fill\_brewer(palette="PiYG")

ggplot(data\_out, aes(x=factor(Date), y=Env\_score, fill=Date)) +

geom\_boxplot(outlier.size=1.5, outlier.shape=21)+ stat\_summary(fun.y="mean", geom="point", shape=23, size=3, fill="black")+

xlab("")+ylab("Env\_score")+scale\_fill\_brewer(palette="PiYG")

#скрипичные графики

install.packages("gcookbook")

library(gcookbook)

p <- ggplot(data\_out, aes(x=factor(Date), y=ESG\_score))

p + geom\_violin()

install.packages("vioplot")

library(vioplot)

col <- c("Grey","Purple","Lightblue", "Lightgreen")

n <- c("ESG\_score","Social\_score", "Gov\_score","Env\_score")

vioplot(data\_out$ESG\_score, data\_out$Social\_score, data\_out$Gov\_score, data\_out$Env\_score,

names = c("ESG\_score","Social\_score", "Gov\_score","Env\_score"),col = col

)

#гистограммы

ggplot(data\_out, aes(x=BVPS)) + geom\_histogram(binwidth=4,fill="darkgreen", col="black" )+xlab("Балансовая стоимость на акицию")+ylab("Количество наблюдений")

ggplot(data\_out, aes(x=Total\_assets)) + geom\_histogram(binwidth = 20000000000, fill="darkgreen", col="black" )+xlab("Активы")+ylab("Количество наблюдений")

ggplot(data\_out, aes(x=ROE)) + geom\_histogram(binwidth=0.1,fill="darkgreen", col="black" )+xlab("Рентабельность собственного капитала")+ylab("Количество наблюдений")

ggplot(data\_out, aes(x=RETURN\_ON\_ASSET)) + geom\_histogram(binwidth=1,fill="darkgreen", col="black" )+xlab("Рентабельность активов")+ylab("Количество наблюдений")

ggplot(data\_out, aes(x=QUICK\_RATIO)) + geom\_histogram(binwidth=0.2,fill="darkgreen", col="black" )+xlab("Коэффициент быстрой ликвидности")+ylab("Количество наблюдений")

ggplot(data\_out, aes(x=ASSET\_GROWTH)) + geom\_histogram(binwidth=12,fill="darkgreen", col="black" )+xlab("Темп роста активов")+ylab("Количество наблюдений")

ggplot(data\_out, aes(x=FNCL\_LVRG)) + geom\_histogram(binwidth=2,fill="darkgreen", col="black" )+xlab("Финансовый рычаг")+ylab("Количество наблюдений")

ggplot(data\_out, aes(x=NIPS)) + geom\_histogram(binwidth=2,fill="darkgreen", col="black" )+xlab("Чистая прибыль на акицию")+ylab("Количество наблюдений")

ggplot(data\_out, aes(x=Price)) + geom\_histogram(binwidth=20,fill="darkgreen", col="black" )+xlab("Цена акции")+ylab("Количество наблюдений")

#sp <- ggplot(data\_out, aes(x=ESG\_score, y=ROE))

#sp + geom\_point() + stat\_smooth(method=lm)

#sp2 <- ggplot(data\_out, aes(x=ESG\_score, y=RETURN\_ON\_ASSET))

#sp2 + geom\_point() + stat\_smooth(method=lm)

#sp3 <- ggplot(data\_out, aes(x=ESG\_score, y=Price))

#sp3 + geom\_point() + stat\_smooth(method=lm, se=TRUE)

summary(data\_out$Industry)

View(a)

a <- aggregate(data\_out$ESG\_score, by=list(data\_out$Industry), FUN=mean)

ggplot(a, aes(x=x, y=reorder(Group.1,x))) +

geom\_point(size=1.5) +

theme\_bw() + xlab("ESG\_score")+ylab("Сектор")+

theme(panel.grid.major.x = element\_blank(),

panel.grid.minor.x = element\_blank(),

panel.grid.major.y = element\_line(colour="grey60", linetype="dashed"))

summary(data\_out$Date)

View(a)

a <- aggregate(data\_out$ESG\_score, by=list(data\_out$Date), FUN=mean)

ggplot(a, aes(x=x, y=reorder(Group.1,x))) +

geom\_point(size=2)+

xlab("ESG\_score")+ylab("Год")

a <- aggregate(data\_out$Social\_score, by=list(data\_out$Date), FUN=mean)

ggplot(a, aes(x=x, y=reorder(Group.1,x))) +

geom\_point(size=2)+

xlab("Социальный фактор")+ylab("Год")

a <- aggregate(data\_out$Gov\_score, by=list(data\_out$Date), FUN=mean)

ggplot(a, aes(x=x, y=reorder(Group.1,x))) +

geom\_point(size=2)+

xlab("Фактор корпоративного управления")+ylab("Год")

a <- aggregate(data\_out$Env\_score, by=list(data\_out$Date), FUN=mean)

ggplot(a, aes(x=x, y=reorder(Group.1,x))) +

geom\_point(size=2)+

xlab("Экологический фактор")+ylab("Год")

View(summary(data\_out$Industry))

b <- summary(data\_out$Industry)

c <- sort(b,decreasing = TRUE)

View(c)

sum(c[1:10])

industry <- c("Equity Real Estate Investment Trusts (REITs)","IT Services", "Capital Markets", "Oil, Gas & Consumable Fuels", "Health Care Providers & Services",

"Insurance", "Health Care Equipment & Supplies", "Machinery", "Banks", "Semiconductors & Semiconductor Equipment")

id <- filter(data\_out,data\_out$Industry==industry)

View(id)

#корреляционная матрица

data\_cl <- na.omit(data\_out)

mat1 <- as.dist(round(cor(data\_cl[,-c(1,2,3,10)]),3))

library(GGally)

ggcorr(data\_cl[,-c(1,2,3,10)], nbreaks = 6,

low = "steelblue",

mid = "white",

high = "darkred",

label = TRUE,

label\_size = 3,

legend.size = 9,

legend.position = "right",

nudge\_x=-1.2,

layout.exp = 2)

