

# Линейные парные и множественные регрессионные модели в R и Python

Справочное руководство

Турунцева М.Ю.

Зямалов В.Е.

Галеева Е.А.


Кириллова М.А.

2022-12-06

# Оглавление

<b>Предварительные замечания</b>	<b>2</b>
Установка пакетов . . . . .	3
Чтение excel-файлов . . . . .	3
Оценивание моделей . . . . .	3
Статистические тесты . . . . .	3
Графики . . . . .	4
Вывод результатов оценивания моделей . . . . .	4
Синтаксис формул . . . . .	4
<b>Сток-Уотсон. Задания E5.1, E7.1 и E8.1</b>	<b>7</b>
Загрузка данных . . . . .	8
Оценивание парной регрессии . . . . .	8
Статистические тесты . . . . .	9
Построение графиков . . . . .	9
Тесты на гетероскедастичность ошибок . . . . .	16
Тесты на автокорреляцию ошибок . . . . .	19
Тест Харке-Бера . . . . .	20
Стабильность оценок модели — Тест Чоу . . . . .	21
Взвешенный МНК . . . . .	21
Оценивание множественных регрессий . . . . .	23
<b>Сток-Уотсон. Задания E6.2 и E7.3</b>	<b>41</b>
Загрузка данных . . . . .	41
Оценки моделей . . . . .	42
<b>Сток-Уотсон. Задания E7.4 и E8.2</b>	<b>49</b>
Загрузка данных . . . . .	49
Оценки моделей . . . . .	50
<b>Список литературы</b>	<b>63</b>

# Предварительные замечания

 Данное руководство предполагает, что его читатели **знакомы** с основами работы в R и Python, умеют устанавливать пакеты, понимают базовые понятия данных языков.

R является языком, изначально разрабатывавшимся для научных вычислений. В подавляющем большинстве случаев функции из различных пакетов возвращают либо одно значение, либо составной объект, представляющий собой совокупность одно- или разнородных вложенных объектов, которые также могут быть составными. К элементам таких объектов можно обращаться при помощи одинарных квадратных скобок `[]` и двойных квадратных скобок `[[ ]]`. Если объекты **именованы**, то к ним можно также обращаться при помощи знака доллара `$`. К методам из установленных пакетов можно обращаться как напрямую, загрузив пакет при помощи `library`, либо при помощи `::`, где слева от `::` стоит имя соответствующего пакета. Для наглядности, чтобы было понятно, из какого пакета та или иная функция, я буду обращаться к ним через `::`. Например, можно загрузить пакет `car`

R

```
library(car)
```

и вызывать функции напрямую

R


```
linearHypothesis(model, ...)
```

Или же можно, не загружая пакет, вызывать функции следующим образом

R

```
car::linearHypothesis(model, ...)
```

Python — язык **общего** назначения, к которому при помощи разных пакетов добавили возможность статистических расчетов. В силу этого, работа со статистическими данными в нем не такая удобная, как в R, но ненамного. Python — язык с сильной ООП составляющей, все объекты в нем являются экземплярами какого-либо класса. Большинство **пользовательских** не встроенных в базовый синтаксис функция являются методами классов, к которым надо обращаться через точку `.`. В python для доступа к методам из установленных модулей мы **обязаны** их импортировать.

 Данное руководство предполагает, что его читатели **попытаются** повторить все описанное в нем своими руками. Повторение — мать учения.

 Когда вы будете видеть фразу “функция принимает на вход то-то и то-то” — помните, что имеется в виду **рассматриваемая** функция из **рассматриваемого** пакета. Так как и в R, и в python огромное

количество альтернативных реализаций одного и того же, то для них — альтернатив — приведенное описание может быть неприменимо!

## Установка пакетов

Для работы нам понадобятся следующие пакеты.

### Чтение excel-файлов

В R обеспечивается пакетом `readxl`.

R

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

В python — пакеты `xlrd` (старые версии Excel, расширение `xls`) и `openpyxl`. Также нам нужны будут пакеты `pandas` для хранения данных и `numpy` для некоторых технических вопросов. Иногда может быть полезен пакет `scipy`.

Консоль

```
pip install pandas numpy scipy xlrd openpyxl
```

### Оценивание моделей

В R есть встроенная функция `lm()`, которой достаточно для наших целей.

В python нужно установить пакет `statsmodels`. На самом деле пакетов в python много, но `statsmodels` позволяет работать с моделями в стиле R.

Консоль

```
pip install statsmodels
```

### Статистические тесты

В R не все так однозначно. В разных пакетах присутствуют разные тесты. Следующая команда устанавливает несколько пакетов, но конкретные нужно выбирать в зависимости от необходимых тестов.

R

```
install.packages(c("car", "aod", "lmtest", "sandwich", "skedastic",  
  "tseries", "whitestrat", "strucchange"))
```

В python для наших целей достаточно использовать модуль `statsmodels.stats.api` из установленного выше пакета `statsmodels`. Для теста Чоу надо установить соответствующий пакет.

Консоль

```
pip install chowtest
```

## Графики

В R имеется встроенная функция `plot`, которой в принципе нам достаточно. Но можно использовать пакет `ggplot2`, позволяющий создавать более сложные графики.

R

```
install.packages("ggplot2")
```

В python установим `matplotlib` и `seaborn`. Второй пакет основан на первом и позволяет тратить меньше сил и нервов при построении графиков. Например, `matplotlib` при построении линии соединяет точки в порядке их следования в наборе данных, а не в порядке возрастания абсциссы, что может привести к странным результатам. `seaborn` же это учитывает и строит то, что мы от него ожидаем. Но оба пакета работают в связке друг с другом.

Консоль

```
pip install matplotlib seaborn
```

## Вывод результатов оценивания моделей

Часто нужно одновременно вывести результат оценивания нескольких моделей. Для этого можно воспользоваться пакетом `stargazer`, который есть и в R, и в python.

R

```
install.packages("stargazer")
```

В python, если вы работаете не в Jupyter, и если вам нужен текстовый вывод таблиц, то вам нужно установить пакет `yatg`. Так как данный документ готовится не в Jupyter, то `yatg` нам пригодится.

Консоль

```
pip install stargazer yatg
```

## Синтаксис формул

В дальнейшем мы будем рассматривать способы оценивания линейных (по параметрам) моделей в R и python. Рассматриваемые методы используют упрощенную версию нотации, впервые, по-видимому, предложенной Уилкинсоном и Роджерсом (Wilkinson and Rogers 1973), описанную Чемберсом, Хастии и Преджибоном (Chambers, Hastie, and Pregibon 1990).

В python формулы записываются как обычные строковые значения в кавычках, но в R формулы — отдельный тип данных, кавычки для которого не нужны. Формула в общем виде представляет собой объясняемую часть — одну переменную или несколько — и объясняющую, разделенные знаком “тильда” `~`.

$Y \sim X \dots$

В нашем случае левая часть выражения будет представлять собой одну переменную. С этим все просто.

С правой частью все сложнее и интереснее. Она, правая часть, состоит из элементов, разделенных знаками + и -. Элемент может представлять собой:

- Переменную, например  $x$ .
- Некоторую функцию от переменной или переменных, например  $\log(x)$  в R или `np.log(x)` в python.
- Составной элемент, включающий в себя переменные и/или функции от переменных, например  $x*z$ . Обратите внимание, что это **не произведение**!

Вообще, в формулах вы будете видеть некоторые математические операторы, но следует помнить, что некоторые из них имеют значение и смысл **отличные** от традиционных! Рассмотрим их подробнее:

- $()$  — скобки призваны группировать элементы и менять порядок обработки других символов. То есть, скобки работают как скобки.
- $+$  — плюс в формулах обозначает добавление элемента в модель. Например, выражение  $y \sim x + z$  обозначает модель, в которой  $y$  является объясняемой переменной, а  $x$  и  $z$  — объясняющими.
- $-$  — минус, как можно догадаться, обозначает удаление элемента из модели. На первый взгляд он не нужен: просто не добавляйте элемент! Но на самом деле, следующие из рассматриваемых операторов в результате своей работы могут создавать элементы, явно в формуле не описанные. Если эти элементы нам не нужны, то при помощи оператора  $-$  их можно убрать из модели.

Также, минус можно использовать, чтобы убрать из формулы свободный член. При оценивании линейной модели он добавляется автоматически, его не надо добавлять вручную. Для его удаления достаточно добавить в формулу  $- 1$  или  $+ 0$ .

- $:$  — двоеточие обозначает член взаимодействия между двумя элементами, причем эти элементы сами могут быть членами взаимодействия. В качестве примера можно привести элементы  $x : z$  или  $x : z : w$ . Элементы, разделенные  $:$  могут быть не только переменными, но и функциями от них, например  $x : \sqrt{z}$ .

Элементы, разделенные  $:$ , могут быть составными, в этом случае мы получим все попарные комбинации. Например,  $(a + b) : (c + d)$  эквивалентно  $a : c + a : d + b : c + b : d$ .

- $*$  — звездочка позволяет добавить в модель член взаимодействия вместе со всеми входящими в него элементами. То есть выражения  $x*z$  и  $x + z + x : z$  эквивалентны.
- $^$  в R и  $**$  в python — показывает максимальную степень взаимодействия между элементами, где под степенью взаимодействия понимается число элементов в члене взаимодействия. Причем итоговая степень не будет превышать число элементов, входящих в выражение. Например:
  - $x^2$  эквивалентно  $x$ .
  - $(x+z)^2$  эквивалентно  $x + z + x : z$ .
  - $(a+b+c)^2$  эквивалентно  $a + b + c + a : b + b : c + a : c$ .
  - $(a+b+c)^3$  эквивалентно  $a + b + c + a : b + b : c + a : c + a : b : c$ .
- $\%in\%$  в R — данный оператор означает, что элемент слева является вложенным в элемент справа. В большинстве случаев запись  $x \%in\% y$  эквивалентна  $y : x$ .
- $/$  — краткая запись,  $x / y$  эквивалентно  $y + x \%in\% y$ .

Как мы видим, некоторые математические символы имеют другое значение. Если же нам нужно использовать их в формуле в **исходном** значении, то надо соответствующий элемент заключить в функ-

цию  $I()$ . То есть  $x*y$  — это член взаимодействия плюс сами переменные ( $x + y + x:y$ ), а  $I(x*y)$  — произведение  $x$  и  $y$ , используемое в качестве переменной.

Следует отметить, что последние два оператора (`%in%` и `/`) в случае непрерывных переменных не имеют большого смысла, так как `:` для таких переменных имеет смысл их произведения. Но если мы будем использовать **факторные** переменные, принимающие строго фиксированное число значений, то в этом случае данные операторы будут полезны, так как они говорят R и python, в каком порядке надо выводить результаты в таблицах и как их надо группировать.

При этом следует помнить, что переменная принимающая строго фиксированное число значений, с точки зрения программы, является непрерывной. Чтобы указать на то, что ее следует трактовать как факторную, следует преобразовать ее в таковую при помощи функции `as.factor` в R или функции `C()` в python.

# Сток-Уотсон. Задания E5.1, E7.1 и E8.1

Напомним, что в R пакеты **можно** импортировать в т.н. глобальное **пространство имен** и вызывать функции **без** указания пакетов при помощи `::!`

Впрочем, пакет `ggplot2` лучше импортировать, так как иначе придется приписывать `ggplot2::` к каждому компоненту команды.

R

```
library(ggplot2)
```

В python мы **должны** импортировать все что нам нужно. Модули для загрузки данных и оценивания моделей.

Python

```
import numpy as np
import pandas as pd
import statsmodels.formula.api as smf
```

Модуль для статистических тестов.

Python

```
import statsmodels.stats.api as sms
```

Модули для графиков.

Python

```
import matplotlib.pyplot as plt
import seaborn as sb
```

Модули для вывода результатов

Python

```
from stargazer.stargazer import Stargazer
import yatg
```



## Загрузка данных

Скачаем файл `cps12.xlsx` и загрузим его:

R

```
dataset <- readxl::read_excel("D:/cps12.xlsx")
head(dataset)
```

```
## # A tibble: 6 x 5
##   year   ahe bachelor female  age
##   <dbl> <dbl>   <dbl>   <dbl> <dbl>
## 1 2012 19.2      0       0    30
## 2 2012 17.5      0       0    29
## 3 2012  8.55     0       0    27
## 4 2012 16.8      0       1    25
## 5 2012 16.3      1       1    27
## 6 2012 16.1      1       0    30
```

Python

```
dataset = pd.read_excel("D:/cps12.xlsx")
dataset.head()
```

```
##   year      ahe  bachelor  female  age
## 0 2012 19.230770      0      0    30
## 1 2012 17.548077      0      0    29
## 2 2012  8.547009      0      0    27
## 3 2012 16.826923      0      1    25
## 4 2012 16.346153      1      1    27
```

## Оценивание парной регрессии

? Оцените регрессию средней зарплаты в час *Ahe* на возраст *Age*. Чему равна оценка свободного члена? Чему равна оценка коэффициента наклона?

R

```
model <- lm(ahe ~ age, data = dataset)
summary(model)
```

```
##
## Call:
## lm(formula = ahe ~ age, data = dataset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.864  -7.381  -2.245   4.799  72.499
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.62605    1.28752   3.593 0.000329 ***
## age          0.51182    0.04323  11.840 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.59 on 7438 degrees of freedom
```

```
## Multiple R-squared:  0.0185, Adjusted R-squared:  0.01837
## F-statistic: 140.2 on 1 and 7438 DF,  p-value: < 2.2e-16
```

## Python



```
model = smf.ols("ahe ~ age", data=dataset)
model_est = model.fit()
print(model_est.summary())
```

```
##                               OLS Regression Results
## =====
## Dep. Variable:                ahe    R-squared:                0.018
## Model:                      OLS      Adj. R-squared:           0.018
## Method:                    Least Squares    F-statistic:           140.2
## Date:                      Br, 06 дек 2022    Prob (F-statistic):       4.72e-32
## Time:                      18:03:00    Log-Likelihood:          -28112.
## No. Observations:          7440    AIC:                    5.623e+04
## Df Residuals:              7438    BIC:                    5.624e+04
## Df Model:                  1
## Covariance Type:            nonrobust
## =====
##               coef      std err          t      P>|t|      [0.025      0.975]
## -----
## Intercept      4.6260      1.288       3.593      0.000       2.102       7.150
## age            0.5118      0.043      11.840      0.000       0.427       0.597
## =====
## Omnibus:                 1953.812    Durbin-Watson:           1.853
## Prob(Omnibus):            0.000    Jarque-Bera (JB):        5216.736
## Skew:                    1.406    Prob(JB):                0.00
## Kurtosis:                 5.987    Cond. No.                313.
## =====
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

## Статистические тесты

❓ Графический анализ гетероскедастичности (квадраты остатков и графики их зависимости). Тестирование гетероскедастичности (тесты Глейзера, Голдфелда-Квандта) и ВМНК.

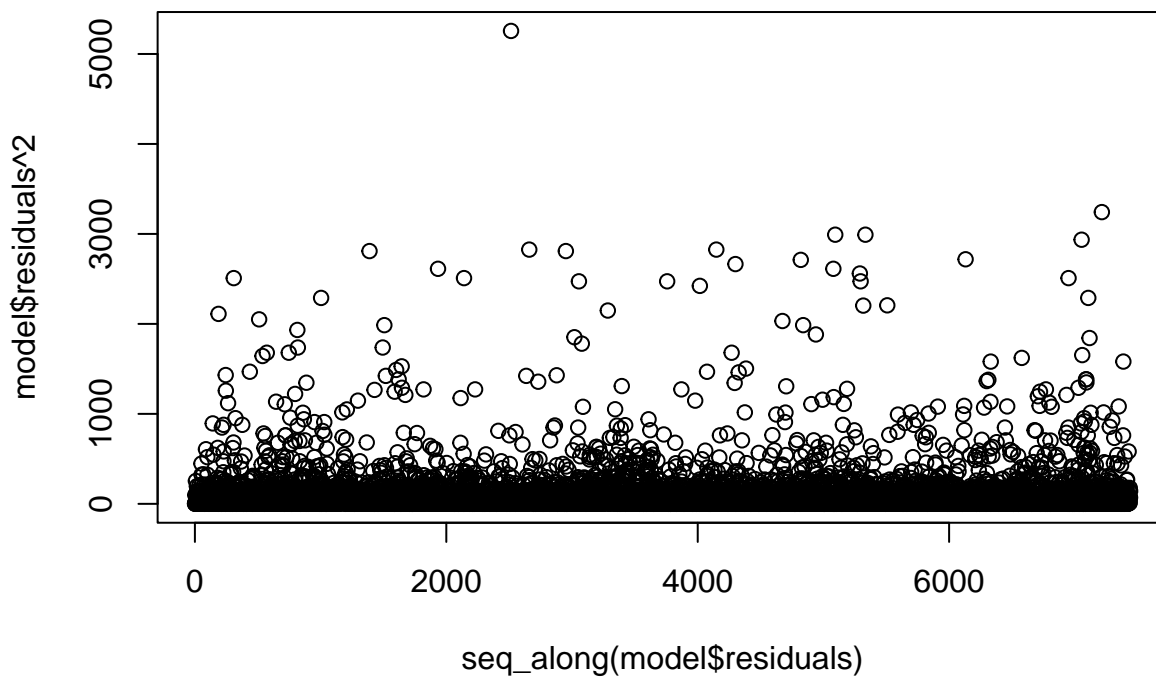
## Построение графиков

Построим график квадратов остатков. Это можно сделать при помощи функции `plot` или `ggplot`.

R



