МИНОБРНАУКИ РОССИИ

РГУ НЕФТИ И ГАЗА (НИУ) ИМЕНИ И.М. ГУБКИНА

ФАКУЛЬТЕТ АВТОМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

КАФЕДРА ИНФОРМАТИКИ

ДИСЦИПЛИНА «ОСНОВЫ АНАЛИЗА БОЛЬШИХ ДАННЫХ И МАШИННОЕ ОБУЧЕНИЕ»

О Т Ч Е Т

по Домашнему Заданию 2

«Анализ данных в среде R»

Выполнила: студентка группы АА-19-05

Данилова М.А.

Проверила: доцент Вишневская Е. А.

Москва 2022

Выбрать тему из предлагаемого списка: 9. Регрессионный анализ

<u>Регрессионный анализ: основные положения.</u>

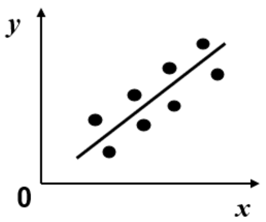
Регрессионные модели используются для прогнозирования непрерывных целевых значений (прогнозирования цен на жилье, прогноза погоды). Под регрессией понимается функциональная зависимость между объясняющими переменными xi и условным математическим ожиданием (средним значением) зависимой переменой y; модель строится с целью прогнозирования этого среднего значения при фиксированных значениях первых. Любая регрессионная модель позволяет обнаружить только количественные зависимости, которые не обязательно отражают причинные. В ходе регрессионного анализа определяют коэффициенты регрессии (β) - величины для каждой независимой переменной, которые представляют силу и тип взаимосвязи независимой переменной по отношению к зависимой.

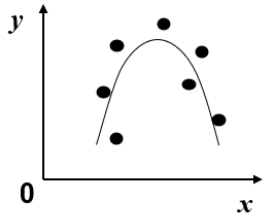
В общем случае, предположим, что мы наблюдаем количественный отклик Y, и несколько разных предикторов X1, X2 , . . .,Xp . Предположим, что есть какая-то взаимосвязь между Y и X = (X1, X2 ,...,Xp), которая в общей форме может быть записана в виде

$Y = f(X) + \epsilon$.

Здесь f – это некоторая фиксированная неизвестная функция переменных X1, X2 ,...,Xp , и ϵ – случайная ошибка, не зависящая от X, с нулевым матожиданием.

Выбор формулы связи переменных называется **спецификацией** уравнения регрессии. В случае парной регрессии выбор формулы обычно осуществляется по графическому изображению реальных статистических данных.





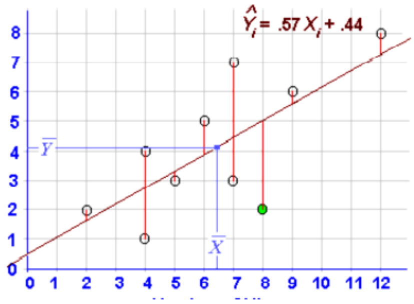
<u>Парная (простая) линейная регрессия</u>. Этот подход для прогнозирования количественного отклика Y на основе единственной предикторной переменной X. Предполагается, что есть приблизительная линейная взаимосвязь между Y и X. Математически можно записать эту взаимосвязь следующим образом:

$Y_i = a + bX_i$, где

Y_i – зависимая переменная и X_i – независимая переменная, a – константа, b – угловой коэффициент, характеризует наклон прямой и показывает, на какую величину в среднем изменится результативный признак Y_i, если переменная X_i увеличится на единицу своего измерения.

Для определения наилучшей линии регрессии используют **метод наименьших квадратов**, те добиваются, чтобы сумма квадратов остатков e была минимальной. Под остатками понимается разность между очередным наблюдением и прогнозом модели.

$\sum e_i^2$ - минимальна



$$\sum_{i=1}^{n} (y_i - (a + b \cdot x_i))^2 \Rightarrow \min_{a,b}$$

В общем случае для n наблюдений решают систему уравнений:

$$\begin{cases} a \cdot n \quad + b \cdot \sum x = \sum y; \\ a \cdot \sum x + b \cdot \sum x^2 = \sum x \cdot y. \end{cases}$$

Толковой интерпретации регрессионных коэффициентов мешает также различие в единицах измерения. Например, если предиктор измеряется в сантиметрах, его вес будет в 100 раз отличаться по весу от предиктора, берущегося в метрах. Чтобы избежать такого, мы должны **стандартизировать** единицы измерения предикторных переменных перед тем, как проводить регрессионный анализ. Стандартизация – это выражение переменных в процентилях.

При использовании линейной регрессии в качестве показателем тесноты связи выступает линейный **коэффициент корреляции** (чем ближе коэффициент по модулю к единице, тем теснее связь).

<u>Множественная регрессия</u> является расширением простой линейной регрессии. Она исследует влияние двух и более предикторов на критерий (Y=B1*X1+B2*X2+B3*X3+…+A).

