**2. Exploratory Graphs**

head(pollution) # Просмотр первых шести строк таблицы pollution

summary(pollution$pm25) # Анализ переменной pm25 из датафрейма pollution

quantile(ppm) # Вычисляет квантильные значения для переменной ppm (pm25)

boxplot(ppm, col = "blue") # Диаграмма размаха (boxplot) для переменной ppm

abline(h=12) # Добавляет горизонтальную линию на уровне y = 12

hist(ppm, col = "green") # Построение гистограммы для переменной ppm

rug(ppm) # Добавляет одномерное распределение данных в виде штрихов

low # Выводит значения переменной low (низкие значения ppm)

high # Выводит значения переменной high (высокие значения ppm)

hist(ppm, col="green", breaks = 100) # Гистограмма с 100 столбцами

rug(ppm) # Добавляет "ковер" под гистограмму

hist(ppm, col = "green") # Гистограмма с зелеными столбцами

abline(v=12, lwd=2) # Вертикальная линия на x = 12 толщиной 2

abline(v=median(ppm), col="magenta", lwd=4) # Вертикальная линия на уровне медианы ppm

names(pollution) # Выводит имена столбцов датафрейма pollution

reg <- table(pollution$region) # Создает таблицу частот для переменной region

reg # Выводит таблицу частот для переменной region

barplot(reg, col="wheat", main="Number of Counties in Each Region") # Столбчатая диаграмма для данных в reg

boxplot(pm25~region, pollution, col="red") # Диаграмма размаха для pm25 по регионам

par(mfrow=c(2,1),mar=c(4,4,2,1)) # Настройка графической области на 2 строки и 1 столбец

east <- subset(pollution, region == "east") # Создает подмножество данных для региона "east"

head(east) # Выводит первые шесть строк подмножества east

hist(east$pm25, col = "green") # Гистограмма для pm25 в регионе "east"

hist(subset(pollution, region == "west")$pm25, col = "green") # Гистограмма для pm25 в регионе "west"

with(pollution, plot(latitude, pm25)) # Scatter plot для latitude и pm25

abline(h=12, lwd=2, lty=2) # Горизонтальная пунктирная линия на уровне y = 12

plot(pollution$latitude, ppm, col=pollution$region) # Scatter plot для latitude и ppm с цветом по регионам

abline(h=12, lwd=2, lty=2) # Горизонтальная пунктирная линия на уровне y = 12

par(mfrow = c(1, 2), mar = c(5, 4, 2, 1)) # Настройка графической области на 1 строку и 2 столбца

west <- subset(pollution, region == "west") # Создает подмножество данных для региона "west"

plot(west$latitude, west$pm25, main = "West") # Scatter plot для latitude и pm25 в регионе "west"

plot(east$latitude, east$pm25, main = "East") # Scatter plot для latitude и pm25 в регионе "east"

**3. Graphics Devices in R**

with(faithful, plot(eruptions, waiting)) # Scatter plot для eruptions и waiting

title(main = "Old Faithful Geyser data") # Добавляет заголовок к графику

dev.cur() # Выводит текущее графическое устройство

pdf(file="myplot.pdf") # Открывает PDF-устройство для сохранения графика

with(faithful, plot(eruptions, waiting)) # Scatter plot в PDF-файле

title(main = "Old Faithful Geyser data") # Добавляет заголовок к графику в PDF

dev.cur() # Проверяет текущее графическое устройство (должно быть PDF)

dev.off() # Закрывает PDF-устройство и сохраняет файл

dev.cur() # Проверяет текущее графическое устройство (основное)

with(faithful, plot(eruptions, waiting)) # Снова строит scatter plot на основном устройстве

title(main = "Old Faithful Geyser data") # Добавляет заголовок к графику

dev.copy(png, file = "geyserplot.png") # Копирует текущий график в PNG-файл

dev.off() # Закрывает PNG-устройство и сохраняет файл

**4. Plotting systems**

head(cars) # Выводит первые шесть строк датафрейма cars

with(cars, plot(speed, dist)) # Scatter plot для speed и dist

text(mean(cars$speed), max(cars$dist), "SWIRL rules!") # Добавляет текст на график

head(state) # Выводит первые шесть строк датафрейма state

table(state$region) # Создает таблицу частот для переменной region

xyplot(Life.Exp ~ Income | region, data = state, layout = c(4, 1)) # Scatter plot с разбивкой по регионам

