Домашнее задание по многофакторному анализу №1 (повторная сдача)

Liliya Golubnikova

2023-01-30

## Загрузка библиотек

#install.packages("faux")  
#install.packages("olsrr")  
library(MASS)  
library(faux)

## Warning: пакет 'faux' был собран под R версии 4.2.2

library(ggplot2)

## Warning: пакет 'ggplot2' был собран под R версии 4.2.2

library(dplyr)

## Warning: пакет 'dplyr' был собран под R версии 4.2.2

library(tidyr)

## Warning: пакет 'tidyr' был собран под R версии 4.2.2

library(faux)  
library(tidyverse)

## Warning: пакет 'tidyverse' был собран под R версии 4.2.2

## Warning: пакет 'tibble' был собран под R версии 4.2.2

## Warning: пакет 'readr' был собран под R версии 4.2.2

## Warning: пакет 'purrr' был собран под R версии 4.2.2

## Warning: пакет 'stringr' был собран под R версии 4.2.2

## Warning: пакет 'forcats' был собран под R версии 4.2.2

library(ggpubr)

## Warning: пакет 'ggpubr' был собран под R версии 4.2.2

library("olsrr")

## Warning: пакет 'olsrr' был собран под R версии 4.2.2

## Задача 1

*Размер выборки*: s = 100  
*Показатели:*  
- креатинин k  
- мочевина m  
- признак W

*Выборочные характеристики:*  
Средние значения:  
- median(k) = 88.5  
- median(m) = 5.4

Стандартные отклонения:  
- sd(k) = 13.25  
- sd(m) = 1.45

Коэффициенты корреляции: - cor(k,m) = 0.6  
- cor(k,W) = 0.0  
- cor(m,W) = 0.0

### Моделируем выборку

# Сделаем df с переменными: k, W, m  
set.seed(1)  
dat2 <- rnorm\_multi(n = 100,   
 mu = c(88.5, 5.4, 5),  
 sd = c(13.25, 1.45, 2),  
 r = c(0.6, 0, 0),   
 varnames = c("k", "m", "W"),  
 empirical = FALSE)  
  