Построение уравнения множественной регрессии начинается с решения вопроса о спецификации модели, который включает 2 круга вопросов: отбор факторов и выбор уравнения регрессии.

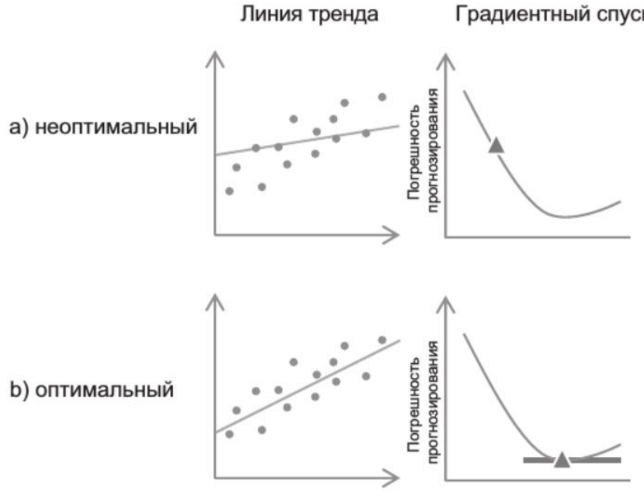
Факторы, включаемые во множественную регрессию, должны отвечать следующим требованиям:

1. Они должны быть **количественно измеримы**. Если необходимо включить в модель качественный фактор, не имеющий количественного измерения, то ему нужно придать количественную определенность.
2. Каждый **фактор должен быть достаточно тесно связан с результатом** (т.е. коэффициент парной линейной корреляции между фактором и результатом должен быть существенным).
3. **Факторы не должны быть сильно коррелированы друг с другом** или находиться в строгой функциональной связи (т.е. они не должны быть интеркоррелированы). Мультиколлинеарность может привести к нежелательным последствиям.

Существуют различные подходы преодоления сильной межфакторной корреляции. Простейший из них – исключение из модели факторов, в наибольшей степени ответственных за мультиколлинеарность. Определение факторов, ответственных за мультиколлинеарность, может быть основано на анализе матрицы межфакторной

корреляции. При этом определяют пару признаков-факторов, которые сильнее всего связаны между собой (коэффициент линейной парной корреляции максимален по модулю). Из этой пары в наибольшей степени ответственным за мультиколлинеарность будет тот признак, который теснее связан с другими факторами модели (имеет более высокие по модулю значения коэффициентов парной линейной корреляции).
Коэффициенты VIF (variance inflation factor) показывают, насколько сильно связаны друг с другом регрессоры модели. Если коэффициенты VIF для всех регрессоров оказались меньше 10 (иногда используют 5), это значит, что существенной мультиколлинеарности в модели не наблюдается. В противном случае стоит сделать вывод о том, что в модели есть мультиколлинеарность.
4. **Отсутствие автокорреляции** – отсутствие независимости остатков. Выявляется с помощью теста Дурбина-Уотсона (обнаруживает автокорреляцию первого порядка).
– Если d=2 – отсутствие автокорреляции.
При выборе формы уравнения множественной регрессии предпочтение отдается линейной функции в виду четкой интерпретации параметров. Параметры уравнения множественной регрессии можно также оценить методом наименьших квадратов, составив и решив систему нормальных линейных уравнений.

Градиентный спуск (gradient descent) используется в случаях, когда параметры уравнения нельзя получить путем решения систем уравнений. Алгоритм градиентного спуска делает первоначальное предположение о наборе весовых составляющих, после чего начинается итеративный процесс их применения к каждому элементу данных для прогнозирования, а затем они перенастраиваются для снижения общей ошибки прогнозирования.
Этот процесс можно сравнивать с пошаговым спуском в овраг в поисках дна. На каждом этапе алгоритм определяет, какое направление даст наиболее крутой спуск, и пересчитывает весовые составляющие. В конечном итоге мы достигнем самой нижней позиции, которая представляет собой точку, в которой погрешность прогнозирования минимальна. Рисунок показывает, как оптимальная линия тренда регрессии соответствует нижней точки градиента.



Кроме регрессии градиентный спуск может также использоваться для оптимизации параметров в других моделях, таких как метод опорных векторов или в нейронных сетях.

<u>Оценка качества уравнения регрессии</u>.

Коэффициент детерминации рассматривают в качестве основного показателя, отражающего меру качества регрессионной модели. Он показывает, какая доля вариации объясняемой переменной y учтена в модели и обусловлена влиянием на нее факторов, включенных в модель:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2},$$

y_i - значения наблюдаемой переменной, $\bar{y}$– среднее значение по наблюдаемым данным, $\hat{y}_i$– модельные значения, построенные по оцененным параметрам.
Чем ближе R-квадрат к 1, тем выше качество регрессионной модели (факторы сильнее влияют на результат).

