Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Уральский федеральный университет имени первого Президента России Б.Н. Ельцина» (УрФУ)

Институт радиоэлектроники и информационных технологий – РтФ

Школа бакалавриата

Оценка работы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Члены комиссии

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

РАЗРАБОТКА СИСТЕМЫ ПРОГНОЗИРОВАНИЯ СТОИМОСТИ ДОМА

Пояснительная записка к курсовому проекту по модулю

«Методы анализа Big Data»

Преподаватель Мирвода С.Г.

Студенты Куштанов Д.И.

Изотов И.Н.

Группа РИ-450004

Екатеринбург

2019

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_Toc534992104)

[1 ОПИСАНИЕ НАБОРА ДАННЫХ 4](#_Toc534992105)

[2 ТРЕНИРОВКА ЛИНЕЙНОЙ МОДЕЛИ 8](#_Toc534992106)

[3 ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ 13](#_Toc534992107)

[4 ИНСТРУКЦИЯ ПОЛЬЗОВАТЕЛЯ 14](#_Toc534992108)

[ЗАКЛЮЧНИЕ 15](#_Toc534992109)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 16](#_Toc534992110)

[ПРИЛОЖЕНИЕ А – КОД RMARDOWN 17](#_Toc534992111)

[ПРИЛОЖЕНИЕ Б – КОД ФУННКЦИИ R 19](#_Toc534992112)

[ПРИЛОЖЕНИЕ В – КОД ОСНОВНОЙ ФУНКЦИИ PYTHON 20](#_Toc534992113)

# ВВЕДЕНИЕ

В текущее время объемы информации растут по экспоненте. Для того, чтобы быстрее реагировать на изменения рынка, получить конкурентные преимущества и повысить эффективность производства, нужно получить, обработать и проанализировать огромное количество данных. Для этого используются специальные программные средства, и одним из таких решений может являться язык программирования для статистической обработки данных и работы с графикой R.

Целью данной работы является разработка системы прогнозирования стоимости домов, с использованием открытого набора данных.

Гипотезой служит предположение о том, что технические и геолокационные характеристики дома вносят влияние на формирование его стоимости.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. изучить набор данных;
2. провести анализ данных;
3. натренировать линейную модель;
4. написать пользовательское приложение.

# 1 ОПИСАНИЕ НАБОРА ДАННЫХ

Для данный работы использовался набор данных стоимостей домов США в определенном регионе [1]. Пример данных изображен на рисунке 1.

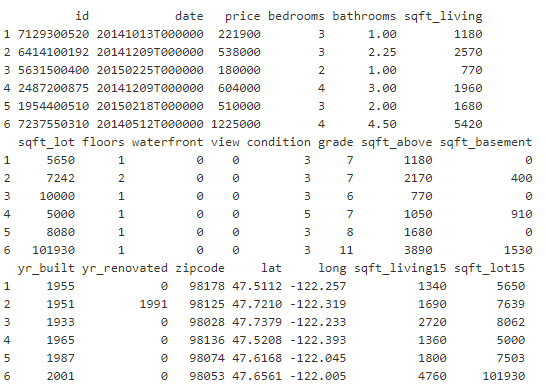


Рисунок 1 – Пример данных

В данном наборе данных колонки имеют следующие значения:

* id – номер дома;
* date – дата продажи дома;
* price – цена;
* bedrooms – число спален;
* bathrooms – число ванных комнат;
* sqft\_living – площадь дома;
* sqft\_lot – площадь участка;
* floors – число этажей;
* waterfront – есть ли выход на набережную;
* view – был ли осмотрен;
* condition – состояние дома;
* grade – оценка дома основанная на King County;
* sqft\_above – площадь дома отдельно от подвала;
* sqft\_basement – площадь подвала;
* yr\_built – год постройи;
* yr\_renovated – год ремонта дома;
* zipcode – почтовый zip код;
* lat – координата широты;
* long – координата долготы;
* sqft\_living15 – жилая площадь на 2015 год;
* sqft\_lot15lot – площадь участка на 2015 год.

На рисунках ниже (рисунки 2, 3, 4, 5) отображены некоторые зависимости атрибутов от цены.

