



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

**Институт информационных технологий (ИИТ)
Кафедра прикладной математики (ПМ)**

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №5
по дисциплине «Языки программирования для статистической обработки
данных»

Студент группы *ИМБО-11-23, Журавлев Ф.А.*

(подпись)

Преподаватель *Трушин СМ*

(подпись)

Москва 2025 г.

1) ЦЕЛЬ И ЗАДАЧИ

Цель практической работы:

Освоить методы анализа корреляций и построения линейных регрессионных моделей с использованием Python, R.

Задачи практической работы:

1. Рассчитать корреляцию между переменными:
 - Вычисление коэффициентов корреляции Пирсона и Спирмена.
 - Построение и визуализация корреляционной матрицы:
 - Python: pandas, seaborn.
 - R: cor(), ggcorrplot.
2. Построить линейную регрессионную модель:
 - Анализ зависимости одной переменной от другой.
 - Интерпретация результатов регрессии (коэффициенты, R^2).
 - Python: statsmodels, sklearn.
 - R: lm().
3. Сравнить подходы к расчёту корреляций и регрессии в Python, R.
4. Выявить преимущества и ограничения каждого инструмента для анализа данных.

2) РЕЗУЛЬТАТЫ ПРАКТИКИ

Шаг 1) Коэффициенты Спирмана и Пирсона.

1.1) Вычисление коэффициентов в Python.

После загрузки исходной таблицы данных в формате .csv, следует сначала рассчитать коэффициент Спирмана и Пирсона, ниже приведен код:

Рисунок 1.1 — вычисление коэффициентов.

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

pearson_corr = df_.corr(method='pearson')
spearman_corr = df_.corr(method='spearman')
print("Корреляция Пирсона:\n", pearson_corr)
print("\nКорреляция Спирмена:\n", spearman_corr)

plt.figure(figsize=(8, 6))
sns.heatmap(pearson_corr, annot=True, cmap='coolwarm')
plt.title("Корреляционная матрица Пирсона")
plt.show()

