**Задание 3. Загрузка данных из Интернет (Web-scraping) средствами R и анализ данных в R.**

Выполнил студент 2 курса

группы 09-715(1)

Санамян Артак.

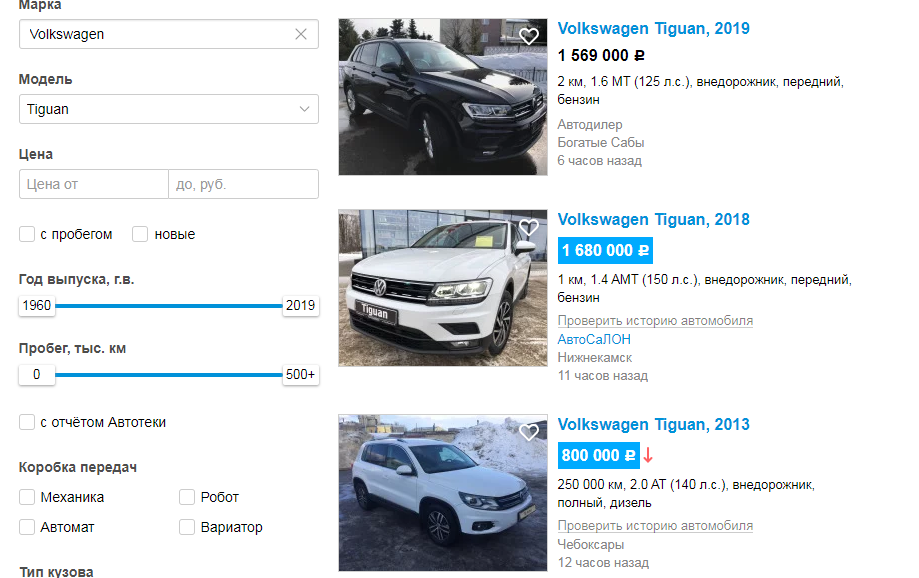
**Текст задания:**

Пусть перед нами стоит задача прогнозирования цены на товары (например, на автомобили) на основе объявлений, публикуемых на сайте **avito.ru** (или на подобных сайтах). Задайте параметры поиска так, чтобы в выборку попало более 50 однородных объектов. Выходным параметром для прогноза является цена. Выберите входные параметры для прогноза.

Далее полученные данные нужно прочитать в R, при необходимости преобразовать к нужному формату и построить модель для прогнозирования. В данной задаче для прогнозирования применим **линейную регрессию**. Можете использовать готовые функции из R или реализовать алгоритм линейной регрессии самостоятельно. Оцените качество построенной модели. Сделайте выводы.

**Описание выполненной работы:**

На сайте **avito.ru** были выбраны автомобили Volkswagen Tiguan, в выборку попало 135 автомобилей.



Для извлечения данных с сайта используем библиотеку **rvest**.

> library(rvest)

Url адрес сайта будем хранить в переменной vw\_url:

> vw\_url= 'https://www.avito.ru/kazan/avtomobili/volkswagen/tiguan?radius=200'

Воспользуемся функцией read\_html(), которая возвращает XML-документ, содержащий всю информацию о веб-странице:

|  |
| --- |
| > pages <- read\_html(vw\_url)  > pages  {xml\_document}  <html>  [1] <head>\n<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"  >\n<script>\n window.dataLayer = [{"pageType":"SearchResults" ...  [2] <body class=" " id="catalog">\n <noscript> <iframe src="//www.google  tagmanager.com/ns.html?id=GTM-KP9Q9H" height="0" width="0" style= ... |
|  |
| |  | | --- | |  | |

На сайте с объявлениями чаще всего бывает несколько страниц. Для начала найдем максимальный номер страницы.

> # функция для получения номера последней страницы

> get\_last\_page\_numb <- function(html){

+

+ pages\_data <- html %>%

+ html\_nodes('.pagination-page') %>%

+ html\_text()

+

+ pages\_data[(length(pages\_data)-1)] %>%

+ unname() %>%

+ as.numeric()

+ }

> # получим номер последней страницы

> (last\_page\_numb <- get\_last\_page\_numb(pages))

[1] 4

Таким образом, мы имеем 4 страницы с объявлениями о продаже автомобилей. Теперь получим список url адресов каждой страницы. Заметим, что url первой страницы имеет вид:

* https://www.avito.ru/kazan/avtomobili/volkswagen/tiguan?radius=200

Url последующих страниц имеет вид:

* [https://www.avito.ru/kazan/avtomobili/volkswagen/ tiguan?p=**page\_numb**&radius=200](https://www.avito.ru/kazan/avtomobili/nissan/almera?p=page_numb&radius=200),

где page\_numb – номер страницы.

Подключим пакет **stringr**, чтобы использовать регулярные выражения:

> library(stringr)

> # получим список страниц

> list\_of\_pages = character();

> list\_of\_pages = append(list\_of\_pages, vw\_url)

> list\_of\_pages = append(list\_of\_pages, str\_c(short\_url, paste0('p=', 2:last\_page\_numb, '&radius=200')));

Список url адресов каждой страницы:

> list\_of\_pages

[1] "https://www.avito.ru/kazan/avtomobili/volkswagen/tiguan?radius=200"

[2] "https://www.avito.ru/kazan/avtomobili/volkswagen/tiguan?p=2&radius=200"

[3] "https://www.avito.ru/kazan/avtomobili/volkswagen/tiguan?p=3&radius=200"

[4] "https://www.avito.ru/kazan/avtomobili/volkswagen/tiguan?p=4&radius=200"

С помощью функции html\_nodes() мы можем обратиться к элементам определенного класса. Например, информация о каждом автомобиле хранится в классе .item\_table-description, чтобы получить все элементы этого класса, необходимо вызвать функцию: html\_nodes(".item\_table-description").