```
plot(seq_along(model$residuals), model$residuals^2)
```

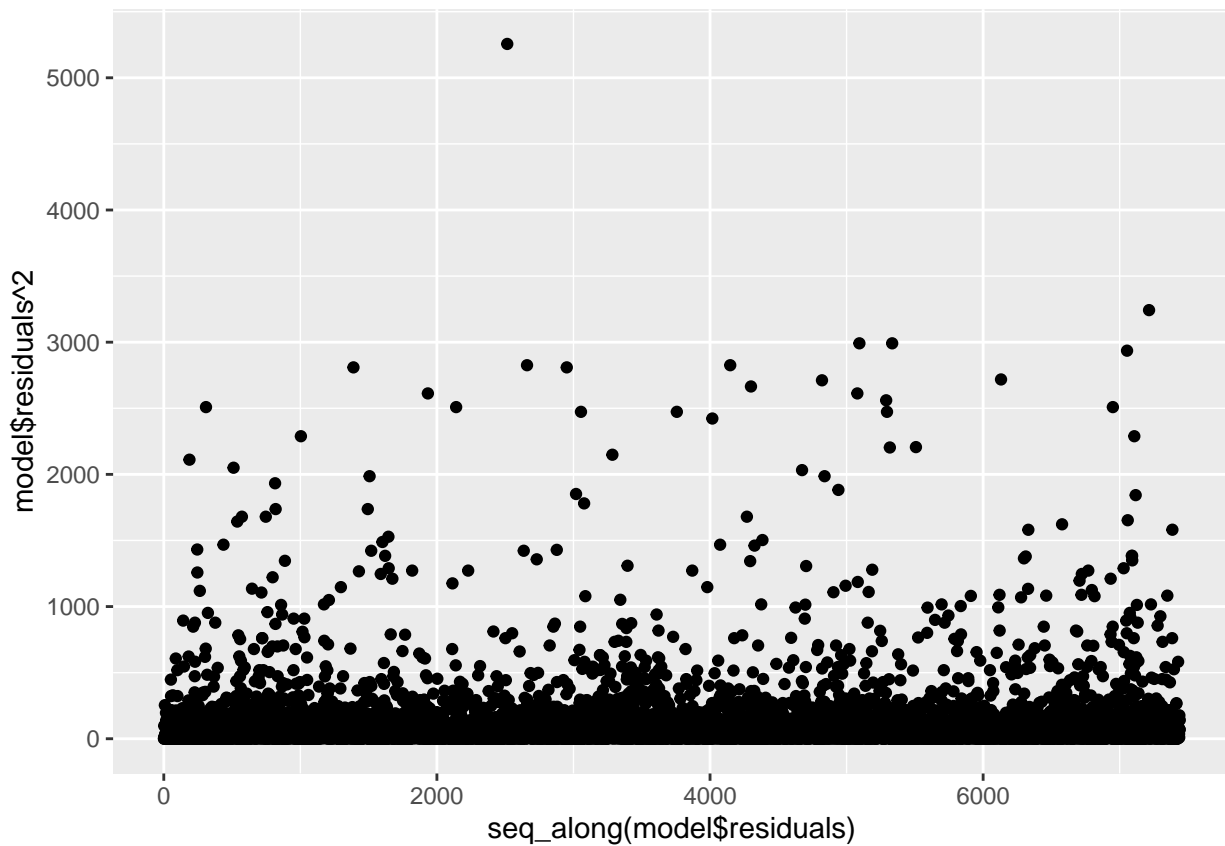


Функция `ggplot` устроена хитрее: сначала создается база, “холст”, на который накладываются слои с содержимым. Функция `aes` служит для задания переменных осей и настройки их отображения. В данном случае мы говорим, что на графике по оси `x` мы откладываем номера наблюдений (последовательность чисел длиной равной числу значений остатков), а по оси `y` для диаграммы рассеяния откладываем квадрат остатков.

R

```
ggplot(dataset, aes(x = seq_along(model$residuals))) +  
  geom_point(aes(y = model$residuals^2))
```



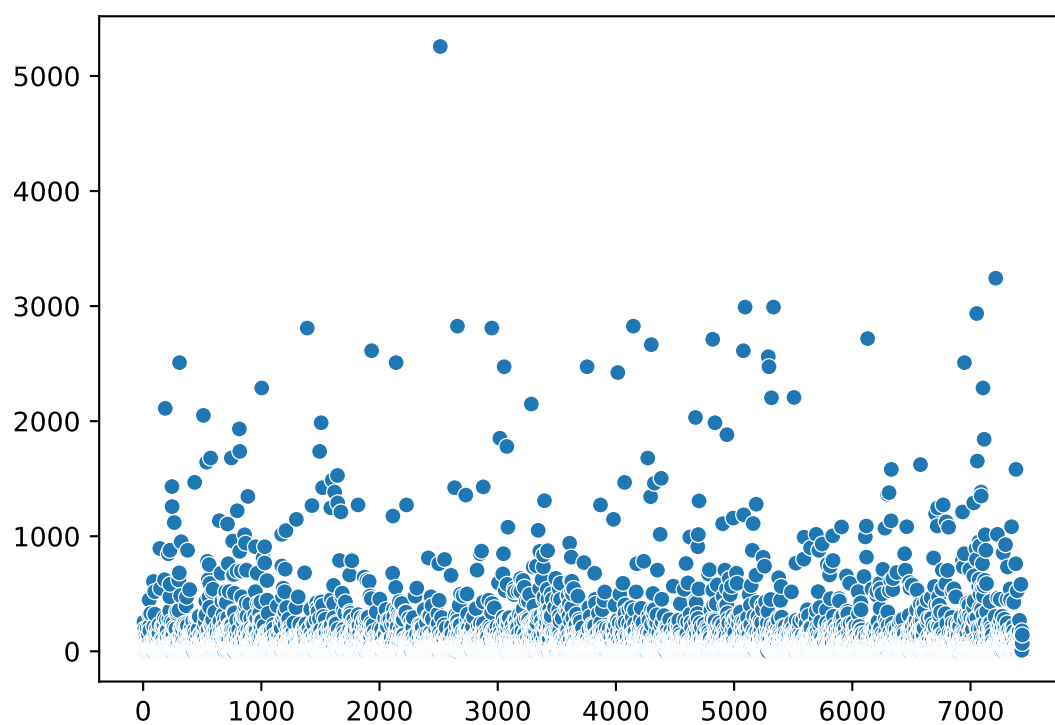


Логика работы с `seaborn` и `matplotlib` похожа на логику `ggplot`. Сначала мы должны очистить хранимый в памяти график. Далее мы накладываем на него слои, после чего показываем.

Python



```
plt.clf()
sb.scatterplot(x = range(len(model_est.resid)), y = model_est.resid ** 2)
plt.show()
```

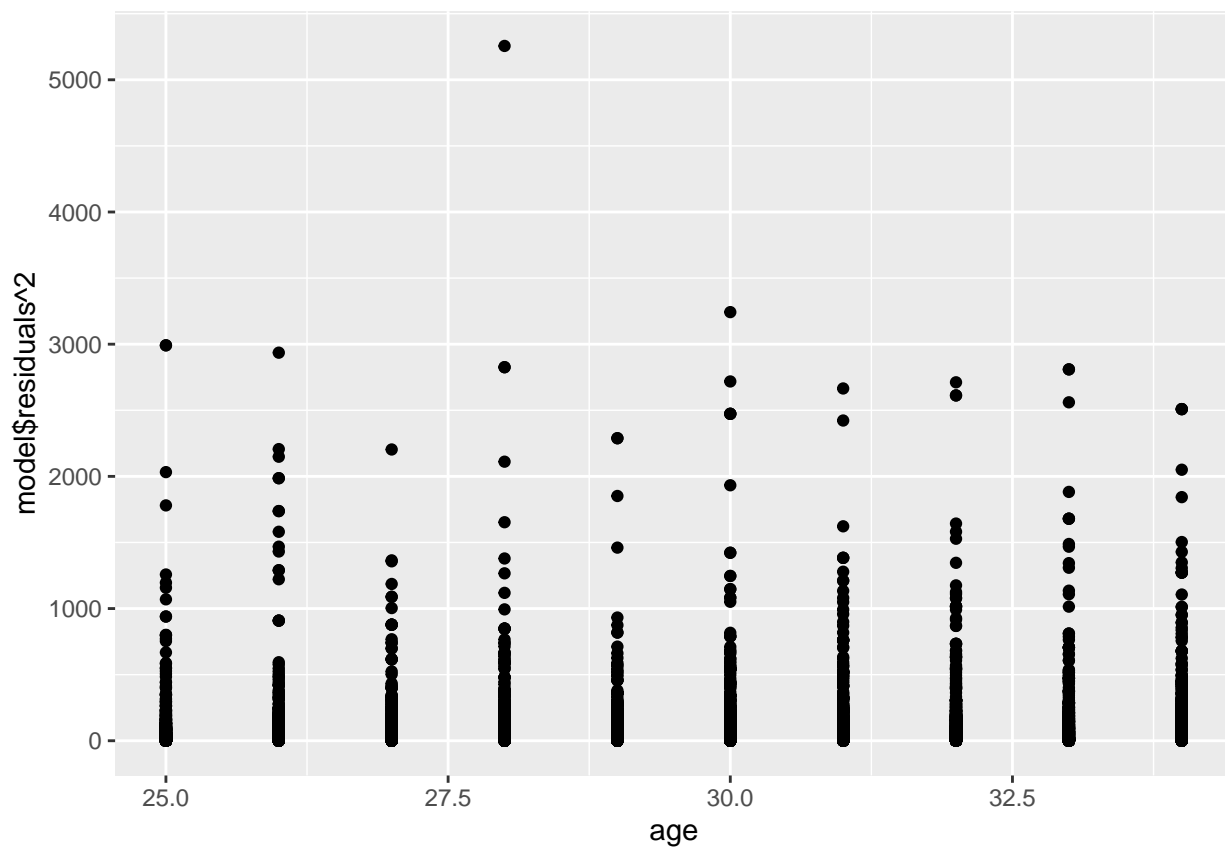


Графики относительно *Age* и *Ahe*.

R



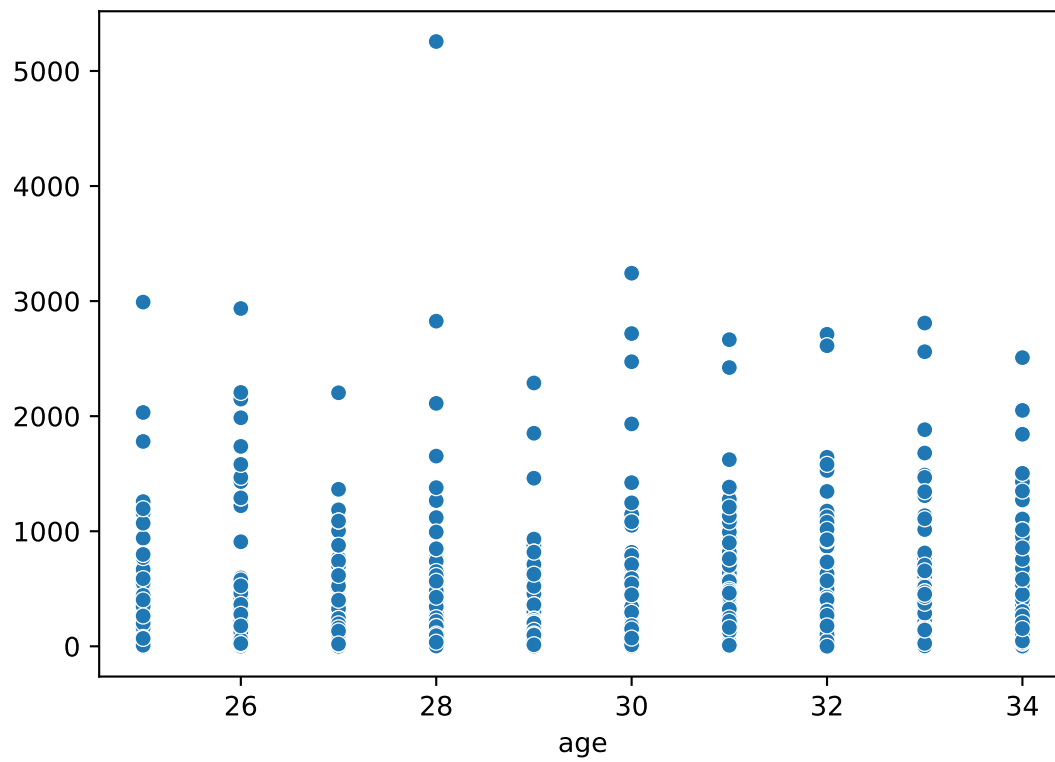
```
ggplot(dataset, aes(x = age)) +  
  geom_point(aes(y = model$residuals^2))
```



Python



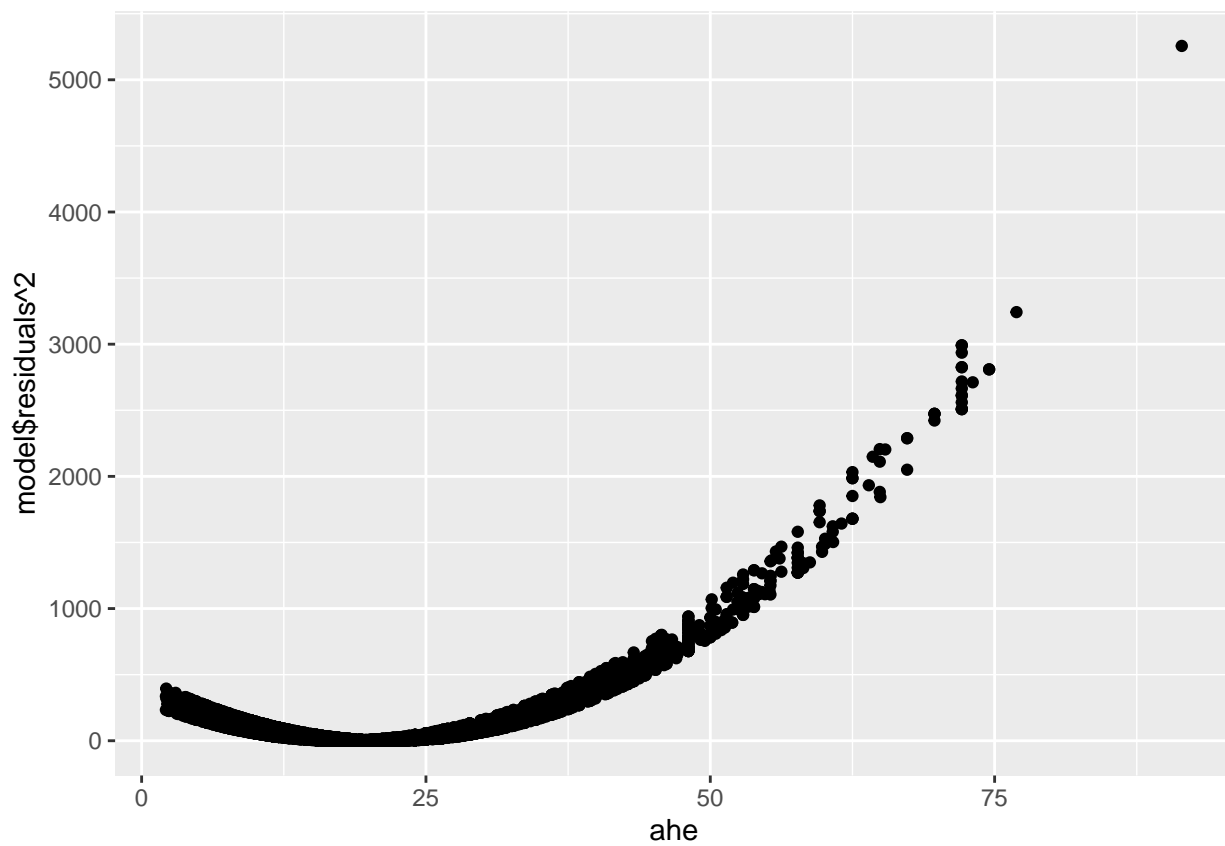
```
plt.clf()
sb.scatterplot(dataset, x = "age", y = model_est.resid ** 2)
plt.show()
```



R



