КЛАСТЕРНЫЙ\_ РЕГРЕССИОННЫЙ И ПОСТРЕГРЕССИОННЫЙ АНАЛИЗ

library(ggplot2)

# Создать файл исходных данных data\_1

data\_1 <- read.csv2("InItial\_Data\_2.csv")

data\_1

data\_1$FO <- as.factor(data\_1$FO)

data\_1$tip\_1 <- as.factor(data\_1$tip\_1)

data\_1$tip\_2 <- as.factor(data\_1$tip\_2)

data\_1$tip\_sv <- as.factor(data\_1$tip\_sv)

View(data\_1)

# Функция описательной статистики summary()

summary(data\_1)

summary(data\_1[,c('mon\_inc\_19', 'expen\_19')])

# Строим гистограммы

hist(data\_1$mon\_inc\_19)

hist(data\_1$expen\_19)

# Рассчитать число групп по формуле Стерджесса

hist(data\_1$mon\_inc\_19, breaks=15)

# Установить точное число групп

x <-data\_1$mon\_inc\_19

hist(x, breaks = seq(min(data\_1$mon\_inc\_19), max(data\_1$mon\_inc\_19), length.out = 11))

# Диаграмма "ящик-усы"

boxplot(expen\_19~FO,data\_1)

boxplot(mon\_inc\_19~FO,data\_1)

# Функции пакета ggplot2

## Построение гистораммы по качественной переменной

# Столбиковая диаграмма по категориальной переменной (по частотам)

p <- ggplot(data = data\_1,

mapping = aes(x = tip\_1))

p + geom\_bar()

# Столбиковая диаграмма по категориальной переменной (по частостям) - 1

## Пример\_1(группировочная переменная tip\_1)

p <- ggplot(data = data\_1,

mapping = aes(x = tip\_1, fill = ..prop..))

p + geom\_bar(mapping = aes(y = ..prop.., group = 1)) + scale\_fill\_gradient()

## Пример 2 (группировочная переменная - FO)

p <- ggplot(data = data\_1,

mapping = aes(x = FO))

p + geom\_bar(mapping = aes(y = ..prop.., group = 1,fill = ..prop..)) + scale\_fill\_gradient()

## по группам регионов, различающихся по уровню развития, с цветовой заливкой

p <- ggplot(data = data\_1,

mapping = aes(x = tip\_1, fill = ..prop..))

p + geom\_bar(mapping = aes(y = ..prop.., group = 1)) + scale\_fill\_gradient(low='coral', high='green')

# Построение гистограммы по количественной (непрерывной) переменной (ВРП на душу населения, тыс.руб.)

p <- ggplot(data = data\_1,

mapping = aes(x = gdp\_c\_18))

p + geom\_histogram()

# еще пример (темп роста ВРП, %)

p <- ggplot(data = data\_1,

mapping = aes(x = gdp\_g\_18))

p + geom\_histogram()

# Еще пример (распределение регионов РФ по величине среднедушевых денежных доходов в месяц,руб)

p <- ggplot(data = data\_1,

mapping = aes(x = mon\_inc\_19))

p + geom\_histogram()

# изменить ширину столбиков

p <- ggplot(data = data\_1,

mapping = aes(x = gdp\_c\_18))

p + geom\_histogram(bins=10)

# Справка

?geom\_histogram

# Наложение гистограмм, составленных по двум категориальным группам

gr\_1 <- c("Среднеразвитые", "Развитые")

p <- ggplot(data = subset(data\_1, subset = tip\_1 %in% gr\_1),

mapping = aes(x = mon\_inc\_19, fill = tip\_1))

p + geom\_histogram(alpha = 0.4, bins = 40)+labs(x="ВРП в расчете на душу населения, тыс.руб.",

y="Число регионов")

subset(data\_1, subset = tip\_1 %in% gr\_1)

## Если кроме

p <- ggplot(data = subset(data\_1, subset = !(tip\_1 %in% gr\_1)),

mapping = aes(x = mon\_inc\_19, fill = tip\_1))

p + geom\_histogram(alpha = 0.4, bins = 40)+labs(x="ВРП в расчете на душу населения, тыс.руб.",

y="Число регионов")

# Визуализация статистических выбросов

## 1-й вариант - Функция boxlot

bp <- boxplot(data\_1$wage\_19,ylab = "Среднемесячная заработная плата работников организаций, руб.")

