

Санкт-Петербургский государственный университет  
Прикладная математика, программирование и искусственный  
интеллект

Отчет по учебной практике 1 (научно-исследовательской работе)  
(семестр 2)

Методы первичной обработки статистических данных

Выполнил:

Гуноев Адам Асланбекович, группа  
22.Б06-мм



Научный руководитель:

Доктор физ.-мат. наук, профессор

Ермаков Михаил Сергеевич.

Кафедра статистического моделирования

*Работа выполнена добросовесно,  
в полном объеме и на  
высоком уровне. Заслужи-  
вает оценки "А" (отлично)*

*MS 127.05.2023.*

Санкт-Петербург

2023

# **Оглавление**

Введение .....	1
Основная часть .....	2
Выбор датасета.....	2
Написание скрипта .....	2
Заключение.....	10
Список литературы .....	11
Приложение.....	12
Код скрипта R .....	12

# **Введение**

Была поставлена задача изучения методов первичной обработки статистических данных на выборочном датасете, прохождения курса по специальному языку программирования для статистического анализа R, а также построения простой линейной регрессии.

# Основная часть

Вся работа была сделана в IDE RStudio. Теория для её выполнения была освоена с помощью книги "Statistics for Business and Economics"<sup>[1]</sup>. Скрипт дан в приложении.

## Выбор датасета

Для работы был выбран датасет "Happiness and Corruption 2015-2020" с сайта "kaggle.com". Он содержит данные 2015-2020 годов 132 стран по таким показателям как "happiness\_score" (счастье), "gdp\_per\_capita" (ВВП на душу населения), "family" (семья), "health" (здоровье), "freedom" (свобода), "generosity" (щедрость), "government\_trust" (доверие государству), "dystopia\_residual" (остаток «Антиутопии»), "continent" (континент), "year" (год), "social\_support" (социальная поддержка), "sri\_score" (уровень восприятия коррупции) и идеально подходит для задачи. Определения этих параметров можно найти на официальном ресурсе<sup>[2]</sup>. Поставим целью изучение зависимости уровня счастья от других показателей датафрейма.

## Написание скрипта

- С помощью функции `library()` подключаем библиотеку `ggplot2`, которая позволит удобно визуализировать данные:

```
1          # Откроем необходимые библиотеки #
2 library(ggplot2)
3
```

- Используя `read.csv()` открываем датасет и сохраняем в переменную `data`. Он автоматически будет представлен как датафрейм:

```
5          # Откроем датасет #
6 data = read.csv("dataset/data.csv")
7
```

Country	happiness_score	gdp_per_capita	family	health	freedom	generosity	government_trust	dystopia_residual
1 Norway	7.537	1.61646318	1.5335236	0.796666503	0.63542259	0.36201224	0.315963835	2.2770267
2 Denmark	7.522	1.48238301	1.5511216	0.792565525	0.62600672	0.35528049	0.400770068	2.3137074
3 Iceland	7.504	1.48063302	1.6105740	0.833552122	0.62716264	0.47554022	0.153526559	2.3227153
4 Switzerland	7.494	1.56497955	1.5169117	0.858131289	0.62007058	0.29054928	0.367007285	2.2767162
5 Finland	7.469	1.44357193	1.5402467	0.809157670	0.61795086	0.24548277	0.382611543	2.4301815
6 Netherlands	7.377	1.50394464	1.4289392	0.810696125	0.58538449	0.47048983	0.282661825	2.2948041
7 Canada	7.316	1.47920442	1.4813490	0.834557652	0.61110091	0.43553972	0.287371516	2.1872644
8 New Zealand	7.314	1.40570605	1.5481951	0.816759706	0.61406213	0.50000513	0.382816702	2.0464563
9 Sweden	7.284	1.49438727	1.4781622	0.830875158	0.61292410	0.38539925	0.384398729	2.0975380
10 Australia	7.284	1.48441493	1.5100420	0.843886793	0.60160738	0.47769925	0.301183730	2.0652108