```
ggplot(dataset, aes(x = age)) +  
  geom_point(aes(y = model$residuals^2))
```

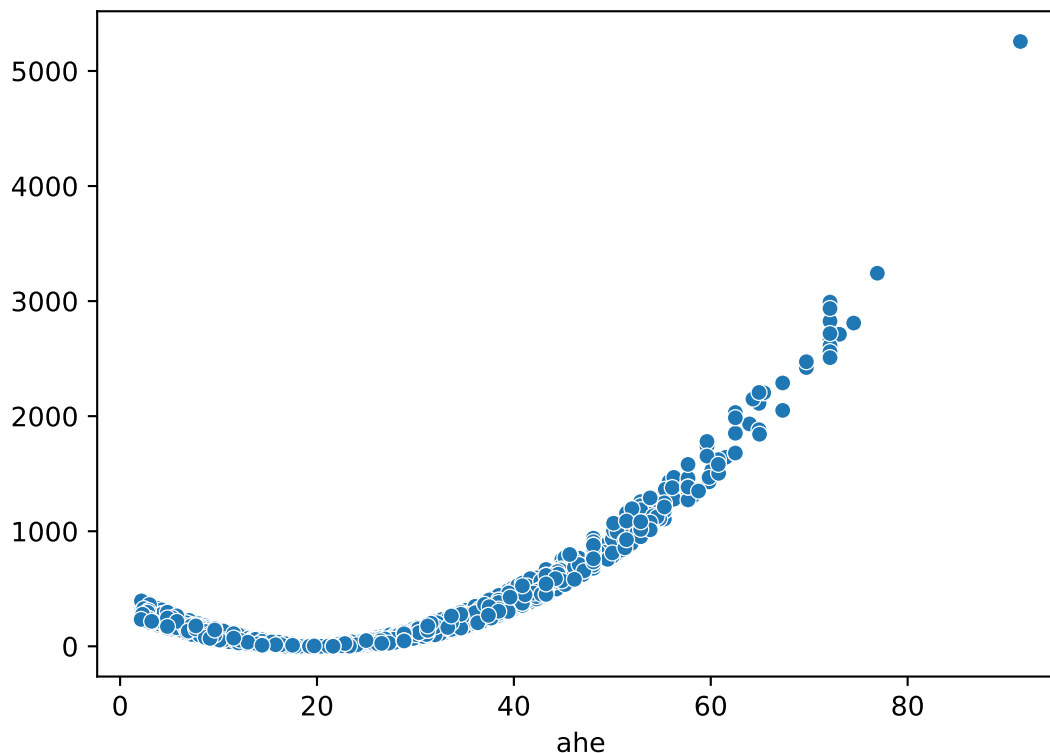


Python



```
plt.clf()
sb.scatterplot(dataset, x = "ahe", y = model_est.resid ** 2)
plt.show()
```





## Тесты на гетероскедастичность ошибок

**Тест Голдфельдта-Квандта.** В R тест принимает на вход оцененную модель, набор переменных для упорядочивания наблюдений и долю наблюдений для отбрасывания.

R

```
lmtest::gqtest(model, order.by = ~ age, data = dataset, fraction = .3)
```

```
##
## Goldfeld-Quandt test
##
## data: model
## GQ = 1.3678, df1 = 2602, df2 = 2602, p-value = 7.912e-16
## alternative hypothesis: variance increases from segment 1 to 2
```

В python тест принимает на вход столбец значений объясняемой переменной и матрицу значений объясняющих переменных с константой, которые можно получить из **не оцененной** модели, номер переменной в exog, по которой надо сортировать данные, конец первой выборки и долю наблюдений, которые надо отбросить. Надо учесть, что данная реализация отбрасывает наблюдения начиная со split-a. Если не указать этот параметр, то наблюдения будут отбрасываться с середины, и мы получим две выборки разного размера.

Python

```
sms.het_goldfeldquandt(y=model.endog, x=model.exog, idx = 1, split=.35,
↳ drop=.3)
```

```
## (1.3232635283092733, 4.958997358175327e-13, 'increasing')
```

### Тест Бройша-Пагана.

$$y = X'\beta + u$$
$$\hat{u}^2 = \gamma_0 + Z'\gamma + v$$
$$nR^2 \sim \chi^2(p)$$

В R тест принимает на вход оцененную модель, набор переменных, потенциально влияющих на дисперсию, представленный в виде формулы, и переменную с данными.

R



```
lmtest::bptest(model, ~ age, data = dataset, studentize = TRUE)
```

```
##  
## studentized Breusch-Pagan test  
##  
## data: model  
## BP = 27.354, df = 1, p-value = 1.694e-07
```

В python тест принимает на вход остатки оцененной модели и матрицу значений объясняющих переменных во **вспомогательной** регрессии квадрата остатков.

Python



```
statsmodels.het_breuschpagan(model_est.resid, model.exog)
```

```
## (27.354094082360767, 1.69405398802106e-07, 27.447655583031647, 1.6583420766689666e-07)
```

### Тест Уайта.

$$y = X'\beta + u$$
$$\hat{u}^2 = \gamma_0 + \sum_i \sum_{j \geq i} \gamma_{ij} z_i z_j + v$$
$$nR^2 \sim \chi^2(p)$$

В R вызывается той же функцией, что и тест Бройша-Пагана, но нужно вручную добавлять во второй аргумент взаимные произведения и квадраты переменных, влияющих на дисперсию. Так как в нашем случае у нас нет второй переменной, то не будет и произведения, только квадрат единственной переменной.

R



```
lmtest::bptest(model, ~ age + I(age^2), data = dataset, studentize = TRUE)
```

```
##  
## studentized Breusch-Pagan test  
##  
## data: model  
## BP = 27.471, df = 2, p-value = 1.083e-06
```

В python тест принимает на вход остатки оцененной модели и матрицу значений объясняющих переменных во **вспомогательной** регрессии квадрата остатков.

Python



```
sm.het_white(model_est.resid, model.exog)
```

```
## (27.47100506683621, 1.083295760596925e-06, 13.780847590728703, 1.0619810952820436e-06)
```

**Тест Глейзера.** С ним есть определенные сложности в том плане, что его реализации несколько отличаются от классического описания из учебников. После оценивания модели и получения модуля остатков  $|\hat{u}|$  оценивается вспомогательная модель

$$|\hat{u}| = \beta_0 + Z'\beta + v$$

где  $Z$  — объясняющие переменные вспомогательного уравнения. Часто в качестве  $Z$  берут переменные исходной модели. Тест имеет две статистики:

$$nR^2 \sim \chi^2(k)$$

$$\frac{ESS_{aux}}{\left(1 - \frac{2}{n}\right)\hat{\sigma}^2} \sim \chi^2(k)$$

где  $ESS_{aux}$  — объясненная сумма квадратов вспомогательной модели,  $\hat{\sigma}^2$  — выборочная дисперсия остатков исходной или вспомогательной модели (Mittelhammer, Judge, and Miller 2000).

В R реализован этот вариант теста, он принимает на вход основную модель. В качестве необязательных аргументов можно указать произвольную матрицу  $Z$ , а также выбрать источник выборочной дисперсии. Результатом является вторая статистика из представленных выше.

R



```
skedastic::glejser(model)
```

```
## # A tibble: 1 x 4
##   statistic p.value parameter alternative
##   <dbl>     <dbl>     <dbl> <chr>
## 1      91.2 1.32e-21         1 greater
```

В python я не смог обнаружить работающей реализации теста. Посчитаем все вручную. Сначала импортируем функцию из `scipy`, нужную для нахождения критических значений распределения  $\chi^2$ .

Python



```
from scipy.stats import chi2
```

А затем рассчитаем тестовую статистику.



```
from scipy.stats import chi2

model_aux = smf.ols("abs(model_est.resid) ~ age", data=dataset)
model_aux_est = model_aux.fit()
stat_aux = model_aux_est.ess / ((1 - 2 / np.pi) * np.var(model_est.resid))
print(f"Stat: {stat_aux:5.4f}, Critical value: {chi2.ppf(0.95,
↳ df=model_aux.df_model):5.4f}, p-value: {1 - chi2.cdf(stat_aux,
↳ df=model_aux.df_model):5.4f}")
```

```
## Stat: 91.1745, Critical value: 3.8415, p-value: 0.0000
```

Рассмотрим также тесты и на другие случаи нарушения условий Гаусса-Маркова, хотя

- в задании это не просили,
- тесты, рассматриваемые далее, чаще используются в случае рассмотрения временных рядов.

## Тесты на автокорреляцию ошибок

Данные тесты следует применять в случае рассмотрения временных рядов. Для межобъектных данных они малоосмысленны. Сейчас мы просто рассмотрим способы их применения, игнорируя данное замечание!

**Тест Дарбина-Уотсона.** В R тест Дарбина-Уотсона принимает на вход оцененную модель, а в python — ее остатки.

R



```
lmtest::dwtest(model)
```

```
##
## Durbin-Watson test
##
## data: model
## DW = 1.8528, p-value = 1.081e-10
## alternative hypothesis: true autocorrelation is greater than 0
```

Python



```
sm.stats.durbin_watson(model_est.resid)
```

```
## 1.852819187401235
```

**Тест Бройша-Годфри.**

$$\hat{u}_t = X' \beta + \sum_{i=1}^p \hat{u}_{t-i} + v_t$$

$$nR^2 \sim \chi^2(p)$$

В R тест принимает на вход формулу для модели, тестируемый порядок автокорреляции и набор данных для оценивания. Формулу можно получить из оцененной модели при помощи функции `formula`.

R



```
lmtest::bptest(formula(model), order = 3, data = dataset)
```

```
##
## Breusch-Godfrey test for serial correlation of order up to 3
##
## data: formula(model)
## LM test = 81.224, df = 3, p-value < 2.2e-16
```

В python тест принимает на вход оцененную модель и тестируемый порядок автокорреляции.

Python



```
sms.acorr_breusch_godfrey(model_est, nlags=3)
```

```
## (81.2243910735383, 1.6764179760766054e-17, 27.355245841858235, 1.377718242092972e-17)
```

## Тест Харке-Бера

Данный тест будет особо важен при изучении временных рядов, когда вы будете иметь дело с большим количеством асимптотических критериев, для которых крайне желательна нормальность остатков оцененной модели.

$$HB = n \left( \frac{S^2}{6} + \frac{(K-3)^2}{24} \right) \underset{asy.}{\sim} \chi^2(2)$$

$$S = \frac{m_3}{m_2^{3/2}}, \quad K = \frac{m_4}{m_2^2}$$

$$m_z = \frac{1}{n} \sum_{i=1}^n \hat{u}_i^z$$

И в R, и в python тест принимает на вход вектор остатков оцененной модели.

R



```
tseries::jarque.bera.test(model$residuals)
```

```
##
## Jarque Bera Test
##
## data: model$residuals
## X-squared = 5216.7, df = 2, p-value < 2.2e-16
```

Python



```
sms.jarque_bera(model_est.resid)
```

```
## (5216.736149641615, 0.0, 1.4061065110013002, 5.986576475959586)
```

## Стабильность оценок модели – Тест Чоу

Часто в реальных данных в определенный момент времени может возникнуть ситуация, когда параметры модели, описывающей связи между переменными, меняются. Этот момент называется точкой структурного сдвига. Оценивание единой модели для всей выборки в данном случае приведет к смещенным оценкам, зачастую не имеющим особого смысла.

Как можно догадаться, данный тест осмыслен преимущественно в случае рассмотрения временных рядов, однако, в отличие от тестов на автокоррелированность, его применение для межобъектных данных можно обосновать. Например, модель может отличаться для низких и высоких значений какой-либо переменной и т.д. Но, в этом случае, данные должны быть специальным образом подготовлены.

Для тестирования подобного эффекта применяется, в том числе, тест Чоу. Его смысл состоит в том, что выборка делится в заранее заданный момент времени, после его RSS общей модели сравнивается с RSS моделей, оцененных на отдельных частях выборки.

$$F = \frac{[RSS_{big} - (RSS_1 + RSS_2)]/(k + 1)}{(RSS_1 + RSS_2)/(n - 2k - 2)} \sim F(k + 1, n - 2k - 2)$$

В R принимает на вход формулу модели, точку потенциального сдвига и переменную с данными.

R

```
strucchange::sctest(formula(model), type = "Chow", point = 1500, data =  
  dataset)
```

```
##  
## Chow test  
##  
## data: formula(model)  
## F = 3.3757, p-value = 0.03425
```

В python надо воспользоваться пакетом chow\_test.

Python

```
from chow_test import chow_test  
chow_test(y_series=pd.Series(model.endog),  
  X_series=pd.DataFrame(model.exog), last_index=1499, first_index=1500,  
  significance=0.05)
```

```
## Fail to reject the null hypothesis of equality of regression coefficients in the two periods.  
## Chow Statistic: 2.2587400010167853, P_value: 0.07947530812960613  
## (2.2587400010167853, 0.07947530812960613)
```

## Взвешенный МНК

Так как тесты на гетероскедастичность показали ее наличие, имеет смысл воспользоваться взвешенным МНК.

Предположим, что дисперсия пропорциональна возрасту. Тогда нужно каждое наблюдение пронормировать на  $\frac{1}{\sqrt{Age_i}}$ .

Функция `lm()` в R может принимать необязательный аргумент `weights`, задающий веса для наблюдений. Причем в R минимизируется  $\sum_{i=1}^n w_i \hat{u}_i^2$ , то есть в `weights` в нашем случае нужно передавать  $\frac{1}{\text{Age}_i}$ .

R



```
wmodel <- lm(ahe ~ age, data = dataset, weights = 1 / dataset$age)
summary(wmodel)
```

```
##
## Call:
## lm(formula = ahe ~ age, data = dataset, weights = 1/dataset$age)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -3.4109 -1.3421 -0.4011  0.8719 13.7028
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.45786    1.26461   3.525 0.000426 ***
## age          0.51749    0.04285  12.076 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.941 on 7438 degrees of freedom
## Multiple R-squared:  0.01923,    Adjusted R-squared:  0.0191
## F-statistic: 145.8 on 1 and 7438 DF,  p-value: < 2.2e-16
```

Python



```
wmodel = smf.wls("ahe ~ age", data=dataset, weights=(1 / dataset.age))
wmodel_est = wmodel.fit()
print(wmodel_est.summary())
```

```
##                               WLS Regression Results
## =====
## Dep. Variable:                ahe    R-squared:                0.019
## Model:                      WLS    Adj. R-squared:           0.019
## Method:                     Least Squares    F-statistic:           145.8
## Date:                       Br, 06 дек 2022    Prob (F-statistic):     2.90e-33
## Time:                       18:03:33    Log-Likelihood:         -28081.
## No. Observations:           7440    AIC:                   5.617e+04
## Df Residuals:               7438    BIC:                   5.618e+04
## Df Model:                   1
## Covariance Type:            nonrobust
## =====
##               coef      std err          t      P>|t|      [0.025      0.975]
## -----
## Intercept      4.4579      1.265        3.525      0.000        1.979        6.937
## age            0.5175      0.043       12.076      0.000        0.433        0.601
## =====
## Omnibus:                 1993.798    Durbin-Watson:           1.852
## Prob(Omnibus):            0.000    Jarque-Bera (JB):       5475.553
## Skew:                    1.423    Prob(JB):               0.00
## Kurtosis:                6.093    Cond. No.               306.
## =====
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

# Оценивание множественных регрессий

? Оцените регрессию средней почасовой зарплаты *Ahe* на возраст *Age*, пол *Female* и образование *Bachelor*.

R



```
model <- lm(ahe ~ age + female + bachelor, data = dataset)
summary(model)
```

```
##
## Call:
## lm(formula = ahe ~ age + female + bachelor, data = dataset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.688  -6.207  -1.708   4.280  75.302
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.86620    1.18760   1.571   0.116
## age          0.51029    0.03952  12.912 <2e-16 ***
## female      -3.81030    0.22960 -16.596 <2e-16 ***
## bachelor     8.31863    0.22739  36.584 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.678 on 7436 degrees of freedom
## Multiple R-squared:  0.1801, Adjusted R-squared:  0.1798
## F-statistic: 544.5 on 3 and 7436 DF,  p-value: < 2.2e-16
```

Python



```
model = smf.ols("ahe ~ age + female + bachelor", data=dataset)
model_est = model.fit()
print(model_est.summary())
```

```
##
## OLS Regression Results
## =====
## Dep. Variable:          ahe      R-squared:                0.180
## Model:                  OLS      Adj. R-squared:           0.180
## Method:                 Least Squares      F-statistic:           544.5
## Date:                   Br, 06 дек 2022      Prob (F-statistic):       6.51e-320
## Time:                   18:03:35      Log-Likelihood:          -27443.
## No. Observations:       7440      AIC:                    5.489e+04
## Df Residuals:           7436      BIC:                    5.492e+04
## Df Model:                3
## Covariance Type:        nonrobust
## =====
##              coef      std err          t      P>|t|      [0.025      0.975]
## -----
## Intercept          1.8662         1.188         1.571      0.116      -0.462         4.194
## age                0.5103         0.040        12.912      0.000         0.433         0.588
## female            -3.8103         0.230       -16.596      0.000        -4.260        -3.360
## bachelor           8.3186         0.227        36.584      0.000         7.873         8.764
## =====
## Omnibus:                1975.582      Durbin-Watson:           1.935
## Prob(Omnibus):           0.000      Jarque-Bera (JB):        6089.399
## Skew:                    1.360      Prob(JB):                 0.00
## Kurtosis:                6.499      Cond. No.                 316.
## =====
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```



? Существенны ли различия в оценках влияния переменной *Age* на *Ahe* в регрессиях из пунктов (1) и (2)? Можно ли сказать, что оценка соответствующего коэффициента в регрессии из пункта (1) смещена из-за пропущенных переменных?

R



```
summary_table <- summary(model)$coefficients
(summary_table["age", 1] - 0.51182) / summary_table["age", 2]
```

```
## [1] -0.03881579
```

Python



```
model_est.t_test("age = 0.51182")
```

```
## <class 'statsmodels.stats.contrast.ContrastResults'>
##                      Test for Constraints
## =====
##              coef      std err          t      P>|t|      [0.025      0.975]
## -----
## c0              0.5103       0.040      -0.039      0.969      0.433      0.588
## =====
```

? Оцените регрессию логарифма редней почасовой зарплаты  $\ln(Ahe)$  на возраст *Age*, пол *Female* и образование *Bachelor*.

R



```
model1 <- lm(log(ahe) ~ age + female + bachelor, data = dataset)
summary(model1)
```

```
##
## Call:
## lm(formula = log(ahe) ~ age + female + bachelor, data = dataset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.28277 -0.28680  0.01372  0.30939  1.85993
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.941423   0.058683  33.08  <2e-16 ***
## age          0.025518   0.001953  13.07  <2e-16 ***
## female      -0.192338   0.011345 -16.95  <2e-16 ***
## bachelor     0.437783   0.011236  38.96  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4782 on 7436 degrees of freedom
## Multiple R-squared:  0.1964, Adjusted R-squared:  0.1961
## F-statistic: 605.7 on 3 and 7436 DF, p-value: < 2.2e-16
```

Python



```
model1 = smf.ols("np.log(ahe) ~ age + female + bachelor", data=dataset)
model1_est = model1.fit()
print(model1_est.summary())
```

```
##                               OLS Regression Results
## =====
## Dep. Variable:                np.log(ahe)    R-squared:                0.196
## Model:                      OLS            Adj. R-squared:          0.196
## Method:                     Least Squares   F-statistic:             605.7
## Date:                       Br, 06 дек 2022  Prob (F-statistic):      0.00
## Time:                       18:03:40        Log-Likelihood:          -5066.6
## No. Observations:           7440            AIC:                    1.014e+04
## Df Residuals:               7436            BIC:                    1.017e+04
## Df Model:                   3
## Covariance Type:            nonrobust
## =====
##               coef      std err          t      P>|t|      [0.025      0.975]
## -----
## Intercept          1.9414        0.059     33.083      0.000        1.826        2.056
## age                0.0255        0.002     13.067      0.000        0.022        0.029
## female            -0.1923        0.011    -16.953      0.000       -0.215       -0.170
## bachelor           0.4378        0.011     38.964      0.000        0.416        0.460
## =====
## Omnibus:                 316.825    Durbin-Watson:           1.936
## Prob(Omnibus):           0.000    Jarque-Bera (JB):        508.141
## Skew:                   -0.375    Prob(JB):                4.56e-111
## Kurtosis:                4.037    Cond. No.                 316.
## =====
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

❓ Являются ли пол и образование факторами, определяющими доход? Проверьте нулевую гипотезу о том, что переменная *Female* может быть исключена из регрессии. Проверьте нулевую гипотезу о том, что переменная *Bachelor* может быть исключена из регрессии. Проверьте нулевую гипотезу о том, что обе переменные *Female* и *Bachelor* могут быть исключены из регрессии.