С помощью функции html\_text() мы можем вытащить текстовую информацию из

html документа. Например, чтобы узнать год выпуска автомобиля, мы обращаемся к классу item-description-title, извлекаем текстовую информацию с помощью html\_text() и удаляем все символы переноса строки с помощью функции str\_trim().

+ #достаем год выпуска

+ tmp <- results %>% html\_nodes(".item-description-title")%>% html\_text()%>%str\_trim()

Используя регулярные выражения, получаем только числа:

+ year\_tmp = as.numeric(str\_extract(tmp, "[:digit:]+"));

Аналогичным образом, мы вытаскиваем информацию о цене (price) и характеристиках автомобиля (info).

Для того чтобы получить ссылки на подробную информацию об автомобиле, мы используем функцию html\_attr(). Например, вызвав html\_attr("href"), мы получим информацию об атрибуте **href.**

+ #вытаскиваем ccылки на подробную информацию об автомобиле

+ ref <- results %>% html\_nodes(".item-description-title-link") %>% html\_attr("href")

Однако ссылки имеют следующий вид:

> ref

[1] "/bogatye\_saby/avtomobili/volkswagen\_tiguan\_2019\_1286588418"

[2] "/nizhnekamsk/avtomobili/volkswagen\_tiguan\_2018\_1274446729"

[3] "/cheboksary/avtomobili/volkswagen\_tiguan\_2013\_926698248"

[4] "/cheboksary/avtomobili/volkswagen\_tiguan\_2018\_998218454"

[5] "/kazan/avtomobili/volkswagen\_tiguan\_2012\_1554074939"

[6] "/cheboksary/avtomobili/volkswagen\_tiguan\_2016\_1279587239"

[7] "/kazan/avtomobili/volkswagen\_tiguan\_2014\_1400097276"

[8] "/kazan/avtomobili/volkswagen\_tiguan\_2012\_1331605371"

[9] "/cheboksary/avtomobili/volkswagen\_tiguan\_2009\_1544939869"

[10] "/nizhnekamsk/avtomobili/volkswagen\_tiguan\_2018\_1219723278"

[11] "/kazan/avtomobili/volkswagen\_tiguan\_2017\_1271370641"

[12] "/ulyanovsk/avtomobili/volkswagen\_tiguan\_2017\_1292755759"

[13] "/nizhnekamsk/avtomobili/volkswagen\_tiguan\_2018\_1225565272"

[14] "/nizhnekamsk/avtomobili/volkswagen\_tiguan\_2018\_1400927632"

[15] "/nizhnekamsk/avtomobili/volkswagen\_tiguan\_2018\_1036456377"

Чтобы получить url адрес необходимой страницы, добавим к полученным ссылкам

> main\_url = 'https://www.avito.ru'

Сложить строки можно с помощью функции paste0():

+ ref <- paste0(main\_url, ref)

Теперь ссылки имеют вид:

> ref

[1] "https://www.avito.ru/bogatye\_saby/avtomobili/volkswagen\_tiguan\_2019\_1286588418"

[2] "https://www.avito.ru/nizhnekamsk/avtomobili/volkswagen\_tiguan\_2018\_1274446729"

[3] "https://www.avito.ru/cheboksary/avtomobili/volkswagen\_tiguan\_2013\_926698248"

[4] "https://www.avito.ru/cheboksary/avtomobili/volkswagen\_tiguan\_2018\_998218454"

[5] "https://www.avito.ru/kazan/avtomobili/volkswagen\_tiguan\_2012\_1554074939"

[6] "https://www.avito.ru/cheboksary/avtomobili/volkswagen\_tiguan\_2016\_1279587239"

[7] "https://www.avito.ru/kazan/avtomobili/volkswagen\_tiguan\_2014\_1400097276"

[8] "https://www.avito.ru/kazan/avtomobili/volkswagen\_tiguan\_2012\_1331605371"

[9] "https://www.avito.ru/cheboksary/avtomobili/volkswagen\_tiguan\_2009\_1544939869"

[10] "https://www.avito.ru/nizhnekamsk/avtomobili/volkswagen\_tiguan\_2018\_1219723278"

[11] "https://www.avito.ru/kazan/avtomobili/volkswagen\_tiguan\_2017\_1271370641"

[12] "https://www.avito.ru/ulyanovsk/avtomobili/volkswagen\_tiguan\_2017\_1292755759"

[13] "https://www.avito.ru/nizhnekamsk/avtomobili/volkswagen\_tiguan\_2018\_1225565272"

[14] "https://www.avito.ru/nizhnekamsk/avtomobili/volkswagen\_tiguan\_2018\_1400927632"

[15] <https://www.avito.ru/nizhnekamsk/avtomobili/volkswagen_tiguan_2018_1036456377>

Таким образом, проходимся по каждой странице объявлений и вытаскиваем год выпуска, цену, характеристики автомобиля, ссылки на подробную информацию об автомобиле:

> for(i in 1:last\_page\_numb)

+ {

+ page <- read\_html(list\_of\_pages[i]);

+ results <- page %>% html\_nodes(".item\_table-description")

+

+ #достаем год выпуска

+ tmp <- results %>% html\_nodes(".item-description-title")%>% html\_text()%>%str\_trim()

+ year\_tmp = as.numeric(str\_extract(tmp, "[:digit:]+"));

+ year = append(year, year\_tmp);