dat2

## k m W  
## 1 80.16834 5.323647 3.759267  
## 2 90.80417 7.511582 5.084232  
## 3 77.30686 6.499876 3.178157  
## 4 109.66241 6.416358 5.316058  
## 5 93.04054 3.049509 3.690831  
## 6 77.43811 7.565422 8.534575  
## 7 94.90735 6.597754 6.433415  
## 8 98.24127 6.672484 6.820348  
## 9 96.13000 5.889298 5.768371  
## 10 84.41469 5.721430 8.364352  
## 11 108.54333 6.535494 3.728527  
## 12 93.63319 6.227653 4.076711  
## 13 80.29929 4.393103 7.864564  
## 14 59.26042 1.875887 3.698607  
## 15 103.32960 7.527116 4.585239  
## 16 87.78850 7.115857 4.214384  
## 17 88.30908 5.029222 4.360014  
## 18 101.10140 4.779935 4.441773  
## 19 99.33192 6.861710 5.988377  
## 20 96.37248 5.869021 4.645339  
## 21 100.80870 4.203842 3.988085  
## 22 98.86297 6.088142 7.686078  
## 23 89.53614 4.737396 4.570841  
## 24 62.16758 3.262177 4.640887  
## 25 96.80094 4.607588 4.799619  
## 26 87.61851 7.433356 6.425333  
## 27 86.46105 4.880973 4.852871  
## 28 69.13555 2.256318 4.924732  
## 29 82.14955 5.208569 3.636679  
## 30 94.01752 6.070357 4.351459  
## 31 106.57754 5.452528 5.120321  
## 32 87.35885 1.973292 3.822211  
## 33 93.68551 5.000130 6.062992  
## 34 87.74350 6.011746 1.963212  
## 35 70.25883 4.123787 5.613116  
## 36 83.00892 4.922792 1.927100  
## 37 83.23289 5.702063 4.398048  
## 38 87.80478 3.977686 3.943440  
## 39 102.99128 7.631103 3.695810  
## 40 98.61232 6.062886 4.886206  
## 41 86.26605 6.072684 1.171281  
## 42 85.06399 6.372240 7.353167  
## 43 97.71753 6.269123 1.670055  
## 44 95.94281 4.873138 4.072939  
## 45 79.28527 6.139366 2.768160  
## 46 79.27869 2.469640 3.498362  
## 47 93.37226 5.090407 9.174333  
## 48 98.70243 5.778464 5.034791  
## 49 87.02413 5.109645 2.427399  
## 50 100.09650 7.351044 1.718789  
## 51 93.76441 5.906341 5.900374  
## 52 80.35967 5.333715 4.962880  
## 53 93.02508 5.618603 4.363863  
## 54 73.55511 4.123872 3.141276  
## 55 107.43410 7.459633 2.025079  
## 56 114.65227 8.460020 2.849615  
## 57 83.81804 2.302579 7.000058  
## 58 74.62168 5.146126 3.757467  
## 59 96.02002 6.332252 2.231146  
## 60 86.74305 4.790432 8.738581  
## 61 120.24823 8.603825 5.850201  
## 62 88.00982 4.916045 4.522706  
## 63 97.66062 5.676286 7.116966  
## 64 88.80550 6.414825 6.772845  
## 65 78.52038 6.734496 3.761514  
## 66 90.98082 5.877413 9.412205  
## 67 64.61696 3.330030 4.489946  
## 68 108.00915 5.311435 2.151011  
## 69 90.55587 5.152023 4.711201  
## 70 117.35845 6.219208 5.415077  
## 71 94.82018 5.517810 9.615957  
## 72 79.06323 5.233101 5.211605  
## 73 96.65709 4.951546 5.913998  
## 74 75.92097 7.640789 4.845694  
## 75 71.87771 4.481158 4.331998  
## 76 92.27523 6.960851 4.930548  
## 77 82.80141 2.367523 6.575279  
## 78 88.45802 6.256798 9.150490  
## 79 89.58559 3.944959 7.054785  
## 80 80.61869 5.945521 7.415817  
## 81 80.93484 5.361288 2.537353  
## 82 86.74006 4.810812 6.967791  
## 83 104.00820 7.962270 5.439850  
## 84 68.36666 3.254429 2.065500  
## 85 96.41403 5.250110 6.042045  
## 86 92.98802 4.535688 4.682491  
## 87 102.63690 5.560272 7.929175  
## 88 84.39740 6.224983 3.467836  
## 89 93.36952 6.225296 4.139576  
## 90 91.96219 6.795081 3.147781  
## 91 81.34154 4.473813 4.645792  
## 92 104.47522 6.893602 5.804024  
## 93 103.85642 6.699298 3.536504  
## 94 97.88667 4.366586 6.660746  
## 95 109.38929 8.845159 2.583834  
## 96 95.88955 6.044894 2.904031  
## 97 71.52693 5.165081 7.882315  
## 98 80.83137 6.000580 2.968305  
## 99 72.27802 4.268010 5.823949  
## 100 82.25092 4.631776 4.237848

#Для первой части эксперимента возьмем переменные k, m из нашей модели:   
  
