**deКОДЫ ЗАМЕНЫ ПРОПУЩЕННЫХ ДАННЫХ (ФАЙЛ Ex\_4)**

library(ggplot2)

# Создать файл исходных данных reg\_1

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

reg\_1

View(reg\_1)

# Создать систему координат

g <- ggplot(data = reg\_1, mapping = aes(x = cpi\_19, y = wage\_19))

g

# Построить диаграмму рассеяния

g <- ggplot(data = reg\_1,

mapping = aes(x = wage\_19,

y = expen\_19))

g+geom\_point()

# Построить диаграмму рассеяния с маркировкой по категориальной переменной

g <- ggplot(data = reg\_1,

mapping = aes(x = wage\_19,

y = expen\_19, color=tip\_1))

g+geom\_point()

# Группы выделяются размером точек (1-й вариант)

g <- ggplot(data = reg\_1,

mapping = aes(x = wage\_19,

y = expen\_19, size=tip\_1))

# Группы выделяются размером точек (2-й вариант)

g+geom\_point()

g <- ggplot(data = reg\_1,

mapping = aes(x = wage\_19,

y = expen\_19, size=gdp\_c\_18))

g+geom\_point()

# Диаграмма рассеяния (цвет/ увеличение размера точек)

g <- ggplot(data = reg\_1,

mapping = aes(x = wage\_19,

y = expen\_19, color=tip\_1))

g+geom\_point(size=4)

# Сглаживание

g <- ggplot(data = reg\_1,

mapping = aes(x = wage\_19, y=expen\_19))

g + geom\_smooth()

# Сглаживание для каждой группы регионов (по категориальной переменной typ\_1)

g <- ggplot(data = reg\_1)+

geom\_smooth(mapping = aes(x = wage\_19, y=expen\_19, linetype=tip\_1, color=tip\_1), se=FALSE, size=2)

g

# Зависимость уровня инфляции (ИПЦ) от темпов экономического роста (gdp\_g\_18) по группам регионов,

# различных по уровню экономического развития

g <- ggplot(data = reg\_1)+

geom\_smooth(mapping = aes(x = gdp\_g\_18, y=cpi\_19, linetype=tip\_1, color=tip\_1), se=FALSE, size=2)

g

# Сглаживание\_метод "lm"

g <- ggplot(data = reg\_1,

mapping = aes(x = wage\_19,

y=expen\_19))

g + geom\_smooth(method = 'lm', col="purple")+geom\_point()

# Сглаживание\_метод "gam"

g <- ggplot(data = reg\_1,

mapping = aes(x = wage\_19,

y=expen\_19))

g + geom\_smooth(method = 'gam', formula = y ~ s(x, bs = "cs", k = 4), col="purple")+geom\_point()

# Сглаживание\_метод "loess"

g <- ggplot(data = reg\_1,

mapping = aes(x = wage\_19,

y = expen\_19,color="purple"))

g + geom\_point(color = "blue") +

geom\_smooth(method = "loess")

# Анализ зависимости по федеральным округам

# Добавляем столбец "FO"

reg\_2 <- read.csv2("FedOkr.csv")

reg\_2

View(reg\_2)

reg\_1$FO<-reg\_2$FO

View(reg\_1)

g <- ggplot(data = reg\_1,

mapping = aes(x = wage\_19,

y = expen\_19,

color = FO,

fill = FO))

g + geom\_point() +

geom\_smooth(method = "lm") +

labs(x = "Среднемесячная начисленная заработная плата работников организаций, руб.",

y = "Среднедушевые денежные расходы, руб.",

title = "Зависимость среднедушевых денежных расходов от уровня заработной платы работников организаций",

subtitle = "По федеральным округам, 2019",

caption = "Источник: Росстат")

# Анализ зависимости по группам регионов, различающихся по уровню экономического развития

g <- ggplot(data = reg\_1,

mapping = aes(x = wage\_19,

y = expen\_19,

color = tip\_1,

fill = tip\_1))

g + geom\_point() +

geom\_smooth(method = "lm") +

labs(x = "Среднемесячная начисленная заработная плата работников организаций, руб.",

y = "Среднедушевые денежные расходы, руб.",

title = "Зависимость среднедушевых денежных расходов от уровня заработной платы работников организаций",

subtitle = "По группам регионов с различным уровнем экономического развития, 2019",

caption = "Источник: Росстат")