+

+ #вытаскиваем цену автомобиля

+ price\_tmp <- results %>% html\_nodes(".about") %>% html\_nodes(".price") %>% html\_text()

+ price\_tmp = as.numeric(str\_replace\_all(price\_tmp, "[ \n\r\t\\\u20bd]+", ""))

+ price = append(price, price\_tmp);

+

+ #вытаскиваем информацию о характеристиках автомобиля

+ info\_tmp <- results %>% html\_nodes(".about")%>% html\_nodes(".specific-params\_block")%>% html\_text()%>%str\_trim()

+ info\_tmp = str\_replace\_all(info\_tmp, "[ ]+", "");

+ info = append(info, info\_tmp);

+

+ #вытаскиваем ccылки на подробную информацию об автомобиле

+ ref <- results %>% html\_nodes(".item-description-title-link") %>% html\_attr("href")

+ ref <- paste0(main\_url, ref)

+ ref\_info = append(ref\_info, ref)

+ }

> #год выпуска

> year

[1] 2019 2018 2013 2018 2012 2016 2014 2012 2009 2018 2017 2017 2018 2018 2018 2018 2011 2017 2013 2009 2019 2019 2012 2012 2012 2008 2014

[28] 2013 2014 2013 2013 2011 2010 2015 2012 2009 2016 2018 2018 2014 2010 2012 2010 2016 2011 2015 2012 2015 2016 2012 2009 2018 2017 2011

[55] 2018 2013 2019 2012 2012 2012 2008 2016 2011 2012 2012 2013 2013 2012 2015 2015 2010 2012 2012 2012 2016 2012 2015 2016 2018 2016 2012

[82] 2011 2012 2016 2009 2010 2012 2015 2009 2015 2009 2015 2012 2016 2018 2012 2012 2019 2011 2011 2012 2015 2019 2019 2008 2017 2018 2015

[109] 2013 2009 2013 2018 2018 2009 2018 2012 2012 2017 2019 2013 2010 2012 2011 2016 2019 2018 2014 2018 2019 2013 2019 2018 2018 2016 2016

> #цена

> price

[1] 1569000 1680000 800000 2390000 790000 1004700 820000 536300 485000 2200000 1623837 1690000 2130000 2130000 2270000 2290000 765000

[18] 1400000 990000 560000 1727500 2467500 825000 770000 740000 500000 790000 682200 721300 685000 965420 765000 570000 800000

[35] 850000 650000 1310130 1799900 1600000 1092500 677000 880900 750000 1289000 680000 1119000 850000 949000 960000 840000 555555

[52] 2290000 1126100 600000 1875000 1100000 1800000 690000 1010000 700000 590000 850000 799000 500000 650000 777700 765000 845000

[69] 970000 1091000 585000 620000 835000 650000 1450000 675555 1024000 1000000 2400000 1277777 700000 699000 855000 1050000 670000

[86] 595000 1000000 985000 600000 1299000 700000 1100000 820000 1199000 1900000 980000 839100 1469000 720000 800000 850000 782300

[103] 1547000 1599000 669000 1600000 1499000 1175000 1050000 600000 860000 1820000 1719000 679000 1959000 826500 658165 1550000 1549000

[120] 980000 485000 850000 870000 1040000 1746130 1370500 1200000 1920000 1399000 669000 1899000 2249900 1780000 1080600 967400

> #информация о характеристиках автомобиля

> info

[1] "2км,1.6 MT (125л.с.),внедорожник,передний,бензин" "1км,1.4 AMT (150л.с.),внедорожник,передний,бензин"

[3] "250000км,2.0 AT (140л.с.),внедорожник,полный,дизель" "1км,2.0 AMT (220л.с.),внедорожник,полный,бензин"

[5] "101000км,2.0 AT (170л.с.),внедорожник,полный,бензин" "26437км,1.4 AMT (150л.с.),внедорожник,передний,бензин"

[7] "63000км,1.4 AT (150л.с.),внедорожник,передний,бензин" "111270км,1.4 MT (122л.с.),внедорожник,передний,бензин"

[9] "150000км,1.4 MT (150л.с.),внедорожник,передний,бензин" "1км,2.0 AMT (150л.с.),внедорожник,полный,дизель"

[11] "32348км,1.4 AMT (150л.с.),универсал,полный,бензин" "35000км,1.4 AMT (150л.с.),внедорожник,полный,бензин"

[13] "1км,2.0 AMT (150л.с.),внедорожник,полный,дизель" "1км,2.0 AMT (150л.с.),внедорожник,полный,дизель"

[15] "1км,2.0 AMT (220л.с.),внедорожник,полный,бензин" "1км,2.0 AMT (220л.с.),внедорожник,полный,бензин"

[17] "123000км,2.0 AT (170л.с.),внедорожник,полный,бензин" "40000км,1.4 MT (150л.с.),внедорожник,полный,бензин"

> #ccылки на подробную информацию об автомобиле

> ref\_info

[1] "https://www.avito.ru/bogatye\_saby/avtomobili/volkswagen\_tiguan\_2019\_1286588418"

[2] "https://www.avito.ru/nizhnekamsk/avtomobili/volkswagen\_tiguan\_2018\_1274446729"

[3] "https://www.avito.ru/cheboksary/avtomobili/volkswagen\_tiguan\_2013\_926698248"

[4] "https://www.avito.ru/cheboksary/avtomobili/volkswagen\_tiguan\_2018\_998218454"

[5] "https://www.avito.ru/kazan/avtomobili/volkswagen\_tiguan\_2012\_1554074939"

[6] "https://www.avito.ru/cheboksary/avtomobili/volkswagen\_tiguan\_2016\_1279587239"