Для ответа на вопрос в отношении отдельных коэффициентов достаточно посмотреть на t-статистики и p-значения в результатах оценивания.

Для совместной гипотезы нужно либо воспользоваться поправкой Бонферрони, либо провести F-тест, что мы и сделаем.

R



```
car::linearHypothesis(model1, c("female=0", "bachelor=0"))
```

```
## Linear hypothesis test
##
## Hypothesis:
## female = 0
## bachelor = 0
##
## Model 1: restricted model
## Model 2: log(ahe) ~ age + female + bachelor
##
##   Res.Df    RSS Df Sum of Sq    F    Pr(>F)
## 1    7438 2077.0
## 2    7436 1700.6   2    376.35 822.8 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Также можно воспользоваться тестом Вальда, который принимает на вход ковариационную матрицу модели, оцененные коэффициенты и номера коэффициентов, которые надо совместно протестировать на равенство нулю (помните про возможное наличие в модели константы). Реализация критерия Вальда из пакета aod требует явно задать ковариационную матрицу. Можно воспользоваться обычной матрицей.

R



```
aod::wald.test(Sigma = vcov(model1), b = coef(model1), Terms = 3:4)
```

```
## Wald test:
## -----
##
## Chi-squared test:
## X2 = 1645.6, df = 2, P(> X2) = 0.0
```

А можно использовать устойчивые к гетероскедастичности ошибки Уайта, функция для расчета которых есть в пакете `sandwich`.

R



```
aod::wald.test(Sigma = sandwich::vcovHC(model1, type = "HC0"), b =
  ↪ coef(model1), Terms = 3:4)
```

```
## Wald test:
## -----
##
## Chi-squared test:
## X2 = 1674.4, df = 2, P(> X2) = 0.0
```

Можно воспользоваться альтернативной реализацией теста Вальда. Проведем тест для обычной ковариационной матрицы.

R



```
lmtest::waldtest(model1, . ~ . - female - bachelor, test = "Chisq")
```

```
## Wald test
##
## Model 1: log(ahe) ~ age + female + bachelor
## Model 2: log(ahe) ~ age
##   Res.Df Df   Chisq Pr(>Chisq)
## 1     7436
## 2     7438 -2 1645.6 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

И для устойчивой к гетероскедастичности.

R



```
lmtest::waldtest(model1, . ~ . - female - bachelor, vcov =
  ↪ sandwich::vcovHC(model1, type = "HC0"), test = "Chisq")
```

```
## Wald test
##
## Model 1: log(ahe) ~ age + female + bachelor
## Model 2: log(ahe) ~ age
##   Res.Df Df   Chisq Pr(>Chisq)
## 1     7436
## 2     7438 -2 1674.4 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

В python метод `f_test` для **оцененной** модели принимает на вход один из трех вариантов входных параметров:

- Матрицу размера  $p \times k$ , где  $p$  — число индивидуальных ограничений,  $k$  — число объясняющих переменных. Ненулевые элементы матрицы показывают, с каким коэффициентом соответствующая переменная входит в соответствующее ограничение. В данном случае подразумевается равенство нулю во всех ограничениях.
- Кортеж из матрицы, аналогичной таковой в предыдущем пункте, и вектор-столбца размера  $p$ , в котором указано, равенство какому числу рассматривается в соответствующем ограничении.
- Строку с явной записью ограничений через запятую.

Python



```
model1_est.f_test("female = 0, bachelor = 0")
```

```
## <class 'statsmodels.stats.contrast.ContrastResults'>
## <F test: F=822.7977573028112, p=1.5e-323, df_denom=7.44e+03, df_num=2>
```

Также можно воспользоваться тестом Вальда, принимающим на вход те же аргументы, что и `f_test`. Если не задавать ковариационную матрицу, то на выходе получим тот же результат, что и для F-теста.

Python



```
model1_est.wald_test("female = 0, bachelor = 0", scalar=True)
```

```
## <class 'statsmodels.stats.contrast.ContrastResults'>
## <F test: F=822.7977573028112, p=1.5e-323, df_denom=7.44e+03, df_num=2>
```

Либо можно воспользоваться ковариационной матрицей Уайта.

Python



```
vcov_white = model1_est.get_robustcov_results(cov_type =
    ↪ "HC0").cov_params()
model1_est.wald_test("female = 0, bachelor = 0", scalar=True,
    ↪ cov_p=vcov_white)
```

```
## <class 'statsmodels.stats.contrast.ContrastResults'>
## <F test: F=837.2126163663196, p=0.0, df_denom=7.44e+03, df_num=2>
```

**?** Оцените регрессию логарифма средней почасовой зарплаты  $\ln(Ahe)$  на логарифм возраста  $\ln(Age)$ , пол *Female* и образование *Bachelor*.

R



```
model2 <- lm(log(ahe) ~ log(age) + female + bachelor, data = dataset)
summary(model2)
```

```
##
## Call:
## lm(formula = log(ahe) ~ log(age) + female + bachelor, data = dataset)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.27827 -0.28691  0.01326  0.30992  1.85737
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.14953    0.19436   0.769   0.442
## log(age)     0.75294    0.05734  13.132 <2e-16 ***
## female      -0.19236    0.01134 -16.957 <2e-16 ***
## bachelor     0.43766    0.01123  38.957 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4782 on 7436 degrees of freedom
## Multiple R-squared:  0.1966, Adjusted R-squared:  0.1962
## F-statistic: 606.4 on 3 and 7436 DF,  p-value: < 2.2e-16
```

## Python



```
model2 = smf.ols("np.log(ahe) ~ np.log(age) + female + bachelor",
                 data=dataset)
model2_est = model2.fit()
print(model2_est.summary())
```

```
##                                OLS Regression Results
## =====
## Dep. Variable:                np.log(ahe)    R-squared:                0.197
## Model:                        OLS           Adj. R-squared:       0.196
## Method:                      Least Squares  F-statistic:             606.4
## Date:                        Br, 06 дек 2022  Prob (F-statistic):      0.00
## Time:                        18:03:51       Log-Likelihood:        -5065.8
## No. Observations:             7440         AIC:                  1.014e+04
## Df Residuals:                 7436         BIC:                  1.017e+04
## Df Model:                     3
## Covariance Type:              nonrobust
## =====
##              coef      std err          t      P>|t|      [0.025      0.975]
## -----
## Intercept      0.1495      0.194        0.769      0.442      -0.231      0.531
## np.log(age)     0.7529      0.057       13.132     0.000       0.641      0.865
## female        -0.1924      0.011      -16.957     0.000      -0.215     -0.170
## bachelor        0.4377      0.011       38.957     0.000       0.416      0.460
## =====
## Omnibus:                 316.790    Durbin-Watson:           1.936
## Prob(Omnibus):           0.000    Jarque-Bera (JB):        508.147
## Skew:                   -0.375    Prob(JB):                4.54e-111
## Kurtosis:                4.037    Cond. No.                131.
## =====
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

? Оцените регрессию логарифма средней почасовой зарплаты  $\ln(Ahe)$  на возраст  $Age$ ,  $Age^2$ , пол  $Female$  и образование  $Bachelor$ .

## R



```
model3 <- lm(log(ahe) ~ age + I(age^2) + female + bachelor, data = dataset)
summary(model3)
```

```
##
## Call:
## lm(formula = log(ahe) ~ age + I(age^2) + female + bachelor, data = dataset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.26705 -0.28919  0.01526  0.31090  1.85221
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.7918819  0.6699502   1.182   0.2372
## age          0.1040449  0.0456313   2.280   0.0226 *
## I(age^2)     -0.0013284  0.0007712  -1.722   0.0850 .
## female       -0.1923983  0.0113436 -16.961 <2e-16 ***
## bachelor      0.4374121  0.0112363  38.928 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4782 on 7435 degrees of freedom
## Multiple R-squared:  0.1967, Adjusted R-squared:  0.1963
## F-statistic: 455.2 on 4 and 7435 DF,  p-value: < 2.2e-16
```

## Python



```
model3 = smf.ols("np.log(ahe) ~ age + I(age**2) + female + bachelor",
                 data=dataset)
model3_est = model3.fit()
print(model3_est.summary())
```

```
##                                OLS Regression Results
## =====
## Dep. Variable:                 np.log(ahe)    R-squared:                 0.197
## Model:                       OLS            Adj. R-squared:          0.196
## Method:                     Least Squares    F-statistic:             455.2
## Date:                       Бт, 06 дек 2022  Prob (F-statistic):       0.00
## Time:                       18:03:54         Log-Likelihood:        -5065.1
## No. Observations:            7440            AIC:                  1.014e+04
## Df Residuals:                7435            BIC:                  1.017e+04
## Df Model:                    4
## Covariance Type:             nonrobust
## =====
##               coef      std err          t      P>|t|      [0.025      0.975]
## -----
## Intercept          0.7919      0.670         1.182      0.237      -0.521      2.105
## age                0.1040      0.046         2.280      0.023       0.015      0.193
## I(age ** 2)       -0.0013      0.001        -1.722      0.085      -0.003      0.000
## female            -0.1924      0.011       -16.961      0.000      -0.215     -0.170
## bachelor           0.4374      0.011        38.928      0.000       0.415      0.459
## =====
## Omnibus:                 316.471    Durbin-Watson:           1.935
## Prob(Omnibus):           0.000    Jarque-Bera (JB):        507.649
## Skew:                    -0.375    Prob(JB):                5.83e-111
## Kurtosis:                4.037    Cond. No.                1.09e+05
## =====
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
## [2] The condition number is large, 1.09e+05. This might indicate that there are
## strong multicollinearity or other numerical problems.
```

? Начертите график функции регрессии между *Age* и *ln(Ahe)* из пунктов 2–4 для мужчин, имеющих высшее образование. Отфильтруем данные.

R



```
dataset_filtered <- dataset[dataset$female == 0 & dataset["bachelor"] == 1,
↵ ]
```

Python



```
dataset_filtered = dataset.loc[(dataset["female"] == 0) & (dataset.bachelor
↵ == 1), ].copy()
```

Оценим на отфильтрованных данных модели 2–4 и рассчитаем подобранные значения  $\ln(\hat{ahe})$  при помощи функции `predict`, принимающей на вход оцененную модель и, опционально, данные для построения прогноза.

R



```
model <- lm(log(ahe) ~ age, data = dataset_filtered)
ahe_2 <- predict(model, dataset_filtered)
summary(model)
```

```
##
## Call:
## lm(formula = log(ahe) ~ age, data = dataset_filtered)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.21588 -0.30193  0.01998  0.33029  1.27284
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.218551   0.114404  19.392  < 2e-16 ***
## age          0.030265   0.003834   7.893 4.81e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4834 on 2002 degrees of freedom
## Multiple R-squared:  0.03018,    Adjusted R-squared:  0.0297
## F-statistic: 62.3 on 1 and 2002 DF,  p-value: 4.807e-15
```

R



```
model <- lm(log(ahe) ~ log(age), data = dataset_filtered)
ahe_3 <- predict(model, dataset_filtered)
summary(model)
```

```
##
## Call:
## lm(formula = log(ahe) ~ log(age), data = dataset_filtered)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.20868 -0.29913  0.01785  0.32939  1.27536
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.09965   0.38168   0.261   0.794
## log(age)     0.89109   0.11266   7.910 4.22e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.4834 on 2002 degrees of freedom
## Multiple R-squared:  0.03031,    Adjusted R-squared:  0.02982
## F-statistic: 62.57 on 1 and 2002 DF,  p-value: 4.222e-15
```

R



```
model <- lm(log(ahe) ~ age + I(age**2), data = dataset_filtered)
ahe_4 <- predict(model, dataset_filtered)
summary(model)
```

```
##
## Call:
## lm(formula = log(ahe) ~ age + I(age^2), data = dataset_filtered)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.20502 -0.29548  0.01795  0.33021  1.27687
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.491548   1.336546   1.116   0.265
## age          0.079886   0.090971   0.878   0.380
## I(age^2)     -0.000839   0.001537  -0.546   0.585
##
## Residual standard error: 0.4835 on 2001 degrees of freedom
## Multiple R-squared:  0.03033,    Adjusted R-squared:  0.02936
## F-statistic: 31.29 on 2 and 2001 DF,  p-value: 4.162e-14
```

Обратите внимание, что к столбцам `dataset`-а можно обращаться как через `[]`, так и через `$`.

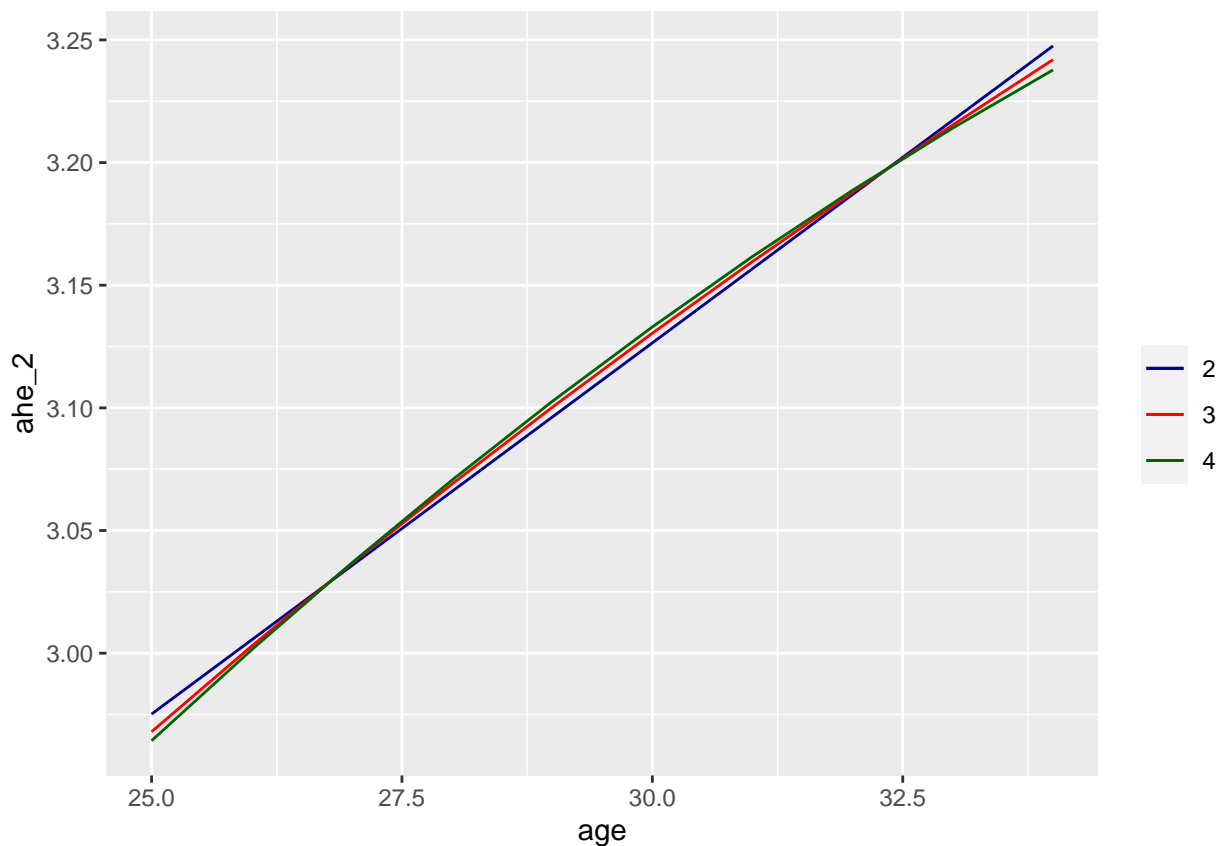
Построим графики. Обратите внимание, что параметр `color` в функциях `aes` определяет имя, под которым соответствующий график появляется в легенде. В функции `scale_color_manual` этим именам ставятся в соответствие цвета графиков.