3. Функция `str()` позволяет рассмотреть типы данных, представленные в датафрейме:

```
'data.frame': 792 obs. of 13 variables:
 $ Country      : chr "Norway" "Denmark" "Iceland" "Switzerland" ...
 $ happiness_score : num 7.54 7.52 7.5 7.49 7.47 ...
 $ gdp_per_capita : num 1.62 1.48 1.48 1.56 1.44 ...
 $ family        : num 1.53 1.55 1.61 1.52 1.54 ...
 $ health         : num 0.797 0.793 0.834 0.858 0.809 ...
 $ freedom        : num 0.635 0.626 0.627 0.62 0.618 ...
 $ generosity     : num 0.362 0.355 0.476 0.291 0.245 ...
 $ government_trust: num 0.316 0.401 0.154 0.367 0.383 ...
 $ dystopia_residual: num 2.28 2.31 2.32 2.28 2.43 ...
 $ continent      : chr "Europe" "Europe" "Europe" "Europe" ...
 $ Year           : int 2015 2015 2015 2015 2015 2015 2015 2015 ...
 $ social_support  : num 0 0 0 0 0 0 0 0 0 ...
 $ cpi_score       : int 88 91 79 86 90 84 83 91 89 79 ...
```

4. Используя `sum(is.na())` проверим наличие пропусков в значениях датафрейма и, если они есть, найдём их количество:

```
14 # Проверим датасет на наличие пропусков #
15 sum(is.na(data))
16
```

Вывод показал, что пропусков нет.

5. Функция `summary()` позволяет рассмотреть краткую информацию по числовым значениям столбцов датафрейма, а именно минимальном, максимальном, среднем и медианном; а также 1-ый и 3-ий квартили:

```
Country      happiness_score gdp_per_capita      family      health      freedom
Length:792    Min.   :2.567   Min.   :0.0000   Min.   :0.000   Min.   :0.0000   Min.   :0.0000
Class :character 1st Qu.:4.591   1st Qu.:0.6442   1st Qu.:0.000   1st Qu.:0.5101   1st Qu.:0.3254
Mode  :character Median :5.486   Median :0.9945   Median :0.000   Median :0.6854   Median :0.4396
                  Mean   :5.473   Mean   :0.9292   Mean   :0.505   Mean   :0.6487   Mean   :0.4270
                  3rd Qu.:6.301   3rd Qu.:1.2287   3rd Qu.:1.040   3rd Qu.:0.8156   3rd Qu.:0.5463
                  Max.   :7.809   Max.   :2.0960   Max.   :1.611   Max.   :1.1410   Max.   :0.7240
generosity    government_trust dystopia_residual continent      Year      social_support
Min.   :0.0000   Min.   :0.00000   Min.   :0.000   Length:792      Min.   :2015   Min.   :0.0000
1st Qu.:0.1258  1st Qu.:0.05286  1st Qu.:0.000   Class :character  1st Qu.:2016  1st Qu.:0.0000
Median :0.1970  Median :0.08900  Median :1.732   Mode  :character  Median :2018   Median :0.1762
Mean   :0.2124  Mean   :0.12572  Mean   :1.379   NA's   :1        Mean   :2018   Mean   :0.6093
3rd Qu.:0.2732  3rd Qu.:0.15425  3rd Qu.:2.237   NA's   :1        3rd Qu.:2019  3rd Qu.:1.2683
Max.   :0.8381  Max.   :0.55191  Max.   :3.602   NA's   :1        Max.   :2020   Max.   :1.6440
cpi_score
Min.   :11.00
1st Qu.:30.00
Median :38.00
Mean   :44.33
3rd Qu.:57.00
Max.   :91.00
```

6. Наша задача - изучить зависимость уровня счастья от остальных показателей, соответственно, мы можем убрать из датафрейма показатели "континент", "год" и "страна", так как они очевидно не влияют на него:

```
25 data$continent = NULL  
26 data$Year      = NULL  
27 data$Country   = NULL
```

7. Нормализуем данные, чтобы все значения находились в одном диапазоне - [0, 1], а затем округлим до 2 знаков после запятой для более удобного восприятия:

```
32 data$happiness_score    = data$happiness_score / 10  
33 data$cpi_score          = data$cpi_score           / 100  
34 data$gdp_per_capita     = data$gdp_per_capita    / 10  
35 data$family              = data$family             / 10  
36 data$social_support     = data$social_support    / 10  
37 data$dystopia_residual = data$dystopia_residual / 10  
38 data = round(data, digits=2)
```

8. С помощью `cor()` рассмотрим корреляции между уровнем счастья и другими показателями:

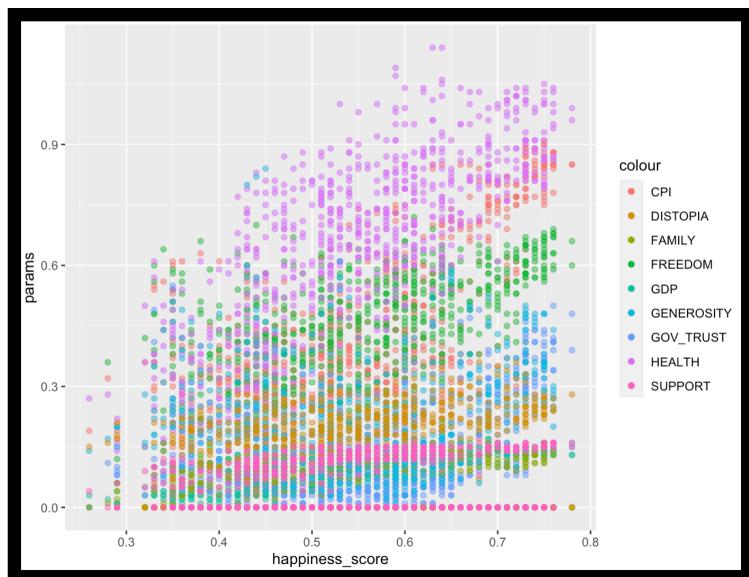
```
47 datacor = round(cor(data, data$happiness_score), digits = 2)  
48 colnames(datacor) = c("happiness_score")  
49 datacor
```

	happiness_score
happiness_score	1.00
gdp_per_capita	0.79
family	0.15
health	0.75
freedom	0.54
generosity	0.15
government_trust	0.45
dystopia_residual	0.17
social_support	0.19
cpi_score	0.69

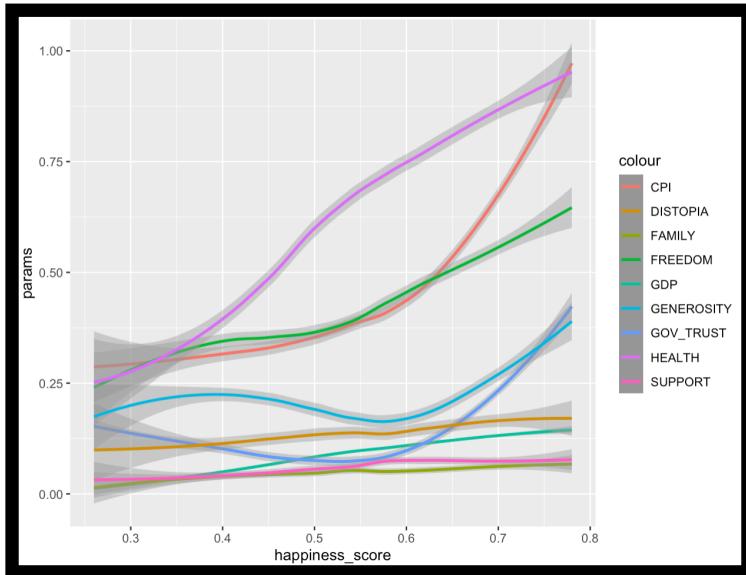
Видим, что больше всего на уровень счастья влияют "gdp\_per\_capita", "health", "cpi\_score" и "freedom". Однако этого недостаточно для точного вывода.

9. Теперь используя `ggplot()` и `geom_point()` построим точечные графики зависимости "happiness\_score" от каждого другого показателя:

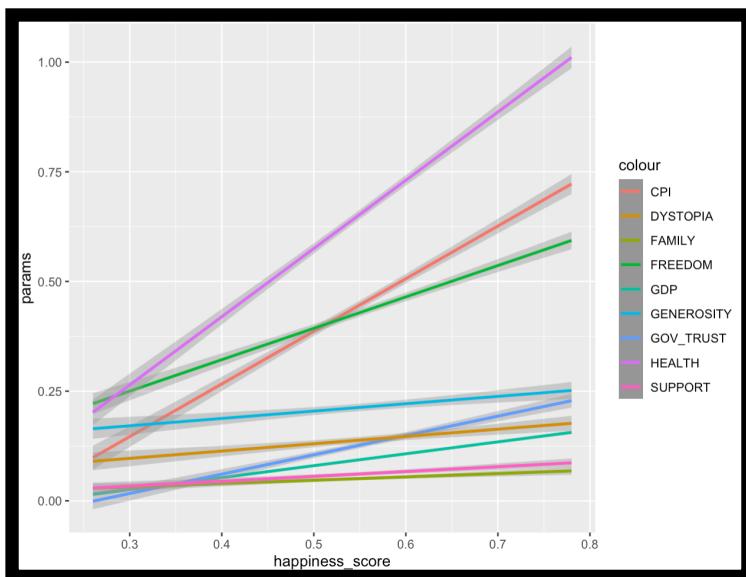
```
53 ggplot(data, aes(happiness_score, params))+  
54   geom_point(aes(happiness_score, cpi_score, col = "CPI", alpha = 0.5))+  
55   geom_point(aes(happiness_score, gdp_per_capita, col = "GDP", alpha = 0.5))+  
56   geom_point(aes(happiness_score, freedom, col = "FREEDOM", alpha = 0.5))+  
57   geom_point(aes(happiness_score, health, col = "HEALTH", alpha = 0.5))+  
58   geom_point(aes(happiness_score, government_trust, col = "GOV_TRUST", alpha = 0.5))+  
59   geom_point(aes(happiness_score, family, col = "FAMILY", alpha = 0.5))+  
60   geom_point(aes(happiness_score, generosity, col = "GENEROSITY", alpha = 0.5))+  
61   geom_point(aes(happiness_score, dystopia_residual, col = "DISTOPIA", alpha = 0.5))+  
62   geom_point(aes(happiness_score, social_support, col = "SUPPORT", alpha = 0.5))
```