# РАБОТА С ПРОПУСКАМИ ДАННЫХ

## Установить пакет VIM

library(VIM)

### Узнать, в каких столбцах (по каким переменным)имеются пропущенные значения

names(which(sapply(reg\_1, anyNA)))

## График распределения пропущенных значений

miss\_plot<-aggr(reg\_1[,c('cost\_hous\_19','prop\_helth\_20')])

miss\_plot

?aggr()

## Для того, чтобы уместились названия переменных на графике, создадим объект reg\_1\_a,

## содержащий эти переменные, придав им короткие названия

reg\_1\_a <- reg\_1[,c('cost\_hous\_19','prop\_helth\_20')]

names(reg\_1\_a) <- c('c\_h\_19', 'p\_h\_20')

aggr(reg\_1\_a)

# Построить график совместного распределения пропущенных значений

?marginplot

marginplot(reg\_1[,c('cost\_hous\_19','prop\_helth\_20')])

**#### РАБОТА С ПРОПУЩЕННЫМИ ДАННЫМИ**

## Установить пакет DMwR

## подключить библиотеку DMwR2

library(DMwR2)

## Установить номера строк с числом пропущенных значений больше 1 %

manyNAs(reg\_1,0.01)

### Способ 1- Удалить строки, в которых число пропущенных значений больше 1 %

reg\_1\_b <- reg\_1[-manyNAs(reg\_1,0.01),]

View(reg\_1\_b)

? manyNAs

### Способ 2 - Замена пропущенных значений путем вменения значения медианы (или средней)

## Установить пакет caret

library(caret)

## Установить число пропусков по переменным cost\_hous\_19 и prop\_helth\_20

sum(is.na(reg\_1$cost\_hous\_19))

sum(is.na(reg\_1$prop\_helth\_20))

## Вменяем значение медианы пропускам (по обоим столбцам)

proc<-preProcess(reg\_1[,c('cost\_hous\_19','prop\_helth\_20')],method='medianImpute')

reg\_1\_c<-reg\_1

reg\_1\_c[,c('cost\_hous\_19','prop\_helth\_20')]<- predict(proc,reg\_1\_c[,c('cost\_hous\_19','prop\_helth\_20')])

## Установить число пропусков по переменным cost\_hous\_19 и prop\_helth\_20 после вменения медианы

sum(is.na(reg\_1\_c$cost\_hous\_19))

sum(is.na(reg\_1\_c$prop\_helth\_20))

### Способ 3 - Замена пропущенных значений методом кластерного анализа (k ближайших соседей)

## загрузить пакет RANN

library(caret)

proc\_1<-preProcess(reg\_1[,c('cost\_hous\_19','prop\_helth\_20')],method='knnImpute')

reg\_1\_d<-reg\_1

reg\_1\_d[,c('cost\_hous\_19','prop\_helth\_20')]<- predict(proc\_1,reg\_1\_d[,c('cost\_hous\_19','prop\_helth\_20')])

## Сравнить описательную статистику по переменным с вмененными значениями

summary (reg\_1\_c[,c('cost\_hous\_19','prop\_helth\_20')])

summary (reg\_1\_d[,c('cost\_hous\_19','prop\_helth\_20')])

# Выполнить действие, обратное стандартизации

mean(reg\_1$cost\_hous\_19,na.rm=TRUE)

sd(reg\_1$cost\_hous\_19,na.rm=TRUE)

reg\_1\_d\_c\_h\_19 <- reg\_1\_d$cost\_hous\_19\*20.28058+52.11807

summary(reg\_1\_d\_c\_h\_19)

head(reg\_1\_d\_c\_h\_19)

## Построить гистограммы

hist(reg\_1\_c$cost\_hous\_19)

hist(reg\_1\_d\_c\_h\_19)