#построим модель

#модель с фиксированными эффектами

View(data\_out)

mod <- lm(data = data\_out,

formula = data\_out$Price~data\_out$BVPS+data\_out$NIPS+data\_out$Total\_assets +

data\_out$FNCL\_LVRG+data\_out$ESG\_score+data\_out$Date)

summary(mod)

plot(mod, which = 1)

#пропущена какая-то степень

#тест Бокса-кокса

boxCox(mod)

summary(powerTransform(mod))

mod2 <- lm(data = data\_out,

formula = log(data\_out$Price)~data\_out$BVPS+data\_out$NIPS+data\_out$Total\_assets +

data\_out$FNCL\_LVRG+data\_out$ESG\_score+data\_out$Date)

summary(mod2)

plot(mod2,1)

boxCox(mod2)

library(car)

crPlots(mod2)

#тест Рамсея для пропуска степени, Н0 - степени не пропущены

library(lmtest)

resettest(mod2, power=2)

resettest(mod2, power=3)

#p-value < любого уровня значимости, значить H0 отвергается, степени пропущены

mod3 <- lm(data = data\_out,

formula = log(data\_out$Price)~data\_out$BVPS+I(data\_out$NIPS^3)+data\_out$Total\_assets +

data\_out$FNCL\_LVRG+data\_out$ESG\_score+data\_out$Date)

summary(mod3)

plot(mod3,1)

crPlots(mod3)

#стало хуже по R2 и по графикам

mod3 <- lm(data = data\_out,

formula = log(data\_out$Price)~data\_out$BVPS+I(data\_out$NIPS^2)+data\_out$Total\_assets +

data\_out$FNCL\_LVRG+data\_out$ESG\_score+data\_out$Date)

summary(mod3)

plot(mod3,1)

crPlots(mod3)

resettest(mod3,power=2)

#опять не лучше

#попробуем взять логарифм активов и левериджа

mod4 <- lm(data = data\_out,

formula = data\_out$Price~data\_out$BVPS+data\_out$NIPS+log(data\_out$Total\_assets) +

log(data\_out$FNCL\_LVRG)+data\_out$ESG\_score+data\_out$Date)

summary(mod4)

crPlots(mod4)

plot(mod4,1)

#R2 увеличился, по графикам стало лучше

#посмотрим нормальность остатков

plot(mod4,2)

plot(mod2,2)

#интересно, что в модели без логарифма остатки нормальны

#мультиколлинеарность

vif(mod4)

vif(mod2)

plot(mod2,3)

plot(mod4,3)

bptest(mod2)

#есть гетероскедастичность

#удалим влиятельные наблюдения

plot(mod4, 4)

cook\_mod <- cooks.distance(mod4)

K <- which(cook\_mod > 4/4511)

K

e <- resid(mod4)

g <- abs(e/sd(e))

barplot(g)

N <- which(g>3)

N

OUT <- c(K,N)

OUT <- unique(OUT)

data\_out2 <- data\_out[-OUT,]

View(data\_out2)

#модель на новой выборке

mod5 <- lm(data = data\_out2,

formula = log(data\_out2$Price)~data\_out2$BVPS+data\_out2$NIPS+log(data\_out2$Total\_assets) +

log(data\_out2$FNCL\_LVRG)+data\_out2$ESG\_score+data\_out2$Date)

summary(mod5)

plot(mod5,1)

crPlots(mod5)

resettest(mod5, power=3)

mod6 <- lm(data = data\_out2,

formula = log(data\_out2$Price)~data\_out2$BVPS+I(data\_out2$NIPS^3)+data\_out2$Total\_assets +

data\_out2$FNCL\_LVRG+data\_out2$ESG\_score+data\_out2$Date)

summary(mod6)

plot(mod6,1)

crPlots(mod6)

resettest(mod6, power=2)

#опять только хуже

#еще раз с логарифмом

mod7 <- lm(data = data\_out2,

formula = log(data\_out2$Price)~data\_out2$BVPS+data\_out2$NIPS+log(data\_out2$Total\_assets) +

log(data\_out2$FNCL\_LVRG)+data\_out2$ESG\_score+data\_out2$Date)

summary(mod7)

plot(mod7,1)

crPlots(mod7)

plot(mod7,2)

plot(mod7,3)

#гетероскедастичность все еще осталась

#Лучше не становится, нет смысла улучшать, mod7 наилучшая

library(sandwich)

cov\_wtite <- vcovHC(mod7, type="HC0")

coeftest(mod7,cov\_wtite)

#добавим фиксированный эффект для отрасли

mod7\_2 <- lm(data = data\_out2,

formula = log(data\_out2$Price)~data\_out2$BVPS+data\_out2$NIPS+log(data\_out2$Total\_assets) +

log(data\_out2$FNCL\_LVRG)+data\_out2$ESG\_score+data\_out2$Date+data\_out2$Industry)

summary(mod7\_2)

cov\_wtite2 <- vcovHC(mod7\_2, type="HC0")

coeftest(mod7\_2,cov\_wtite2)

#влияние ESG значимо

##### Модель для фактора E ######

mode <- lm(data = data\_out,

formula = data\_out$Price~data\_out$BVPS+data\_out$NIPS+data\_out$Total\_assets +

data\_out$FNCL\_LVRG+data\_out$Env\_score+data\_out$Date)

summary(mode)

plot(mode,1)

vif(mode)

boxCox(mode)

crPlots(mode)

mode2 <- lm(data = data\_out,

formula = log(data\_out$Price)~data\_out$BVPS+data\_out$NIPS+log(data\_out$Total\_assets) +

log(data\_out$FNCL\_LVRG)+data\_out$Env\_score+data\_out$Date)

summary(mode2)

boxCox(mode2)

plot(mode2,1)

plot(mode2,2)

plot(mode2,3)

plot(mode2, 4)

cook\_mod <- cooks.distance(mode2)

