



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 №7

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:

Научиться анализировать временные ряды, выявлять тренды и сезонность, а также использовать визуальные инструменты для их интерпретации в Python, R.

Задачи практической работы:

1. Выполнить анализ временных рядов:
 - Загрузить данные временных рядов в Python, R.
 - Выявить тренды и сезонность.
 - Python: использование pandas (rolling, resample).
 - R: использование пакетов forecast, zoo.
2. Выполнить визуализацию временных рядов:
 - Построение графиков трендов, сезонности и остатков:
 - Python: matplotlib и seaborn.
 - R: ggplot2 и forecast.
3. Сравнить результаты анализа и визуализации в Python, R
4. Определить, какие инструменты наиболее подходят для анализа временных рядов в зависимости от задач.

2) РЕЗУЛЬТАТЫ ПРАКТИКИ

Шаг 1) Временный ряды в Python

1.1) Анализ и построение временных рядов.

После загрузки исходной таблицы данных в формате .csv, следует проанализировать временные ряды и написать код, благодаря которому мы сможем их сравнивать.

Рисунок 1.1 — Ресемплирование временного ряда

```
import pandas as pd
import matplotlib.pyplot as plt

ds['Date'] = pd.to_datetime(ds['Date'])
# 3. Ресемплирование по месяцам (средние значения)
dsm = ds.resample('ME').agg({
    'ABV': 'mean',
    'IBU': 'mean',
    'SRM': 'mean',
    'pH': 'mean',
    'Rating': 'mean',
    'Production_Volume': 'sum', # Суммарный объем за месяц
    'Beer_Style': lambda x: x.mode()[0], # Самый частый сорт
    'Season': lambda x: x.mode()[0] # Самый частый сезон
})

plt.figure(figsize=(12, 6))
```

Проведем ресемплирование временного ряда, а затем построим график, благодаря которому сможем сравнивать объем, сорт и сезон, для определения популярности пивных напитков.

Рисунок 1.2 — Построение графика.

```
# График производства
plt.subplot(2, 1, 1)
plt.plot(dsm.index, dsm['Production_Volume'], marker='o', color='b')
plt.title('Объем производства пива по месяцам')
plt.ylabel('Литры')
plt.grid(True)

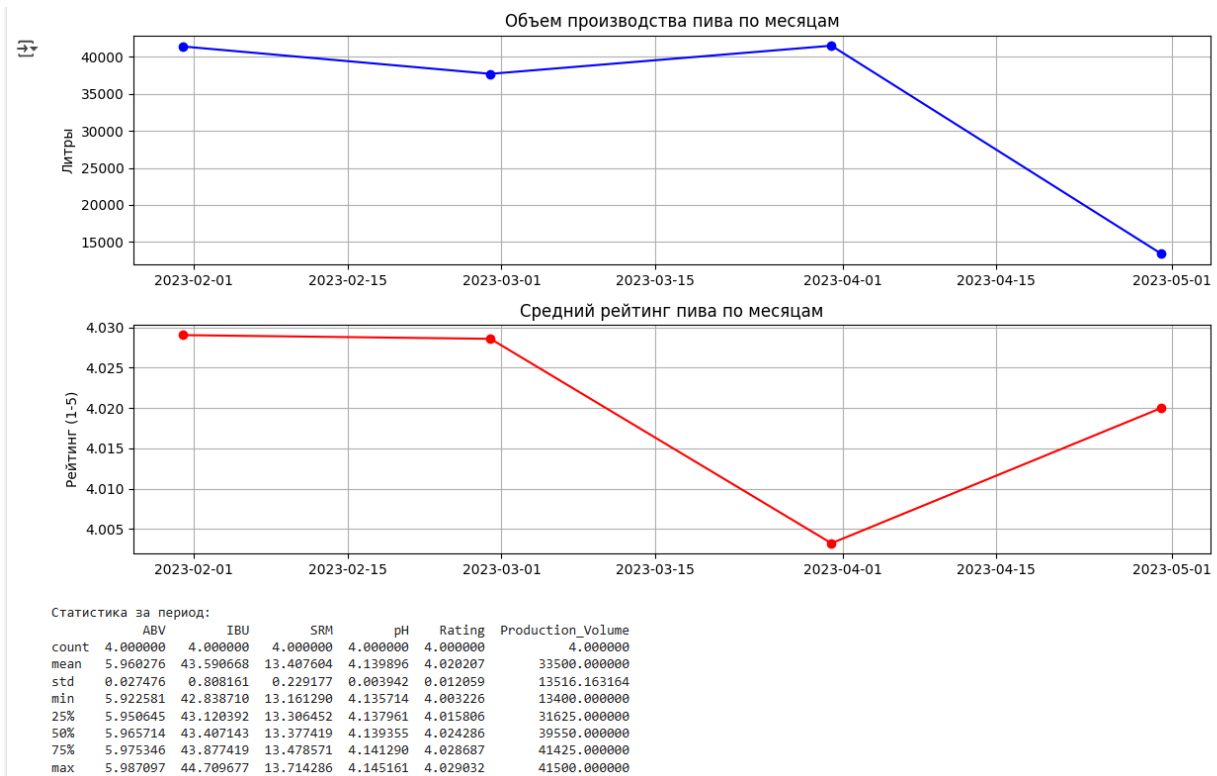
# График рейтинга
plt.subplot(2, 1, 2)
plt.plot(dsm.index, dsm['Rating'], marker='o', color='r')
plt.title('Средний рейтинг пива по месяцам')
plt.ylabel('Рейтинг (1-5)')
plt.grid(True)

plt.tight_layout()
plt.show()

# 5. Вывод статистик
print("\nСтатистика за период:")
print(dsm.describe())
```