plt.figure(figsize=(8, 6))
sns.heatmap(spearman_corr, annot=True, cmap='coolwarm')
plt.title("Корреляционная матрица Спирмана")
plt.show()
```

Далее рассмотрим итоговую таблицу, где показаны все значения:

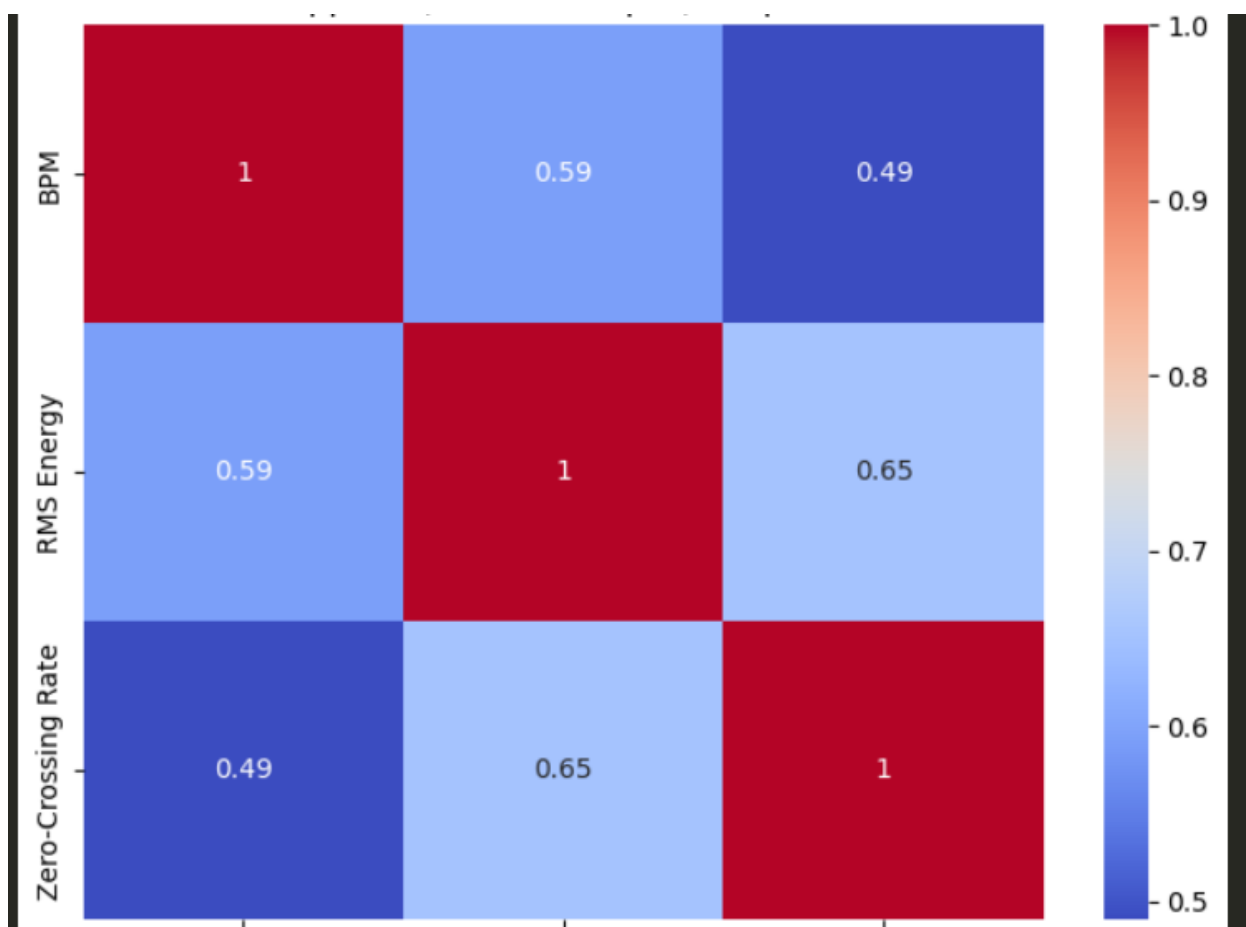
Рисунок 1.2 — Таблица значений.

Корреляция Пирсона:			
	BPM	RMS Energy	Zero-Crossing Rate
BPM	1.000000	0.591573	0.489428
RMS Energy	0.591573	1.000000	0.651859
Zero-Crossing Rate	0.489428	0.651859	1.000000

Корреляция Спирмена:			
	BPM	RMS Energy	Zero-Crossing Rate
BPM	1.000000	0.604318	0.498168
RMS Energy	0.604318	1.000000	0.590596
Zero-Crossing Rate	0.498168	0.590596	1.000000

Для наглядности можно вывести карту значений, которая более понятно покажет коэффициенты:

Рисунок 1.3 — Карта значений.



Эта карта показывает коэффициент Спирмана, и можно заметить, что она показывает то же самое, что и таблица.

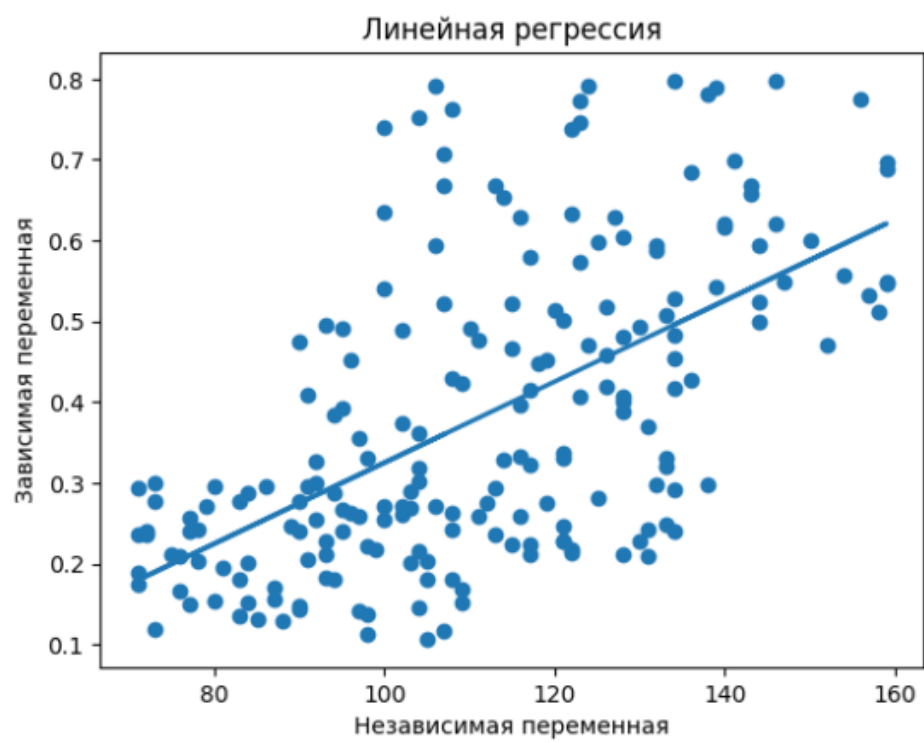
Шаг 2) Линейная регрессия.

2.1) Линейная регрессия в Python.

Рисунок 2.1 — Код линейной регрессии и ее визуализации.



Рисунок 2.2 — График линейной регрессии.



3)

Вычисление коэффициентов в R.

Рисунок 3.1 — Вычисление коэффициентов и построение матрицы.