xyplot(Life.Exp ~ Income | region, data = state, layout = c(2,2)) # Scatter plot с разбивкой по регионам

head(mpg) # Выводит первые шесть строк датафрейма mpg

dim(mpg) # Выводит размерность датафрейма mpg

table(mpg$model) # Создает таблицу частот для переменной model

qplot(displ, hwy, data = mpg) # Scatter plot для displ и hwy с помощью qplot

**5. Base plotting system**

head(airquality) # Выводит первые шесть строк датафрейма airquality

range(airquality$Ozone, na.rm = TRUE) # Выводит диапазон значений переменной Ozone

hist(airquality$Ozone) # Гистограмма для переменной Ozone

table(airquality$Month) # Создает таблицу частот для переменной Month

boxplot(Ozone~Month, airquality) # Диаграмма размаха для Ozone по месяцам

boxplot(Ozone~Month, airquality, xlab = "Month", ylab = "Ozone (ppb)", col.axis = "blue", col.lab = "red") # Диаграмма размаха с настройками

title(main="Ozone and Wind in New York City") # Добавляет заголовок к графику

with(airquality, plot(Wind, Ozone)) # Scatter plot для Wind и Ozone

title(main="Ozone and Wind in New York City") # Добавляет заголовок к графику

length(par()) # Выводит количество параметров графической системы

names(par()) # Выводит имена всех параметров графической системы

par()$pin # Выводит размеры текущего графика в дюймах

par("fg") # Выводит текущий цвет переднего плана

par("pch") # Выводит текущий символ для точек

par("lty") # Выводит текущий тип линии

plot(airquality$Wind, airquality$Ozone, type = "n") # Создает пустой график с осями

title(main = "Wind and Ozone in NYC") # Добавляет заголовок к пустому графику

may <- subset(airquality, Month == 5) # Создает подмножество данных для мая

points(may$Wind,may$Ozone,col="blue",pch=17) # Добавляет точки для мая на график

notmay <- subset(airquality, Month != 5) # Создает подмножество данных для остальных месяцев

points(notmay$Wind,notmay$Ozone,col="red",pch=8) # Добавляет точки для остальных месяцев

legend("topright", pch = c(17, 8), col = c("blue", "red"), legend = c("May", "Other Months")) # Добавляет легенду

abline(v = median(airquality$Wind), lty=2, lwd=2) # Добавляет вертикальную пунктирную линию

par(mfrow = c(1, 2)) # Настраивает графическую область на 1 строку и 2 столбца

plot(airquality$Wind, airquality$Ozone, main = "Ozone and Wind") # Scatter plot для Wind и Ozone

plot(airquality$Ozone, airquality$Solar.R, main = "Ozone and Solar Radiation") # Scatter plot для Ozone и Solar.R

par(mfrow = c(1, 3), mar = c(4, 4, 2, 1), oma = c(0, 0, 2, 0)) # Настраивает графическую область на 1 строку и 3 столбца

plot(airquality$Wind, airquality$Ozone, main = "Ozone and Wind") # Scatter plot для Wind и Ozone

plot(airquality$Solar.R, airquality$Ozone, main = "Ozone and Solar Radiation") # Scatter plot для Solar.R и Ozone

plot(airquality$Temp, airquality$Ozone, main = "Ozone and Temperature") # Scatter plot для Temp и Ozone

mtext("Ozone and Weather in New York City", outer = TRUE) # Добавляет общий заголовок

**6. Lattice Plotting System**

> head(airquality) # Просмотр первых строк данных airquality

> xyplot(Ozone ~ Wind, data = airquality) # Построение графика Ozone от Wind с использованием Lattice

> xyplot(Ozone ~ Wind, data = airquality, pch=8, col="red", main="Big Apple Data") # График с настройками символов, цвета и заголовка

> xyplot(Ozone ~ Wind | as.factor(Month), data = airquality, layout=c(5,1)) # График с разбивкой по месяцам

> xyplot(Ozone ~ Wind | Month, data = airquality, layout=c(5,1)) # Аналогичный график с автоматическим преобразованием Month в фактор