<u>Значимость уравнения регрессии и отдельных параметров</u>.

Проверить значимость уравнения регрессии – значит установить, соответствует ли аналитическая модель экспериментальным данным, и достаточно ли включенных в уравнение объясняющих переменных для описания зависимой переменной. Оценка значимости уравнения регрессии в целом производится на основе **F-критерия Фишера** (чем больше значение параметра — тем лучше).
Для проверки значимости коэффициента регрессии **применяется t -распределение Стьюдента** (если есть основания считать, что между величинами Y и X нет линейной зависимости, то коэффициент статистически незначим-слишком близок к 0).

Если между изучаемыми явлениями существуют нелинейные соотношения, то они выражаются с помощью соответствующих нелинейных функций (полиномы различных степеней, гипербола, степенная, показательная, экспоненциальная регрессии и тд).

<u>Для сравнения регрессионных моделей по степени точности предсказаний используются метрики оценки.</u>

MSE (Mean Squared Error) *и*змеряет среднюю сумму квадратной разности между фактическим значением и прогнозируемым значением для всех точек данных. Самая популярная метрика, используемая для задач регрессии. Усиливается влияние ошибок по квадратуре от исходного значения.

$$MSE = \frac{1}{n}\sum_{t=1}^{n} e_t^2, where \quad e_t = original_t - predict_t$$

Чем меньше MSE, тем точнее наше предсказание. Оптимум достигается в точке 0. Является дифференцируемой, что позволяет более эффективно использовать для поиска экстремумов с помощью математических методов.

Root Mean Squared Error (RMSE) **-** *к*орень от квадратной ошибки. Ее легко интерпретировать, поскольку она имеет те же единицы, что и исходные значения (в отличие от MSE). Также она оперирует меньшими величинами по абсолютному значению, что может быть полезно для вычисления на компьютере.

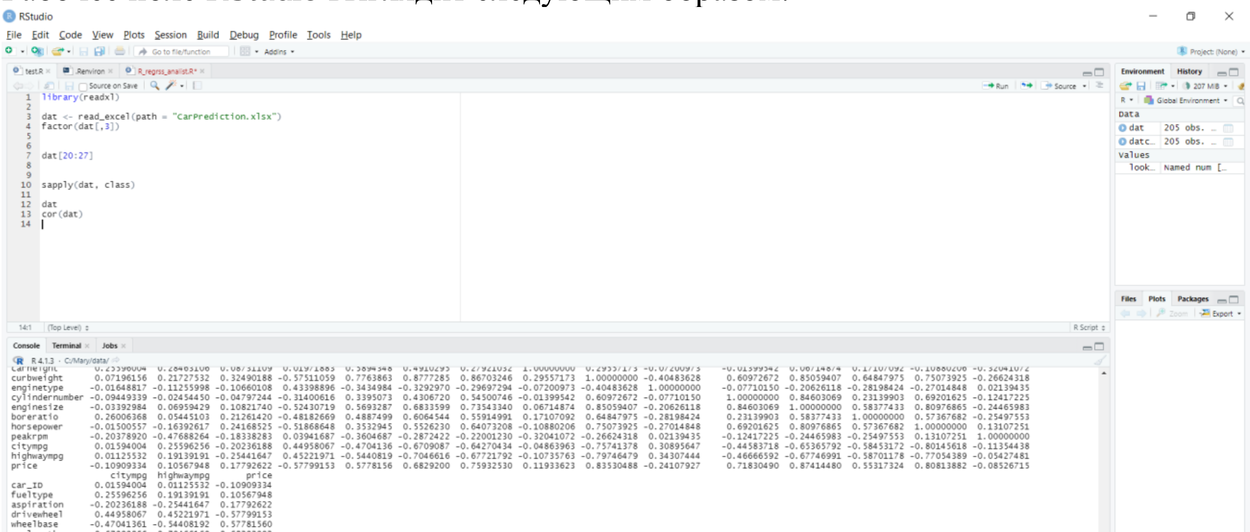
$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n} e_t^2}, \quad where \quad e_t = original_t - predict_t$$

Итак, **основными этапами регрессионного анализа являются:**
1. Выбор вида уравнения регрессии (спецификация модели).
2. Выбор независимых переменных, оказывающих существенное влияние на зависимую переменную.
3. Оценка параметров уравнения регрессии (параметризация модели).
4. Оценка статистической надежности регрессионной модели (верификация).

Данные с подходящей структурой для выбранного метода.
Для анализа используем данные автомобильной компании Geely Auto, представленные на сайте Kaggle: https://www.kaggle.com/datasets/hellbuoy/car-price-prediction
Задача будет состоять в том, чтобы определить взаимосвязь между различными параметрами автомобилей и их ценой на рынке.