R



```
ggplot(dataset_filtered, aes(x = age)) +
  geom_line(aes(y = ahe_2, color = "2")) +
  geom_line(aes(y = ahe_3, color = "3")) +
  geom_line(aes(y = ahe_4, color = "4")) +
  scale_color_manual(name = "", values = c("2" = "darkblue", "3" = "red",
    ↪ "4" = "darkgreen"))
```





## Python



```
model = smf.ols("np.log(ahe) ~ age", data=dataset_filtered)
model_est = model.fit()
ahe_2 = model_est.predict(dataset_filtered)
print(model_est.summary())
```

```
##                               OLS Regression Results
## =====
## Dep. Variable:                np.log(ahe)    R-squared:                0.030
## Model:                      OLS            Adj. R-squared:         0.030
## Method:                     Least Squares   F-statistic:           62.30
## Date:                       Br, 06 дек 2022  Prob (F-statistic):    4.81e-15
## Time:                       18:03:59        Log-Likelihood:        -1385.7
## No. Observations:            2004           AIC:                  2775.
## Df Residuals:                2002           BIC:                  2787.
## Df Model:                    1
## Covariance Type:             nonrobust
## =====
##               coef      std err          t      P>|t|      [0.025      0.975]
## -----
## Intercept      2.2186      0.114     19.392      0.000      1.994      2.443
## age            0.0303      0.004      7.893      0.000      0.023      0.038
## =====
## Omnibus:                 96.346    Durbin-Watson:           1.964
## Prob(Omnibus):            0.000    Jarque-Bera (JB):        133.227
## Skew:                    -0.450    Prob(JB):                1.18e-29
## Kurtosis:                 3.886    Cond. No.                 316.
## =====
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```



```
model = smf.ols("np.log(ahe) ~ np.log(age)", data=dataset_filtered)
model_est = model.fit()
ahe_3 = model_est.predict(dataset_filtered)
print(model_est.summary())
```

```
##                               OLS Regression Results
## =====
## Dep. Variable:                np.log(ahe)    R-squared:                0.030
## Model:                      OLS            Adj. R-squared:          0.030
## Method:                     Least Squares   F-statistic:             62.57
## Date:                       Br, 06 дек 2022  Prob (F-statistic):      4.22e-15
## Time:                       18:04:02        Log-Likelihood:          -1385.6
## No. Observations:           2004           AIC:                   2775.
## Df Residuals:                2002          BIC:                   2786.
## Df Model:                    1
## Covariance Type:             nonrobust
## =====
##               coef      std err          t      P>|t|      [0.025      0.975]
## -----
## Intercept           0.0996      0.382        0.261      0.794      -0.649      0.848
## np.log(age)          0.8911      0.113        7.910      0.000        0.670      1.112
## =====
## Omnibus:                 96.243    Durbin-Watson:           1.965
## Prob(Omnibus):            0.000    Jarque-Bera (JB):        132.906
## Skew:                     -0.450    Prob(JB):                1.38e-29
## Kurtosis:                 3.884    Cond. No.                130.
## =====
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```



```
model = smf.ols("np.log(ahe) ~ age + I(age ** 2)", data=dataset_filtered)
model_est = model.fit()
ahe_4 = model_est.predict(dataset_filtered)
print(model_est.summary())
```

```
##                               OLS Regression Results
## =====
## Dep. Variable:                np.log(ahe)    R-squared:                0.030
## Model:                      OLS            Adj. R-squared:          0.029
## Method:                     Least Squares   F-statistic:             31.29
## Date:                       Br, 06 дек 2022  Prob (F-statistic):      4.16e-14
## Time:                       18:04:05        Log-Likelihood:          -1385.6
## No. Observations:           2004           AIC:                   2777.
## Df Residuals:                2001          BIC:                   2794.
## Df Model:                    2
## Covariance Type:             nonrobust
## =====
##               coef      std err          t      P>|t|      [0.025      0.975]
## -----
## Intercept           1.4915      1.337        1.116      0.265      -1.130      4.113
## age                  0.0799      0.091        0.878      0.380      -0.099      0.258
## I(age ** 2)         -0.0008      0.002       -0.546      0.585      -0.004      0.002
## =====
## Omnibus:                 96.180    Durbin-Watson:           1.965
## Prob(Omnibus):            0.000    Jarque-Bera (JB):        132.725
## Skew:                     -0.450    Prob(JB):                1.51e-29
## Kurtosis:                 3.883    Cond. No.                1.12e+05
## =====
##
## Notes:
```

```
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.  
## [2] The condition number is large, 1.12e+05. This might indicate that there are  
## strong multicollinearity or other numerical problems.
```

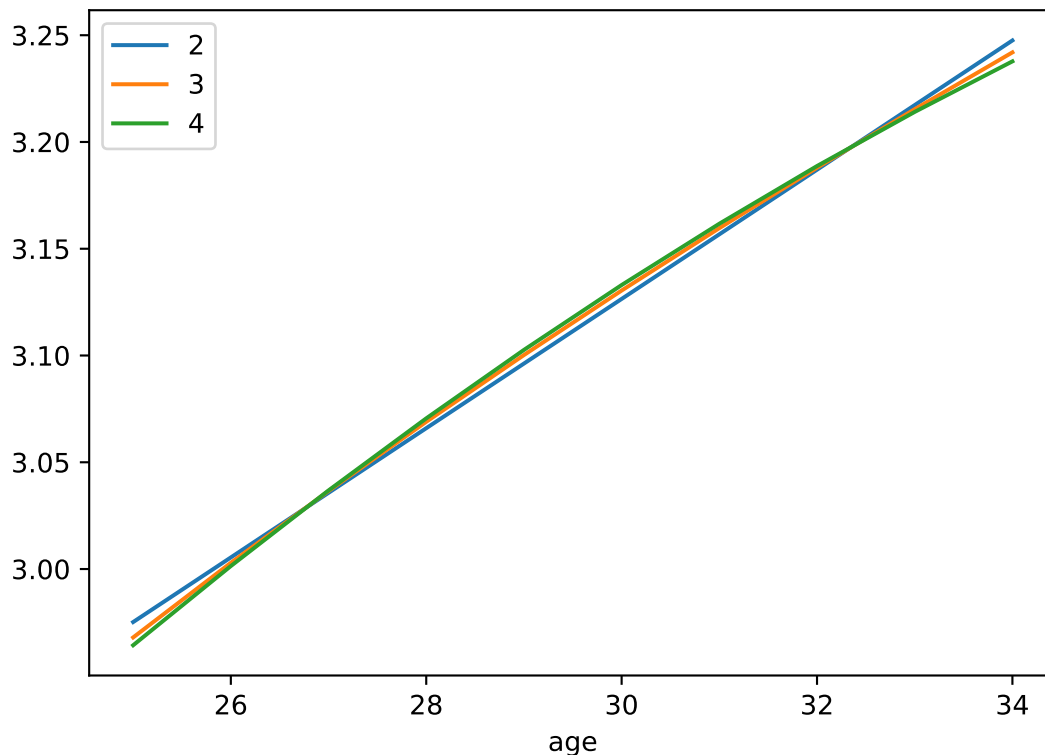
Обратите внимание, что к столбцам dataset-а можно обращаться как через [], так и через точку.

Построим графики.

Python



```
plt.clf()  
sb.lineplot(x=dataset_filtered.age, y=ahe_2, label="2")  
sb.lineplot(x=dataset_filtered.age, y=ahe_3, label="3")  
sb.lineplot(x=dataset_filtered.age, y=ahe_4, label="4")  
plt.show()
```



? Оцените регрессию логарифма средней почасовой зарплаты  $\ln(Ahe)$  на возраст  $Age$ ,  $Age^2$ , пол  $Female$ , образование  $Bachelor$  и компоненту взаимодействия  $Female \times Bachelor$ . Что измеряет коэффициент при компоненте взаимодействия?

R



```
model4 <- lm(log(ahe) ~ age + I(age^2) + female + bachelor +  
  female:bachelor, data = dataset)  
summary(model4)
```

```
##
## Call:
## lm(formula = log(ahe) ~ age + I(age^2) + female + bachelor +
##     female:bachelor, data = dataset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.28762 -0.29105  0.01547  0.31522  1.83505
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.8037407   0.6693037   1.201   0.2298
## age           0.1043224   0.0455869   2.288   0.0221 *
## I(age^2)      -0.0013316   0.0007705  -1.728   0.0840 .
## female        -0.2423732   0.0170102 -14.249 < 2e-16 ***
## bachelor       0.4004463   0.0146306  27.370 < 2e-16 ***
## female:bachelor 0.0898571   0.0228090   3.940 8.24e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4777 on 7434 degrees of freedom
## Multiple R-squared:  0.1984, Adjusted R-squared:  0.1978
## F-statistic: 367.9 on 5 and 7434 DF, p-value: < 2.2e-16
```

Или то же самое можно записать как

R



```
model4 <- lm(log(ahe) ~ age + I(age^2) + female*bachelor, data = dataset)
summary(model4)
```

```
##
## Call:
## lm(formula = log(ahe) ~ age + I(age^2) + female * bachelor, data = dataset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.28762 -0.29105  0.01547  0.31522  1.83505
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.8037407   0.6693037   1.201   0.2298
## age           0.1043224   0.0455869   2.288   0.0221 *
## I(age^2)      -0.0013316   0.0007705  -1.728   0.0840 .
## female        -0.2423732   0.0170102 -14.249 < 2e-16 ***
## bachelor       0.4004463   0.0146306  27.370 < 2e-16 ***
## female:bachelor 0.0898571   0.0228090   3.940 8.24e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4777 on 7434 degrees of freedom
## Multiple R-squared:  0.1984, Adjusted R-squared:  0.1978
## F-statistic: 367.9 on 5 and 7434 DF, p-value: < 2.2e-16
```

Python



```
model4 = smf.ols("np.log(ahe) ~ age + I(age**2) + female*bachelor",
    data=dataset)
model4_est = model4.fit()
print(model4_est.summary())
```

```
##                               OLS Regression Results
## =====
## Dep. Variable:                np.log(ahe)    R-squared:                0.198
```

```
## Model: OLS Adj. R-squared: 0.198
## Method: Least Squares F-statistic: 367.9
## Date: Вт, 06 дек 2022 Prob (F-statistic): 0.00
## Time: 18:04:12 Log-Likelihood: -5057.4
## No. Observations: 7440 AIC: 1.013e+04
## Df Residuals: 7434 BIC: 1.017e+04
## Df Model: 5
## Covariance Type: nonrobust
## =====
##               coef      std err          t      P>|t|      [0.025      0.975]
## -----
## Intercept      0.8037      0.669        1.201      0.230     -0.508      2.116
## age            0.1043      0.046        2.288      0.022      0.015      0.194
## I(age ** 2)    -0.0013      0.001       -1.728      0.084     -0.003      0.000
## female        -0.2424      0.017     -14.249      0.000     -0.276     -0.209
## bachelor       0.4004      0.015     27.370      0.000      0.372      0.429
## female:bachelor 0.0899      0.023      3.940      0.000      0.045      0.135
## =====
## Omnibus:      319.786   Durbin-Watson:      1.933
## Prob(Omnibus): 0.000   Jarque-Bera (JB):    511.678
## Skew:         -0.379   Prob(JB):           7.77e-112
## Kurtosis:      4.038   Cond. No.           1.09e+05
## =====
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
## [2] The condition number is large, 1.09e+05. This might indicate that there are
## strong multicollinearity or other numerical problems.
```

? Оценив все эти регрессии (и любые другие, которые вы хотите оценить), сделайте выводы об эффекте влияния возраста на доходы для молодых работников.

Выведем оценки всех оцененных выше моделей.

R



```
stargazer::stargazer(model1, model2, model3, model4, type = "text")
```

```
##
##
## -----
##                                     Dependent variable:
## -----
##                                     log(ahe)
##               (1)               (2)               (3)
## (4)
## -----
## age                0.026***                0.104**
##   0.104**
##               (0.002)                (0.046)
##   (0.046)
##
## log(age)                0.753***
##               (0.057)
##
## I(age2)                -0.001*
##   -0.001*
##               (0.001)
##
## female              -0.192***              -0.192***              -0.192***
##   -0.242***
##               (0.011)              (0.011)              (0.011)
##   (0.017)
##
```

```
## bachelor          0.438***          0.438***          0.437***
↳ 0.400***
##          (0.011)          (0.011)          (0.011)
↳ (0.015)
##
## female:bachelor
↳ 0.090***
##
↳ (0.023)
##
## Constant          1.941***          0.150          0.792
↳ 0.804
##          (0.059)          (0.194)          (0.670)
↳ (0.669)
##
## -----
## Observations          7,440          7,440          7,440
↳ 7,440
## R2          0.196          0.197          0.197
↳ 0.198
## Adjusted R2          0.196          0.196          0.196
↳ 0.198
## Residual Std. Error    0.478 (df = 7436)    0.478 (df = 7436)    0.478 (df = 7435)
↳ 0.478 (df = 7434)
## F Statistic          605.726*** (df = 3; 7436) 606.413*** (df = 3; 7436) 455.156*** (df = 4; 7435)
↳ 367.940*** (df = 5; 7434)
##
↳ =====
## Note:
↳ *p<0.1; **p<0.05; ***p<0.01
```

## Python



```
print(yatg.html_2_ascii_table(Stargazer([model1_est, model2_est,
↳ model3_est, model4_est]).render_html()))
```

```
## |          |          |          |
↳ |          |          |          |
##
↳ |-----+-----+-----+-----+
## |          | Dependent variable:np.log(ahe) |          |
↳ |          |          |          |
## |          |          |          |
↳ |          |          |          |
## |          | (1)          | (2)          | (3)
↳ | (4)          |          |          |
## |          |          |          |
↳ |          |          |          |
## | I(age ** 2) |          |          | -0.001*
↳ | -0.001*     |          |          |
## |          |          |          | (0.001)
↳ | (0.001)     |          |          |
## | Intercept   | 1.941***    | 0.150        | 0.792
↳ | 0.804       |          |          |
## |          | (0.059)     | (0.194)      | (0.670)
↳ | (0.669)     |          |          |
## | age         | 0.026***    |          | 0.104**
↳ | 0.104**     |          |          |
## |          | (0.002)     |          | (0.046)
↳ | (0.046)     |          |          |
## | bachelor    | 0.438***    | 0.438***     | 0.437***
↳ | 0.400***    |          |          |
## |          | (0.011)     | (0.011)      | (0.011)
↳ | (0.015)     |          |          |
## | female      | -0.192***   | -0.192***    | -0.192***
↳ | -0.242***   |          |          |
## |          | (0.011)     | (0.011)      | (0.011)
↳ | (0.017)     |          |          |
```

```
## | female:bachelor | | |
↪ | 0.090*** | | |
## | | | | |
↪ | (0.023) | | |
## | np.log(age) | | 0.753*** | |
↪ | | | | |
## | | | | (0.057) | |
↪ | | | | | |
## | | | | | |
↪ | | | | | |
## | Observations | 7,440 | 7,440 | 7,440
↪ | 7,440 | | |
## | R2 | 0.196 | 0.197 | 0.197
↪ | 0.198 | | |
## | Adjusted R2 | 0.196 | 0.196 | 0.196
↪ | 0.198 | | |
## | Residual Std. Error | 0.478 (df=7436) | 0.478 (df=7436) | 0.478 (df=7435)
↪ | 0.478 (df=7434) | | |
## | F Statistic | 605.726*** (df=3; 7436) | 606.413*** (df=3; 7436) | 455.156***
↪ (df=4; 7435) | 367.940*** (df=5; 7434) | |
## | | | | |
↪ | | | | |
## | Note: | *p<0.1; **p<0.05; ***p<0.01 | |
↪ | | | | |
```

? Различается ли эффект влияния переменной *Age* на доходы для мужчин и женщин? Специфицируйте и оцените регрессию, которую вы можете использовать для ответа на этот вопрос.