K <- which(cook\_mod > 4/4511)

K

e <- resid(mode2)

g <- abs(e/sd(e))

barplot(g)

N <- which(g>3)

N

OUT <- c(K,N)

OUT <- unique(OUT)

data\_out3 <- data\_out[-OUT,]

View(data\_out3)

#новая модель

mode3 <- lm(data = data\_out3,

formula = log(data\_out3$Price)~data\_out3$BVPS+data\_out3$NIPS+log(data\_out3$Total\_assets) +

log(data\_out3$FNCL\_LVRG)+data\_out3$Env\_score+data\_out3$Date)

summary(mode3)

plot(mode3,1)

plot(mode3,2)

crPlots(mode3)

plot(mode3,3)

#введем робастные ошибки

cov\_wtite3 <- vcovHC(mode3, type="HC0")

coeftest(mode3,cov\_wtite3)

#добавим отрасть

mode4 <- lm(data = data\_out3,

formula = log(data\_out3$Price)~data\_out3$BVPS+data\_out3$NIPS+log(data\_out3$Total\_assets) +

log(data\_out3$FNCL\_LVRG)+data\_out3$Env\_score+data\_out3$Date+data\_out3$Industry)

summary(mode4)

cov\_wtite41 <- vcovHC(mode4, type="HC0")

coeftest(mode4,cov\_wtite41)

#### Модель для фактора S #####

mods <- lm(data = data\_out,

formula = data\_out$Price~data\_out$BVPS+data\_out$NIPS+data\_out$Total\_assets +

data\_out$FNCL\_LVRG+data\_out$Social\_score+data\_out$Date)

summary(mods)

plot(mods,1)

vif(mods)

boxCox(mods)

crPlots(mods)

mods2 <- lm(data = data\_out,

formula = log(data\_out$Price)~data\_out$BVPS+data\_out$NIPS+log(data\_out$Total\_assets) +

log(data\_out$FNCL\_LVRG)+data\_out$Social\_score+data\_out$Date)

summary(mods2)

boxCox(mods2)

plot(mods2,1)

plot(mods2,2)

plot(mods2,3)

plot(mods2, 4)

cook\_mod <- cooks.distance(mods2)

K <- which(cook\_mod > 4/4511)

K

e <- resid(mods2)

g <- abs(e/sd(e))

barplot(g)

N <- which(g>3)

N

OUT <- c(K,N)

OUT <- unique(OUT)

data\_out4 <- data\_out[-OUT,]

View(data\_out4)

#новая модель

mods3 <- lm(data = data\_out4,

formula = log(data\_out4$Price)~data\_out4$BVPS+data\_out4$NIPS+log(data\_out4$Total\_assets) +

log(data\_out4$FNCL\_LVRG)+data\_out4$Social\_score+data\_out4$Date)

summary(mods3)

plot(mods3,1)

plot(mods3,2)

crPlots(mods3)

plot(mods3,3)

#введем робастные ошибки

cov\_wtite4 <- vcovHC(mods3, type="HC0")

coeftest(mods3,cov\_wtite4)

#добавим отрасль

mods4 <- lm(data = data\_out4,

formula = log(data\_out4$Price)~data\_out4$BVPS+data\_out4$NIPS+log(data\_out4$Total\_assets) +

log(data\_out4$FNCL\_LVRG)+data\_out4$Social\_score+data\_out4$Date+data\_out4$Industry)

summary(mods4)

cov\_wtite5 <- vcovHC(mods4, type="HC0")

coeftest(mods4,cov\_wtite5)

#### Модель для фактора G####

modg <- lm(data = data\_out,

formula = data\_out$Price~data\_out$BVPS+data\_out$NIPS+data\_out$Total\_assets +

data\_out$FNCL\_LVRG+data\_out$Gov\_score+data\_out$Date)

summary(modg)

plot(modg,1)

vif(modg)

boxCox(modg)

crPlots(modg)

modg2 <- lm(data = data\_out,

formula = log(data\_out$Price)~data\_out$BVPS+data\_out$NIPS+log(data\_out$Total\_assets) +

log(data\_out$FNCL\_LVRG)+data\_out$Gov\_score+data\_out$Date)

summary(modg2)

boxCox(modg2)

plot(modg2,1)

plot(modg2,2)

plot(modg2,3)

crPlots(modg2)

plot(modg2, 4)

cook\_mod <- cooks.distance(modg2)

K <- which(cook\_mod > 4/4511)

K

e <- resid(modg2)

g <- abs(e/sd(e))

barplot(g)

N <- which(g>3)

N

OUT <- c(K,N)

OUT <- unique(OUT)

data\_out5 <- data\_out[-OUT,]

View(data\_out5)

#новая модель

modg3 <- lm(data = data\_out5,

formula = log(data\_out5$Price)~data\_out5$BVPS+data\_out5$NIPS+log(data\_out5$Total\_assets) +

log(data\_out5$FNCL\_LVRG)+data\_out5$Gov\_score+data\_out5$Date)

summary(modg3)

plot(modg3,1)

plot(modg3,2)

crPlots(modg3)

plot(modg3,3)

#введем робастные ошибки

cov\_wtite6 <- vcovHC(modg3, type="HC0")

coeftest(modg3,cov\_wtite6)

#добавим отрасль

modg4 <- lm(data = data\_out5,

formula = log(data\_out5$Price)~data\_out5$BVPS+data\_out5$NIPS+log(data\_out5$Total\_assets) +

log(data\_out5$FNCL\_LVRG)+data\_out5$Gov\_score+data\_out5$Date+data\_out5$Industry)

summary(modg4)

cov\_wtite7 <- vcovHC(modg4, type="HC0")

coeftest(modg4,cov\_wtite7)