dat <- subset(dat2, select = c(k,m))  
dat

## k m  
## 1 80.16834 5.323647  
## 2 90.80417 7.511582  
## 3 77.30686 6.499876  
## 4 109.66241 6.416358  
## 5 93.04054 3.049509  
## 6 77.43811 7.565422  
## 7 94.90735 6.597754  
## 8 98.24127 6.672484  
## 9 96.13000 5.889298  
## 10 84.41469 5.721430  
## 11 108.54333 6.535494  
## 12 93.63319 6.227653  
## 13 80.29929 4.393103  
## 14 59.26042 1.875887  
## 15 103.32960 7.527116  
## 16 87.78850 7.115857  
## 17 88.30908 5.029222  
## 18 101.10140 4.779935  
## 19 99.33192 6.861710  
## 20 96.37248 5.869021  
## 21 100.80870 4.203842  
## 22 98.86297 6.088142  
## 23 89.53614 4.737396  
## 24 62.16758 3.262177  
## 25 96.80094 4.607588  
## 26 87.61851 7.433356  
## 27 86.46105 4.880973  
## 28 69.13555 2.256318  
## 29 82.14955 5.208569  
## 30 94.01752 6.070357  
## 31 106.57754 5.452528  
## 32 87.35885 1.973292  
## 33 93.68551 5.000130  
## 34 87.74350 6.011746  
## 35 70.25883 4.123787  
## 36 83.00892 4.922792  
## 37 83.23289 5.702063  
## 38 87.80478 3.977686  
## 39 102.99128 7.631103  
## 40 98.61232 6.062886  
## 41 86.26605 6.072684  
## 42 85.06399 6.372240  
## 43 97.71753 6.269123  
## 44 95.94281 4.873138  
## 45 79.28527 6.139366  
## 46 79.27869 2.469640  
## 47 93.37226 5.090407  
## 48 98.70243 5.778464  
## 49 87.02413 5.109645  
## 50 100.09650 7.351044  
## 51 93.76441 5.906341  
## 52 80.35967 5.333715  
## 53 93.02508 5.618603  
## 54 73.55511 4.123872  
## 55 107.43410 7.459633  
## 56 114.65227 8.460020  
## 57 83.81804 2.302579  
## 58 74.62168 5.146126  
## 59 96.02002 6.332252  
## 60 86.74305 4.790432  
## 61 120.24823 8.603825  
## 62 88.00982 4.916045  
## 63 97.66062 5.676286  
## 64 88.80550 6.414825  
## 65 78.52038 6.734496  
## 66 90.98082 5.877413  
## 67 64.61696 3.330030  
## 68 108.00915 5.311435  
## 69 90.55587 5.152023  
## 70 117.35845 6.219208  
## 71 94.82018 5.517810  
## 72 79.06323 5.233101  
## 73 96.65709 4.951546  
## 74 75.92097 7.640789  
## 75 71.87771 4.481158  
## 76 92.27523 6.960851  
## 77 82.80141 2.367523  
## 78 88.45802 6.256798  
## 79 89.58559 3.944959  
## 80 80.61869 5.945521  
## 81 80.93484 5.361288  
## 82 86.74006 4.810812  
## 83 104.00820 7.962270  
## 84 68.36666 3.254429  
## 85 96.41403 5.250110  
## 86 92.98802 4.535688  
## 87 102.63690 5.560272  
## 88 84.39740 6.224983  
## 89 93.36952 6.225296  
## 90 91.96219 6.795081  
## 91 81.34154 4.473813  
## 92 104.47522 6.893602  
## 93 103.85642 6.699298  
## 94 97.88667 4.366586  
## 95 109.38929 8.845159  
## 96 95.88955 6.044894  
## 97 71.52693 5.165081  
## 98 80.83137 6.000580  
## 99 72.27802 4.268010  
## 100 82.25092 4.631776

### Оценим получившуюся модель

summary(dat)

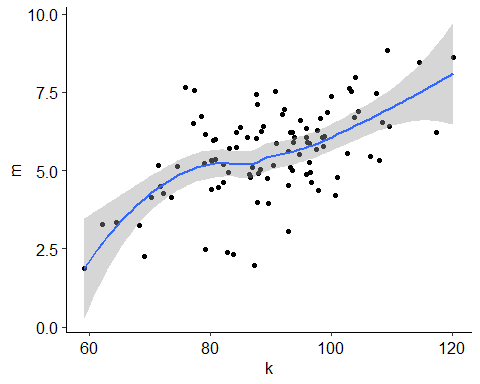
## k m   
## Min. : 59.26 Min. :1.876   
## 1st Qu.: 81.95 1st Qu.:4.788   
## Median : 90.07 Median :5.647   
## Mean : 89.94 Mean :5.530   
## 3rd Qu.: 97.67 3rd Qu.:6.383   
## Max. :120.25 Max. :8.845

cor(dat)

## k m  
## k 1.000000 0.555491  
## m 0.555491 1.000000

### Визуализация исходных данных

#head(dat, 4)  
  
theme\_set(theme\_pubr())  
ggplot(dat, aes(x = k, y = m)) +  
 geom\_point() +  
 stat\_smooth()