R



```
model <- lm(ahe ~ female*age, data=dataset)
summary(model)
```

```
##
## Call:
## lm(formula = ahe ~ female * age, data = dataset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.619  -7.287  -2.204   4.814  71.549
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.30515    1.68342   1.963  0.0496 *
## female       3.60858    2.58661   1.395  0.1630
## age          0.59293    0.05645  10.503 <2e-16 ***
## female:age   -0.20823    0.08689  -2.397  0.0166 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.51 on 7436 degrees of freedom
## Multiple R-squared:  0.03328,    Adjusted R-squared:  0.03289
## F-statistic: 85.34 on 3 and 7436 DF,  p-value: < 2.2e-16
```

Python



```
model = smf.ols("ahe ~ female*age", data=dataset)
model_est = model.fit()
print(model_est.summary())
```

```
##
## ===== OLS Regression Results =====
## Dep. Variable:                ahe    R-squared:                0.033
```

```
## Model: OLS Adj. R-squared: 0.033
## Method: Least Squares F-statistic: 85.34
## Date: Вт, 06 дек 2022 Prob (F-statistic): 2.76e-54
## Time: 18:04:17 Log-Likelihood: -28056.
## No. Observations: 7440 AIC: 5.612e+04
## Df Residuals: 7436 BIC: 5.615e+04
## Df Model: 3
## Covariance Type: nonrobust
## =====
##              coef      std err          t      P>|t|      [0.025      0.975]
## -----
## Intercept      3.3052        1.683        1.963      0.050        0.005        6.605
## female         3.6086        2.587        1.395      0.163       -1.462        8.679
## age            0.5929        0.056       10.503      0.000        0.482        0.704
## female:age     -0.2082        0.087       -2.397      0.017       -0.379       -0.038
## =====
## Omnibus:            1912.305   Durbin-Watson:           1.845
## Prob(Omnibus):      0.000   Jarque-Bera (JB):       5016.515
## Skew:               1.384   Prob(JB):              0.00
## Kurtosis:           5.920   Cond. No.              775.
## =====
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

? Различается ли эффект влияния переменной *Age* на доходы для людей имеющих среднее и высшее образование?

R



```
model <- lm(ahe ~ bachelor*age, data=dataset)
summary(model)
```

```
##
## Call:
## lm(formula = ahe ~ bachelor * age, data = dataset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24.077  -6.387  -1.774   4.105  76.409
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.31392    1.73520   2.486  0.01294 *
## bachelor     0.02023    2.39806   0.008  0.99327
## age          0.38333    0.05822   6.584 4.89e-11 ***
## bachelor:age  0.26097    0.08051   3.241  0.00119 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.849 on 7436 degrees of freedom
## Multiple R-squared:  0.1509, Adjusted R-squared:  0.1506
## F-statistic: 440.6 on 3 and 7436 DF, p-value: < 2.2e-16
```

Python



```
model = smf.ols("ahe ~ bachelor*age", data=dataset)
model_est = model.fit()
print(model_est.summary())
```

```
## OLS Regression Results
## =====
## Dep. Variable: ahe R-squared: 0.151
```



```

## Model:                                OLS    Adj. R-squared:            0.151
## Method:                               Least Squares    F-statistic:            440.6
## Date:                                Вт, 06 дек 2022    Prob (F-statistic):      1.66e-263
## Time:                                18:04:20          Log-Likelihood:          -27573.
## No. Observations:                    7440          AIC:                    5.515e+04
## Df Residuals:                        7436          BIC:                    5.518e+04
## Df Model:                            3
## Covariance Type:                     nonrobust
## =====
##               coef      std err          t      P>|t|      [0.025      0.975]
## -----
## Intercept                4.3139        1.735        2.486      0.013        0.912        7.715
## bachelor                 0.0202        2.398        0.008      0.993       -4.681        4.721
## age                     0.3833        0.058        6.584      0.000        0.269        0.497
## bachelor:age             0.2610        0.081        3.241      0.001        0.103        0.419
## =====
## Omnibus:                     2042.778    Durbin-Watson:           1.942
## Prob(Omnibus):              0.000    Jarque-Bera (JB):       6342.505
## Skew:                       1.405    Prob(JB):               0.00
## Kurtosis:                   6.545    Cond. No.               840.
## =====
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```



Так как прочие задания не предусматривают оценивания моделей, то опустим их.

## Сток-Уотсон. Задания E6.2 и E7.3

Напомним, что в R пакеты **можно** импортировать в т.н. глобальное **пространство имен** и вызывать функции **без** указания пакетов при помощи `::`!

В python мы **должны** импортировать все что нам нужно. Модули для загрузки данных и оценивания моделей.

Python



```
import numpy as np
import pandas as pd
import statsmodels.formula.api as smf
```

Модуль для статистических тестов.

Python



```
import statsmodels.stats.api as sms
```

Модуль для одновременного вывода результатов нескольких моделей.

Python



```
from stargazer.stargazer import Stargazer
import yatg
```

## Загрузка данных

Скачаем файл `CollegeDistance.xls` и загрузим его:

R



```
dataset <- readxl::read_excel("D:/CollegeDistance.xls")
head(dataset)
```

```
## # A tibble: 6 x 14
##   female black hispanic bytest dadcoll momcoll ownhome urban cue80 stwmf~1 dist
##   <dbl> <dbl>   <dbl> <dbl>   <dbl>   <dbl>   <dbl> <dbl> <dbl>   <dbl> <dbl>
## 1     0     0       0  39.2     1     0       1     1   6.2    8.09  0.2
## 2     1     0       0  48.9     0     0       1     1   6.2    8.09  0.2
## 3     0     0       0  48.7     0     0       1     1   6.2    8.09  0.2
## 4     0     1       0  40.4     0     0       1     1   6.2    8.09  0.2
## 5     1     0       0  40.5     0     0       0     1   5.6    8.09  0.4
```

```
## 6      0      0      0 54.7      0      0      1      1 5.6      8.09      0.4
## # ... with 3 more variables: tuition <dbl>, ed <dbl>, incomehi <dbl>, and
## # abbreviated variable name 1: stwmfg80
```

## Python



```
dataset = pd.read_excel("D:/CollegeDistance.xls")
dataset.head()
```

```
##      female  black  hispanic  bytest  ...  dist  tuition  ed  incomehi
## 0          0      0          0  39.15  ...  0.2  0.88915  12          1
## 1          1      0          0  48.87  ...  0.2  0.88915  12          0
## 2          0      0          0  48.74  ...  0.2  0.88915  12          0
## 3          0      1          0  40.40  ...  0.2  0.88915  12          0
## 4          1      0          0  40.48  ...  0.4  0.88915  13          0
##
## [5 rows x 14 columns]
```

## Оценки моделей

❓ Оцените регрессию числа полных лет обучения *Ed* от расстояния до ближайшего колледжа *Dist*. Чему равен оцененный коэффициент наклона?

В R для этих целей идеально подходит **встроенная** функция `lm()`. Она требует минимум два аргумента: формулу и источник данных.

## R



```
model_pair <- lm(ed ~ dist, data = dataset)
summary(model_pair)
```

```
##
## Call:
## lm(formula = ed ~ dist, data = dataset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.9559 -1.8091 -0.6624  2.0515  4.4844
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  13.95586    0.03772  369.945  <2e-16 ***
## dist        -0.07337    0.01375  -5.336   1e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.807 on 3794 degrees of freedom
## Multiple R-squared:  0.00745,    Adjusted R-squared:  0.007188
## F-statistic: 28.48 on 1 and 3794 DF,  p-value: 1.004e-07
```

В python нам сначала надо создать объект класса `ols`, конструктор которого, `ols(:)`, принимает такие же аргументы, как и `lm()` в R.



```
model_pair = smf.ols("ed ~ dist", data=dataset)
model_pair_est = model_pair.fit()
print(model_pair_est.summary())
```

```
##                               OLS Regression Results
## =====
## Dep. Variable:                ed    R-squared:                0.007
## Model:                      OLS    Adj. R-squared:           0.007
## Method:                    Least Squares    F-statistic:           28.48
## Date:                      Br, 06 дек 2022    Prob (F-statistic):      1.00e-07
## Time:                      18:04:34    Log-Likelihood:         -7632.2
## No. Observations:          3796    AIC:                   1.527e+04
## Df Residuals:              3794    BIC:                   1.528e+04
## Df Model:                  1
## Covariance Type:            nonrobust
## =====
##               coef      std err          t      P>|t|      [0.025      0.975]
## -----
## Intercept      13.9559      0.038     369.945      0.000      13.882      14.030
## dist           -0.0734      0.014     -5.336      0.000      -0.100      -0.046
## =====
## Omnibus:                 7187.794    Durbin-Watson:           1.769
## Prob(Omnibus):           0.000    Jarque-Bera (JB):         361.676
## Skew:                   0.410    Prob(JB):                 2.90e-79
## Kurtosis:               1.729    Cond. No.                 3.73
## =====
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

? Оцените регрессию *Ed* на *Dist*, включив дополнительные переменные для контроля за характеристиками студентов и их семей, а также местного рынка труда. В частности, включите в качестве дополнительных регрессоров переменные *Bytest*, *Female*, *Black*, *Hispanic*, *Incomehi*, *Ownhome*, *DadColl*, *Cue80*, и *Stwmfg80*. Каково оцененное влияние *Dist* на *Ed*?



```
model_mult <- lm(ed ~ dist + bytest + female + black + hispanic + incomehi
  + ownhome + dadcoll + momcoll + cue80 + stwmfg80, data = dataset)
summary(model_mult)
```

```
##
## Call:
## lm(formula = ed ~ dist + bytest + female + black + hispanic +
##      incomehi + ownhome + dadcoll + momcoll + cue80 + stwmfg80,
##      data = dataset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.2752 -1.1429 -0.2216  1.1733  5.0559
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.861373   0.249705   35.487 < 2e-16 ***
## dist        -0.030804   0.012338   -2.497  0.01258 *
## bytest       0.092447   0.003167   29.187 < 2e-16 ***
## female       0.143378   0.050454    2.842  0.00451 **
## black        0.353808   0.071235    4.967 7.11e-07 ***
## hispanic     0.402351   0.074264    5.418 6.41e-08 ***
## incomehi     0.366595   0.060679    6.042 1.67e-09 ***
## ownhome      0.145642   0.066641    2.185  0.02892 *
```

```
## dadcoll      0.569915    0.073718    7.731 1.36e-14 ***
## momcoll      0.379184    0.081550    4.650 3.44e-06 ***
## cue80        0.024418    0.009609    2.541 0.01109 *
## stwmfg80     -0.050204    0.019801   -2.535 0.01127 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.538 on 3784 degrees of freedom
## Multiple R-squared:  0.2829, Adjusted R-squared:  0.2809
## F-statistic: 135.7 on 11 and 3784 DF,  p-value: < 2.2e-16
```

## Python



```
model_mult = smf.ols("ed ~ dist + bytest + female + black + hispanic +
↳ incomehi + ownhome + dadcoll + momcoll + cue80 + stwmfg80",
↳ data=dataset)
model_mult_est = model_mult.fit()
print(model_mult_est.summary())
```

```
##                               OLS Regression Results
## =====
## Dep. Variable:                ed      R-squared:                0.283
## Model:                      OLS      Adj. R-squared:          0.281
## Method:                     Least Squares      F-statistic:          135.7
## Date:                       Br, 06 дек 2022      Prob (F-statistic):    1.92e-263
## Time:                       18:04:36      Log-Likelihood:        -7015.1
## No. Observations:           3796      AIC:                  1.405e+04
## Df Residuals:               3784      BIC:                  1.413e+04
## Df Model:                   11
## Covariance Type:            nonrobust
## =====
##               coef      std err          t      P>|t|      [0.025      0.975]
## -----
## Intercept      8.8614      0.250      35.487      0.000      8.372      9.351
## dist          -0.0308      0.012      -2.497      0.013     -0.055     -0.007
## bytest         0.0924      0.003      29.187      0.000      0.086      0.099
## female         0.1434      0.050       2.842      0.005      0.044      0.242
## black          0.3538      0.071       4.967      0.000      0.214      0.493
## hispanic       0.4024      0.074       5.418      0.000      0.257      0.548
## incomehi       0.3666      0.061       6.042      0.000      0.248      0.486
## ownhome        0.1456      0.067       2.185      0.029      0.015      0.276
## dadcoll        0.5699      0.074       7.731      0.000      0.425      0.714
## momcoll        0.3792      0.082       4.650      0.000      0.219      0.539
## cue80          0.0244      0.010       2.541      0.011      0.006      0.043
## stwmfg80      -0.0502      0.020      -2.535      0.011     -0.089     -0.011
## =====
## Omnibus:                116.663      Durbin-Watson:           1.928
## Prob(Omnibus):          0.000      Jarque-Bera (JB):        98.499
## Skew:                   0.326      Prob(JB):                4.08e-22
## Kurtosis:               2.554      Cond. No.                539.
## =====
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Так как мы оценили модель множественной регрессии, имеет смысл рассмотреть критерий VIF.

$$VIF_j = \frac{1}{1 - R_j^2}$$

Он позволяет выявить наличие мультиколлинеарности в модели.

R



```
car::vif(model_mult)
```

```
##      dist   bytest   female   black hispanic incomehi ownhome dadcoll
## 1.111543 1.251421 1.012485 1.265679 1.127367 1.207018 1.054777 1.405525
## momcoll   cue80 stwmfg80
## 1.279515 1.216229 1.170605
```

В python есть функция, вычисляющая значение VIF для одной переменной, которую можно найти в пакете statsmodels.

Python



```
from statsmodels.stats.outliers_influence import variance_inflation_factor
↳ as vif
```

При помощи несложного программирования (а если вы считаете в питоне, то оно несложное) можно вывести VIF для всех переменных: надо взять матрицу значений объясняющих переменных из **не оцененной** модели `model`, найти число столбцов через `.shape[1]` и для каждого из них рассчитать VIF, не забывая, что нумерация начинается с 0, и что в модели с константой на первом месте стоит именно она.

Функция, которую мы импортировали под именем `vif` принимает на вход матрицу значений объясняющих переменных и номер столбца, в котором хранится переменная, для которой мы считаем VIF.

Python



```
for i in range(1, model_mult.exog.shape[1]):
    vif_est = vif(model_mult.exog, i)
    print(f"{model_mult_est.params.index[i]:8}: {vif_est:5.4f}")
```

```
## dist      : 1.1115
## bytest    : 1.2514
## female    : 1.0125
## black     : 1.2657
## hispanic  : 1.1274
## incomehi  : 1.2070
## ownhome   : 1.0548
## dadcoll   : 1.4055
## momcoll   : 1.2795
## cue80     : 1.2162
## stwmfg80  : 1.1706
```

**?** Отличается ли существенно предполагаемое влияние *Dist* на *Ed* в регрессии из пункта (2) от предполагаемого влияния *Dist* на *Ed* в регрессии из пункта (1)? Основываясь на этом выводе, что вы можете сказать о наличии смещения из-за пропущенной переменной оценок регрессии из пункта (1)?

Сохраним таблицу с коэффициентами и соответствующими статистиками из `summary` и найдем значение t-статистики для нулевой гипотезы  $\beta_{dist} = -0.07337$ .

R



```
summary_table <- summary(model_mult)$coefficients
(summary_table["dist", 1] + 0.07337) / summary_table["dist", 2]
```

```
## [1] 3.45071
```

В python же можно воспользоваться методом `t_test` **оцененной** модели, который принимает на вход те же аргументы, что и описанный выше `f_test`. В результате получим таблицу со значениями t-статистик для каждого ограничения.

Python



```
model_mult_est.t_test("dist = -0.07337")
```

```
## <class 'statsmodels.stats.contrast.ContrastResults'>
##                               Test for Constraints
## =====
##               coef      std err          t      P>|t|      [0.025      0.975]
## -----
## c0             -0.0308      0.012      3.450      0.001     -0.055     -0.007
## =====
```

❓ Есть и другие факторы, которые влияют число полных лет обучения. Изменится ли оценка влияния расстояния до ближайшего колледжа на число полных лет обучения, если контролировать на эти другие факторы?

В данном задании требуется одновременно вывести результаты нескольких оцененных моделей. Выведем наши две оцененные модели. В R укажем тип вывода `text`, иначе получим  $\LaTeX$  код.

R



```
stargazer::stargazer(model_pair, model_mult, type = "text")
```

```
##
## =====
##                               Dependent variable:
##               -----
##                               ed
##               (1)                (2)
## -----
## dist             -0.073***      -0.031**
##                   (0.014)        (0.012)
##
## bytest                        0.092***
##                               (0.003)
##
## female                        0.143***
##                               (0.050)
##
## black                      0.354***
##                               (0.071)
##
## hispanic                  0.402***
##                               (0.074)
##
## incomehi                0.367***
##                               (0.061)
##
```

```
## ownhome                0.146**
##                        (0.067)
##
## dadcoll                0.570***
##                        (0.074)
##
## momcoll               0.379***
##                        (0.082)
##
## cue80                 0.024**
##                        (0.010)
##
## stwmfg80             -0.050**
##                        (0.020)
##
## Constant              13.956***
##                        (0.038)
##                        8.861***
##                        (0.250)
## -----
## Observations          3,796
## R2                    0.007
## Adjusted R2           0.007
## Residual Std. Error   1.807 (df = 3794)
## F Statistic           28.476*** (df = 1; 3794)
##                      135.733*** (df = 11; 3784)
## =====
## Note:                  *p<0.1; **p<0.05; ***p<0.01
```

В python есть порт пакета `stargazer`, но он не умеет выводить таблицы в текстовом виде, только в html и  $\text{\LaTeX}$ . Если вы работаете в Jupyter, то достаточно следующей команды:

Python



```
Stargazer([model_pair_est, model_mult_est])
```

```
## <stargazer.stargazer.Stargazer object at 0x00000152E8E58E50>
```

Если же вы работаете в чем-то ином — консоли или R Markdown (в котором данный мануал и написан) — то можно воспользоваться, например, `yatg`.

Python



```
print(yatg.html_2_ascii_table(Stargazer([model_pair_est,
    ↪ model_mult_est]).render_html()))
```

```
## | | | |
## |-----|-----|-----|
## | | Dependent variable:ed | |
## | | (1) | (2) |
## | Intercept | 13.956*** | 8.861*** |
## | | | (0.250) |
## | | | (0.038) |
## | black | | 0.354*** |
## | | | (0.071) |
## | bytest | | 0.092*** |
## | | | (0.003) |
## | cue80 | | 0.024** |
## | | | (0.010) |
## | dadcoll | | 0.570*** |
## | | | (0.074) |
## | dist | -0.073*** | -0.031** |
## | | | (0.012) |
## | female | (0.014) | 0.143*** |
```



##			(0.050)
##	hispanic		0.402***
##			(0.074)
##	incomehi		0.367***
##			(0.061)
##	momcoll		0.379***
##			(0.082)
##	ownhome		0.146**
##			(0.067)
##	stwmfg80		-0.050**
##			(0.020)
##	Observations	3,796	3,796
##	R2	0.007	0.283
##	Adjusted R2	0.007	0.281
##	Residual Std. Error	1.807 (df=3794)	1.538 (df=3784)
##	F Statistic	28.476*** (df=1; 3794)	135.733*** (df=11; 3784)
##	Note:	*p<0.1; **p<0.05; ***p<0.01	



Так как прочие задания не предусматривают оценивания моделей, то опустим их.