###выгрузим все красиво

library(stargazer)

stargazer(coeftest(mod7,cov\_wtite),coeftest(mode3,cov\_wtite3),coeftest(mods3,cov\_wtite4),coeftest(modg3,cov\_wtite6),type = "text",column.labels=c("Модель 1", "Модель 2","Модель 3", "Модель 4"),

df=FALSE, digits=3, out = "models1.doc" )

stargazer(mod7,mode3,mods3,modg3, type = "text", digits = 3, out = "models2.doc" )

stargazer(coeftest(mod7\_2,cov\_wtite2),coeftest(mode4,cov\_wtite41),coeftest(mods4,cov\_wtite5),coeftest(modg4,cov\_wtite7),type = "text",column.labels=c("Модель 1", "Модель 2","Модель 3", "Модель 4"),

df=FALSE, digits=3, out = "models3.doc" )

stargazer(coeftest(mod7\_2,cov\_wtite2),coeftest(mode4,cov\_wtite41),type = "text",column.labels=c("Модель 1", "Модель 2","Модель 3", "Модель 4"),

df=FALSE, digits=3, out = "models32.doc" )

stargazer(coeftest(mods4,cov\_wtite5),coeftest(modg4,cov\_wtite7),type = "text",column.labels=c("Модель 1", "Модель 2","Модель 3", "Модель 4"),

df=FALSE, digits=3, out = "models33.doc" )

stargazer(mod7\_2,mode4,mods4,modg4,type = "text",column.labels=c("Модель 1", "Модель 2","Модель 3", "Модель 4"),

df=FALSE, digits=3, out = "models4.doc" )

######Построение портфелей#####

full\_data <- read\_excel("Уни/8 семестр/Диплом/full\_data.xlsx")

View(full\_data)

data <- full\_data[,c(1,2,3,4,5,6,7,8,10,13,14,15,66)]

View(data)

data$`Identifier (RIC)` <- as.factor(data$`Identifier (RIC)`)

#Топ 25 компаний и их устойчивость в топе

data2019 <- filter(data, data$Date==2019)

data2019 <- data2019[order(-data2019$ESG\_score),]

top2019 <- data2019[c(1:25),c(1,2,4)]

View(top2019)

data2018 <- filter(data, data$Date==2018)

data2018 <- data2018[order(-data2018$ESG\_score),]

top2018 <- data2018[c(1:25),c(1,2,4)]

View(top2018)

data2017 <- filter(data, data$Date==2017)

data2017 <- data2017[order(-data2017$ESG\_score),]

top2017 <- data2017[c(1:25),c(1,2,4)]

View(top2017)

data2016 <- filter(data, data$Date==2016)

data2016 <- data2016[order(-data2016$ESG\_score),]

top2016 <- data2016[c(1:25),c(1,2,4)]

View(top2016)

data2015 <- filter(data, data$Date==2015)

data2015 <- data2015[order(-data2015$ESG\_score),]

top2015 <- data2015[c(1:25),c(1,2,4)]

View(top2015)

data2014 <- filter(data, data$Date==2014)

data2014 <- data2014[order(-data2014$ESG\_score),]

top2014 <- data2014[c(1:25),c(1,2,4)]

View(top2014)

x <- c(top2019$`Company Name`,top2018$`Company Name`)

View(x)

x <- unique(x)

y <- c(top2017$`Company Name`,top2016$`Company Name`)

View(y)

y <- unique(y)

y <- c(top2018$`Company Name`,top2017$`Company Name`)

View(y)

y <- unique(y)

y <- c(top2016$`Company Name`,top2015$`Company Name`)

View(y)

y <- unique(y)

y <- c(top2019$`Company Name`,top2015$`Company Name`)

View(y)

y <- unique(y)

###Отберем наилучший и наихудший кваритили по ESG

bad\_q2014 <- quantile(data2014$ESG\_score, 0.25)

bad\_q2014

bad\_q2014\_num <- which(data2014$ESG\_score<bad\_q2014)

bad\_q2014\_num

best\_q2014 <- quantile(data2014$ESG\_score, 0.75)

best\_q2014

best\_q2014\_num <- which(data2014$ESG\_score>best\_q2014)

best\_q2014\_num

ticker\_w2014 <- data2014$`Identifier (RIC)`[bad\_q2014\_num]

ticker\_b2014 <- data2014$`Identifier (RIC)`[best\_q2014\_num]

#скачаем котировки акций

D <- new.env()

sDate <- as.Date("2014-10-30")

eDate <- as.Date("2015-10-30")

num\_ticker <- length(ticker\_b2014)

Temp\_D <- list()

counter <- 1L

for(i in ticker\_b2014) {

getSymbols(

i,

env = D,

reload.Symbols = FALSE,

from = sDate,

to = eDate,

verbose = FALSE,

warnings = TRUE,

src = "yahoo",

symbol.lookup = TRUE)

print(counter)

Temp\_D[[i]] <- Cl(D[[i]])

if (counter == length(ticker\_b2014))

{

#Merge all the objects of the list into one object.

Daily\_Price <- do.call(merge, Temp\_D)

}

else

{

counter <- counter + 1

}

}

View(Daily\_Price)

best2015 <- Daily\_Price

#для первого квартиля

D <- new.env()

sDate <- as.Date("2014-10-30")

eDate <- as.Date("2015-10-30")

num\_ticker <- length(ticker\_w2014)

Temp\_D <- list()

counter <- 1L

for(i in ticker\_w2014) {

getSymbols(

i,

env = D,

reload.Symbols = FALSE,

from = sDate,

to = eDate,

verbose = FALSE,

warnings = TRUE,

src = "yahoo",

symbol.lookup = TRUE)

print(counter)

Temp\_D[[i]] <- Cl(D[[i]])

if (counter == length(ticker\_w2014))

{

#Merge all the objects of the list into one object.