[7] "https://www.avito.ru/kazan/avtomobili/volkswagen\_tiguan\_2014\_1400097276"

[8] "https://www.avito.ru/kazan/avtomobili/volkswagen\_tiguan\_2012\_1331605371"

[9] "https://www.avito.ru/cheboksary/avtomobili/volkswagen\_tiguan\_2009\_1544939869"

[10] "https://www.avito.ru/nizhnekamsk/avtomobili/volkswagen\_tiguan\_2018\_1219723278"

[11] "https://www.avito.ru/kazan/avtomobili/volkswagen\_tiguan\_2017\_1271370641"

[12] "https://www.avito.ru/ulyanovsk/avtomobili/volkswagen\_tiguan\_2017\_1292755759"

[13] "https://www.avito.ru/nizhnekamsk/avtomobili/volkswagen\_tiguan\_2018\_1225565272"

[14] "https://www.avito.ru/nizhnekamsk/avtomobili/volkswagen\_tiguan\_2018\_1400927632"

[15] "https://www.avito.ru/nizhnekamsk/avtomobili/volkswagen\_tiguan\_2018\_1036456377"

[16] "https://www.avito.ru/nizhnekamsk/avtomobili/volkswagen\_tiguan\_2018\_1515826608"

Некоторые автомобили являются битыми, сохраним информацию об этом в переменную is\_broken:

> #заполняем признак, является ли автомобиль битым, 1 - битый, 0 - нет

> is\_broken = ifelse(substr(info, 0, 5) == 'Битый', 1, 0)

> is\_broken

[1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

[69] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Т0еперь удалим из переменной info информацию о том, что автомобиль является битым:

> info = str\_replace(info, 'Битый,', '')

Характеристики автомобиля перечисляются через запятую, разобьем строку с помощью функции str\_split():

> split\_info <- str\_split(info, ',');

> split\_info

[[1]]

[1] "2км" "1.6 MT (125л.с.)" "внедорожник" "передний" "бензин"

[[2]]

[1] "1км" "1.4 AMT (150л.с.)" "внедорожник" "передний" "бензин"

[[3]]

[1] "250000км" "2.0 AT (140л.с.)" "внедорожник" "полный" "дизель"

[[4]]

[1] "1км" "2.0 AMT (220л.с.)" "внедорожник" "полный" "бензин"

[[5]]

[1] "101000км" "2.0 AT (170л.с.)" "внедорожник" "полный" "бензин"

Вытащим следующую информацию:

> mileage = numeric(); #пробег, км

> engine\_capacity = numeric(); #объем двигателя, л

> power = numeric(); #мощность л.с.

> transmission = character(); #трансмиссия (механика/автомат)

> body\_type = character(); #тип кузова

> drive\_unit = character(); #привод

> engine\_type = character(); #тип двигателя

Заметим, что если длина разбитой строки = 5, значит, в характеристиках есть информация о пробеге автомобиля, иначе – автомобиль новый, его пробег устанавливаем 0 км. С помощью регулярных выражений вытаскиваем нужную информацию:

> for(i in 1:length(split\_info))

+ {

+ str\_info = split\_info[[i]];

+ index = 1;

+

+ if(length(str\_info) == 5)

+ {

+ mileage = append(mileage, as.numeric(str\_extract(str\_info[1], "[:digit:]+")));

+ index = 2;

+ }

+ else

+ {

+ mileage = append(mileage, 0);

+ }

+

+ k = str\_extract(str\_info[index], "[(][:digit:]+");

+ k = as.numeric(str\_replace(k, '[(]', ''));

+ power = append(power, k);

+ y = str\_extract(str\_info[index], "[MA]?T");

+ transmission = append(transmission, y);

+ z = str\_extract(str\_info[index], "[:digit:]+[.][:digit:]+");

+ engine\_capacity = append(engine\_capacity, as.numeric(z))

+ body\_type = append(body\_type, str\_info[index+1])

+ drive\_unit = append(drive\_unit, str\_info[index+2])

+ engine\_type = append(engine\_type, str\_info[index+3])

+ }

> #пробег, км

> mileage

[1] 2 1 250000 1 101000 26437 63000 111270 150000 1 32348 35000 1 1 1 1 123000 40000 74904

[20] 140000 0 0 91670 94000 113245 200000 89493 68000 79695 121000 92506 154000 130000 65000 105000 146000 55361 25300

[39] 21000 77455 112000 111021 159784 79374 93000 92000 64000 49500 58000 108000 150500 1 30478 166336 1 61800 3

[58] 100000 91000 90000 89000 37500 136000 114000 160000 83556 75300 105000 16000 86000 162000 119000 110151 161000 31000 140000

[77] 49221 33000 4032 65000 100500 71000 120000 50000 110000 103000 121000 144080 72000 42800 150000 96456 107000 18000 16500

[96] 67000 123010 1 100000 122000 90000 103764 1 1 126000 35700 0 47500 81000 107000 82000 1 1 116113

[115] 1 134751 159750 26000 1 40000 169737 174500 70000 39501 0 10197 43960 12000 1 75320 1 8 1

[134] 21294 115130

> #мощность л.с.