Столбцы таблицы (все данные-целые числа):
car_ID – ид автомобиля
fueltype – тип топлива (1=газ, 2=дизельное)
aspiration – ускорение (стандарт, турбо)
drivewheel – ведущее колесо (переднее или заднее)
wheelbase – база шасси
carlength – длина авто
carwidth – ширина авто
carheight – высота авто
curbweight – снаряженная масса
enginetype – тип двигателя
cylindernumber – количество цилиндров
enginesize – размер двигателя
boreratio – коэффициент проходимости
horsepower – число лошадиных сил
peakrpm – пиковые обороты
citympg – ситимиль на галлон
highwaympg – шоссемиль на галлон
price – цена, целевая ячейка

Для анализа в среде R потребуется RStudio, а также инструмент RTools.
Рабочее поле RStudio выглядит следующим образом:

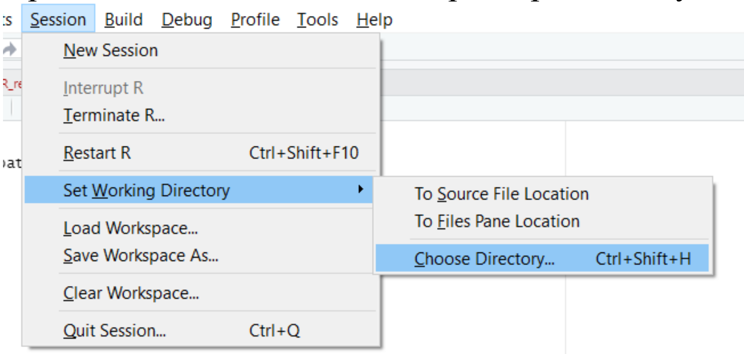


Этап 1. Чтение данных.

Для чтения данных из Excel установим необходимый пакет

```
install.packages("readxl")
library(readxl)
```

Прочитаем данные из Excel, предварительно указав рабочую директорию:




```
dat <- read_excel(path = "CarPrediction.xlsx")
```

Посмотрим на полученные данные:

```
dat
```

```
   car_ID fueltype aspiration drivewheel wheelbase carlength carwidth carheight curbweight enginetype cylindernumber enginesize
    <dbl>    <dbl>      <dbl>      <dbl>     <dbl>     <dbl>    <dbl>     <dbl>      <dbl>      <dbl>          <dbl>      <dbl>
 1      1        1          1          1      88.6      169.     64.1      48.8       2548          3              4        130
 2      2        1          1          1      88.6      169.     64.1      48.8       2548          3              4        130
 3      3        1          1          1      94.5      171.     65.5      52.4       2823          5              6        152
 4      4        1          1          2      99.8      177.     66.2      54.3       2337          7              4        109
 5      5        1          1          3      99.4      177.     66.4      54.3       2824          7              5        136
 6      6        1          1          2      99.8      177.     66.3      53.1       2507          7              5        136
 7      7        1          1          2     106.       193.     71.4      55.7       2844          7              5        136
 8      8        1          1          2     106.       193.     71.4      55.7       2954          7              5        136
 9      9        1          2          2     106.       193.     71.4      55.9       3086          7              5        131
10     10        1          2          3      99.5      178.     67.9      52         3053          7              5        131
# ... with 195 more rows, and 6 more variables: boreratio <dbl>, horsepower <dbl>, peakrpm <dbl>, citympg <dbl>, highwaympg <dbl>
#   price <dbl>
> |
```

Из-за большого числа столбцов не все из них отобразились. Можно указать в квадратных скобках число, указывающее номера столбцов таблицы.

```
dat[12:18]
```

```
   enginesize boreratio horsepower peakrpm citympg highwaympg  price
        <dbl>     <dbl>      <dbl>   <dbl>   <dbl>      <dbl>  <dbl>
 1        130      3.47        111    5000      21         27 13495
 2        130      3.47        111    5000      21         27 16500
 3        152      2.68        154    5000      19         26 16500
 4        109      3.19        102    5500      24         30 13950
 5        136      3.19        115    5500      18         22 17450
 6        136      3.19        110    5500      19         25 15250
 7        136      3.19        110    5500      19         25 17710
 8        136      3.19        110    5500      19         25 18920
 9        131      3.13        140    5500      17         20 23875
10        131      3.13        160    5500      16         22 17859.
# ... with 195 more rows
```

Этап 2. Очистка данных.

Проверим типы данных в столбцах

```
sapply(dat, class)
```

```
      car_ID       fueltype     aspiration     drivewheel      wheelbase      carlength       carwidth 
carheight      curbweight     enginetype cylindernumber 
   "numeric"      "numeric"      "numeric"      "numeric"      "numeric"      "numeric"      "numeric" 
"numeric"      "numeric"      "numeric"      "numeric" 
  enginesize      boreratio     horsepower        peakrpm        citympg     highwaympg          price 
   "numeric"      "numeric"      "numeric"      "numeric"      "numeric"      "numeric"      "numeric" 
```

Все столбцы имеют тип numeric – число.