## Сток-Уотсон. Задания E7.4 и E8.2

Напомним, что в R пакеты **можно** импортировать в т.н. глобальное **пространство имен** и вызывать функции **без** указания пакетов при помощи `::`!

Впрочем, пакет `ggplot2` лучше импортировать, так как иначе придется приписывать `ggplot2::` к каждому компоненту команды.

R

```
library(ggplot2)
```

В python мы **должны** импортировать все что нам нужно. Модули для загрузки данных и оценивания моделей.

Python

```
import numpy as np
import pandas as pd
import statsmodels.formula.api as smf
```

Модули для графиков.

Python

```
import matplotlib.pyplot as plt
import seaborn as sb
```

## Загрузка данных

Скачаем файл `Growth.xlsx` и загрузим его. Причем по условию задачи нам надо убрать из данных Мальту.

R

```
dataset <- readxl::read_excel("D:/Growth.xlsx")
dataset <- dataset[dataset$country_name != "Malta", ]
head(dataset)
```

```
## # A tibble: 6 x 8
##   country_name growth    oil rgdp60 tradeshare yearsschool rev_coups assassinat-1
##   <chr>         <dbl> <dbl> <dbl>      <dbl>      <dbl>      <dbl>      <dbl>
## 1 India         1.92     0    766.      0.141      1.45      0.133      0.867
```

```
## 2 Argentina      0.618      0 4462.      0.157      4.99      0.933      1.93
## 3 Japan          4.30       0 2954.      0.158      6.71      0        0.200
## 4 Brazil         2.93       0 1784.      0.160      2.89      0.100     0.100
## 5 United States  1.71       0 9895.      0.161      8.66      0        0.433
## 6 Bangladesh     0.708      0  952.      0.221      0.790     0.306     0.175
## # ... with abbreviated variable name 1: assassinations
```

## Python



```
dataset = pd.read_excel("D:/Growth.xlsx")
```

```
## C:\Users\vadim\AppData\Roaming\Python\Python311\site-packages\openpyxl\worksheet\_reader.py:312:
↳ UserWarning: Unknown extension is not supported and will be removed
## warn(msg)
```

## Python



```
dataset = dataset.loc[dataset.country_name != "Malta", ]
dataset.head()
```

```
##      country_name  growth  oil  ...  yearsschool  rev_coups  assassinations
## 0      India      1.915168    0  ...           1.45  0.133333      0.866667
## 1    Argentina      0.617645    0  ...           4.99  0.933333      1.933333
## 2      Japan      4.304759    0  ...           6.71  0.000000      0.200000
## 3      Brazil      2.930097    0  ...           2.89  0.100000      0.100000
## 4  United States      1.712265    0  ...           8.66  0.000000      0.433333
##
## [5 rows x 8 columns]
```

## Оценки моделей

❓ Оцените регрессию *Growth* от переменных *TradeShare*, *YearsSchool*, *Rev\_Coups*, *Assassinations* и *RGDP60*. Постройте 95%-й доверительный интервал для коэффициента при *TradeShare*. Является ли этот коэффициент статистически значимым на 5%-м уровне значимости?

## R



```
model <- lm(growth ~ tradeshare + yearsschool + rev_coups + assassinations +
↳ rgdp60, data = dataset)
summary(model)
```

```
##
## Call:
## lm(formula = growth ~ tradeshare + yearsschool + rev_coups +
##      assassinations + rgdp60, data = dataset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.6897 -0.9459 -0.0565  0.8286  5.1534
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.6268915   0.7830280   0.801   0.42663
## tradeshare     1.3408193   0.9600631   1.397   0.16786
## yearsschool    0.5642445   0.1431131   3.943   0.00022 ***
## rev_coups     -2.1504256   1.1185900  -1.922   0.05947 .
##
```

```
## assassinations 0.3225844 0.4880043 0.661 0.51121
## rgdp60 -0.0004613 0.0001508 -3.059 0.00336 **
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.594 on 58 degrees of freedom
## Multiple R-squared: 0.2911, Adjusted R-squared: 0.23
## F-statistic: 4.764 on 5 and 58 DF, p-value: 0.001028
```

## Python



```
model = smf.ols("growth ~ tradeshare + yearsschool + rev_coups +
↳ assassinations + rgdp60", data=dataset)
model_est = model.fit()
print(model_est.summary())
```

```
##                                OLS Regression Results
## =====
## Dep. Variable:                growth    R-squared:                0.291
## Model:                      OLS        Adj. R-squared:         0.230
## Method:                     Least Squares    F-statistic:             4.764
## Date:                       Br, 06 дек 2022    Prob (F-statistic):      0.00103
## Time:                       18:05:01    Log-Likelihood:         -117.49
## No. Observations:            64          AIC:                   247.0
## Df Residuals:                58          BIC:                   259.9
## Df Model:                    5
## Covariance Type:             nonrobust
## =====
##                                coef    std err          t      P>|t|      [0.025    0.975]
## -----
## Intercept                    0.6269    0.783        0.801    0.427    -0.941    2.194
## tradeshare                   1.3408    0.960        1.397    0.168    -0.581    3.263
## yearsschool                  0.5642    0.143        3.943    0.000    0.278    0.851
## rev_coups                   -2.1504    1.119       -1.922    0.059    -4.390    0.089
## assassinations               0.3226    0.488        0.661    0.511    -0.654    1.299
## rgdp60                      -0.0005    0.000       -3.059    0.003    -0.001   -0.000
## =====
## Omnibus:                      7.569    Durbin-Watson:           2.130
## Prob(Omnibus):                0.023    Jarque-Bera (JB):        8.597
## Skew:                         0.494    Prob(JB):                0.0136
## Kurtosis:                     4.499    Cond. No.                2.59e+04
## =====
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
## [2] The condition number is large, 2.59e+04. This might indicate that there are
## strong multicollinearity or other numerical problems.
```

Проверьте гипотезу о том, что *YearsSchool*, *Rev\_Coups*, *Assassinations* и *RGDP60* можно одновременно исключить из регрессии. Чему равно р-значение F-статистики?

## R



```
car::linearHypothesis(model, c("yearsschool=0", "rev_coups=0",
↳ "assassinations=0", "rgdp60=0"))
```

```
## Linear hypothesis test
##
## Hypothesis:
## yearsschool = 0
## rev_coups = 0
## assassinations = 0
## rgdp60 = 0
```

```
##
## Model 1: restricted model
## Model 2: growth ~ tradeshare + yearsschool + rev_coups + assassinations +
##   rgdp60
##
##   Res.Df    RSS Df Sum of Sq    F    Pr(>F)
## 1      62 198.53
## 2      58 147.31  4    51.217 5.0414 0.001481 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## Python



```
model_est.f_test("yearsschool = 0, rev_coups = 0, assassinations = 0, rgdp60
↪ = 0")
```

```
## <class 'statsmodels.stats.contrast.ContrastResults'>
## <F test: F=5.041359542406447, p=0.0014806697273907743, df_denom=58, df_num=4>
```

? Оцените следующие пять регрессий переменной *Growth* на

- *tradeshare* и *yearsschool*.
- *tradeshare* и логарифм логарифм *yearsschool*.
- *tradeshare*, логарифм *yearsschool*, *rev\_coups*, *assassinations* и логарифм *rgdp60*.
- *tradeshare*, логарифм *yearsschool*, их произведение, *rev\_coups*, *assassinations* и логарифм *rgdp60*.
- *growth* на *tradeshare*, его квадрат и куб, логарифм *yearsschool*, *rev\_coups*, *assassinations* и логарифм *rgdp60*.

## R



```
model1 <- lm(growth ~ tradeshare + yearsschool, data=dataset)
summary(model1)
```

```
##
## Call:
## lm(formula = growth ~ tradeshare + yearsschool, data = dataset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.4896 -0.9538 -0.3304  0.7680  5.5663
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.1222     0.6627  -0.184  0.85426
## tradeshare    1.8978     0.9361   2.027  0.04699 *
## yearsschool   0.2430     0.0837   2.903  0.00514 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.691 on 61 degrees of freedom
## Multiple R-squared:  0.1606, Adjusted R-squared:  0.1331
## F-statistic: 5.836 on 2 and 61 DF, p-value: 0.004796
```



```
model1 = smf.ols("growth ~ tradeshare + yearsschool", data=dataset)
model1_est = model1.fit()
print(model1_est.summary())
```

```
##                                OLS Regression Results
## =====
## Dep. Variable:                growth    R-squared:                0.161
## Model:                        OLS        Adj. R-squared:        0.133
## Method:                      Least Squares    F-statistic:            5.836
## Date:                        Br, 06 дек 2022    Prob (F-statistic):      0.00480
## Time:                        18:05:07    Log-Likelihood:          -122.90
## No. Observations:            64        AIC:                    251.8
## Df Residuals:                61        BIC:                    258.3
## Df Model:                    2
## Covariance Type:              nonrobust
## =====
##                                coef    std err          t      P>|t|    [0.025    0.975]
## -----
## Intercept                    -0.1222     0.663     -0.184     0.854    -1.447     1.203
## tradeshare                   1.8978     0.936     2.027     0.047     0.026     3.770
## yearsschool                   0.2430     0.084     2.903     0.005     0.076     0.410
## =====
## Omnibus:                      10.227    Durbin-Watson:           2.156
## Prob(Omnibus):                 0.006    Jarque-Bera (JB):        10.906
## Skew:                          0.745    Prob(JB):                 0.00428
## Kurtosis:                      4.368    Cond. No.                 24.9
## =====
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```



```
model2 <- lm(growth ~ tradeshare + log(yearsschool), data=dataset)
summary(model2)
```

```
##
## Call:
## lm(formula = growth ~ tradeshare + log(yearsschool), data = dataset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.6162 -0.9887 -0.2074  0.7479  5.2958
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.1857    0.5643  -0.329   0.7432
## tradeshare     1.7490    0.8600   2.034   0.0463 *
## log(yearsschool) 1.0163    0.2231   4.556 2.56e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.558 on 61 degrees of freedom
## Multiple R-squared:  0.2872, Adjusted R-squared:  0.2638
## F-statistic: 12.29 on 2 and 61 DF, p-value: 3.282e-05
```



```
model2 = smf.ols("growth ~ tradeshare + np.log(yearsschool)", data=dataset)
model2_est = model2.fit()
print(model2_est.summary())
```

```
##                                OLS Regression Results
## =====
## Dep. Variable:                growth    R-squared:                0.287
## Model:                        OLS        Adj. R-squared:        0.264
## Method:                       Least Squares    F-statistic:             12.29
## Date:                         Br, 06 дек 2022    Prob (F-statistic):      3.28e-05
## Time:                         18:05:10    Log-Likelihood:          -117.67
## No. Observations:              64        AIC:                    241.3
## Df Residuals:                  61        BIC:                    247.8
## Df Model:                      2
## Covariance Type:              nonrobust
## =====
##                                coef      std err          t      P>|t|      [0.025      0.975]
## -----
## Intercept                    -0.1857      0.564      -0.329      0.743      -1.314      0.943
## tradeshare                   1.7490      0.860       2.034      0.046       0.029      3.469
## np.log(yearsschool)          1.0163      0.223       4.556      0.000       0.570      1.462
## =====
## Omnibus:                      14.404    Durbin-Watson:           2.127
## Prob(Omnibus):                0.001    Jarque-Bera (JB):        16.497
## Skew:                         1.006    Prob(JB):                0.000262
## Kurtosis:                     4.461    Cond. No.                8.69
## =====
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

R



```
model3 <- lm(growth ~ tradeshare + log(yearsschool) + rev_coups +
  ↪ assassinations + log(rgdp60), data=dataset)
summary(model3)
```

```
##
## Call:
## lm(formula = growth ~ tradeshare + log(yearsschool) + rev_coups +
##     assassinations + log(rgdp60), data = dataset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.2482 -0.9214 -0.0839  0.7938  4.1158
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   11.7459     2.9198   4.023 0.000168 ***
## tradeshare     1.1035     0.8332   1.325 0.190528
## log(yearsschool) 2.1613     0.3627   5.960 1.59e-07 ***
## rev_coups     -2.2995     1.0045  -2.289 0.025719 *
## assassinations  0.2277     0.4337   0.525 0.601501
## log(rgdp60)    -1.6211     0.3985  -4.068 0.000145 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.4 on 58 degrees of freedom
## Multiple R-squared:  0.4532, Adjusted R-squared:  0.406
## F-statistic: 9.613 on 5 and 58 DF, p-value: 1.012e-06
```

Python



```
model3 = smf.ols("growth ~ tradeshare + np.log(yearsschool) + rev_coups +
  ↪ assassinations + np.log(rgdp60)", data=dataset)
model3_est = model3.fit()
print(model3_est.summary())
```

```
##                                OLS Regression Results
## =====
## Dep. Variable:                growth    R-squared:                0.453
## Model:                      OLS        Adj. R-squared:         0.406
## Method:                     Least Squares    F-statistic:             9.613
## Date:                       Br, 06 дек 2022    Prob (F-statistic):      1.01e-06
## Time:                       18:05:12        Log-Likelihood:          -109.18
## No. Observations:           64            AIC:                    230.4
## Df Residuals:               58            BIC:                    243.3
## Df Model:                   5
## Covariance Type:            nonrobust
## =====
##                                coef      std err          t      P>|t|      [0.025      0.975]
## -----
## Intercept                   11.7459      2.920        4.023    0.000        5.901     17.591
## tradeshare                   1.1035      0.833        1.325    0.191       -0.564      2.771
## np.log(yearsschool)          2.1613      0.363        5.960    0.000        1.435      2.887
## rev_coups                   -2.2995      1.004       -2.289    0.026       -4.310     -0.289
## assassinations               0.2277      0.434        0.525    0.602       -0.640      1.096
## np.log(rgdp60)              -1.6211      0.399       -4.068    0.000       -2.419     -0.823
## =====
## Omnibus:                     3.780    Durbin-Watson:           1.966
## Prob(Omnibus):               0.151    Jarque-Bera (JB):         2.930
## Skew:                        0.379    Prob(JB):                 0.231
## Kurtosis:                   3.724    Cond. No.                 136.
## =====
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

R



```
model4 <- lm(growth ~ tradeshare*log(yearsschool) + rev_coups +
  ↳ assassinations + log(rgdp60), data=dataset)
summary(model4)
```

```
##
## Call:
## lm(formula = growth ~ tradeshare * log(yearsschool) + rev_coups +
##     assassinations + log(rgdp60), data = dataset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.0681 -0.9281 -0.0606  0.7652  4.1068
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    11.4985     2.9599   3.885 0.000269 ***
## tradeshare       1.8828     1.4753   1.276 0.207051
## log(yearsschool)  2.5247     0.6736   3.748 0.000418 ***
## rev_coups       -2.3502     1.0127  -2.321 0.023900 *
## assassinations   0.2242     0.4359   0.514 0.608998
## log(rgdp60)     -1.6414     0.4018  -4.085 0.000139 ***
## tradeshare:log(yearsschool) -0.6901     1.0756  -0.642 0.523706
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.407 on 57 degrees of freedom
## Multiple R-squared:  0.4571, Adjusted R-squared:  0.3999
## F-statistic: 7.998 on 6 and 57 DF, p-value: 2.803e-06
```





```
model4 = smf.ols("growth ~ tradeshare*np.log(yearsschool) + rev_coups +
↳ assassinations + np.log(rgdp60)", data=dataset)
model4_est = model4.fit()
print(model4_est.summary())
```

```
##                               OLS Regression Results
## =====
## Dep. Variable:                growth    R-squared:                0.457
## Model:                      OLS        Adj. R-squared:         0.400
## Method:                     Least Squares    F-statistic:             7.998
## Date:                       Br, 06 дек 2022    Prob (F-statistic):      2.80e-06
## Time:                       18:05:14    Log-Likelihood:          -108.95
## No. Observations:           64          AIC:                    231.9
## Df Residuals:               57          BIC:                    247.0
## Df Model:                   6
## Covariance Type:            nonrobust
## =====
##                               coef    std err          t      P>|t|      [0.025      0.975]
## -----
## Intercept                   11.4985     2.960      3.885     0.000     5.571    17.426
## tradeshare                   1.8828     1.475      1.276     0.207    -1.071     4.837
## np.log(yearsschool)          2.5247     0.674      3.748     0.000     1.176     3.874
## tradeshare*np.log(yearsschool) -0.6901     1.076     -0.642     0.524    -2.844     1.464
## rev_coups                   -2.3502     1.013     -2.321     0.024    -4.378    -0.322
## assassinations               0.2242     0.436     0.514     0.609    -0.649     1.097
## np.log(rgdp60)              -1.6414     0.402     -4.085     0.000    -2.446    -0.837
## =====
## Omnibus:                     4.112    Durbin-Watson:           1.979
## Prob(Omnibus):               0.128    Jarque-Bera (JB):         3.221
## Skew:                        0.423    Prob(JB):                 0.200
## Kurtosis:                    3.701    Cond. No.                  138.
## =====
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```