> power

[1] 125 150 140 220 170 150 150 122 150 150 150 150 150 150 220 220 170 150 140 150 150 220 170 170 170 150 150 150 122 122 122 170 150 122

[35] 170 170 180 150 150 170 140 170 170 170 122 170 170 150 122 170 150 220 125 122 150 170 150 150 140 150 150 122 170 150 150 150 122 140

[69] 122 170 170 150 170 122 180 122 150 150 180 180 150 122 170 150 170 150 140 170 150 170 170 140 170 150 150 170 200 125 170 140 170 122

[103] 150 150 170 150 125 170 170 150 150 150 150 170 180 200 140 150 150 170 170 170 170 150 150 125 210 180 150 122 150 150 150 122 150

> #трансмиссия (механика/автомат)

> transmission

[1] "MT" "MT" "AT" "MT" "AT" "MT" "AT" "MT" "MT" "MT" "MT" "MT" "MT" "MT" "MT" "MT" "AT" "MT" "AT" "MT" "MT" "MT" "AT" "AT" "AT" "MT" "MT"

[28] "MT" "MT" "MT" "MT" "AT" "MT" "MT" "AT" "AT" "AT" "MT" "MT" "AT" "AT" "AT" "AT" "AT" "MT" "AT" "AT" "MT" "MT" "AT" "MT" "MT" "MT" "MT"

[55] "AT" "AT" "MT" "AT" "AT" "MT" "MT" "MT" "AT" "MT" "MT" "MT" "MT" "AT" "MT" "AT" "AT" "AT" "AT" "MT" "AT" "MT" "MT" "MT" "MT" "AT" "MT"

[82] "MT" "AT" "AT" "AT" "MT" "AT" "AT" "MT" "AT" "AT" "AT" "AT" "MT" "MT" "AT" "AT" "MT" "AT" "AT" "AT" "MT" "AT" "MT" "AT" "MT" "AT" "AT"

[109] "AT" "MT" "MT" "MT" "MT" "AT" "MT" "AT" "MT" "AT" "MT" "AT" "AT" "AT" "AT" "MT" "MT" "MT" "MT" "MT" "MT" "MT" "MT" "AT" "MT" "MT" "MT"

> #объем двигателя, л

> engine\_capacity

[1] 1.6 1.4 2.0 2.0 2.0 1.4 1.4 1.4 1.4 2.0 1.4 1.4 2.0 2.0 2.0 2.0 2.0 1.4 2.0 1.4 1.4 2.0 2.0 2.0 2.0 1.4 1.4 1.4 1.4 1.4 1.4 2.0 1.4 1.4

[35] 2.0 2.0 2.0 2.0 1.4 2.0 2.0 2.0 2.0 2.0 1.4 2.0 2.0 1.4 1.4 2.0 1.4 2.0 1.4 1.4 1.4 2.0 1.4 1.4 2.0 1.4 1.4 1.4 2.0 1.4 1.4 1.4 1.4 2.0

[69] 1.4 2.0 2.0 1.4 2.0 1.4 2.0 1.4 1.4 1.4 2.0 2.0 1.4 1.4 2.0 1.4 2.0 1.4 2.0 2.0 1.4 2.0 2.0 2.0 2.0 1.4 2.0 2.0 2.0 1.4 2.0 2.0 2.0 1.4

[103] 1.4 1.4 2.0 1.4 1.4 2.0 2.0 1.4 1.4 1.4 1.4 2.0 2.0 2.0 1.4 1.4 1.4 2.0 2.0 2.0 2.0 1.4 1.4 1.4 2.0 2.0 1.4 1.4 1.4 2.0 1.4 1.4 1.4

> #тип кузова

> body\_type

[1] "внедорожник" "внедорожник" "внедорожник" "внедорожник" "внедорожник" "внедорожник" "внедорожник" "внедорожник" "внедорожник"

[10] "внедорожник" "универсал" "внедорожник" "внедорожник" "внедорожник" "внедорожник" "внедорожник" "внедорожник" "внедорожник"

[19] "внедорожник" "внедорожник" "внедорожник" "внедорожник" "универсал" "внедорожник" "внедорожник" "внедорожник" "внедорожник"

[28] "внедорожник" "внедорожник" "внедорожник" "универсал" "внедорожник" "внедорожник" "универсал" "внедорожник" "внедорожник"

[37] "внедорожник" "внедорожник" "внедорожник" "внедорожник" "внедорожник" "внедорожник" "внедорожник" "универсал" "внедорожник"

> #привод

> drive\_unit

[1] "передний" "передний" "полный" "полный" "полный" "передний" "передний" "передний" "передний" "полный" "полный" "полный"

[13] "полный" "полный" "полный" "полный" "полный" "полный" "полный" "полный" "передний" "полный" "полный" "полный"

[25] "полный" "полный" "передний" "передний" "передний" "передний" "передний

> #тип двигателя

> engine\_type

[1] "бензин" "бензин" "дизель" "бензин" "бензин" "бензин" "бензин" "бензин" "бензин" "дизель" "бензин" "бензин" "дизель" "дизель" "бензин"

[16] "бензин" "бензин" "бензин" "дизель" "бензин" "бензин" "бензин" "бензин" "бензин" "бензин" "бензин" "бензин" "бензин" "бензин" "бензин"

[31] "бензин" "бензин" "бензин" "бензин" "бензин" "бензин" "бензин" "дизель" "бензин" "бензин" "дизель" "бензин" "бензин" "бензин" "бензин"

[46] "бензин" "бензин" "бензин" "бензин" "бензин" "бензин" "бензин" "бензин" "бензин" "бензин" "бензин" "бензин" "бензин" "дизель" "бензин"

[61] "бензин" "бензин" "бензин" "бензин" "бензин" "бензин" "бензин" "дизель" "бензин" "бензин" "бензин" "бензин" "бензин" "бензин" "бензин"

Запишем в переменную petdis информацию о типе двигателя:

> petdis = numeric(); #тип двигателя - бензин(0) /дизель (1)