Daily\_Price <- do.call(merge, Temp\_D)

}

else

{

counter <- counter + 1

}

}

View(Daily\_Price)

worst2015 <- Daily\_Price

#Построим портфель

#сначала посмотрим на какую-нибудь акцию

library(PortfolioAnalytics)

library(ROI)

library(ROI.plugin.glpk)

library(DEoptim)

library(quantmod)

best2015 <- na.omit(best2015)

Returns20151 <-Return.calculate(best2015, method = "log")

Returns20151

A <- Returns20151$A.Close

plot(A)

chart.RollingPerformance(A, width = 30, FUN = "sd") #скольязящее среднее для дисперсии

#вывод - есть гетероскедастичность

chart.CumReturns(A) #видим как мы росли

chart.Drawdown(A) #куммулятивные провалы

chart.Histogram(A, methods = c("add.density"))

chart.Histogram(A, methods = c("add.density", "add.normal"))

colnames(Returns20151)

Port <- portfolio.spec(assets = colnames(Returns20151))

#средние доходности

Returns20151 <- na.omit(Returns20151)

(colMeans(Returns20151))

#линейные ограничения

Port <- add.constraint(Port, type = "full\_invsetment") #сумма весов 1

Port <- add.constraint(Port, type = "long\_only")

#Port <- add.constraint(Port, type="diversification", div\_target=1-(1/dim(Returns20191)[2]))

Port <- add.constraint(Port, type = "return", name = "mean", return\_target = 0.001)

#добавим цель - минимизация дисперсии

Port <- add.objective(Port, type = "risk", name = "StdDev")

#метод оптимизации

Result <- optimize.portfolio(portfolio = Port, R = Returns20151, optimize\_method = "ROI")

Result$weights

Result$objective\_measures

w1 <- (Result$weights)

1-sum(w1^2)

#построим плохой портфель

worst2015 <- na.omit(worst2015)

Returns20152 <-Return.calculate(worst2015, method = "log")

BIO <- Returns20152$BIO.Close

plot(BIO)

chart.RollingPerformance(BIO, width = 30, FUN = "sd") #скольязящее среднее для дисперсии

#вывод - есть гетероскедастичность

chart.CumReturns(BIO) #видим как мы росли

chart.Drawdown(BIO) #куммулятивные провалы

chart.Histogram(BIO, methods = c("add.density"))

chart.Histogram(BIO, methods = c("add.density", "add.normal"))

colnames(Returns20152)

Port2 <- portfolio.spec(assets = colnames(Returns20152))

#средние доходности

Returns20152 <- na.omit(Returns20152)

mean(colMeans(Returns20152))

#линейные ограничения

Port2 <- add.constraint(Port2, type = "full\_invsetment") #сумма весов 1

Port2 <- add.constraint(Port2, type = "long\_only")

#Port <- add.constraint(Port, type="diversification", div\_target=1-(1/dim(Returns20191)[2]))

Port2 <- add.constraint(Port2, type = "return", name = "mean", return\_target = 0.001)

#добавим цель - минимизация дисперсии

Port2 <- add.objective(Port2, type = "risk", name = "StdDev")

#метод оптимизации

Result2 <- optimize.portfolio(portfolio = Port2, R = Returns20152, optimize\_method = "ROI")

Result2$weights

Result2$objective\_measures

w2 <- (Result2$weights)

1-sum(w2^2)

#найдем доходность портфеля для каждого периода

Day\_return1 <- Return.portfolio(R = Returns20151, weights = w1)

plot(Day\_return1)

Day\_return2 <- Return.portfolio(R = Returns20152, weights = w2)

plot(Day\_return2)

#скачаем цены акций за 2016-2018 года для хорошего портфеля

D <- new.env()

sDate <- as.Date("2015-10-30")

eDate <- as.Date("2018-10-30")

num\_ticker <- length(ticker\_b2014)

Temp\_D <- list()

counter <- 1L

for(i in ticker\_b2014) {

getSymbols(

i,

env = D,

reload.Symbols = FALSE,

from = sDate,

to = eDate,

verbose = FALSE,

warnings = TRUE,

src = "yahoo",

symbol.lookup = TRUE)

print(counter)

Temp\_D[[i]] <- Cl(D[[i]])

if (counter == length(ticker\_b2014))

{

#Merge all the objects of the list into one object.

Daily\_Price <- do.call(merge, Temp\_D)

}

else

{

counter <- counter + 1

}

}

View(Daily\_Price)

best2016\_2018 <- Daily\_Price

# и для плохого портфеля

D <- new.env()

sDate <- as.Date("2015-10-30")

eDate <- as.Date("2018-10-30")

num\_ticker <- length(ticker\_w2014)

Temp\_D <- list()

counter <- 1L

for(i in ticker\_w2014) {

getSymbols(

i,

env = D,

reload.Symbols = FALSE,

from = sDate,

to = eDate,

verbose = FALSE,

warnings = TRUE,

src = "yahoo",

symbol.lookup = TRUE)

print(counter)

Temp\_D[[i]] <- Cl(D[[i]])

if (counter == length(ticker\_w2014))

{

#Merge all the objects of the list into one object.