Вот, что нам показывает Python :

Рисунок 1.3 — Сам график.



По графику можно сделать пару выводов:

1)Производство пива "Антрац" сокращается, возможно, из-за низкого рейтинга (1.5) или других факторов.

2)Параметры пива (ABV, IBU, SRM, pH) стабильны, но рейтинг низкий, что требует анализа причин недовольства потребителей.

Далее займемся декомпозицией временного ряда, посмотрим сезонность, остатки и тренды, для этого представлен код ниже:

Рисунок 1.4 — Код декомпозиции,сезонности,остатков и трендов.

```
from statsmodels.tsa.seasonal import seasonal_decompose
decomposition = seasonal_decompose(ds['Rating'],model = 'additive',period = 12)
trend = decomposition.trend
seasonal = decomposition.seasonal
residual = decomposition.resid
decomposition.plot()
plt.show()
```

↩

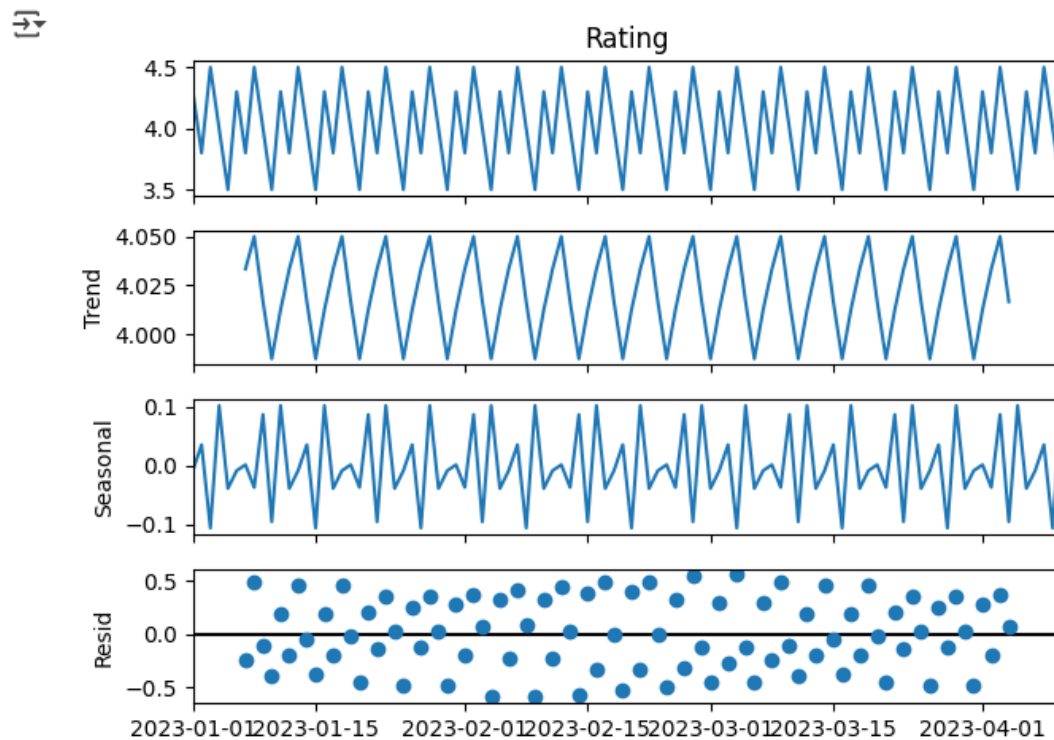
После появляется график, на котором можно будет заметить следующие признаки:

1)Положительный тренд — рейтинг постепенно растёт (с ~4.0 до ~4.05), что может говорить об улучшении качества или восприятия бренда.

2)Сезонные колебания — падения в середине февраля и марта (± 0.1), возможно, из-за изменений спроса или внешних факторов.

3) График показывает, что, несмотря на небольшие сезонные колебания, общая тенденция рейтинга — положительная.

Рисунок 1.5 — График трендов, сезонности и остатков.

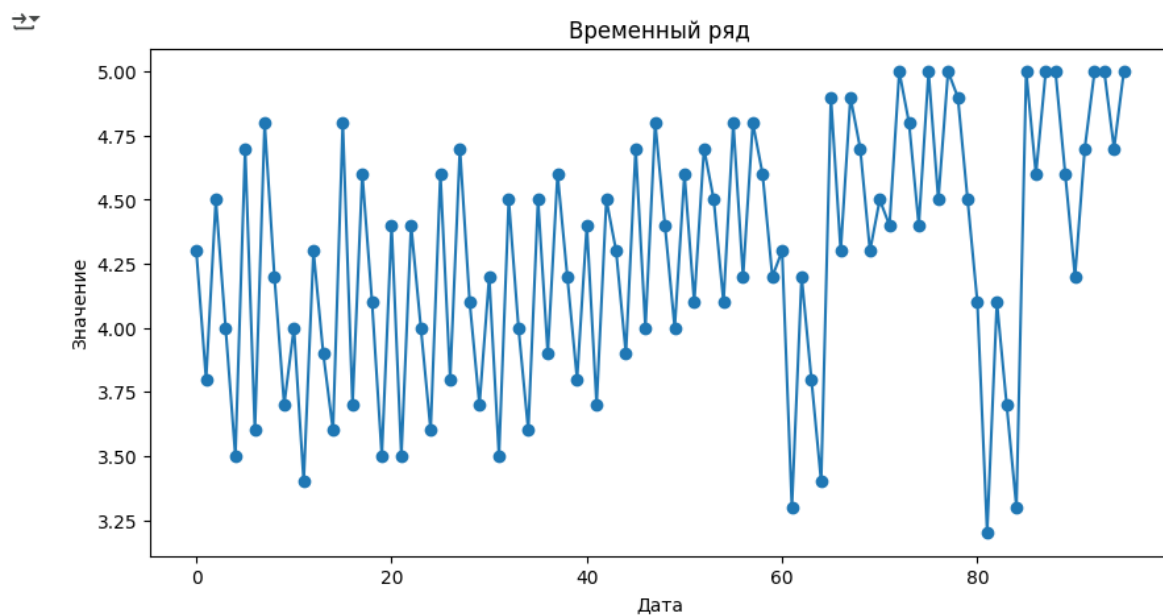


И в завершении построим самый простой временной ряд, который связан со столбцом «Rating» и визуализируем его.