> for(i in 1:length(engine\_type))

+ {

+ if(engine\_type[i] == 'бензин')

+ {

+ petdis[i] = 0;

+ }

+ else

+ {

+ petdis[i] = 1;

+ }

+

+ }

> petdis

[1] 0 0 1 0 0 0 0 0 0 1 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1

[69] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0

Запишем в переменную bt информацию о типе кузова:

bt = numeric(); #тип кузова - хетчбэк, внедорожник(0) / универсал, минивэн(1)

for(i in 1:length(body\_type))

{

if(body\_type[i] == 'хетчбэк' || body\_type[i] == 'внедорожник')

{

bt[i] = 0;

}

else

{

bt[i] = 1;

}

}

[1] 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0

[69] 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Построим модель для прогнозирования. Входные параметры для модели прогноза:

* Год выпуска;
* Признак битый / нет;
* Пробег, км;
* Мощность, л.с.;
* Объем двигателя, л;
* Признак тип кузова: (хетчбэк, внедорожник) / нет;
* Признак тип двигателя: бензин/газ;

Выходной параметр – цена.

Для удобства разделим пробег и цену на 1000:

> mileage = mileage / 1000;

> price = price / 1000;

> # сохраним преобразованные поля как data.frame

> parse\_data = data.frame("year" = year, "is\_broken" = is\_broken, "mileage" = mileage, "power" = power,

+ "engine\_capacity" = engine\_capacity, "bt" = bt, "petdis" = petdis,

+ "price" = price);

> parse\_data

year is\_broken mileage power engine\_capacity bt petdis price

1 2019 0 0.002 125 1.6 1 0 1569.000

2 2018 0 0.001 150 1.4 1 0 1680.000

3 2013 0 250.000 140 2.0 1 1 800.000

4 2018 0 0.001 220 2.0 1 0 2390.000

5 2012 0 101.000 170 2.0 1 0 790.000

6 2016 0 26.437 150 1.4 1 0 1004.700

7 2014 0 63.000 150 1.4 1 0 820.000

8 2012 0 111.270 122 1.4 1 0 536.300

9 2009 0 150.000 150 1.4 1 0 485.000

10 2018 0 0.001 150 2.0 1 1 2200.000

11 2017 0 32.348 150 1.4 2 0 1623.837

12 2017 0 35.000 150 1.4 1 0 1690.000

13 2018 0 0.001 150 2.0 1 1 2130.000

14 2018 0 0.001 150 2.0 1 1 2130.000

15 2018 0 0.001 220 2.0 1 0 2270.000

16 2018 0 0.001 220 2.0 1 0 2290.000

17 2011 0 123.000 170 2.0 1 0 765.000

18 2017 0 40.000 150 1.4 1 0 1400.000

19 2013 0 74.904 140 2.0 1 1 990.000

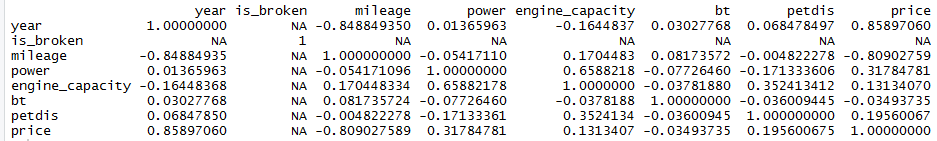
20 2009 0 140.000 150 1.4 1 0 560.000

Вычислим попарные коэффициенты корреляции между предикторами:

> # вычислим попарные коэффициенты корреляции между предикторами

> r = cor(parse\_data)

> r



Как видим, имеется сильная отрицательная корреляция между пробегом и годом выпуска (-0.848), сильная положительная корреляция между годом выпуска и ценой (0.858), сильная отрицательная корреляция между пробегом и ценой(-0.809).

> # построим регрессионную модель

> mymodel = lm(price ~ year + is\_broken + mileage + power + engine\_capacity + bt + petdis, data = parse\_data)

> mymodel

Call:

lm(formula = price ~ year + is\_broken + mileage + power + engine\_capacity +

bt + petdis, data = parse\_data)

Coefficients:

(Intercept) year is\_broken mileage power engine\_capacity

-1.924e+05 9.557e+01 NA -2.492e+00 8.330e+00 -7.835e+01

bt petdis

6.444e+00 3.856e+02

> summary(mymodel)

Call:

lm(formula = price ~ year + is\_broken + mileage + power + engine\_capacity +

bt + petdis, data = parse\_data)

Residuals:

Min 1Q Median 3Q Max

-366.31 -112.62 -4.56 125.00 601.83

Coefficients: (1 not defined because of singularities)

Estimate Std. Error t value Pr(>|t|)

(Intercept) -1.924e+05 1.837e+04 -10.471 < 2e-16 \*\*\*

year 9.557e+01 9.106e+00 10.495 < 2e-16 \*\*\*

is\_broken NA NA NA NA

mileage -2.492e+00 5.275e-01 -4.724 6.00e-06 \*\*\*

power 8.330e+00 1.150e+00 7.243 3.63e-11 \*\*\*

engine\_capacity -7.835e+01 9.146e+01 -0.857 0.393

bt 6.444e+00 4.998e+01 0.129 0.898

petdis 3.856e+02 6.644e+01 5.804 4.81e-08 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 172.2 on 128 degrees of freedom

Multiple R-squared: 0.8903, Adjusted R-squared: 0.8851

F-statistic: 173.1 on 6 and 128 DF, p-value: < 2.2e-16

Заметим, что коэффициент детерминации = 0.8903, «исправленный» коэффициент детерминации = 0.8851.