10. `geom_smooth()` с методом `loess` построит график скользящей зависимости:



11. `geom_smooth()` с методом `lm` построит график простой линейной зависимости:



Как видно, уровень счастья имеет положительную корреляцию с другими показателями. И она более-менее линейна. Значит мы можем попробовать создать модель простой линейной регрессии.

12. Перед созданием модели подготовим данные. Используем `set.seed()` чтобы каждый раз при запуске скрипта разделение было одинаковое. Разделим датасет на обучающий и тестовый наборы в соотношении 7:3. Для этого используем `sample()`, чтобы создать вектор из значений TRUE и FALSE в произвольном порядке с соотношением 7:3. Затем создаём dataфреймы `data_train` и `data_test`, в первый заносим строки исходного dataфрейма в соответствии с индексами значений TRUE вектора `sample`; во второй - с индексами FALSE.

```

89 set.seed(1)
90 sample = sample(c(TRUE, FALSE), nrow(data), replace = TRUE, prob = c(0.7, 0.3))
91 data_train = data[sample, ]
92 data_test = data[!sample, ]

```

13. С помощью *lm()* создаём модель линейной регрессии, используя все признаки из обучающего датасета:

```

96 model1 = lm(happiness_score ~ gdp_per_capita + health + freedom +
97               dystopia_residual + government_trust + social_support +
98               cpi_score + generosity, data_train)

```

14. Используем *summary()* для просмотра информации по модели:

#### Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.179218	0.009397	19.072	< 2e-16 ***
gdp_per_capita	1.489590	0.102674	14.508	< 2e-16 ***
health	0.094819	0.016950	5.594	3.52e-08 ***
freedom	0.127146	0.018854	6.744	3.94e-11 ***
dystopia_residual	0.348853	0.027791	12.553	< 2e-16 ***
government_trust	0.025164	0.026522	0.949	0.343
social_support	0.471894	0.054126	8.718	< 2e-16 ***
cpi_score	0.025795	0.019717	1.308	0.191
generosity	0.104816	0.020275	5.170	3.30e-07 ***

Как видно из полученных данных, в данном случае (при использовании всех признаков) параметры "government\_trust" и "cpi\_score" почти не влияют на результат предсказаний, их значение оценки и стандартной ошибки малы и почти одинаковы.

```

64 ggplot(data, aes(happiness_score, params))+
65   geom_smooth(aes(happiness_score, cpi_score,
66               col = "CPI" ), method = loess)+ 
67   geom_smooth(aes(happiness_score, gdp_per_capita,
68               col = "GDP" ), method = loess)+ 
69   geom_smooth(aes(happiness_score, freedom,
70               col = "FREEDOM" ), method = loess)+ 
71   geom_smooth(aes(happiness_score, health,
72               col = "HEALTH" ), method = loess)+ 
73   geom_smooth(aes(happiness_score, government_trust,
74               col = "GOV_TRUST" ), method = loess)+ 
75   geom_smooth(aes(happiness_score, family,
76               col = "FAMILY" ), method = loess)+ 
77   geom_smooth(aes(happiness_score, generosity,
78               col = "GENEROSITY"), method = loess)+ 
79   geom_smooth(aes(happiness_score, dystopia_residual,
80               col = "DISTOPIA" ), method = loess)+ 
81   geom_smooth(aes(happiness_score, social_support,
82               col = "SUPPORT" ), method = loess)