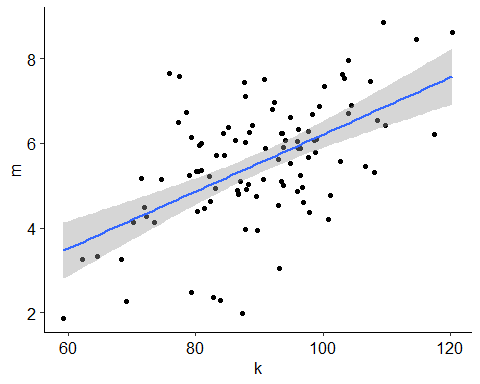


### Построим модель линейной регрессии креатинина k на мочевину m

# Линейная модель  
model <- lm(m ~ k, data = dat)  
summary(model)

##   
## Call:  
## lm(formula = m ~ k, data = dat)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.3826 -0.5606 -0.0467 0.7614 3.0553   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.52769 0.92385 -0.571 0.569   
## k 0.06735 0.01018 6.613 1.99e-09 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.206 on 98 degrees of freedom  
## Multiple R-squared: 0.3086, Adjusted R-squared: 0.3015   
## F-statistic: 43.74 on 1 and 98 DF, p-value: 1.99e-09

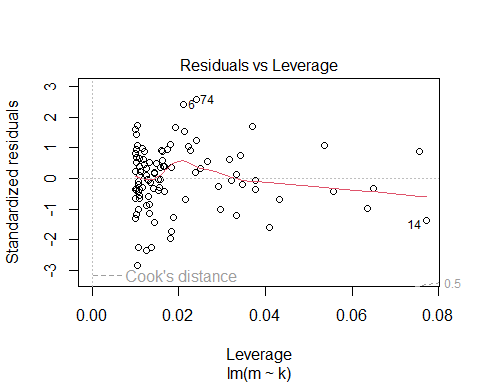
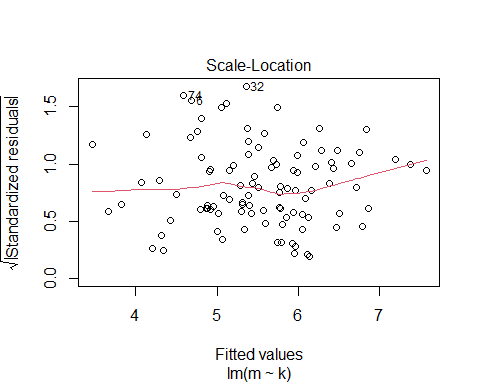
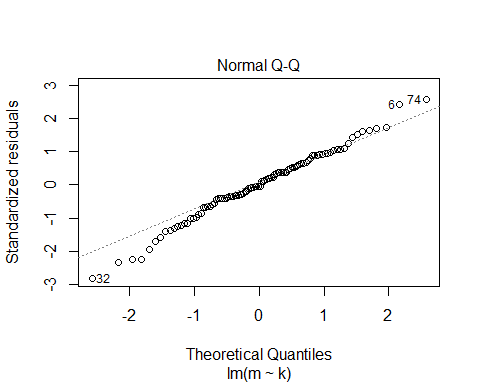
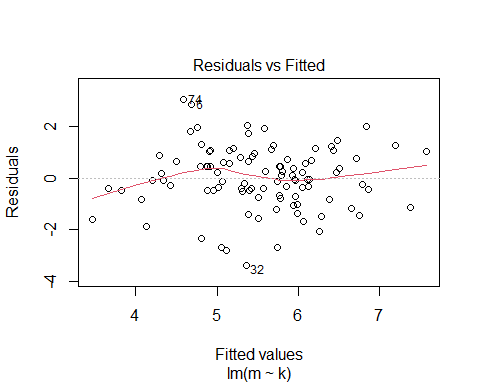
ggplot(dat, aes(k, m)) +  
 geom\_point() +  
 stat\_smooth(method = lm)