Рисунок 1.6 — Временной ряд.

```
plt.figure(figsize = (10,5))
plt.plot(ds.index,ds['Rating'],marker = 'o')
plt.title("Временный ряд")
plt.xlabel("Дата")
plt.ylabel("Значение")
plt.show()
```

Рисунок 1.7 — График временного ряда.



Шаг 2) Временные ряды в R.

2.1) Построение множественной линейной регрессии.

Рисунок 2.1 — функции *descomp* и *fit*.

```

ds <- read.csv(file.choose())
ds$Date <- as.Date(ds$Date, format = "%Y-%m-%d")
ds_ts <- ts(ds$Rating,start = c(2023,1),frequency = 12)

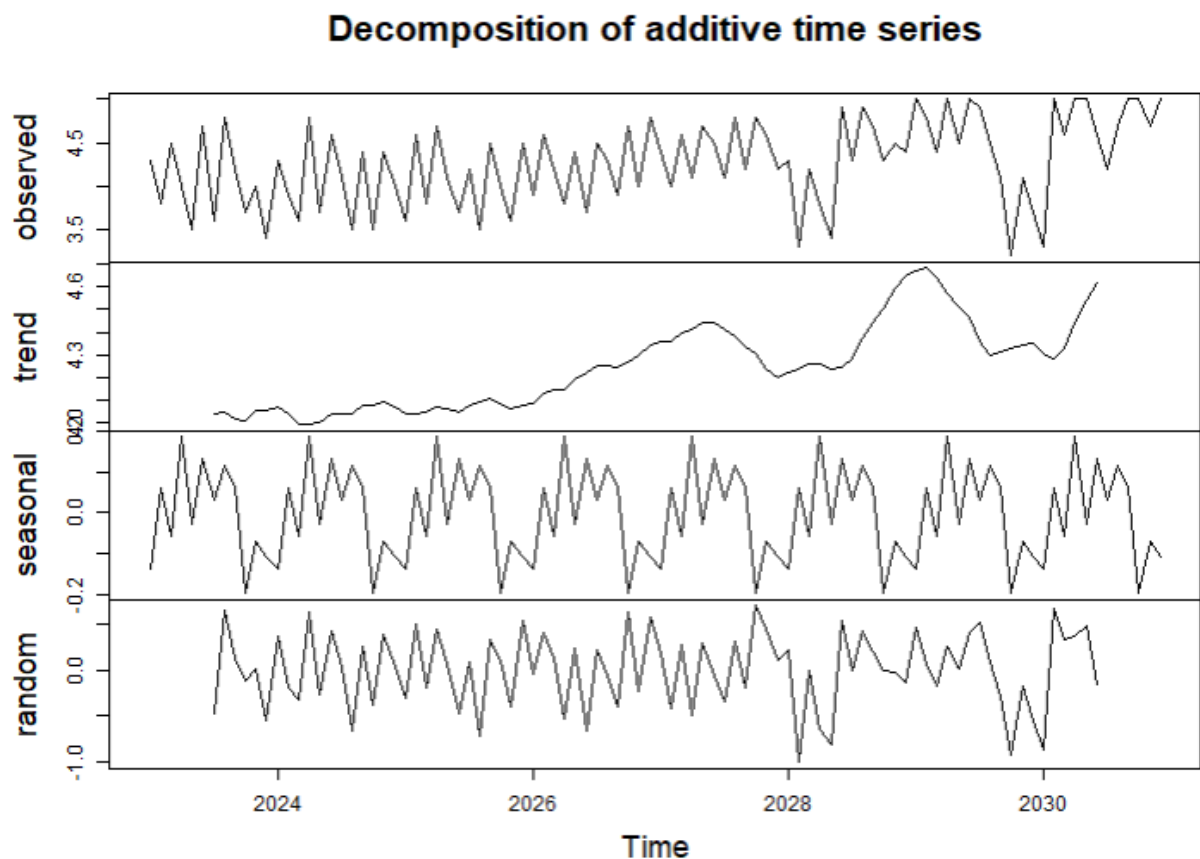
decomp <- decompose(ds_ts,type = "additive")
type = "multiplicative"
plot(decomp)

fit <- stl(ds_ts,s.window = "periodic")
plot(fit)

```

И получим следующих два друг на друга похожих графика.