Удалим столбец с идентификатором, поскольку этот фактор не должен влиять на цену авто

```
cars <- dat[,-1]
cars
```

fueltype	aspiration	drivewheel	wheelbase	carlength	carwidth	carheight	curbweight	enginetype	cylindernumber	enginesize	boreratio	horsepower	peakrpm	citympg	highwaympg	price
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	1	1	88.6	169.	64.1	48.8	2548	3	4	130	3.47	111	5000	21	27	13495
1	1	1	88.6	169.	64.1	48.8	2548	3	4	130	3.47	111	5000	21	27	16500
1	1	1	94.5	171.	65.5	52.4	2823	5	6	152	2.68	154	5000	19	26	16500
1	1	2	99.8	177.	66.2	54.3	2337	7	4	109	3.19	102	5500	24	30	13950
1	1	3	99.4	177.	66.4	54.3	2824	7	5	136	3.19	115	5500	18	22	17450
1	1	2	99.8	177.	66.3	53.1	2507	7	5	136	3.19	110	5500	19	25	15250
1	1	2	106.	193.	71.4	55.7	2844	7	5	136	3.19	110	5500	19	25	17710
1	1	2	106.	193.	71.4	55.7	2954	7	5	136	3.19	110	5500	19	25	18920
1	2	2	106.	193.	71.4	55.9	3086	7	5	131	3.13	140	5500	17	20	23875
1	2	3	99.5	178.	67.9	52	3053	7	5	131	3.13	160	5500	16	22	17859.

... with 195 more rows

Проверим, есть ли «пустые» ячейки в данных (NA):

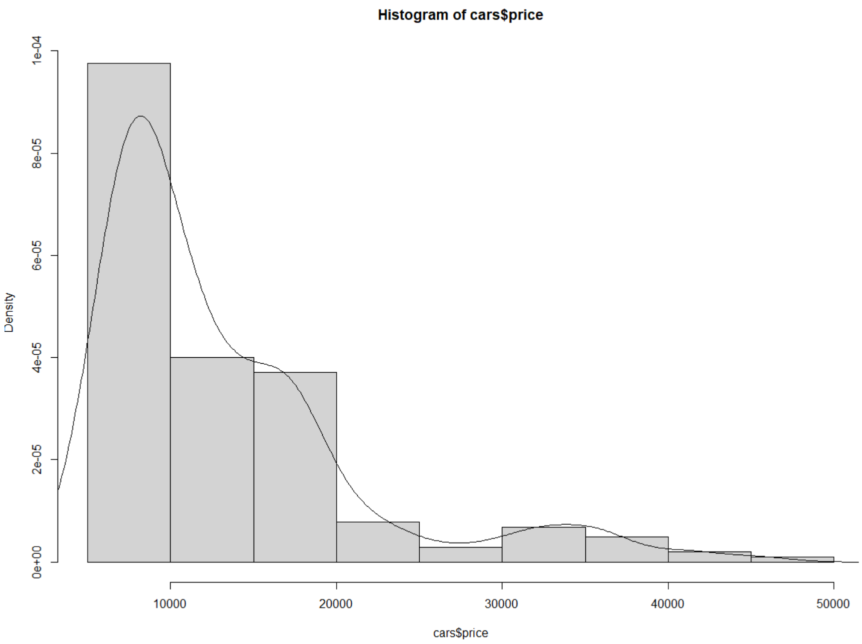
```
find_na <- function(data){
  sum =0
  for(i in 1:nrow(data)){
    for (j in 1:ncol(data))
    {
    sum<-sum+is.na(cars[i,j])
    }
  }
  print(sum)
}

find_na(cars)
```

Результат: 0
те таких данных нет, все значения представляют собой числа.

Этап 3. Визуализация данных. Выбор независимых переменных

Отобразим распределение цены авто с помощью hist

```
hist(cars$price, freq=F)
lines(density(cars$price))
```



Поскольку в таблице много столбцов, для выявления менее значимых факторов посчитаем матрицу взаимных корреляций всех переменных между собой и округлим результат до двух цифр после запятой. В данном случае нас будет интересовать последний столбец – корреляция цены с другими параметрами.
Для этого воспользуемся пакетом

```
install.packages("psych")
```

```
correl <- round(cor(cars), 2)
correl[,ncol(correl)]
```