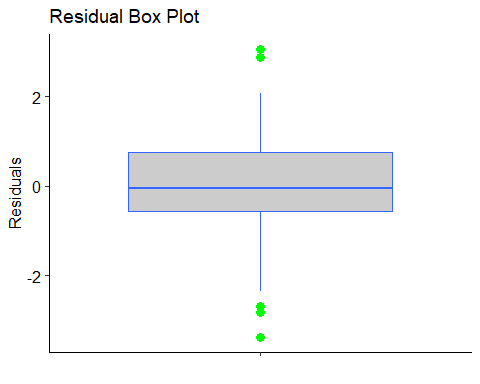
*Анализ результата:*  
1) Уровень значимости статистической модели: уровень p-value = 1.99e-09, что меньше 0.05 => модель статистически значима.  
2) R^2 = 0.3086 => 30,8% вариации переменной m можно объяснить переменной-предиктором k.  
3) Оценка коэффициента k = 0.06735. Каждое дополнительное увеличение k на 1 единицу связано со средним увеличением m на 0.06735.  
4) Cоласно значению p-value переменная k является статистически значимой (p-value = 1.99e-09).  
**Оценочное уравнение регрессии:** m = -0.52769 + 0.06735\*(k).

### Построим диагностические графики по полученной модели линейной регрессии:

plot(model)



ols\_plot\_resid\_box(model)



1. Для определения того, имеют ли остатки нелинейные закономерности оценим *график Residuals vs Fitted*: красная линия отклоняется от идеальной горизонтальной линии, но не сильно. Можем сделать вывод, что остатки следуют примерно линейному образцу и что для этого набора данных подходит модель линейной регрессии.
2. Для проверки нормальности распределения остатков регрессионной модели оценим *график Normal Q-Q*: большинство точек попадает на диагональную линию, однако, наблюдаются некоторые отклонения. Таким образом, согласно графику нельзя считать, что распределение остатков нормальное.
3. Для проверки предположения о ровной дисперсии (“гомоскедастичности”) остатков в регрессионной модели оценим *график масштаба-предположения (Scale-Location)*: красная линия проходит не совсем горизонтально на графике, но она не отклоняется слишком сильно ни в одной точке => предположение о равной дисперсии не нарушается.
4. Для оценки наличия влиятельных наблюдений оценим *график остатков кредитного плеча (Residuals vs Leverage)*: некоторые наблюдения находится близко к границе расстояния Кука, но не выходит за пределы пугктирной линии => в нашем наборе нет чрезмерно влиятельных точек.
5. Оценим нормальность распределения с помощью *графика Residual box-plot*: остатки находятся вокруг нуля, усы почти одинакового размера, есть выбросы (отмечены зеленым), но незначительные. Делаем вывод, что распределение нормальное.

### Проверка гипотезы о нормальном распределении остатков с помощью статистических тестов

**Гипотеза**  
**H0:** Распределение нормальное.  
**H1:** Распределение не нормальное.

ols\_test\_normality(model)

## -----------------------------------------------  
## Test Statistic pvalue   
## -----------------------------------------------  
## Shapiro-Wilk 0.9887 0.5649   
## Kolmogorov-Smirnov 0.0811 0.5256   
## Cramer-von Mises 6.1179 0.0000   
## Anderson-Darling 0.376 0.4059   
## -----------------------------------------------

*Примечание: тип используемого теста зависит от количества наблюдений. Так как у нас более 50 наблюдений, воспользуемся результатами Колмогорова-Смирнова (Kolmogorov-Smirnov).*

*Результат:*  
Уровень значимости p-value = 0.5256, что больше 0.05 => *H0 принимается* => Распределение нормальное.

**Общий вывод:** на основе выполненных тестов можно считать, что распределение остатков нормальное, модель гомоскедастична, для данного набора данных подходит модель линейной регрессии.

#### Как полученные оценки связаны с выборочными характеристиками (средним, дисперсией, корреляцией) выборки S?

1. Коэффициент детерминации (R-квадрат) — это доля дисперсии зависимой переменной, объясняемая рассматриваемой моделью зависимости, то есть объясняющими переменными. В частности, для модели парной линейной регрессии коэффициент детерминации равен квадрату обычного коэффициента корреляции между y и x. В нашем случае R^2 = 0.3016, а cor(k,m) = 0.6.
2. Наилучшая прямая линейной регрессии проходит через точку (; ) и имеет угловой коэффициент и свободный член, соответственной:

= ( \* ) / , где — коэффициент корреляции

= - \*

#### Оценим исходную полную модель с признаками W, k, m

summary(dat2)