Daily\_Price <- do.call(merge, Temp\_D)

}

else

{

counter <- counter + 1

}

}

View(Daily\_Price)

worst2016\_2018 <- Daily\_Price

#построим портфели с заданными весами на этом промежутке

Returns20161 <- Return.calculate(best2016\_2018, method = "log")

ret20161 <- Return.portfolio(R = Returns20161, weights = w1)

charts.PerformanceSummary(R = ret20161)

sum(ret20161)

var(ret20161)

Table\_return <- table.CalendarReturns(ret20161)

Table\_return

StdDev.annualized(ret20161)

year\_ret20161 <- periodReturn(ret20161)

year\_ret20161

?periodReturn

Returns20162 <- Return.calculate(worst2016\_2018, method = "log")

ret20162 <- Return.portfolio(R = Returns20162, weights = w2)

charts.PerformanceSummary(R = ret20162)

sum(ret20162)

var(ret20162)

#####Портфели с равными весами#####

library(readxl)

best\_2018 <- read\_excel("best\_2018.xlsx")

View(best\_2018)

str(best\_2018)

d <- best\_2018[,1]

d <- as.data.frame(d)

b <- best\_2018[,-1]

str(b)

ymd(d)

best <- as.xts(b, order.by = as.Date(d$Date))

best\_2018 <- na.omit(best\_2018)

str(best\_2018)

Ret161 <- Return.calculate(best, method = "log")

View(Ret161)

Ret161 <- na.omit(Ret161)

P3<- Return.portfolio(Ret161, weights = rep((1/125),times = 125))

charts.PerformanceSummary(R = P3)

sum(P3)

mean(P3)

var(P3)

Table\_return1 <- table.CalendarReturns(P3)

Table\_return1

StdDev.annualized(P3)

#для плохого портфеля

worst\_2018 <- read\_excel("worst\_2018.xlsx")

View(worst\_2018)

str(worst\_2018)

d <- worst\_2018[,1]

#d <- as.data.frame(d)

w <- worst\_2018[,-1]

str(w)

worst <- as.xts(w, order.by = as.Date(d$Date))

str(worst)

Ret162 <- Return.calculate(worst, method = "log")

View(Ret162)

Ret162 <- na.omit(Ret162)

P4<- Return.portfolio(Ret162, weights = rep((1/105),times = 105))

charts.PerformanceSummary(R = P4)

sum(P4)

mean(P4)

var(P4)

Table\_return2 <- table.CalendarReturns(P4)

Table\_return2

StdDev.annualized(P4)

dim(Table\_return1)

t.test(Table\_return1[,13],Table\_return2[,13], var.equal = FALSE)

####С короткими позициями####

best\_2015 <- read\_excel("best\_2015.xlsx")

View(best\_2015)

str(best\_2015)

d <- best\_2015[,1]

b <- best\_2015[,-1]

str(b)

best <- as.xts(b, order.by = as.Date(d$Date))

str(best)

Ret151 <- Return.calculate(best, method = "log")

View(Ret151)

Ret151 <- na.omit(Ret151)

colnames(Ret151)

Port5 <- portfolio.spec(assets = colnames(Ret151))

#средние доходности

max((colMeans(Ret151)))

min((colMeans(Ret151)))

#линейные ограничения

Port5 <- add.constraint(Port5, type = "full\_invsetment") #сумма весов 1

Port5 <- add.constraint(Port5, type = "return", name = "mean", return\_target = 0.002)

#добавим цель - минимизация дисперсии

Port5 <- add.objective(Port5, type = "risk", name = "StdDev")

#метод оптимизации

Result5 <- optimize.portfolio(portfolio = Port5, R = Ret151, optimize\_method = "ROI")

Result5$weights

Result5$objective\_measures

w5 <- (Result5$weights)

#для плохого портфеля

worst\_2015 <- read\_excel("worst\_2015.xlsx")

View(worst\_2015)

str(worst\_2015)

d <- worst\_2015[,1]

b <- worst\_2015[,-1]

str(b)

worst <- as.xts(b, order.by = as.Date(d$Date))

str(worst)

Ret152 <- Return.calculate(worst, method = "log")

View(Ret152)

Ret152 <- na.omit(Ret152)

colnames(Ret152)

Port6 <- portfolio.spec(assets = colnames(Ret152))

#средние доходности

max((colMeans(Ret152)))

min((colMeans(Ret152)))

#линейные ограничения

Port6 <- add.constraint(Port6, type = "full\_invsetment") #сумма весов 1

Port6 <- add.constraint(Port6, type = "return", name = "mean", return\_target = 0.002)

#добавим цель - минимизация дисперсии

Port6 <- add.objective(Port6, type = "risk", name = "StdDev")

#метод оптимизации

Result6 <- optimize.portfolio(portfolio = Port6, R = Ret152, optimize\_method = "ROI")

Result6$weights

Result6$objective\_measures

w6 <- (Result6$weights)

#посмотрим как вели весбя портфели на продолжительном периоде

ret20161 <- Return.portfolio(R = Ret161, weights = w5)

charts.PerformanceSummary(R = ret20161)

sum(ret20161)

var(ret20161)

Table\_return3 <- table.CalendarReturns(ret20161)

Table\_return3

StdDev.annualized(ret20161)

ret20162 <- Return.portfolio(R = Ret162, weights = w6)

charts.PerformanceSummary(R = ret20162)

sum(ret20162)

var(ret20162)

Table\_return4 <- table.CalendarReturns(ret20162)

Table\_return4

StdDev.annualized(ret20162)

?t.test

library(stats)

t.test(Table\_return3[,13],Table\_return4[,13],var.equal = FALSE)

####Регрессии с финансовыми показателями#####

View(full\_data)

m <- lm(data=data\_out, formula = data\_out$ROE~data\_out$ESG\_score+data\_out$Total\_assets+data\_out$FNCL\_LVRG+data\_out$QUICK\_RATIO+data\_out$ASSET\_GROWTH+data\_out$Date)

summary(m)

plot(m, which = 1)

#пропущена какая-то степень

crPlots(m)

#попробуем взять логарифм активов, ликвидности, роста и рычага

m2 <- lm(data=data\_out, formula = data\_out$ROE~data\_out$ESG\_score+log(data\_out$Total\_assets)+log(data\_out$FNCL\_LVRG)+log(data\_out$QUICK\_RATIO)+log(data\_out$ASSET\_GROWTH)+data\_out$Date)

summary(m2)

stargazer(m2, type = "text")

crPlots(m2)

plot(m2,2)

#остатки не нормальны

plot(m2,3)

plot(m2, 4)

cook\_mod <- cooks.distance(m2)

K <- which(cook\_mod > 4/2851)