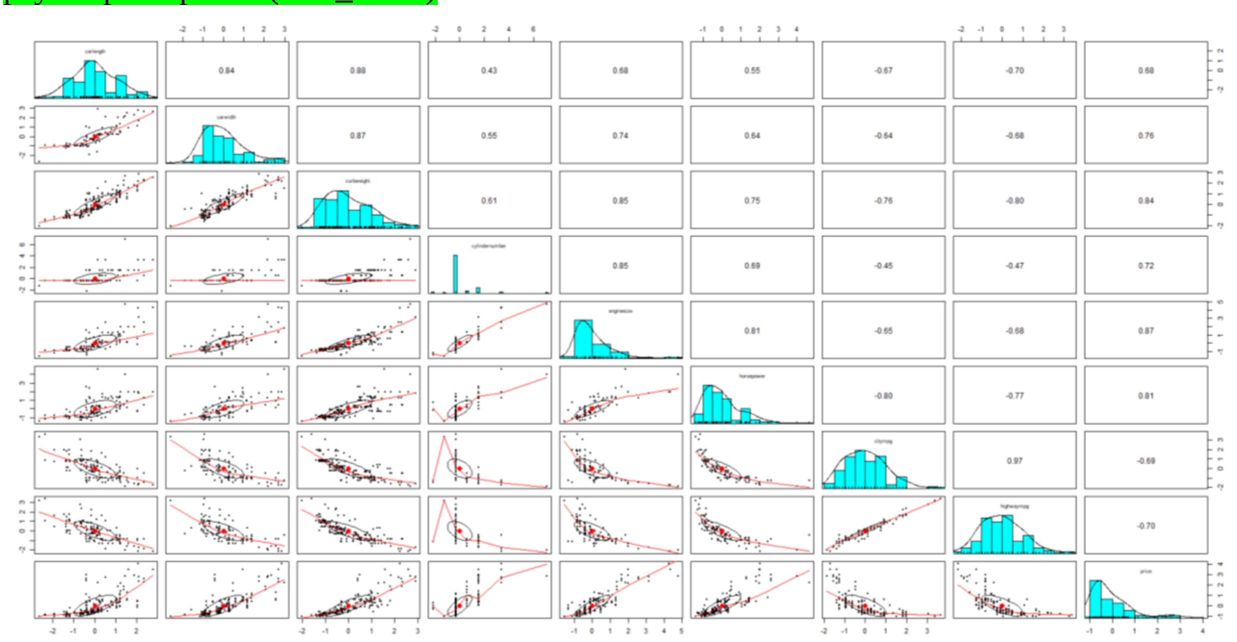
```
     fueltype     aspiration     drivewheel      wheelbase      carlength
         0.11           0.18          -0.58           0.58           0.68
     carwidth      carheight     curbweight     enginetype cylindernumber
         0.76           0.12           0.84          -0.24           0.72
   enginesize      boreratio     horsepower        peakrpm        citympg
         0.87           0.55           0.81          -0.09          -0.69
   highwaympg          price
        -0.70           1.00
```

Видно, что столбцы fueltype, aspiration, carheight, enginetype, peakrpm, drivewheel, wheelbase, boreratio по модулю меньше 0.6, что говорит о слабо-умеренной связи с ценой, поэтому также исключим эти факторы из рассмотрения.

```
cars_clean <- cars[,c(5:6,8,10:11,13, 15:17)]
```

Альтернативный способ представления данных - использование функции pairs.panels пакета psych, которая возвращает одновременно и диаграммы распределения данных, и значения коэффициентов корреляции.

```
psych::pairs.panels(cars_clean)
```



Видно, что все параметры имеют нормальное распределение, что также соответствует требованиям регрессии.

<u>Этап 4. Подготовка данных.</u>

Поскольку некоторые представленные параметры отличаются в 1000 раз (имеют разные единицы измерения), их необходимо стандартизировать.

```
cars_sc <- scale(cars_clean)
cars_sc
```