```

15. Создадим новую модель без этих параметров:

```
103 model2 = lm(happiness_score ~ gdp_per_capita + health + freedom +  
104     dystopia_residual + social_support + generosity, data_train)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.175937	0.009213	19.097	< 2e-16 ***
gdp_per_capita	1.562445	0.093426	16.724	< 2e-16 ***
health	0.100765	0.016697	6.035	2.93e-09 ***
freedom	0.142114	0.017585	8.082	4.12e-15 ***
dystopia_residual	0.348469	0.027841	12.516	< 2e-16 ***
social_support	0.461001	0.054050	8.529	< 2e-16 ***
generosity	0.112643	0.020017	5.627	2.92e-08 ***

Замечаем, что точность большинства признаков заметно повысилась. У тех, у которых понизилась, изменения не столь велики.

16. Теперь рассмотрим предсказания каждой из этих моделей используя *predict()*:

```
109 test_predicted_values1 = round(data.frame(hs = data_test$happiness_score, predicted = predict(model1, data_test[2:10])), digits = 2)  
110 test_predicted_values2 = round(data.frame(hs = data_test$happiness_score, predicted = predict(model2, data_test[2:10])), digits = 2)
```

	hs	predicted
4	0.75	0.72
6	0.74	0.71
7	0.73	0.71
15	0.70	0.69
17	0.69	0.67
18	0.69	0.70
20	0.67	0.64
21	0.66	0.69
29	0.65	0.61
35	0.61	0.64
37	0.61	0.63

	hs	predicted
4	0.75	0.71
6	0.74	0.71
7	0.73	0.71
15	0.70	0.68
17	0.69	0.67
18	0.69	0.70
20	0.67	0.64
21	0.66	0.69
29	0.65	0.62
35	0.61	0.64
37	0.61	0.63

17. Для наглядности суммируем все отклонения каждой модели:

```
112 n1 = sum(abs(test_predicted_values1[ , 1] - test_predicted_values1[ , 2]))  
113 n2 = sum(abs(test_predicted_values2[ , 1] - test_predicted_values2[ , 2]))
```

```
> n1  
[1] 8.74  
> n2  
[1] 8.97
```

Как видим, вторая модель ошибается сильнее, хоть и не критично. Из этого можно сделать вывод, что пусть признаки "cpi\_score", "government\_trust" и имеют очень слабое влияние на результат предсказаний, в отличие от остальных, всё же их удаление немножко повысило ошибку модели.

# **Заключение**

В процессе выполнения работы было проведено знакомство с простейшими задачами статистики на примере выбранного датасета с самостоятельно поставленной проблемой для исследования, начальное знакомство с пакетом статистических программ R, а также построение простой линейной регрессии и проверка её предсказаний.

# **Список литературы**

[1] - "Statistics for Business and Economics" - Дэвид Р. Андерсон, Томас А. Уильямс, Деннис Р. Суини.

[2] - Приложение к одному из докладов, из которых собраны данные датасета - [https://happiness-report.s3.amazonaws.com/2023/WHR+23\\_Statistical\\_Appendix.pdf](https://happiness-report.s3.amazonaws.com/2023/WHR+23_Statistical_Appendix.pdf)

# Приложение

## Код скрипта R

```
#          Откроем необходимые библиотеки      #

library(ggplot2)

#          Откроем датасет      #

data = read.csv("dataset/data.csv")

#          Рассмотрим данные      #

str(data)
head(data, 3)

#          Проверим датасет на наличие пропусков      #

sum(is.na(data))

#          Рассмотрим отдельные параметры      #

summary(data)

# Избавимся от параметров, не имеющих значение #
# А именно континент, год и название страны #
# (счастье населения явно от них не зависит) #

data$continent = NULL
data$Year     = NULL
data$Country  = NULL

#          Нормализуем все параметры      #

#          Будем придерживаться диапазона [0, 1]      #

data$happiness_score = data$happiness_score / 10
data$cpi_score        = data$cpi_score / 100
data$gdp_per_capita   = data$gdp_per_capita / 10
data$family            = data$family / 10
data$social_support    = data$social_support / 10
data$dystopia_residual = data$dystopia_residual / 10
data = round(data, digits=2)
```

```

#          Рассмотрим данные ещё раз          #
summary(data)

# Рассмотрим корреляции между уровнем счастья #
# и другими параметрами                      #

datacor = round(cor(data, data$happiness_score), digits = 2)
colnames(datacor) = c("happiness_score")