> p <- xyplot(Ozone~Wind,data=airquality) # Сохранение графика в переменную p

> p # Отображение графика из переменной p

> names(p) # Просмотр имен элементов объекта p

> p[["formula"]] # Получение формулы, используемой в графике

> p[["x.limits"]] # Получение пределов оси X

> table(f) # Создание таблицы для фактора f

> xyplot(y ~ x | f, layout = c(2, 1)) # График с разбивкой по фактору f

> myedit("plot1.R") # Редактирование файла plot1.R

> source(pathtofile("plot1.R"),local=TRUE) # Запуск кода из файла plot1.R

> myedit("plot2.R") # Редактирование файла plot2.R

> source(pathtofile("plot2.R"),local=TRUE) # Запуск кода из файла plot2.R

> str(diamonds) # Просмотр структуры данных diamonds

> table(diamonds$color) # Таблица частот для переменной color

> table(diamonds$color,diamonds$cut) # Таблица сопряженности color и cut

> myedit("myLabels.R") # Редактирование файла myLabels.R

> source(pathtofile("myLabels.R"),local=TRUE) # Запуск кода из файла myLabels.R

>xyplot(price~carat|color\*cut,data=diamonds,strip=FALSE,pch=20,xlab=myxlab,ylab=myylab,main=mymain) # График с настройками осей и заголовка

>xyplot(price~carat|color\*cut,data=diamonds,pch=20,xlab=myxlab,ylab=myylab,main=mymain) # График с настройками символов и осей

**7. Working with Colors**

> sample(colors(),10) # Выбор 10 случайных цветов из списка

> pal <- colorRamp(c("red","blue")) # Создание палитры от красного к синему

> pal(0) # Получение цвета для значения 0

> pal(1) # Получение цвета для значения 1

> pal(seq(0,1,len=6)) # Получение цветов для последовательности значений

> p1 <- colorRampPalette(c("red","blue")) # Создание палитры с градиентом от красного к синему

> p1(2) # Получение двух цветов из палитры

> p1(6) # Получение шести цветов из палитры

> 0xcc # Преобразование шестнадцатеричного числа в десятичное

> p2 <- colorRampPalette(c("red","yellow")) # Создание палитры с градиентом от красного к желтому

> p2(2) # Получение двух цветов из палитры

> p2(10) # Получение десяти цветов из палитры

> showMe(p1(20)) # Отображение 20 цветов из палитры p1

> showMe(p2(20)) # Отображение 20 цветов из палитры p2

> showMe(p2(2)) # Отображение 2 цветов из палитры p2

> p1 # Просмотр функции палитры p1

> ?rgb # Справка по функции rgb

> p3 <- colorRampPalette(c("blue","green"),alpha=.5) # Создание палитры с прозрачностью

> p3(5) # Получение пяти цветов из палитры p3

> plot(x,y,pch=19,col=rgb(0,.5,.5)) # Построение графика с цветом rgb

> plot(x,y,pch=19,col=rgb(0,.5,.5,.3)) # Построение графика с прозрачным цветом

> cols <- brewer.pal(3, "BuGn") # Получение цветов из палитры BuGn

> showMe(cols) # Отображение цветов

> pal <- colorRampPalette(cols) # Создание палитры на основе цветов

> showMe(pal(20)) # Отображение 20 цветов из палитры

> image(volcano, col = pal(20)) # Отображение изображения с использованием палитры

> image(volcano, col = p1(20)) # Отображение изображения с использованием палитры p1

**8: GGPlot2 Part1**

> str(mpg) # Просмотр структуры данных mpg

> qplot(displ, hwy, data = mpg) # Построение графика с использованием qplot

> qplot(displ, hwy, data = mpg, color = drv) # График с цветом по переменной drv

> qplot(displ, hwy, data = mpg, color=drv, geom = c("point", "smooth")) # График с точками и сглаживанием

> qplot(y=hwy, data = mpg, color = drv) # График распределения hwy с цветом по drv

> qplot(drv,hwy,data=mpg,geom="boxplot") # Построение boxplot

> qplot(drv,hwy,data=mpg,geom="boxplot",color=manufacturer) # Boxplot с цветом по производителю

> qplot(hwy, data = mpg, fill = drv) # Гистограмма с заполнением по drv

> qplot(displ, hwy, data = mpg, facets = . ~ drv) # График с разбивкой по drv

> qplot(hwy, data = mpg, facets = drv ~ ., binwidth = 2) # Гистограмма с разбивкой по drv