	carlength	carwidth	curbweight	cylindernumber	enginesize	horsepower	citympg	highwaympg	price
[1,]	-0.42547990	-0.84271939	-0.014530711	-0.3520252	0.07426712	0.17405669	-0.64497414	-0.54472526	0.027324254
[2,]	-0.42547990	-0.84271939	-0.014530711	-0.3520252	0.07426712	0.17405669	-0.64497414	-0.54472526	0.403473402
[3,]	-0.23094769	-0.19010076	0.513624571	1.4983638	0.60257108	1.26144842	-0.95068443	-0.68993810	0.403473402
[4,]	0.20674978	0.13620856	-0.419769855	-0.3520252	-0.43002303	-0.05353693	-0.18640871	-0.10908672	0.084278617
[5,]	0.20674978	0.22943979	0.515545136	0.5731693	0.21835002	0.27520941	-1.10353957	-1.27078948	0.522389106
[6,]	0.26348834	0.18282417	-0.093273862	0.5731693	0.21835002	0.14876851	-0.95068443	-0.83515095	0.247005370
[7,]	1.51173669	2.56022061	0.553956429	0.5731693	0.21835002	0.14876851	-0.95068443	-0.83515095	0.554934457
[8,]	1.51173669	2.56022061	0.765218542	0.5731693	0.21835002	0.14876851	-0.95068443	-0.83515095	0.706395512
[9,]	1.51173669	2.56022061	1.018733078	0.5731693	0.09828094	0.90741391	-1.25639471	-1.56121517	1.326634789
[10,]	0.33643792	0.92867404	0.955354444	0.5731693	0.09828094	1.41317750	-1.40924986	-1.27078948	0.573606350
[11,]	0.22296080	-0.51641007	-0.308377105	-0.3520252	-0.45403684	-0.07882511	-0.33926385	-0.25429957	0.394711192
[12,]	0.22296080	-0.51641007	-0.308377105	-0.3520252	-0.45403684	-0.07882511	-0.33926385	-0.25429957	0.456672533
[13,]	0.22296080	-0.51641007	0.296600764	1.4983638	0.89073688	0.42693849	-0.64497414	-0.39951241	0.963003084
[14,]	0.22296080	-0.51641007	0.402231821	1.4983638	0.89073688	0.42693849	-0.64497414	-0.39951241	0.979901631
[15,]	1.21183287	0.46251787	0.959195573	1.4983638	0.89073688	0.42693849	-0.79782928	-0.83515095	1.413005143
[16,]	1.21183287	0.46251787	1.295294390	1.4983638	1.97135863	1.96951746	-1.40924986	-1.27078948	2.188460708
[17,]	1.60089729	0.92867404	1.583379089	1.4983638	1.97135863	1.96951746	-1.40924986	-1.27078948	3.509676768
[18,]	1.86027356	2.32714253	1.823449672	1.4983638	1.97135863	1.96951746	-1.56210500	-1.56121517	2.954528191
[19,]	-2.67070582	-2.61411281	-2.050329255	-1.2772198	-1.58268622	-1.41909864	3.32925959	3.23080872	-1.017131151
[20,]	-1.47109053	-1.07579747	-1.308991295	-0.3520252	-0.88628554	-0.86275868	1.95356330	1.77868026	-0.873931609
[21,]	-1.23603077	-1.07579747	-1.241771531	-0.3520252	-0.88628554	-0.86275868	1.95356330	1.77868026	-0.838882770
[22,]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Этап 5. Разделение данных на наборы для обучения и тестирования.

Используем для построения модели около 70% представленных данных, и для тестирования качества модели в дальнейшем оставим около 30%.

train <- 1:140
test <- 141:(nrow(cars_sc))

Этап 6. Построение линейной модели.

В качестве модели выберем линейную множественную регрессию. Возьмем 6 параметров с наибольшим коэффициентов корреляции с ценой:

model <- lm(price ~ horsepower+enginesize+carwidth+carlength+curbweight+cylindernumber, data = as.data.frame(cars_sc[train,]))

До знака тильды в функции указывается целевая переменная, после – список выбранных факторов.

Чтобы посмотреть сведения о линейной аппроксимации используется функция summary(model)

А также сопоставим качество прогноза с истинным значением

pred <- predict(model, cars_clean[test,])
cor(pred, cars_clean$price[test])

```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)     0.04346    0.04028   1.079   0.2826    
horsepower      0.30821    0.06828   4.514 1.39e-05 ***
enginesize      0.56910    0.12771   4.456 1.75e-05 ***
carwidth        0.15089    0.08121   1.858   0.0654 .  
carlength       0.05934    0.08810   0.674   0.5017    
curbweight     -0.01079    0.14411  -0.075   0.9404    
cylindernumber -0.07165    0.08213  -0.872   0.3846    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4705 on 133 degrees of freedom
Multiple R-squared:  0.836,	Adjusted R-squared:  0.8286 
F-statistic:   113 on 6 and 133 DF,  p-value: < 2.2e-16

> pred <- predict(model, cars_clean[test,])
> cor(pred, cars_clean$price[test])
[1] 0.7366096
> 
```

Перейдем теперь к расшифровке полученных результатов.

Intercept — точка пересечения прямой с осью координат, те остаточный член.

Estimate – коэффициенты линейной регрессии.

R-squared — коэффициент детерминации; указывает, насколько тесной является связь между факторами регрессии и зависимой переменной. Чем ближе к 1, тем ярче выражена зависимость. В данном случае равен 0.836, что является неплохим результатом.
F-statistic — используется для оценки значимости модели регрессии в целом (чем больше значение параметра, тем лучше).
t value — критерий, основанный на t распределении Стьюдента. Значение параметра в линейной регрессии указывает на значимость фактора, можно считать, что при t>2 фактор является значимым для модели.
p-value — вероятность истинности нуль гипотезы, которая гласит, что независимые переменные не объясняют динамику зависимой переменной. Если значение p-value ниже порогового уровня (0.05), то нуль гипотеза ложная. Чем ниже — тем лучше.