#          Рассмотрим графики зависимости         #
ggplot(data, aes(happiness_score, freedom))+
  geom_point(aes(happiness_score, cpi_score,      col = "CPI"      ), alpha = 0.5)+
  geom_point(aes(happiness_score, gdp_per_capita, col = "GDP"      ), alpha = 0.5)+
  geom_point(aes(happiness_score, freedom,        col = "FREEDOM"   ), alpha = 0.5)+
  geom_point(aes(happiness_score, health,         col = "HEALTH"    ), alpha = 0.5)+
  geom_point(aes(happiness_score, government_trust, col = "GOV_TRUST" ), alpha = 0.5)+
  geom_point(aes(happiness_score, family,          col = "FAMILY"    ), alpha = 0.5)+
  geom_point(aes(happiness_score, generosity,      col = "GENEROSITY"), alpha = 0.5)+
  geom_point(aes(happiness_score, dystopia_residual, col = "DISTOPIA" ), alpha = 0.5)+
  geom_point(aes(happiness_score, social_support,   col = "SUPPORT"   ), alpha = 0.5)

ggplot(data, aes(happiness_score, freedom))+
  geom_smooth(aes(happiness_score, cpi_score,      col = "CPI"      ), method = loess)+
  geom_smooth(aes(happiness_score, gdp_per_capita, col = "GDP"      ), method = loess)+
  geom_smooth(aes(happiness_score, freedom,        col = "FREEDOM"   ), method = loess)+
  geom_smooth(aes(happiness_score, health,         col = "HEALTH"    ), method = loess)+
  geom_smooth(aes(happiness_score, government_trust, col = "GOV_TRUST" ), method = loess)+
  geom_smooth(aes(happiness_score, family,          col = "FAMILY"    ), method = loess)+
  geom_smooth(aes(happiness_score, generosity,      col = "GENEROSITY"), method = loess)+
  geom_smooth(aes(happiness_score, dystopia_residual, col = "DISTOPIA" ), method = loess)+
  geom_smooth(aes(happiness_score, social_support,   col = "SUPPORT"   ), method = loess)

ggplot(data, aes(happiness_score, freedom))+
  geom_smooth(aes(happiness_score, cpi_score,      col = "CPI"      ), method = lm)+
  geom_smooth(aes(happiness_score, gdp_per_capita, col = "GDP"      ), method = lm)+
  geom_smooth(aes(happiness_score, freedom,        col = "FREEDOM"   ), method = lm)+
  geom_smooth(aes(happiness_score, health,         col = "HEALTH"    ), method = lm)+
  geom_smooth(aes(happiness_score, government_trust, col = "GOV_TRUST" ), method = lm)+
  geom_smooth(aes(happiness_score, family,          col = "FAMILY"    ), method = lm)+
  geom_smooth(aes(happiness_score, generosity,      col = "GENEROSITY"), method = lm)+
```

```
geom_smooth(aes(happiness_score, dystopia_residual, col = "DYSTOPIA" ), method = lm)+  
geom_smooth(aes(happiness_score, social_support, col = "SUPPORT" ), method = lm)
```

```
#      Разделим данные на обучающий      #  
#      и тестовый наборы      #  
set.seed(1)  
sample = sample(c(TRUE, FALSE), nrow(data), replace = TRUE, prob = c(0.7, 0.3))  
data_train = data[sample, ]  
data_test = data[!sample, ]
```

```
#      Создадим модель линейной регрессии      #  
model1 = lm(happiness_score ~ gdp_per_capita + health + freedom +  
dystopia_residual + government_trust + social_support +  
cpi_score + generosity, data_train)  
summary(model1)
```

```
# Наблюдаем, что "cpi_score и government_trust" #  
#      хуже всех предсказывают результат;      #  
#      Попробуем их убрать      #  
model2 = lm(happiness_score ~ gdp_per_capita + health + freedom +  
dystopia_residual + social_support + generosity, data_train)  
summary(model2)
```

```
#      Рассмотрим предсказания моделей      #  
test_predicted_values1 = round(data.frame(hs = data_test$happiness_score, predicted = predict(model1,  
data_test[2:10])), digits = 2)  
test_predicted_values2 = round(data.frame(hs = data_test$happiness_score, predicted = predict(model2,  
data_test[2:10])), digits = 2)  
  
#      Для наглядности, сравним суммы ошибок      #  
n1 = sum(test_predicted_values1[, 1] - test_predicted_values1[, 2])  
n2 = sum(test_predicted_values2[, 1] - test_predicted_values2[, 2])
```

---