Построим регрессионную модель, где предикторы год, признак битый / нет, пробег.

> # построим регрессионную модель

> mymodel = lm(price ~ year + is\_broken + mileage, data = parse\_data)

> mymodel

Call:

lm(formula = price ~ year + is\_broken + mileage, data = parse\_data)

Coefficients:

(Intercept) year is\_broken mileage

-196277.01 98.10 NA -2.63

> summary(mymodel)

Call:

lm(formula = price ~ year + is\_broken + mileage, data = parse\_data)

Residuals:

Min 1Q Median 3Q Max

-544.05 -165.47 -0.29 117.15 721.74

Coefficients: (1 not defined because of singularities)

Estimate Std. Error t value Pr(>|t|)

(Intercept) -1.963e+05 2.587e+04 -7.587 5.22e-12 \*\*\*

year 9.810e+01 1.282e+01 7.651 3.70e-12 \*\*\*

is\_broken NA NA NA NA

mileage -2.630e+00 7.409e-01 -3.549 0.000536 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 250.4 on 132 degrees of freedom

Multiple R-squared: 0.7607, Adjusted R-squared: 0.757

F-statistic: 209.8 on 2 and 132 DF, p-value: < 2.2e-16

Заметим, что коэффициент детерминации уменьшился до 0.7607, «исправленный» коэффициент детерминации уменьшился до 0.757. Поэтому воспользуемся первой моделью.

> # спрогнозируем цену автомобилей по нашей модели

> predictPrice = predict(mymodel, predict\_data)

> predictPrice = as.data.frame(predictPrice)

> predictPrice

predictPrice

1 1786.9558

2 1688.8584

3 540.9711

4 1688.8584

5 834.6753

6 1423.1433

7 1130.7987

8 807.6698

9 411.5269

10 1688.8584

> # истинная цена автомобилей

> truePrice = parse\_data$price

> truePrice

1569.000 1680.000 800.000 2390.000 790.000 1004.700 820.000 536.300 485.000 2200.000 1623.837 1690.000 2130.000 2130.000 2270.000

[16] 2290.000 765.000 1400.000 990.000 560.000

Вычислим ошибку прогноза в процентах. Найдем, на сколько процентов отличается разница между истинной и предсказанной ценами автомобилей.

> # ошибка прогноза

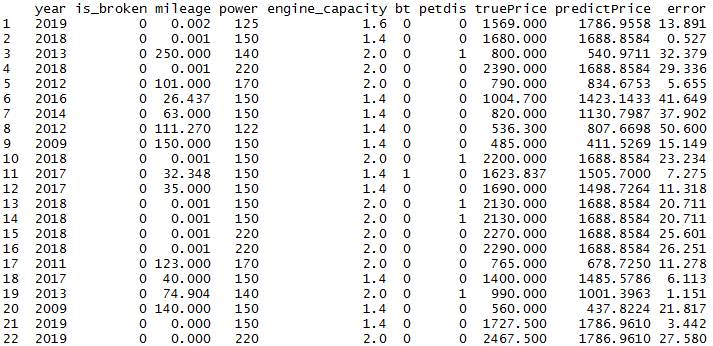
> error = abs(truePrice - predictPrice)\*100/truePrice;

> error= round(error, 3)

> colnames(error) = "error"

> max(error)

[1] 64.006



Выберем те автомобили, для которых ошибка прогноза составила больше 50%:

> for(i in 1:nrow(error))

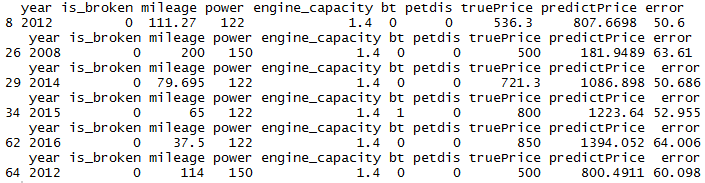
+ {

+ if(error[i,] > 50)

+ print(tableResults[i,]);

+ }

Получим 6 автомобилей:



Самая большая ошибка прогноза (64.006%) у автомобиля 62.

**Код программы:**

# назначим рабочий директорий

setwd("C:/Users/admin.WIN-OSD1NEIENB0/Documents/Rprojects/task3")

library(rvest)

library(stringr)

main\_url = 'https://www.avito.ru'

vw\_url= 'https://www.avito.ru/kazan/avtomobili/volkswagen/tiguan?radius=200'

short\_url = str\_replace(vw\_url, 'radius=200', '');

pages <- read\_html(vw\_url)

pages

write\_xml(pages, "pages.txt")

# функция для получения номера последней страницы

get\_last\_page\_numb <- function(html){

pages\_data <- html %>%

html\_nodes('.pagination-page') %>%

html\_text()

pages\_data[(length(pages\_data)-1)] %>%

unname() %>%

as.numeric()

}

# получим номер последней страницы

(last\_page\_numb <- get\_last\_page\_numb(pages))

# получим список страниц

list\_of\_pages = character();

list\_of\_pages = append(list\_of\_pages, vw\_url)

list\_of\_pages = append(list\_of\_pages, str\_c(short\_url, paste0('p=', 2:last\_page\_numb, '&radius=200')));

list\_of\_pages

year = numeric(); #год выпуска

price = numeric(); #цена

info = character(); #характеристики

ref\_info = character(); #ссылки

is\_broken = numeric(); #признак, является ли автомобиль битым, (1 - является, 0 - нет)

# color = character(); #цвет автомобиля

for(i in 1:last\_page\_numb)