## 2-й вариант - пакет ggplot2

ggplot(data\_1) +

aes(y = wage\_19) +

geom\_boxplot(fill = "green") +

theme\_minimal()

## Наложение кривой нормального распределения на гистограмму (в пакете ggplot2)

p <- ggplot(data = data\_1,

mapping = aes(x = mon\_inc\_19))

p + geom\_histogram(color='blue',fill="lightblue")

x <- data\_1$mon\_inc\_19

p <- ggplot(data = data\_1,

mapping = aes(x = mon\_inc\_19))

p + geom\_histogram(aes(y=..density..), color='blue',fill="lightblue") +

stat\_function(fun = dnorm, args=list(mean=mean(x), sd=sd(x)), color='red', size=2)

# То же самое через базовые функции

### Простоить гистограмму и занестия её значения в переменную h

h<-hist(x, breaks=10, col="lightblue", xlab="Среднемесячные денежные доходы населения, руб.",

main="Распределение регионов РФ по величине среднедушевых денежных доходов населения, 2019 г.")

### В xfit заносим значения ряда (переменной X, см.выше) от минимального до максимального с разбивкой на 50 групп

### для построения гладкой функции

### стандартизируем xfit (разность середины 1-го и 2-го столюиков гистограммы умножается на доину исходного ряда x)/

### приводим в единую размерность с гистограммой

xfit<-seq(min(x),max(x),length=50)

yfit<-dnorm(xfit,mean=mean(x),sd=sd(x))

yfit <- yfit\*diff(h$mids[1:2])\*length(x)

lines(xfit, yfit, col="red", lwd=5)

### посмотри результат

h$mids

h$mids[1:2]

# График плотности распределения

## по переменной Среднемесячный денежный доход. руб.

x <- data\_1$mon\_inc\_19

d <- density(x) # returns the density data

plot(d) # plots the results

# Filled Density Plot

d <- density(x)

plot(d, main="Диагональ картинки")

polygon(d, col="red", border="blue")

# Comparing Groups VIA Kernal Density

## Загрузить пакеты: SM, VIOPLOT

library(sm)

# create value labels

z <- data\_1$tip\_1

# plot densities

## Распредленение среднедушевых денежных доходов по группам регионов

comp <- sm.density.compare(x, z, xlab="Среднедушевые денежные доходы населения, руб.", size=2)

legend("topright", comp$levels, col = comp$col, lty = comp$lty, lwd = comp$lwd)

## Распредленение среднемесячной заработной платы по группам регионов

comp\_1<-sm.density.compare(data\_1$wage\_19,data\_1$tip\_1,xlab="Заработная плата в организациях,

в среднем за месяц, руб.", size=2)

legend("topright", comp\_1$levels, col = comp\_1$col, lty = comp\_1$lty, lwd = comp\_1$lwd)

?legend

# Скрипичные диаграммы

# Скрипичные диаграммы в пакете ggplot2

# VIOLIN GGPLOT

## см.выше: z,x

p <- ggplot(data\_1, aes(z, x))

p + geom\_violin()

p <- ggplot(data\_1, aes(z, x))

p + geom\_violin(aes(fill = z)) +

geom\_boxplot(width=.1)

# Перевернуть подписи по оси X

p + theme(axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust=1)) +

geom\_violin(aes(fill = z)) +

geom\_boxplot(width=.1)

# Убрать легенду

p + guides(fill=FALSE) +

geom\_violin(aes(fill = z)) +

geom\_boxplot(width=.1)

# Выбросы

bp <- boxplot(x)

bp$out

# Тесты Диксона и Граббса (пакет outliers)

library(outliers)

# Тест Диксона(создадим для ЦФО массив данных по переменной gdp\_c\_18. Это 14 наблюдений!)

gdp\_c\_18\_CFO <- data\_1[data\_1$FO =='ЦФО',]$gdp\_c\_18

dixon.test(gdp\_c\_18\_CFO,type=22,opposite=TRUE,two.sided=FALSE)

dixon.test(gdp\_c\_18\_CFO,type=22,opposite=FALSE,two.sided=FALSE)

?dixon.test()

# Тест Граббса

## Для всего массива данных data\_1, тест на выбросы в противоположных хвостах (type=11)

grubbs.test(data\_1$gdp\_c\_18,type=11,opposite=TRUE)

?grubbs.test

# Исключаем строку с выбросом gdp\_c\_18 = 6950415,5

data\_1\_3 <- data\_1[data\_1$gdp\_c\_18 != 6950415.5,]

# Исключаем 2 строки с выбросами gdp\_c\_18 = 6950415,5 и gdp\_c\_18 = 112553,4

data\_1\_4 <- data\_1[!(data\_1$gdp\_c\_18 %in% c(112553.4, 6950415.5)),]