**9: GGPlot2 Part2**

> g <- ggplot(mpg,aes(displ,hwy)) # Создание объекта ggplot

> g+geom\_point() # Добавление точек к графику

> g+geom\_point()+geom\_smooth() # Добавление сглаживания к графику

> g+geom\_point()+geom\_smooth(method="lm") # Добавление линейной модели

> g+geom\_point()+geom\_smooth(method="lm") + facet\_grid(.~drv) # Добавление разбивки по drv

> g+geom\_point()+geom\_smooth(method="lm") + facet\_grid(.~drv)+ ggtitle("Swirl Rules!") # Добавление заголовка

> g+geom\_point(color="pink",size=4,alpha=1/2) # Настройка цвета, размера и прозрачности точек

> g + geom\_point(aes(color = drv), size = 4, alpha = 1/2) # Настройка цвета по drv

> g + geom\_point(aes(color = drv)) + labs(title="Swirl Rules!") + labs(x="Displacement", y="Hwy Mileage") # Добавление заголовка и подписей осей

> g + geom\_point(aes(color = drv),size=2,alpha=1/2) + geom\_smooth(size=4,linetype=3,method="lm",se=FALSE) # Настройка стиля линии сглаживания

> g + geom\_point(aes(color = drv)) + theme\_bw(base\_family="Times") # Изменение темы графика

> plot(myx, myy, type = "l", ylim = c(-3,3)) # Построение линейного графика

> g <- ggplot(testdat, aes(x = myx, y = myy)) # Создание объекта ggplot

> g + geom\_line() # Добавление линии к графику

> g + geom\_line() + ylim(-3,3) # Установка пределов оси Y

> g + geom\_line() + coord\_cartesian(ylim=c(-3,3)) # Установка пределов оси Y с использованием coord\_cartesian

> g <- ggplot(mpg,aes(x=displ,y=hwy,color=factor(year))) # Создание графика с цветом по году

> g + geom\_point() # Добавление точек

> g + geom\_point() + facet\_grid(drv~cyl,margins=TRUE) # Добавление разбивки по drv и cyl

> g + geom\_point() + facet\_grid(drv~cyl,margins=TRUE)+geom\_smooth(method="lm",size=2,se=FALSE,color="black") # Добавление линейной модели

> g + geom\_point() + facet\_grid(drv~cyl,margins=TRUE)+geom\_smooth(method="lm",size=2,se=FALSE,color="black")+labs(x="Displacement",y="Highway Mileage",title="Swirl Rules!") # Добавление подписей и заголовка

**10: GGPlot2 Extras**

> qplot(price,data=diamonds) # Гистограмма цены

> qplot(price,data=diamonds,binwidth=18497/30) # Гистограмма с настройкой ширины бинов

> qplot(price,data=diamonds,binwidth=18497/30,fill=cut) # Гистограмма с заполнением по cut

> qplot(price,data=diamonds,geom="density") # График плотности

> qplot(price,data=diamonds,geom="density",color=cut) # График плотности с цветом по cut

> qplot(carat,price,data=diamonds) # График carat от price

> qplot(carat,price,data=diamonds, shape=cut) # График с формой точек по cut

> qplot(carat,price,data=diamonds, color=cut) # График с цветом по cut

> qplot(carat,price,data=diamonds, color=cut) + geom\_smooth(method="lm") # Добавление линейной модели

> qplot(carat,price,data=diamonds, color=cut, facets=.~cut) + geom\_smooth(method="lm") # График с разбивкой по cut

> g <- ggplot(diamonds,aes(depth,price)) # Создание объекта ggplot

> g+geom\_point(alpha=1/3) # Добавление точек с прозрачностью

> cutpoints <- quantile(diamonds$carat,seq(0,1,length=4),na.rm=TRUE) # Вычисление квантилей

> diamonds$car2 <- cut(diamonds$carat,cutpoints); stageVariable("diamonds$car2",diamonds$car2) # Создание новой переменной car2

> g <- ggplot(diamonds,aes(depth,price)) # Создание объекта ggplot

> g+geom\_point(alpha=1/3)+facet\_grid(cut~car2) # График с разбивкой по cut и car2

>g+geom\_point(alpha=1/3)+facet\_grid(cut~car2)+geom\_smooth(method="lm",size=3,color="pink") # Добавление линейной модели

> ggplot(diamonds,aes(carat,price))+geom\_boxplot()+facet\_grid(.~cut) # Boxplot с разбивкой по cut