{

page <- read\_html(list\_of\_pages[i]);

results <- page %>% html\_nodes(".item\_table-description")

#hide-elements

# write\_xml(page, "page.txt")

Sys.sleep(1)

#достаем год выпуска

tmp <- results %>% html\_nodes(".item-description-title")%>% html\_text()%>%str\_trim()

year\_tmp = as.numeric(str\_extract(tmp, "[:digit:]+"));

year = append(year, year\_tmp);

Sys.sleep(1)

#вытаскиваем цену автомобиля

price\_tmp <- results %>% html\_nodes(".about") %>% html\_nodes(".price") %>% html\_text()

#\_bold-price

price\_tmp = as.numeric(str\_replace\_all(price\_tmp, "[ \n\r\t\\\u20bd]+", ""))

price = append(price, price\_tmp);

Sys.sleep(1)

#вытаскиваем информацию о характеристиках автомобиля

info\_tmp <- results %>% html\_nodes(".about")%>% html\_nodes(".specific-params\_block")%>% html\_text()%>%str\_trim()

#\_bold-price

info\_tmp = str\_replace\_all(info\_tmp, "[ ]+", "");

info = append(info, info\_tmp);

Sys.sleep(1)

#вытаскиваем ccылки на подробную информацию об автомобиле

ref <- results %>% html\_nodes(".item-description-title-link") %>% html\_attr("href")

ref <- paste0(main\_url, ref)

ref\_info = append(ref\_info, ref)

}

write\_xml(page, "page.txt")

#год выпуска

year

#цена

price

#информация о характеристиках автомобиля

info = str\_replace\_all(info, "\"", "")

info

#ccылки на подробную информацию об автомобиле

ref\_info = str\_replace\_all(ref\_info, "\"", "")

ref\_info

#заполняем признак, является ли автомобиль битым, 1 - битый, 0 - нет

is\_broken = ifelse(substr(info, 0, 5) == 'Битый', 1, 0)

is\_broken

info = str\_replace(info, 'Битый,', '')

info

split\_info <- str\_split(info, ',');

split\_info

mileage = numeric(); #пробег, км

engine\_capacity = numeric(); #объем двигателя, л

power = numeric(); #мощность л.с.

transmission = character(); #трансмиссия (механика/автомат)

body\_type = character(); #тип кузова

drive\_unit = character(); #привод

engine\_type = character(); #тип двигателя

for(i in 1:length(split\_info))

{

str\_info = split\_info[[i]];

index = 1;

if(length(str\_info) == 5)

{

mileage = append(mileage, as.numeric(str\_extract(str\_info[1], "[:digit:]+")));

index = 2;

}

else

{

mileage = append(mileage, 0);

}

k = str\_extract(str\_info[index], "[(][:digit:]+");

k = as.numeric(str\_replace(k, '[(]', ''));

power = append(power, k);

y = str\_extract(str\_info[index], "[MA]?T");

transmission = append(transmission, y);

z = str\_extract(str\_info[index], "[:digit:]+[.][:digit:]+");

engine\_capacity = append(engine\_capacity, as.numeric(z))

body\_type = append(body\_type, str\_info[index+1])

drive\_unit = append(drive\_unit, str\_info[index+2])

engine\_type = append(engine\_type, str\_info[index+3])

}

#пробег, км

mileage

#мощность л.с.

power

#трансмиссия (механика/автомат)

transmission

#объем двигателя, л

engine\_capacity

#тип кузова

body\_type

#привод

drive\_unit

#тип двигателя

engine\_type

petdis = numeric(); #тип двигателя - бензин(0) / дизель (1)

for(i in 1:length(engine\_type))

{

if(engine\_type[i] == 'бензин')

{

petdis[i] = 0;

}

else

{

petdis[i] = 1;

}

}

petdis

bt = numeric(); #тип кузова - хетчбэк, внедорожник(0) / универсал, минивэн(1)

for(i in 1:length(body\_type))

{

if(body\_type[i] == 'хетчбэк' || body\_type[i] == 'внедорожник')

{

bt[i] = 0;

}

else

{

bt[i] = 1;

}

}

bt

mileage = mileage / 1000;

price = price / 1000;

# сохраним преобразованные поля как data.frame

parse\_data = data.frame("year" = year, "is\_broken" = is\_broken, "mileage" = mileage, "power" = power,

"engine\_capacity" = engine\_capacity, "bt" = bt, "petdis" = petdis,

"price" = price);

parse\_data

# информация о данных

summary(parse\_data)

# вычислим попарные коэффициенты корреляции между предикторами

r = cor(parse\_data)

r

# построим регрессионную модель

mymodel = lm(price ~ year + is\_broken + mileage + power + engine\_capacity + bt + petdis, data = parse\_data)

mymodel

summary(mymodel)

# построим регрессионную модель

mymodel = lm(price ~ year + is\_broken + mileage, data = parse\_data)

mymodel

summary(mymodel)

# данные для прогноза

predict\_data = parse\_data

predict\_data = predict\_data[-8]

predict\_data

# спрогнозируем цену автомобилей по нашей модели

predictPrice = predict(mymodel, predict\_data)

predictPrice = as.data.frame(predictPrice)

predictPrice

# истинная цена автомобилей

truePrice = parse\_data$price

truePrice

# ошибка прогноза

error = abs(truePrice - predictPrice)\*100/truePrice;

error= round(error, 3)

colnames(error) = "error"

max(error)

# таблица результата прогноза

tableResults = data.frame(predict\_data, truePrice, predictPrice, error)

tableResults

for(i in 1:nrow(error))

{

if(error[i,] > 50)

print(tableResults[i,]);

}