# Тест Граббса на 2 выброса в одном хвосте для ЦФО

grubbs.test(gdp\_c\_18\_CFO,type=20,opposite=TRUE,two.sided=FALSE)

grubbs.test(gdp\_c\_18\_CFO,type=20,opposite=FALSE,two.sided=FALSE)

# Кластерный анализ

library(fastcluster)

data\_1\_2 <- scale(data\_1[,c('gdp\_c\_18', 'lab\_inc\_19', 'wage\_19', 'cpi\_19', 'expen\_19', 'pov\_19')])

data\_1\_2

# !!! Использование hclust из пакета stats

fit2 <- stats::hclust(dist(data\_1\_2), method="ward.D2")

plot(fit2)

# Дендрограмма

fit <- hclust(dist(data\_1\_2), method="ward.D2")

plot(fit)

groups <- cutree(fit, k=3)

rect.hclust(fit, k=3, border="red")

groups

table(groups)

# Линейные модели

group\_1 <- data\_1\_2[groups == 1,]

group\_2 <- data\_1\_2[groups == 2,]

group\_3 <- data\_1\_2[groups == 3,]

group\_1 <- as.data.frame(group\_1)

group\_2 <- as.data.frame(group\_2)

group\_3 <- as.data.frame(group\_3)

## Получим таблицы по кластерам с наименование регионов.

# В data\_1\_2 нет столбца с наименование регионов.

# Поэтому берем строки по группам из data\_1 (Строки data\_1 и data\_1\_2 совпадают!)

data\_1\_group\_1 <- data\_1[groups == 1,]

data\_1\_group\_2 <- data\_1[groups == 2,]

data\_1\_group\_3 <- data\_1[groups == 3,]

## Посмотреть таблицы по группам регионов

View(data\_1\_group\_1)

View(data\_1\_group\_2)

View(data\_1\_group\_3)

## Сопоставление средних значений переменных по кластерам

summary(data\_1\_group\_1)

summary(data\_1\_group\_2)

summary(data\_1\_group\_3)

head(group\_1)

head(group\_2)

head(group\_3)

# Матрица парных коэффициентов корелляции

cor(group\_1)

# Линейная модель

lm\_1 <- lm(expen\_19 ~ gdp\_c\_18+lab\_inc\_19+pov\_19, data=group\_1)

summary(lm\_1)

plot(lm\_1)

##!График меняем, перейдя в нижнюю консоль и нажимая ENTER

# Матрица парных коэффициентов корреляции

cor(group\_2)

lm\_2 <- lm(expen\_19 ~ lab\_inc\_19+cpi\_19+pov\_19, data=group\_2)

summary(lm\_2)

plot(lm\_2)

lm\_2 <- lm(expen\_19 ~ lab\_inc\_19+pov\_19, data=group\_2)

summary(lm\_2)

# Пострегрессионный индексный анализ

## Создание таблицы для пострегрессионого индексного анализа

# Найти средние значение по каждому столбцу исходных данных

lm\_1\_model <- as.data.frame(t(as.data.frame(lapply(lm\_1$model, mean))))

lm\_2\_model <- as.data.frame(t(as.data.frame(lapply(lm\_2$model, mean))))

# Задать название единственному столбцу (lm\_1 и lm\_2, соответственно)

names(lm\_1\_model) <- c('lm\_1')

names(lm\_2\_model) <- c('lm\_2')

# Установить названия строк в виде --<переменная>

rownames(lm\_1\_model) <- lapply(rownames(lm\_1\_model), function(x) sprintf('--%s',x))

rownames(lm\_2\_model) <- lapply(rownames(lm\_2\_model), function(x) sprintf('--%s',x))

# Отобразить

lm\_1\_model

lm\_2\_model

# Занести коэффициенты как таблицу данных

lm\_1\_coeff <- as.data.frame(lm\_1$coefficients)

lm\_2\_coeff <- as.data.frame(lm\_2$coefficients)

# Задать название единственному столбцу (lm\_1 и lm\_2, соответственно)

names(lm\_1\_coeff) <- c('lm\_1')

names(lm\_2\_coeff) <- c('lm\_2')

# Отобразить

lm\_1\_coeff

lm\_2\_coeff

# Соединить всё вместе в одну таблицу lm\_1\_2\_table

lm\_1\_2\_table = merge(

rbind(lm\_1\_model, lm\_1\_coeff),

rbind(lm\_2\_model, lm\_2\_coeff),

by=0,sort=T,all=T

)

# Задать названия колонкам

names(lm\_1\_2\_table) <- c('--', 'lm\_1', 'lm\_2')

lm\_1\_2\_table