```
library(readr)
df <- read_csv("D:/Documents/Learning/3/R/5/df_without_genre.csv")
# Установите библиотеку ggcorrplot, если она еще не установлена
# install.packages("ggcorrplot")

# Вычисление корреляций
pearson_corr <- cor(df, method = "pearson")
spearman_corr <- cor(df, method = "spearman")

# Вывод результатов
cat("Корреляция Пирсона:\n")
print(pearson_corr)
cat("\nКорреляция Спирмена:\n")
print(spearman_corr)

# Визуализация корреляции Пирсона
library(ggcorrplot)
ggcorrplot(pearson_corr, lab = TRUE, title = "Корреляционная матрица Пирсона")
```

Рисунок 3.2 — Корреляция Пирсона и корреляция Спирмана.

```
Корреляция Пирсона:
              BPM RMS.Energy Zero.Crossing.Rate
BPM           1.0000000  0.5915732           0.4894280
RMS.Energy     0.5915732  1.0000000           0.6518594
Zero.Crossing.Rate 0.4894280  0.6518594           1.0000000

Корреляция Спирмена:
              BPM RMS.Energy Zero.Crossing.Rate
BPM           1.0000000  0.6043181           0.4981683
RMS.Energy     0.6043181  1.0000000           0.5905963
Zero.Crossing.Rate 0.4981683  0.5905963           1.0000000
```

Таблица в R идентична таблице в Python.

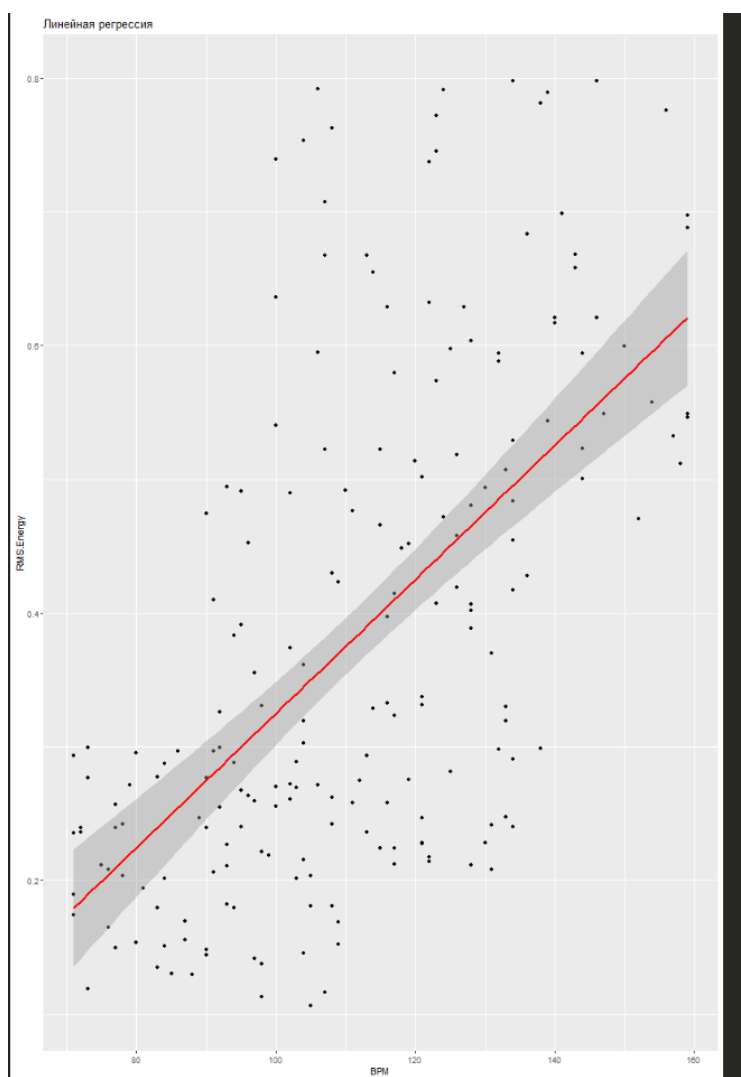
Далее реализуем линейную зависимость на языке R.

Рисунок 3.3 — Линейная зависимость в R.

```
model <- lm(BPM ~ RMS.Energy, data = df)
print(summary(model))
library(ggplot2)
print(ggplot(df, aes(x = BPM, y = RMS.Energy)) +
  geom_point() +
  geom_smooth(method = "lm", color = "red") +
  labs(title = "Линейная регрессия"))
```

И посмотрим на получившийся график линейной зависимости.

Рисунок 3.4 — График линейной зависимости в R.



3) Сравнения результатов

И там, и там результаты получились одинаковые.

Если же рассматривать линейную зависимость, то видно, даже по графику, что они совпали.

4) Выводы

В 5 практической работе были получены результаты корреляций Спирмана и Пирсона, а также построена линейная модель регрессии от двух переменных.