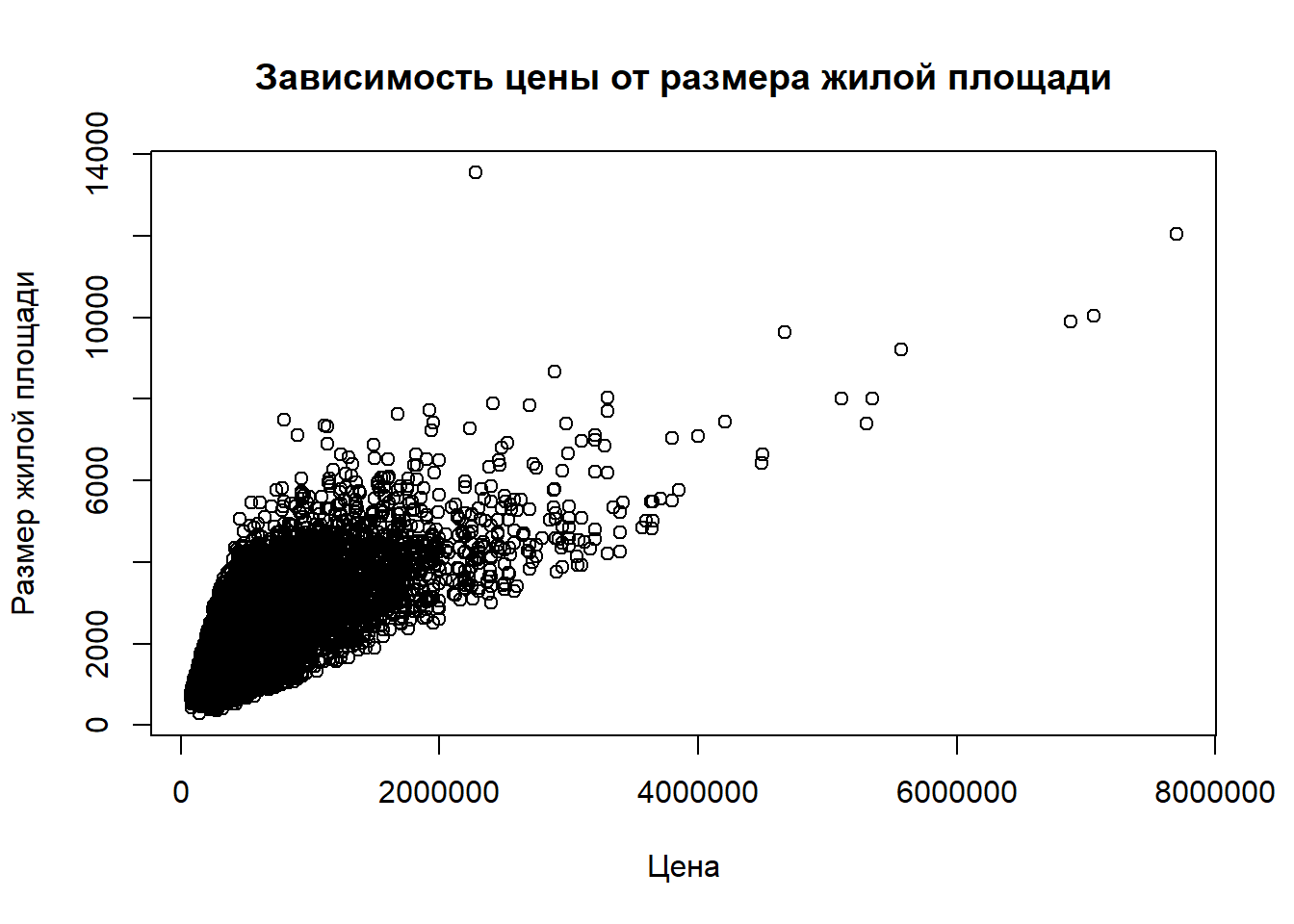


Рисунок 2 – Зависимость цены от размера жилой площади

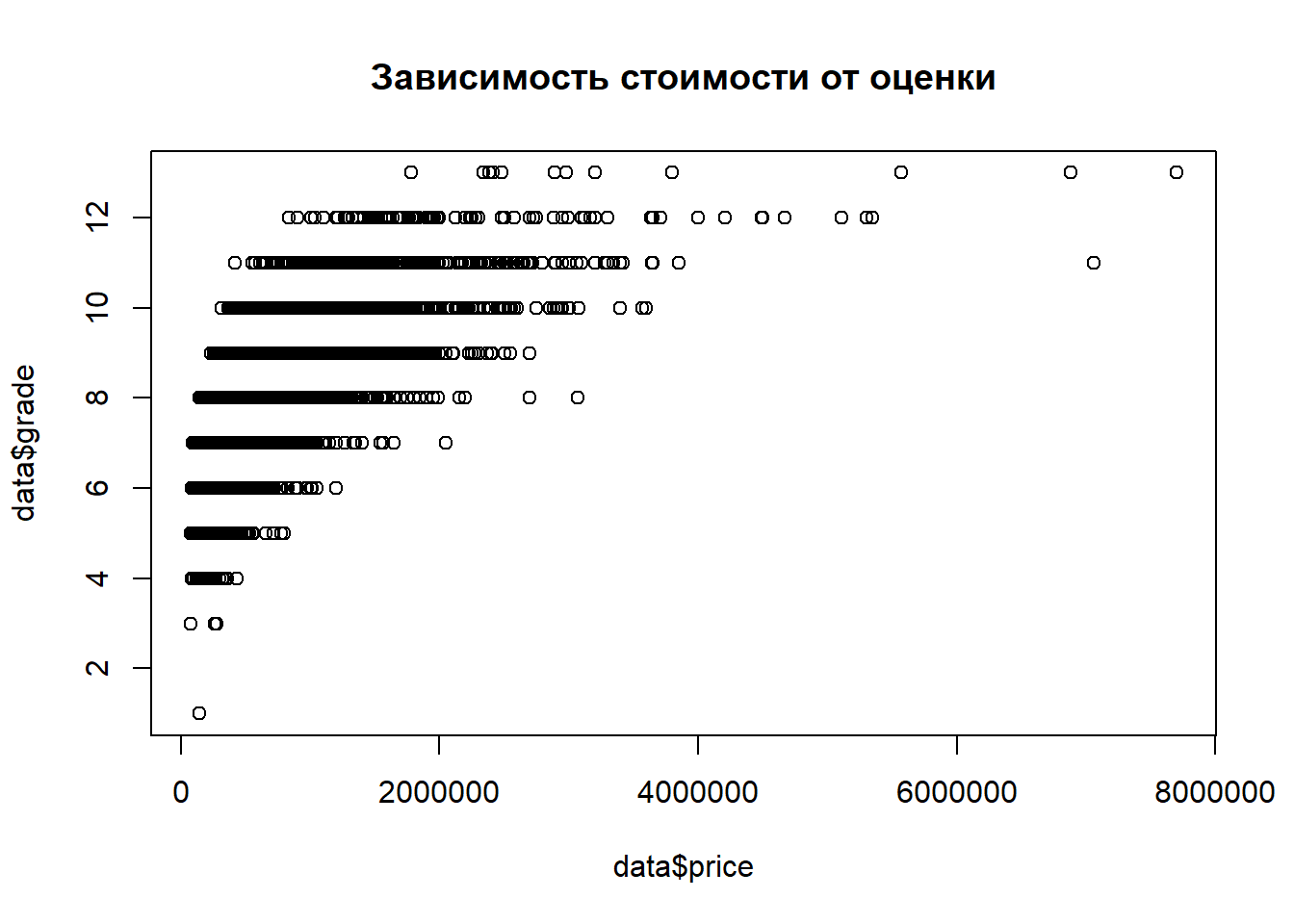


Рисунок 3 – Зависимость цены от оценки