```
model5 <- lm(growth ~ tradeshare + I(tradeshare^2) + I(tradeshare^3) +
↳ log(yearsschool) + rev_coups + assassinations + log(rgdp60),
data=dataset)
summary(model5)
```

```
##
## Call:
## lm(formula = growth ~ tradeshare + I(tradeshare^2) + I(tradeshare^3) +
##   log(yearsschool) + rev_coups + assassinations + log(rgdp60),
##   data = dataset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.2438 -0.8381 -0.1331  0.6846  4.2607
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   12.9291     3.0985   4.173 0.000106 ***
## tradeshare     -5.7019     9.7551  -0.585 0.561226
## I(tradeshare^2)  8.4879    17.4350   0.487 0.628280
## I(tradeshare^3) -2.7597     9.2498  -0.298 0.766535
## log(yearsschool)  2.1332     0.3670   5.813 3.05e-07 ***
## rev_coups     -2.0355     1.0259  -1.984 0.052168 .
## assassinations  0.1021     0.4435   0.230 0.818747
## log(rgdp60)    -1.5843     0.4079  -3.884 0.000274 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.401 on 56 degrees of freedom
## Multiple R-squared:  0.4711, Adjusted R-squared:  0.405
## F-statistic: 7.126 on 7 and 56 DF,  p-value: 4.38e-06
```

## Python



```
model5 = smf.ols("growth ~ tradeshare + I(tradeshare ** 2) + I(tradeshare
  ↳ ** 3) + np.log(yearsschool) + rev_coups + assassinations +
  ↳ np.log(rgdp60)", data=dataset)
model5_est = model5.fit()
print(model5_est.summary())
```

```
##                      OLS Regression Results
## =====
## Dep. Variable:          growth      R-squared:                0.471
## Model:                  OLS        Adj. R-squared:            0.405
## Method:                 Least Squares    F-statistic:              7.126
## Date:                   Br, 06 дек 2022    Prob (F-statistic):       4.38e-06
## Time:                   18:05:17         Log-Likelihood:           -108.12
## No. Observations:       64             AIC:                    232.2
## Df Residuals:           56             BIC:                    249.5
## Df Model:               7
## Covariance Type:        nonrobust
## =====
##                      coef      std err          t      P>|t|      [0.025      0.975]
## -----
## Intercept               12.9291      3.098      4.173      0.000      6.722     19.136
## tradeshare              -5.7019      9.755     -0.585      0.561     -25.244     13.840
## I(tradeshare ** 2)       8.4879     17.435      0.487      0.628     -26.439     43.414
## I(tradeshare ** 3)      -2.7597      9.250     -0.298      0.767     -21.289     15.770
## np.log(yearsschool)      2.1332      0.367      5.813      0.000      1.398      2.868
## rev_coups               -2.0355      1.026     -1.984      0.052     -4.091      0.020
## assassinations           0.1021      0.444      0.230      0.819     -0.786      0.991
## np.log(rgdp60)          -1.5843      0.408     -3.884      0.000     -2.402     -0.767
## =====
## Omnibus:                7.017      Durbin-Watson:           2.030
## Prob(Omnibus):          0.030      Jarque-Bera (JB):        6.867
## Skew:                   0.544      Prob(JB):                0.0323
## Kurtosis:               4.179      Cond. No.:               994.
## =====
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

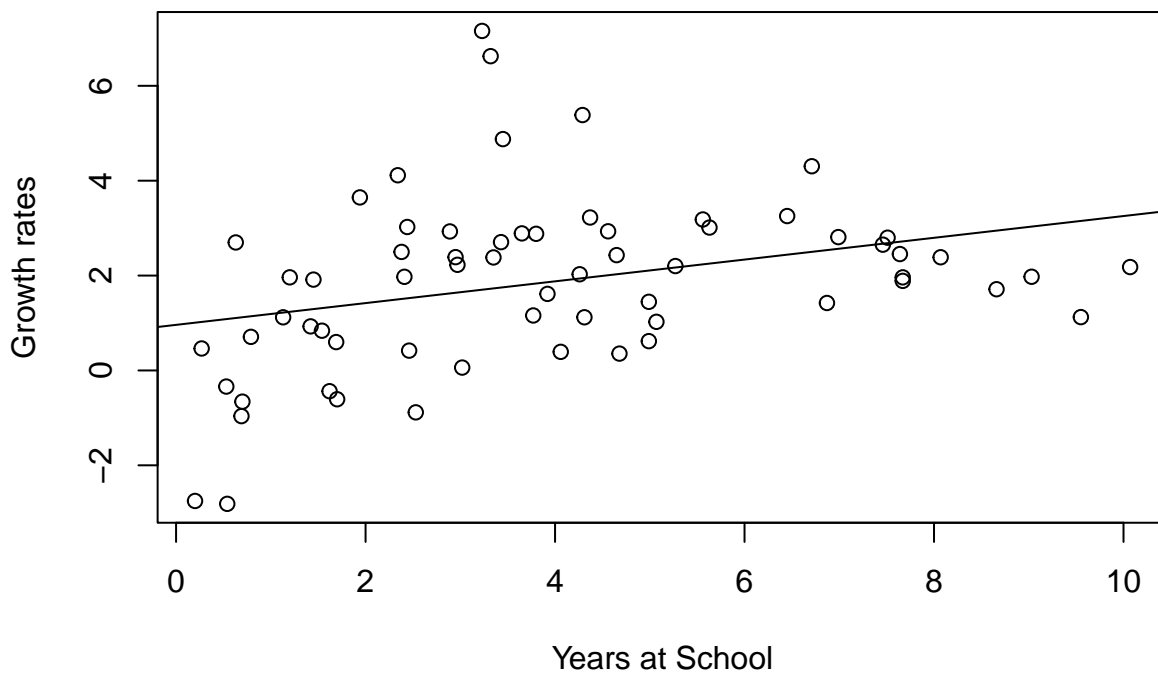
❓ Постройте диаграмму рассеяния переменных *Growth* и *YearsSchool*. Кажется ли полученное соотношение линейным или нет? Объясните. Используйте график для объяснения, почему регрессия (2) подходит для описания данных лучше, чем регрессия (1).

Построим диаграмму рассеяния *Growth* на *YearsSchool*. И добавим на нее линию тренда.

## R



```
plot(dataset$yearsschool, dataset$growth,
      xlab="Years at School", ylab="Growth rates")
abline(lm(growth ~ yearsschool, data = dataset))
```

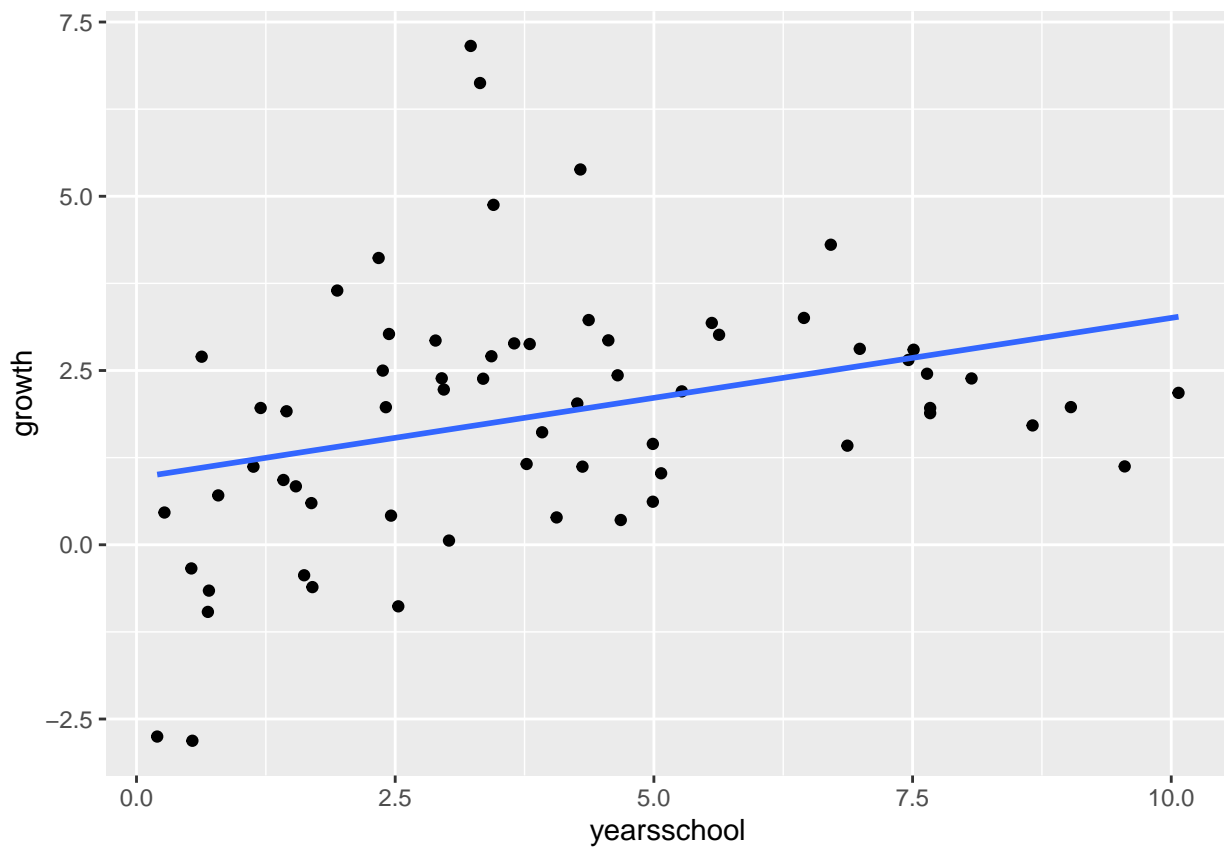


То же самое, но при помощи ggplot2.

R

```
ggplot(dataset, aes(x = yearsschool, y = growth)) +  
  geom_point() +  
  geom_smooth(method = lm, se = FALSE)
```



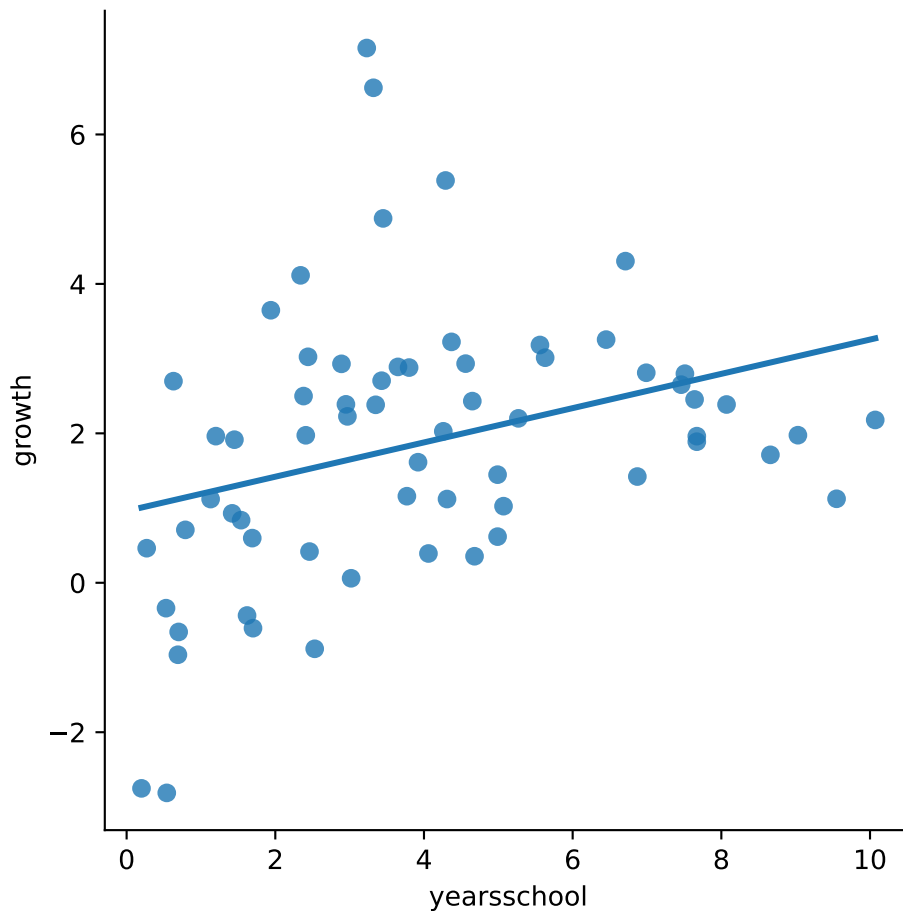


И в python.

Python



```
plt.clf()
sb.scatterplot(dataset, x="yearsschool", y="growth")
sb.lmplot(dataset, x="yearsschool", y="growth", ci=None)
```



Если график не был показан, то выполните следующую команду.

Python



```
plt.show()
```

❓ Проверьте, равны ли коэффициенты при *Assassinations* и *Rev\_Coups* нулю, используя регрессию (3).

R



```
car::linearHypothesis(model3, c("assassinations=0", "rev_coups=0"))
```

```
## Linear hypothesis test
##
## Hypothesis:
## assassinations = 0
## rev_coups = 0
##
## Model 1: restricted model
## Model 2: growth ~ tradeshare + log(yearsschool) + rev_coups + assassinations +
##          log(rgdp60)
##
##   Res.Df    RSS Df Sum of Sq    F Pr(>F)
## 1      60 124.72
## 2      58 113.64  2    11.081 2.828 0.06731 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Python



```
model3_est.f_test("assasinations=0, rev_coups=0")
```

```
## <class 'statsmodels.stats.contrast.ContrastResults'>
## <F test: F=2.827961081370163, p=0.06731102549280178, df_denom=58, df_num=2>
```

? Используя регрессию (4), скажите, существует ли свидетельство того, что эффект влияния *TradeShare* на *Growth* зависит от уровня образования в стране?

R



```
car::linearHypothesis(model4, c("tradeshare:log(yearsschool) = 0"))
```

```
## Linear hypothesis test
##
## Hypothesis:
## tradeshare:log(yearsschool) = 0
##
## Model 1: restricted model
## Model 2: growth ~ tradeshare * log(yearsschool) + rev_coups + assassinations +
##           log(rgdp60)
##
##      Res.Df    RSS Df Sum of Sq    F Pr(>F)
## 1       58 113.64
## 2       57 112.82  1   0.81478 0.4116 0.5237
```

Python



```
model4_est.f_test("tradeshare:np.log(yearsschool) = 0")
```

```
## <class 'statsmodels.stats.contrast.ContrastResults'>
## <F test: F=0.41164801932892225, p=0.5237059925456938, df_denom=57, df_num=1>
```

? Используя регрессию (5), скажите, существует ли свидетельство нелинейности эффекта влияния *TradeShare* на *Growth*?

R



```
car::linearHypothesis(model5, c("I(tradeshare^2) = 0", "I(tradeshare^3) =
  0"))
```

```
## Linear hypothesis test
##
## Hypothesis:
## I(tradeshare^2) = 0
## I(tradeshare^3) = 0
##
## Model 1: restricted model
## Model 2: growth ~ tradeshare + I(tradeshare^2) + I(tradeshare^3) + log(yearsschool) +
##           rev_coups + assassinations + log(rgdp60)
##
##      Res.Df    RSS Df Sum of Sq    F Pr(>F)
## 1       58 113.64
## 2       56 109.91  2   3.7251 0.949 0.3933
```



```
model5_est.f_test("I(tradeshare ** 2) = 0, I(tradeshare ** 3) = 0")
```

```
## <class 'statsmodels.stats.contrast.ContrastResults'>  
## <F test: F=0.948996151151372, p=0.3932662154476029, df_denom=56, df_num=2>
```

# Список литературы

- Chambers, J., T. Hastie, and D. Pregibon. 1990. "Statistical Models in S." In *Compstat*, edited by Konstantin Momirović and Vesna Mildner, 317–21. Heidelberg: Physica-Verlag HD. [https://doi.org/10.1007/978-3-642-50096-1\\_48](https://doi.org/10.1007/978-3-642-50096-1_48).
- Mittelhammer, Ron C., George G. Judge, and Douglas J. Miller. 2000. *Econometric Foundations*. Cambridge: Cambridge University Press.
- Wilkinson, G. N., and C. E. Rogers. 1973. "Symbolic Description of Factorial Models for Analysis of Variance." *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 22 (3): 392–99. <https://doi.org/10.2307/2346786>.



# Предметный указатель

- Автокорреляция, 19
  - тест Бройша-Годфри, 19
  - тест Дарбина-Уотсона, 19
- Гетероскедастичность, 9
  - взвешенный МНК, 21
  - графики, 9
  - тест Бройша-Пагана, 17
  - тест Глейзера, 18
  - тест Голдфельдта-Квандта, 16
  - тест Уайта, 17
- Гетероскедстичность
  - ошибки Уайта, 26, 27
- Значимость
  - F-тест, 25, 51, 60
  - t-тест, 24, 45, 46
  - критерий Вальда, 25
- Мультиколлинеарность
  - VIF, 44
- Регрессия
  - компоненты взаимодействия, 34
  - логарифмы, 27
  - множественная, 23, 43, 50
  - парная, 8, 42
  - полиномы, 28
- Тест Харке-Бера, 20
- Тест Чоу, 21
- Формулы, 4
  - включение, 5
  - двоеточие, 5
  - звездочка, 5
  - минус, 5
  - плюс, 5
  - скобки, 5
  - степень, 5