Рисунок 2.1— график функции *decomp*.



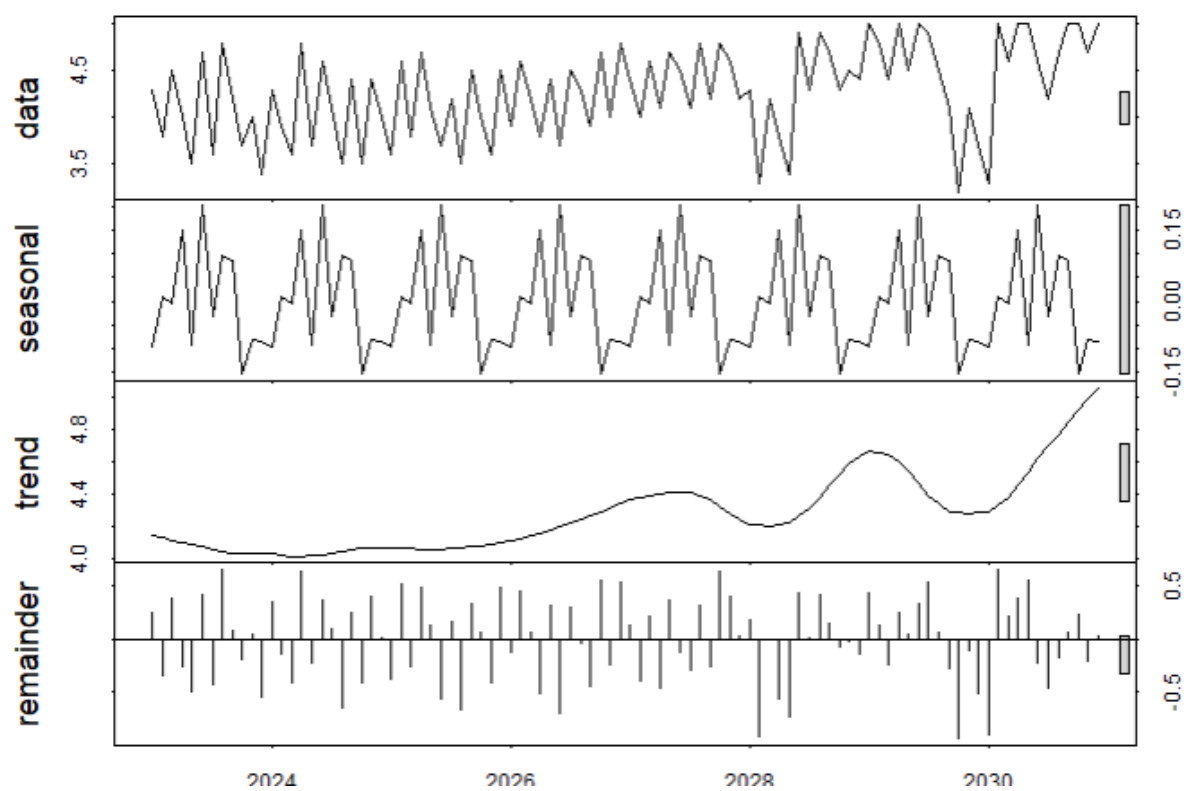


Рисунок 2.2—График функции *fit*.

3 СРАВНЕНИЕ РЕЗУЛЬТАТОВ

Результаты получились одинаковые на двух языках программирования. Python, по моему мнению, кажется удобнее и доступнее ввиду большей опытности в нем, однако R тоже является хорошим инструментом

4 ВЫВОДЫ

В 7 практической работе проведена работа со способами реализации временных рядов, рассмотрели тренды, сезонность, остатки, графики. А также они были проанализированы и реализованы на разных языках программирования для статистической обработки данных.