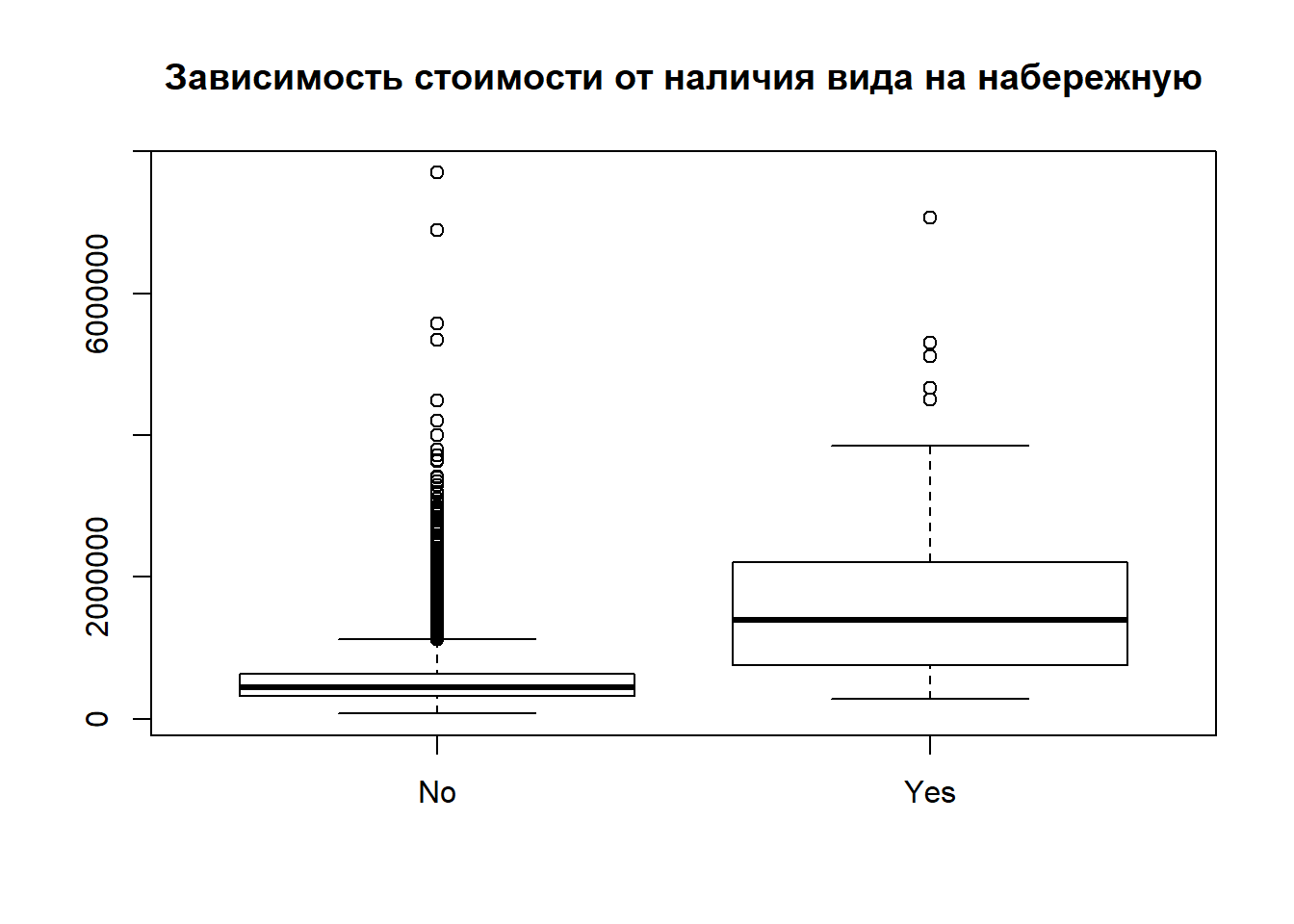


Рисунок 4 – Зависимость цены от наличия вида на набережную.

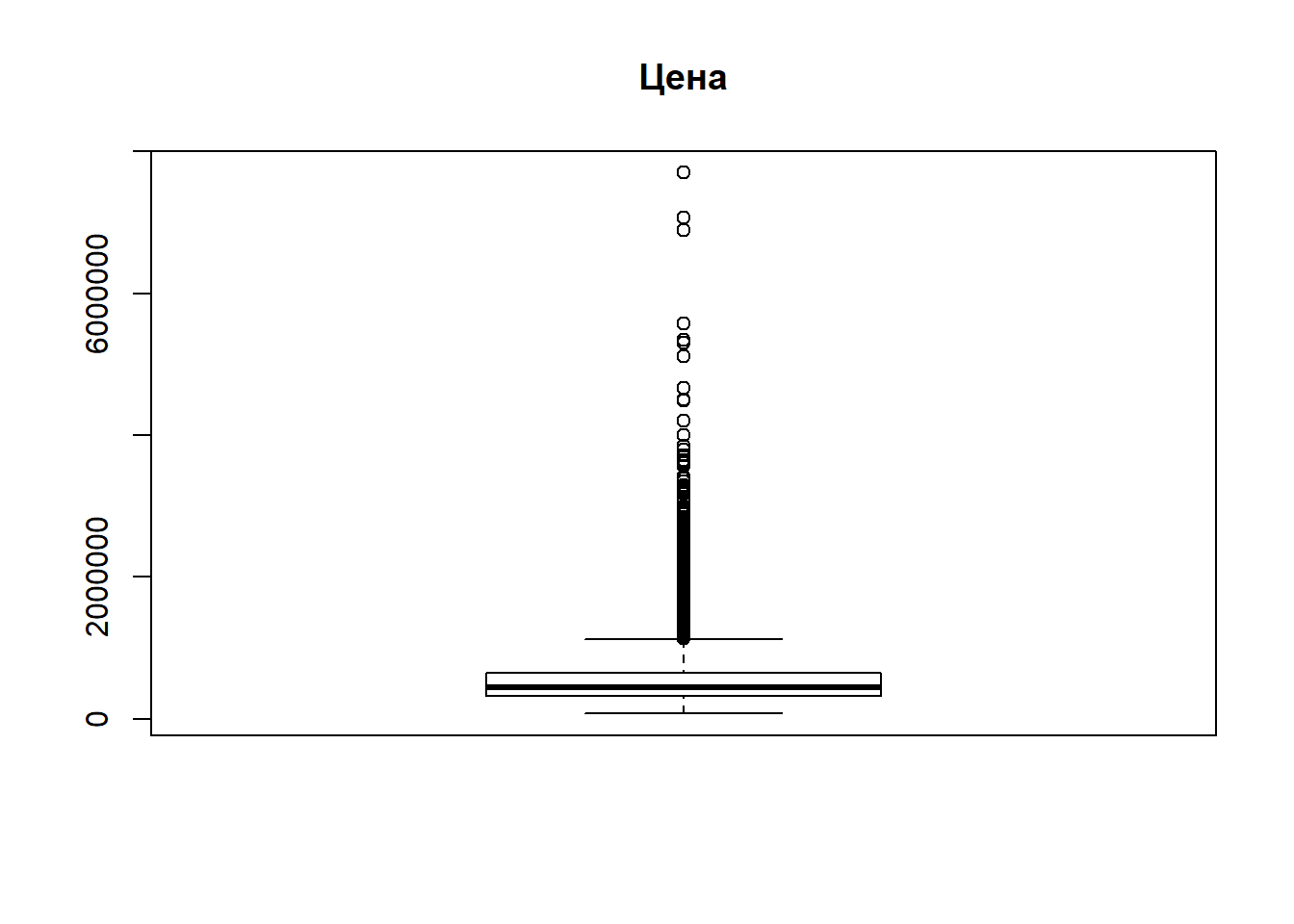


Рисунок 5 – График стоимости

# 2 ТРЕНИРОВКА ЛИНЕЙНОЙ МОДЕЛИ

Для тренировки модели была проведена чистка данных (результат на рисунке 6) и построена линейная модель на основе всех данных, чтобы определить наиболее значимые параметры (рисунок 7).