K

e <- resid(m2)

g <- abs(e/sd(e))

barplot(g)

N <- which(g>3)

N

OUT <- c(K,N)

OUT <- unique(OUT)

data\_out4 <- data\_out[-OUT,]

#новая модель

m3 <- lm(data=data\_out4, formula = data\_out4$ROE~data\_out4$ESG\_score+log(data\_out4$Total\_assets)+log(data\_out4$FNCL\_LVRG)+log(data\_out4$QUICK\_RATIO)+log(data\_out4$ASSET\_GROWTH)+data\_out4$Date)

plot(m3,1)

plot(m3,2)

plot(m3,3)

#есть небольшая гетероскедастичность, введем робастные ошибки

cov\_wtite8 <- vcovHC(m3, type="HC0")

coeftest(m3,cov\_wtite8)

#####Модель для ROA#####

data\_out$RETURN\_ON\_ASSET <- data\_out$RETURN\_ON\_ASSET/100

m4 <- lm(data=data\_out, formula = data\_out$RETURN\_ON\_ASSET~data\_out$ESG\_score+log(data\_out$Total\_assets)+log(data\_out$FNCL\_LVRG)+log(data\_out$QUICK\_RATIO)+log(data\_out$ASSET\_GROWTH)+data\_out$Date)

summary(m4)

stargazer(m4, type = "text")

crPlots(m4)

plot(m4,2)

#остатки не нормальны

plot(m4,3)

plot(m4, 4)

cook\_mod <- cooks.distance(m4)

K <- which(cook\_mod > 4/2851)

K

e <- resid(m4)

g <- abs(e/sd(e))

barplot(g)

N <- which(g>3)

N

OUT <- c(K,N)

OUT <- unique(OUT)

data\_out5 <- data\_out[-OUT,]

#новая модель

m5 <- lm(data=data\_out5, formula = data\_out5$RETURN\_ON\_ASSET~data\_out5$ESG\_score+log(data\_out5$Total\_assets)+log(data\_out5$FNCL\_LVRG)+log(data\_out5$QUICK\_RATIO)+log(data\_out5$ASSET\_GROWTH)+data\_out5$Date)

summary(m5)

plot(m5,1)

plot(m5,2)

plot(m5,3)

#есть небольшая гетероскедастичность, введем робастные ошибки

cov\_wtite9 <- vcovHC(m5, type="HC0")

coeftest(m5,cov\_wtite9)

#выведем все красиво

stargazer(coeftest(m3,cov\_wtite8), coeftest(m5,cov\_wtite9),type = "text",column.labels=c("Модель 9", "Модель 10"),

df=FALSE, digits=3, out = "models4.doc" )

stargazer(m3,m5,type = "text",column.labels=c("Модель 1", "Модель 2"),

df=FALSE, digits=3, out = "models5.doc" )

####Модель Фамы-Френча####

library(dplyr)

library(xts)

library(ggplot2)

library(PortfolioAnalytics)

library(readxl)

library(openxlsx)

library(lubridate)

library(car)

library(stargazer)

#загрузим данные для модели Фамы-Френча, они взяты с https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data\_library.html

library(readxl)

#данные

esg\_score <- read\_excel("esg score.xlsx")

sp <- read\_excel("sp.xlsx")

FF <- read\_excel("FF3.xlsx")

data\_FF <- mutate\_all(FF, as.numeric)

str(data\_FF)

#переименую одно имя

names(data\_FF)[2] <- "Mkt\_RF"

names(data\_FF)

#поделю на 100 чтобы получить доходности

data\_FF2 <- mutate(data\_FF, Mkt\_RF = Mkt\_RF/100, SMB = SMB/100, HML = HML/100, RF = RF/100)

View(data\_FF2)

#превратим в xts

data\_FF3 <- data\_FF2

data\_FF3$Date <- ymd(data\_FF2$Date)

#обработка данных

Time <- sp$Date

data <- select(sp, - Date)

data2 <- xts(data, order.by = as.Date(Time))

#перейдем к доходностям

r <- CalculateReturns(data2, method = "log")

r <- na.omit(r)

#объединим с ФФ

data\_FF4 <- select(data\_FF3, -Date)

data\_FF5 <- xts(data\_FF4, order.by = data\_FF3$Date)

data\_all <- merge(r, data\_FF5)

#удалим пропуски

data\_all <- na.omit(data\_all)

dim(data\_all)

head(data\_all)

#отберем 2020 год

data\_2020 <- window(data\_all, start = "2020-01-01", end = "2020-12-31")

dim(data\_2020)

#для проверки построим обычную модель без ESG

names(data\_2020)

mod <- lm(MSFT - RF.1 ~ Mkt\_RF + SMB + HML, data\_2020)

summary(mod)

#все хорошо

#создадим ESG фактор

esg\_2020 <- filter(esg\_score, Date == 2020)

up <- filter(esg\_2020, ESG\_score >

quantile(esg\_2020$ESG\_score, 0.7))

Names\_up <- up$`Identifier (RIC)`

data\_2020\_up <- data\_2020[, colnames(data\_2020) %in% Names\_up]

#вычислим доходность портфеля при равенстве весов

ESG\_high <- rowMeans(data\_2020\_up)

#аналогично для нижних 30%

down <- filter(esg\_2020, ESG\_score <

quantile(esg\_2020$ESG\_score, 0.3))

Names\_down <- down$`Identifier (RIC)`

data\_2020\_down <- data\_2020[, colnames(data\_2020) %in% Names\_down]

#вычислим доходность портфеля при равенстве весов

ESG\_down <- rowMeans(data\_2020\_down)

#Вычислим разницу верхнего и нижнего

ESG\_factor <- ESG\_high - ESG\_down

#переведем в форму ряда

ESG\_factor <- xts(ESG\_factor, order.by = date(data\_2020))

#добавим в данные

data\_2020\_2 <- merge(data\_2020, ESG\_factor)

dim(data\_2020\_2)