В данном случае видно, что у некоторых переменных t-value значительно ниже 2, поэтому стоит исключить их из рассмотрения также.
Кроме того, вычислим значение vif (отвечает за мультиколлинеарность, если больше 5-10)

```
install.packages("usdm")
install.packages("car")
library(car)
car::vif(model)
```

```
> car::vif(model)
  horsepower   enginesize     carwidth    carlength   curbweight cylindernumber
    3.425295    13.321158     5.139714     5.979082    16.220994       5.550123
>
```

По результатам видно, что в модели присутствует мультиколлинеарность.
С учетом сказанного выше, исключим из модели незначимые факторы (3 последних).

```
model <- lm(price ~ horsepower+enginesize+carwidth, data =
as.data.frame(cars_sc[train,]))
summary(model)
pred <- predict(model, cars_clean[test,])
cor(pred, cars_clean$price[test])
car::vif(model)
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.04001    0.03982   1.005 0.316787
horsepower   0.29112    0.06294   4.625 8.61e-06 ***
enginesize   0.51423    0.06813   7.548 5.75e-12 ***
carwidth     0.20151    0.05303   3.800 0.000218 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4691 on 136 degrees of freedom
Multiple R-squared:  0.8333,	Adjusted R-squared:  0.8296
F-statistic: 226.6 on 3 and 136 DF,  p-value: < 2.2e-16

> pred <- predict(model, cars_clean[test,])
> cor(pred, cars_clean$price[test])
[1] 0.7853768
> car::vif(model)
horsepower enginesize   carwidth
  2.928091   3.813672   2.204699
```

Внесенные изменения повлияли на качество модели: исчезла мультиколлинеарность (vif<5), коэффициенты (кроме остаточного члена) являются значимыми (t value > 2), кроме того, качество прогнозирования улучшилось (0.78>0.73).
Остаточный член также можно исключить:

```
model2 <- lm(price ~ horsepower+enginesize+carwidth+0, data =
as.data.frame(cars_sc[train,]))
summary(model2)
pred <- predict(model2, cars_clean[test,])
cor(pred, cars_clean$price[test])
car::vif(model2)
```

```
Coefficients:
           Estimate Std. Error t value Pr(>|t|)    
horsepower  0.29397    0.06288   4.675 6.95e-06 ***
enginesize  0.51583    0.06811   7.573 4.87e-12 ***
carwidth    0.20002    0.05301   3.773 0.000239 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4691 on 137 degrees of freedom
Multiple R-squared:  0.834,	Adjusted R-squared:  0.8303 
F-statistic: 229.4 on 3 and 137 DF,  p-value: < 2.2e-16

> pred <- predict(model, cars_clean[test,])
> cor(pred, cars_clean$price[test])
[1] 0.7853819
> car::vif(model)
horsepower enginesize   carwidth 
  2.944758   3.834604   2.206691 
```

Определим с помощью дисперсионного анализа является ли отличие двух последних моделей значимым или нет.

```
anova(model2,model)
```

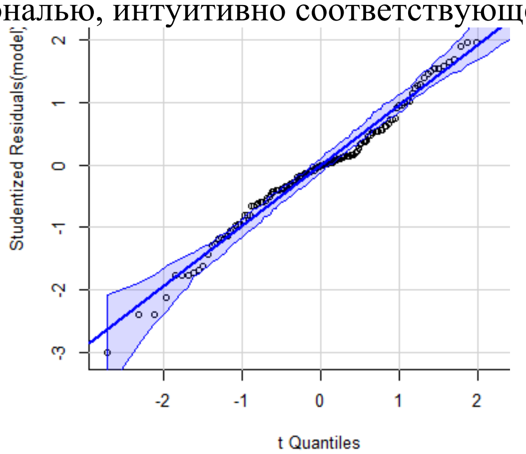
```
Model 1: price ~ horsepower + enginesize + carwidth
Model 2: price ~ horsepower + enginesize + carwidth + 0
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    136 29.930                           
2    137 30.152 -1  -0.22218 1.0096 0.3168
> 
```

Величина 0.3168>0.05, поэтому можно утверждать, что с вероятностью 95% отличие моделей не значимо и мы в праве выбрать любую модель.

После проверки значимости у полученной регрессионной модели выполняется анализ остатков (разница между прогнозируемым значением и фактическим значением), который должен следовать нормальному распределению.

```
car::qqPlot(model2, simulate = TRUE)
```

На графике остаточного QQ точки данных расположены практически по диагональной линии, стремящейся быть прямой, и непосредственно пересекаются диагональю, интуитивно соответствующей нормальному распределению.



Таким образом, полученная регрессионная модель:
price = horsepower*0.29397 + enginesize*0.51583 + carwidth*0.20002,
те цена в большей степени определяется числом лошадиных сил, шириной машины и типом двигателя.
Полученную модель можно оптимизировать, выбирая другие зависимости (нелинейные).

Список использованных источников:

1. Теоретический материал_регрессионный парн.анализ.pdf (vyatsu.ru)
2. http://main.isuct.ru/files/publ/PUBL_ALL/167.pdf
3. EconometricsWithR.pdf - Яндекс.Документы (yandex.ru)
4. https://habr.com/ru/post/207750/
5. http://qsar4u.com/files/rintro/03.html
6. https://www.kaggle.com/datasets/hellbuoy/car-price-prediction