Рисунок 6 – Очищенная зависимость стоимости от вида на набережную

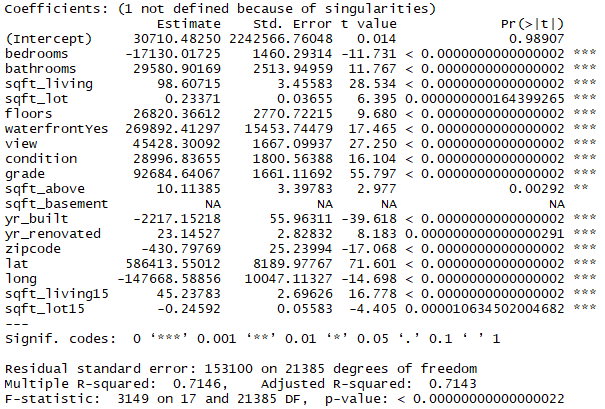


Рисунок 7 – Зависимость влияния параметров на цену

Для построения более точной модели были убраны наименее значимые столбцы и создан дополнительный avgZipCodePrice являющийся средней ценой по почтовому индексу. Результаты показаны на рисунках 8, 9, 10.

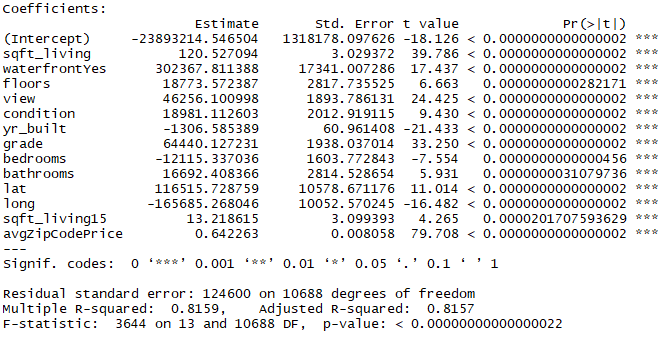


Рисунок 8 – Коэффициенты новой модели

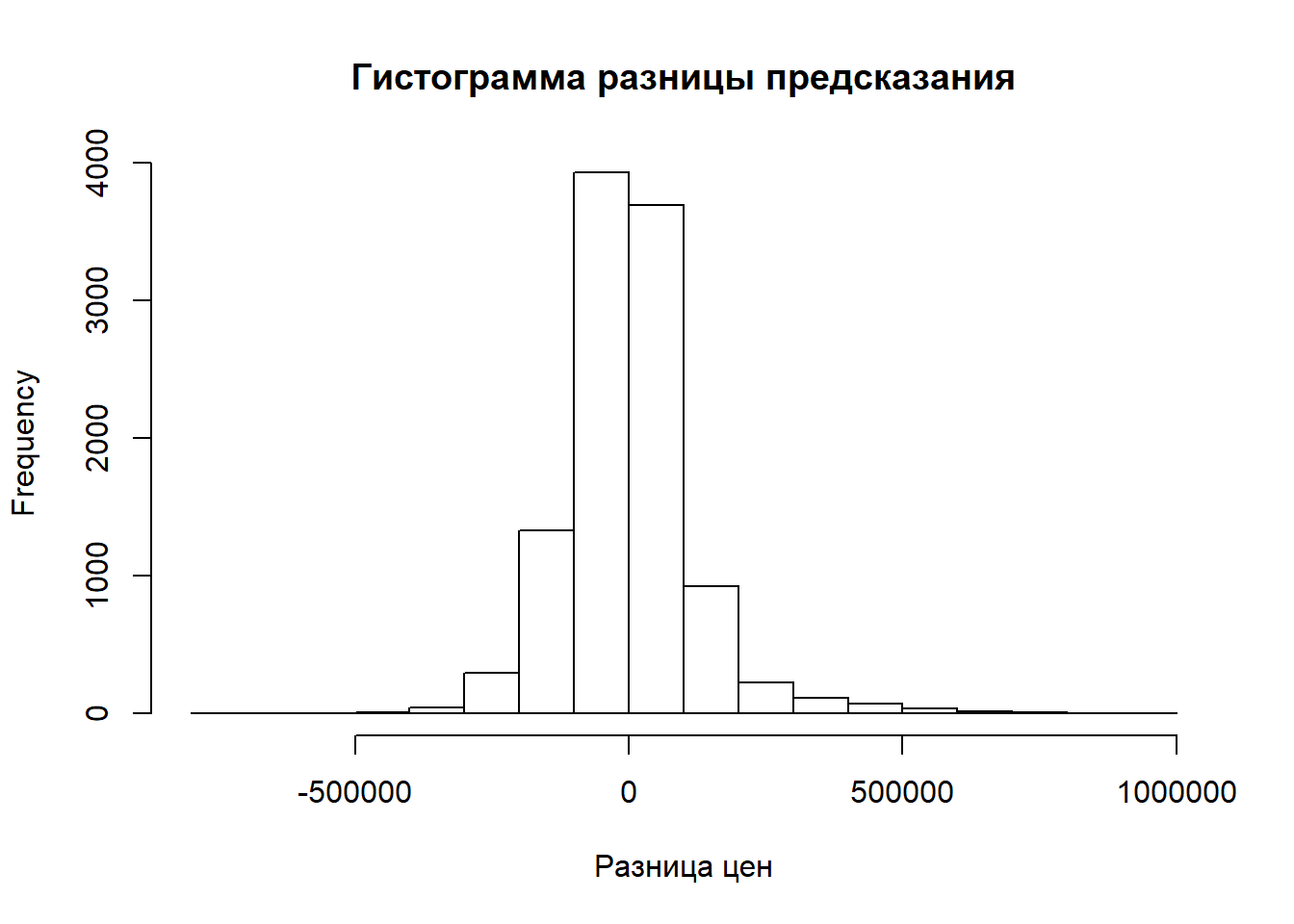


Рисунок 9 – Гистограмма зависимости предсказанных значений от проверяемых

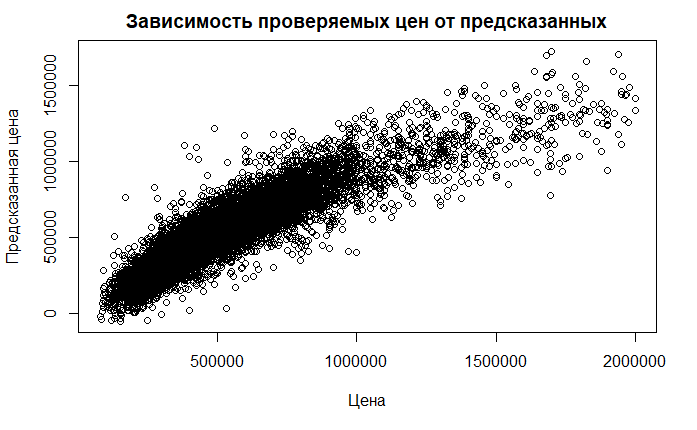


Рисунок 10 – Зависимость предсказанных значений от проверяемых