#проверочная регрессия

mod2 <- lm(MSFT - RF.1 ~ Mkt\_RF + SMB + HML + ESG\_factor, data\_2020\_2)

summary(mod2)

#поищем значимые

head(names(data\_2020\_2))

mod3 <- lm(JNJ - RF.1 ~ Mkt\_RF + SMB + HML + ESG\_factor, data\_2020\_2)

summary(mod3)

#построим оценки для всех и выгрузим

#первые 486 - это акции

K <- matrix(0, nrow = 486, ncol = 5)

t\_value <- matrix(0, nrow = 486, ncol = 5)

Stars <- matrix(0, nrow = 486, ncol = 5)

q <- qt(0.975, 247)

R2 <- NULL

#запустим цикл

for(i in 1:486) {

mod4 <- lm(data\_2020\_2[,i] - RF.1 ~ Mkt\_RF +

SMB + HML + ESG\_factor, data\_2020\_2)

S <- summary(mod4)

K[i,] <- S$coefficients[,1]

t\_value[i,] <- S$coefficients[,3]

Stars[i,] <- abs(S$coefficients[,3]) > q

R2[i] <- S$r.squared

}

#сколько раз ESG фактор значим?

sum(Stars[,5])

#значит, ESG фактор хорошо прогнозирует доходность

#сколько раз значима альфа?

sum(Stars[,1])

#ни разу

#проделаем все то же самое без R2 и посмотрим на сколько он

#меняется при добавлении ESG

K2 <- matrix(0, nrow = 486, ncol = 4)

t\_value2 <- matrix(0, nrow = 486, ncol = 4)

Stars2 <- matrix(0, nrow = 486, ncol = 4)

q <- qt(0.975, 247)

R2\_2 <- NULL

#запустим цикл

for(i in 1:486) {

mod4 <- lm(data\_2020\_2[,i] - RF.1 ~ Mkt\_RF +

SMB + HML, data\_2020\_2)

S <- summary(mod4)

K2[i,] <- S$coefficients[,1]

t\_value2[i,] <- S$coefficients[,3]

Stars2[i,] <- abs(S$coefficients[,3]) > q

R2\_2[i] <- S$r.squared

}

#изменен R2

R2\_delta <- R2 - R2\_2

#среднее изменение R2

mean(R2\_delta)

#изменение R2 в моделях, где ESG значим

R2\_delta\_star <- R2\_delta\*Stars[,5]

R2\_delta\_star <- R2\_delta\_star[R2\_delta\_star != 0]

mean(R2\_delta\_star)

#выгрузим таблицу со значимыми коэффициентами (буду ставить нули там,

#где нет значимости)

Data\_result <- data.frame(K\*Stars, R2\_delta = R2\_delta\*Stars[,5])

#первый столбец не выгружаю, там нули

Data\_result <- Data\_result[, -1]

colnames(Data\_result) <- c("Mkt\_RF", "SMB", "HML", "ESG", "R2\_d")

rownames(Data\_result) <- colnames(r)

#выгрузка

stargazer(Data\_result, type = "html",

summary = FALSE, out = "Data\_result.html")

####ESG-граница

library(dplyr)

library(xts)

library(ggplot2)

library(PortfolioAnalytics)

#данные

esg\_score <- read\_excel("esg score.xlsx")

sp <- read\_excel("sp.xlsx")

esg\_score <- na.omit(esg\_score)

sp <- na.omit(sp)

#обработка данных

Time <- sp$Date

data <- select(sp, - Date)

#переведем в формат дат

data2 <- xts(data, order.by = as.Date(Time))

#отберем 2020 год

data\_2020 <- window(data2, start = "2020-01-01", end = "2020-12-31")

#составим топ ESG за 2020

esg\_2020 <- filter(esg\_score, Date == 2020)

#отберем 100 лучших активов

up\_100 <- filter(esg\_2020, ESG\_score >

quantile(esg\_2020$ESG\_score, 0.8))

dim(up\_100)

#получился 101

#достанем их имена

Names <- up\_100$`Identifier (RIC)`

#отберем их доходности

data\_2020\_100 <- data\_2020[, colnames(data\_2020) %in% Names]

dim(data\_2020\_100)

#перейдем к доходностям

r <- CalculateReturns(data\_2020\_100, method = "log")

r <- na.omit(r)

#вектор средних

r\_mean <- colMeans(r)

#ковариационная матрица

V <- var(r)

#их ESG параметры

ESG <- up\_100[Names %in% colnames(data\_2020), 4]

#отклонения от среднего ESG

ESG\_norm <- ESG$ESG\_score - mean(ESG$ESG\_score)

#оптимальные параметры

SD <- mean(sqrt(diag(V)))

ESG\_target <- seq(-20,

20, length.out = 100)

SR <- NULL

#запустим цикл

for(i in 1:100) {

#получим оптимальные веса

ESG\_stand <- ESG\_norm - ESG\_target[i]

#получим веса

Pi <- - (ESG\_stand %\*% solve(V) %\*% r\_mean)/(ESG\_stand %\*% solve(V) %\*% ESG\_stand)

Teta <- 1/SD \* sqrt(r\_mean %\*% solve(V) %\*% t(r\_mean + Pi %\*% ESG\_stand))

Teta <- as.numeric(Teta)

x <- solve(V) %\*% t(r\_mean + Pi %\*% ESG\_stand)/ Teta

SR[i] <- sum(r\_mean\*x)/SD}

#граница

data\_2 <- data.frame(ESG\_target, SR)

ggplot(data\_2, aes(ESG\_target, SR)) + geom\_line() + theme\_bw()

#рыночный порфтель: проверка

z <- solve(V) %\*% r\_mean

w <- z/sum(z)

SD3 <- sqrt(t(w) %\*% V %\*% w)

M <- t(w) %\*% r\_mean

M/SD3