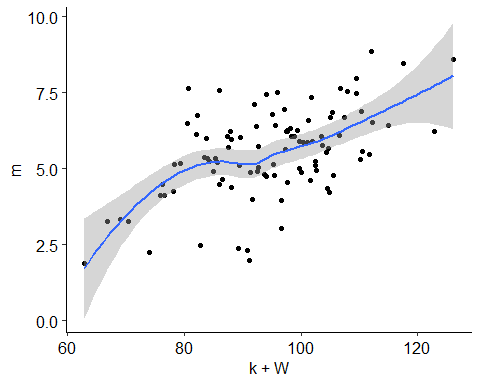
## k m W   
## Min. : 59.26 Min. :1.876 Min. :1.171   
## 1st Qu.: 81.95 1st Qu.:4.788 1st Qu.:3.698   
## Median : 90.07 Median :5.647 Median :4.646   
## Mean : 89.94 Mean :5.530 Mean :4.924   
## 3rd Qu.: 97.67 3rd Qu.:6.383 3rd Qu.:6.002   
## Max. :120.25 Max. :8.845 Max. :9.616

cor(dat2)

## k m W  
## k 1.0000000000 0.55549098 -0.0006654046  
## m 0.5554909819 1.00000000 -0.0415564479  
## W -0.0006654046 -0.04155645 1.0000000000

### Визуализация исходных данных

#head(dat, 4)  
  
theme\_set(theme\_pubr())  
ggplot(dat2, aes(x = k + W, y = m)) +  
 geom\_point() +  
 stat\_smooth()

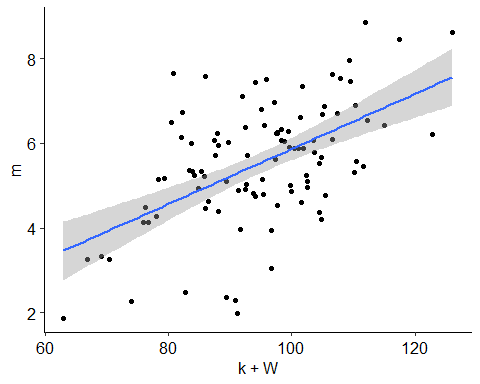


### Построим модель линейной регрессии креатинина и признака (W): k + W на мочевину m

# Линейная модель  
model <- lm(m ~ k + W, data = dat2)  
summary(model)

##   
## Call:  
## lm(formula = m ~ k + W, data = dat2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.4168 -0.5155 0.0060 0.7445 3.0528   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.37464 0.97896 -0.383 0.703   
## k 0.06735 0.01022 6.587 2.32e-09 \*\*\*  
## W -0.03102 0.06350 -0.488 0.626   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.211 on 97 degrees of freedom  
## Multiple R-squared: 0.3103, Adjusted R-squared: 0.296   
## F-statistic: 21.82 on 2 and 97 DF, p-value: 1.5e-08

ggplot(dat2, aes(k + W, m)) +  
 geom\_point() +  
 stat\_smooth(method = lm)



*Анализ результата:*  
1) Уровень значимости статистической модели: уровень p-value = 1.5e-08, что меньше 0.05 => модель статистически значима. Вывод не изменился по сравнению с ранее построенной моделью.  
2) R^2 = 0.3103 => 31% вариации переменной m можно объяснить переменными-предикторами k, W. В ранее построенной модели (m = -0.52769 + 0.06735\*(k)) R^2 ниже (30,8%). Модифицированный коэффициент детерминации в первой модели Adjusted R^2 = 0.3015, в новой модели Adjusted R^2 = 0.296 -> снизился.  
3) Важно отметить, что переменная k является статистически значимой (p-value = 2.32e-09), а переменная W (p-value = 0.626) - нет.

Источники:  
[1] <https://handbook.mathpsy.com/?page_id=238>  
[2] <https://www.codingprof.com/5-ways-to-check-the-normality-of-residuals-in-r-examples/>  
[3] <https://milnepublishing.geneseo.edu/natural-resources-biometrics/chapter/chapter-7-correlation-and-simple-linear-regression/>  
[4] <https://www.codecamp.ru/blog/lm-function-in-r/>