Для приложения было принято решение использовать меньшее число параметров, для этого была создана дополнительная модель. Ее результат показан на рисунках 11, 12, 13.

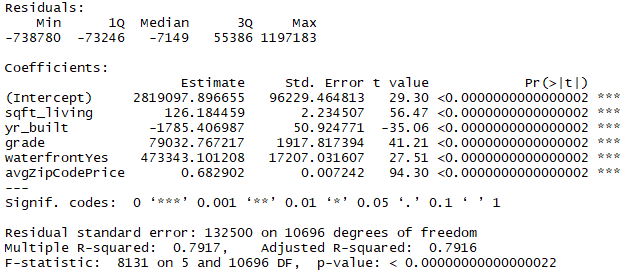


Рисунок 11 – Коэффициенты модели с меньшим числом параметров



Рисунок 12 – Гистограмма зависимости предсказанных значений от проверяемых модели с меньшим числом параметров



Рисунок 13 – Зависимость предсказанных значений от проверяемых модели с меньшим числом параметров

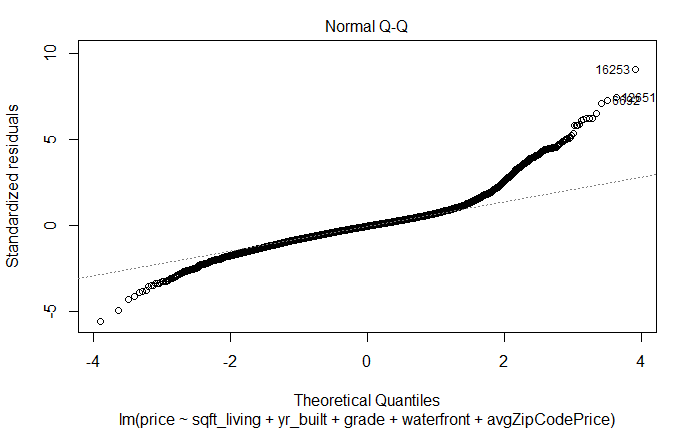


Рисунок 14 – Проверка зависимости при помощи графика Q-Q

# 3 ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ

Приложение было написано на языке Python. Его интерфейс изображен на рисунке 12.

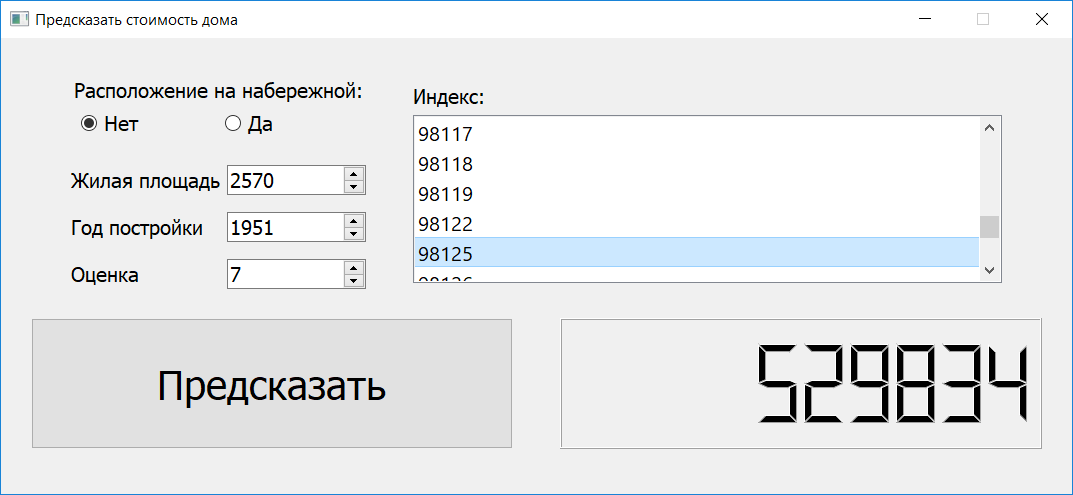


Рисунок 15 – Макет приложения

Результат вызова функции из R с теми же параметрами, что представлены на изображение макета, показан на рисунке 13. Результаты вывода у них совпадают.



Рисунок 16 – Вызов функции предсказания в R

Данное приложение получает значения почтовых индексов, а при нажатии кнопки предсказать вызывает функции R для расчета цены.

# 4 ИНСТРУКЦИЯ ПОЛЬЗОВАТЕЛЯ

1. Запустить приложение;
2. Ввести требуемые параметры в окна ввода (возможные вводимые параметры ограничены граничными значениями). Выбрать необходимый почтовый индекс;
3. Нажить на кнопку «Предсказать»;
4. Получить значение в окне справа от нажимаемой кнопки.

# ЗАКЛЮЧНИЕ

В результате выполнения данной работы был изучен открытый набор данных стоимости домов, путем изучения представленной информации на сайте источника и отображением графиков и параметров при помощи функция языка программирования R в программе Rstudio. После была построена линейная модель по этим данным для предсказания стоимости домов, используя для этого язык Rmarkdown в котором были построены возможные зависимости между ценой и другими параметрами, проведена очистка от выбросов и изучены значения наиболее влияющие факторы на цену дома. Значение для оптимального числа параметров в линейной модели составило 0,82.

Для отображения возможного применения данных расчетов было реализовано приложение языке Python, в котором была реализована функциональность в виде обращения к функциям языка R для применения и расчета линейной модели. В данном случае использовалась линейная модель с меньшим числом параметров, в котором .

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. House Sales in King County, USA. Predict house price using regression – https://www.kaggle.com/harlfoxem/housesalesprediction

2.

# ПРИЛОЖЕНИЕ А – КОД RMARDOWN

---

title: "Курсовая работа"

author: "Изотов Илья, Куштанов Дмитрий РИ-450004"

---

```{r}

options(scipen=10000)

data <- read.csv("kc\_house\_data.csv", header=TRUE, sep=",")

head(data)

summary(data)

data <- data[,-c(1,2)]

data$waterfront <- factor(data$waterfront, levels = c(0,1))

levels(data$waterfront) <- c("No", "Yes")

hist(data$price, xlab = "Цена", main = "Цена")

hist(data$sqft\_living, xlab = "Размер жилой площади", main = "Жилая площадь")

hist(data$yr\_built, xlab = "Год постройки", main = "Год постройки")

```

###Визулизируем возможные зависимости

```{R}

plot(data$price, data$sqft\_living,'p',main = "Зависимость стоимости от размера жилой площади")

plot(data$price, data$grade,'p',main = "Зависимость стоимости от оценки")

plot(data$waterfront, data$price,'p',main = "Зависимость стоимости от наличия вида на набережную")

boxplot(data$price, main = "Цена")

```

###Очистка от выбросов

```{r}

summary(data[data$waterfront=="No", ]$price)

data <- data[data$sqft\_living<13000,]

data <- data[(data$price<2\*10^6),]

data <- data[(data$grade<=12 & data$grade>3),]

plot(data$waterfront, data$price,'p',main = "Зависимость стоимости от наличия вида на набережную")

plot(sort(data$price), main = "Цена")

```

###Нахождение значимых параметров

```{r}

boxplot(data$price, main = "Цена")

lm.data <- lm(price~., data = data)

summary(lm.data)

plot(lm.data)

```

```{R}

plot(x= data$price, xlab = "Цена", ylab = "Размер жилой площади", y = data$sqft\_living,'p',main = "Зависимость стоимости от размера жилой площади")

plot(x= data$price,xlab = "Цена", ylab = "Оценка", y =data$grade,'p',main = "Зависимость стоимости от оценки")

```

###Добавление дополнительного параметра средней цены по почтовому индексу и разделение данных

```{r}

data$avgZipCodePrice <- ave(data$price,data$zipcode)

odds <- seq(1, nrow(data), by=2)

data.in <- data[odds, ]

data.out <- data[-odds, ]

```

###Тренировка первой модели

```{r}

lm.data.in <- lm(price~sqft\_living+waterfront+floors+view+condition+yr\_built+grade+bedrooms+bathrooms+lat+long+sqft\_living15+avgZipCodePrice, data = data.in)

data.predict.out <- predict(lm.data.in, data.out)

cor(data.out$price, data.predict.out)

summary(lm.data.in)

plot(x = data.out$price, xlab = "Цена", ylab = "Предсказанная цена",y = data.predict.out, main = "Зависимость проверяемых цен от предсказанных")

hist(data.out$price - data.predict.out, xlab = "Разница цен", main = "Гистограмма разницы предсказания")

```

###Уменьшение числа параметров для тренировки второй модели

```{r}

lm.data.in1 <- lm(price~sqft\_living+yr\_built+grade+waterfront+avgZipCodePrice, data = data.in)

data.predict.out1 <- predict(lm.data.in1, data.out)

cor(data.out$price, data.predict.out1)

summary(lm.data.in1)

plot(x = data.out$price, xlab = "Цена", ylab = "Предсказанная цена",y = data.predict.out1, main = "Зависимость проверяемых цен от предсказанных")

hist(data.out$price - data.predict.out1, xlab = "Разница цен", main = "Гистограмма разницы предсказания")

plot(lm.data.in1)

```

# ПРИЛОЖЕНИЕ Б – КОД ФУННКЦИИ R

# Функция получения списка потовых индексов

# Outputs: Возвращает уникальные номера почтовых индексов

# Для выполнения необходимо нахождение используемового набора данных в текущей директории

fun.zip<-function()

{

data <- read.csv("kc\_house\_data.csv", header=TRUE, sep=",");

data <- data[,-c(1,2)];

data <- data[data$sqft\_living<13000,];

data<- data[data$price<2\*10^6,];

data <- data[(data$grade<=12 & data$grade>3),]

data$avgZipCodePrice <- ave(data$price,data$zipcode);

return (unique(data$zipcode));

}

# Функция предсказания цены

# Inputs: sqft\_living = площадь жилья

# yr\_built = год постройки здания

# grade = оценка здания

# waterfront = Наличие выхода к воде (0 - нет, 1 - да)

# zipcode = почтовый индекс

# Outputs: Возвращает предсказанную цену

# Для выполнения необходимо нахождение набора данных в текущей директории

fun.predict<-function(sqft\_living, yr\_built, grade, waterfront, zipcode){

data <- read.csv("kc\_house\_data.csv", header=TRUE, sep=",");

data <- data[,-c(1,2)];

data <- data[data$sqft\_living<13000,];

data<- data[data$price<2\*10^6,];

data <- data[(data$grade<=12 & data$grade>3),]

data$avgZipCodePrice <- ave(data$price,data$zipcode)

avg <-mean(data[data$zipcode==zipcode,]$price)

lm.data.in <- lm(price~sqft\_living+yr\_built+grade+waterfront+avgZipCodePrice, data = data)

ret <- coef(lm.data.in)[1] + coef(lm.data.in)[2]\*sqft\_living +

coef(lm.data.in)[3]\*yr\_built + coef(lm.data.in)[4]\*grade +

coef(lm.data.in)[5]\*waterfront + coef(lm.data.in)[6]\*avg;

return (as.integer(ret))

}

# ПРИЛОЖЕНИЕ В – КОД ОСНОВНОЙ ФУНКЦИИ PYTHON

import sys

from PyQt5 import QtWidgets

import design

import numpy as np

import rpy2.robjects as robjects

class pyApp(QtWidgets.QMainWindow, design.Ui\_MainWindow, QtWidgets.QDialogButtonBox):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.setupUi(self)

list\_r = robjects.r(f"fun.zip()")

list\_r = np.asarray(list\_r)

list\_r = np.array(map(str, list\_r))

list\_r = list\_r.tolist()

list\_r = list(list\_r)

list\_r.sort()

self.list\_r = list\_r

self.listZip.addItems(self.list\_r)

self.pushButtonPredict.clicked.connect(self.Predict)

self.listZip.setCurrentRow(0)

def Predict(self):

sqftLiving = self.spinBoxSqftLiving.value()

yrBuilt = self.spinBoxYrBuilt.value()

grade = self.spinBoxGrade.value()

zipcode = self.listZip.currentRow()

zipcode = self.list\_r[zipcode]

if self.radioButtonYes.isChecked():

waterfront = 1

elif self.radioButtonNo.isChecked():

waterfront = 0

predictFromR = robjects.r(f"fun.predict({sqftLiving}, {yrBuilt}, {grade}, {waterfront}, {zipcode})")

predictFromR = int(predictFromR[0])

if predictFromR < 0:

self.lcdNumberResult.setProperty("intValue", 0)

msgBox = QtWidgets.QMessageBox()

msgBox.setText('Ошибка предсказания')

msgBox.setWindowTitle('Ошибка')

msgBox.exec()

else:

self.lcdNumberResult.setProperty("intValue", predictFromR)

def main():

robjects.r("source('R.R')")

app = QtWidgets.QApplication(sys.argv)

window = pyApp()

window.show()

app.exec\_()

if \_\_name\_\_ == '\_\_main\_\_':

main()