|  |
| --- |
|  |
| 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 №3

по дисциплине «Языки программирования для статистической обработки данных»

|  |  |
| --- | --- |
| Студент группы | *ИМБО-11-23 Шамков В.Н.* |
|  | (подпись) |
| Преподаватель | *Трушин СМ* |
|  | (подпись) |

Москва 2025 г.

# СОДЕРЖАНИЕ

[1 ЦЕЛЬ И ЗАДАЧИ 3](#_bookmark0)

1. [РЕЗУЛЬТАТЫ ПРАКТИЧЕСКОЙ РАБОТЫ 4](#_bookmark1)
   1. [Графическое представление в Python 4](#_bookmark1)
   2. [Графическое представление в Rstudio 6](#_bookmark2)
   3. [Графическое представление в GLARUS BI 9](#_bookmark3)
2. [ИТОГИ И ВЫВОДЫ: 10](#_bookmark4)

# 1 ЦЕЛЬ И ЗАДАЧИ

## Цель практической работы:

Освоить методы построения базовых графиков в Python, R и Glarus BI, а также научиться визуализировать данные с использованием различных инструментов.

## Задачи практической работы:

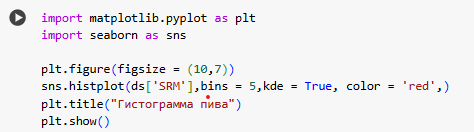
1. Построить базовые графики:
   * Гистограммы, линейные графики, боксплоты в Python (matplotlib, seaborn).
   * Те же графики в R (ggplot2).
2. Создать визуализации в Glarus BI:
   * Столбчатые диаграммы, гистограммы и тепловые карты (древо видную карту в случае если тепловые карты получаются некор ректные) .
   * Использовать интерактивные элементы Glarus BI для настройки графиков.
3. Сравнить визуализации, созданные в Python, R и Glarus BI.
4. Проанализировать, в каких ситуациях каждый инструмент наиболее эффективен.

## РЕЗУЛЬТАТЫ ПРАКТИЧЕСКОЙ РАБОТЫ

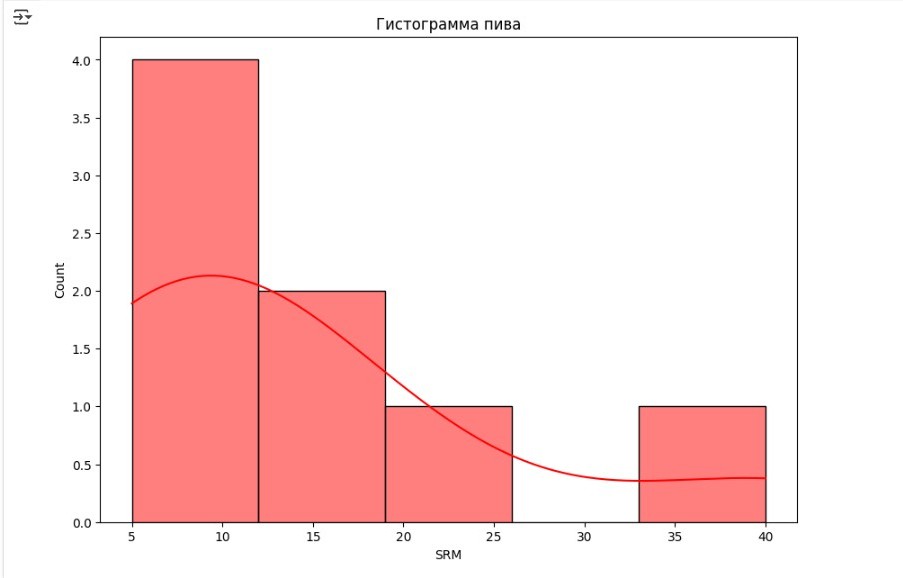
### Графическое представление в Python.

После загрузки данных в Python, первым делом необходимо построить гистограмму по исходному набору данных, построим гистограмму по столб- цу SRM. Ниже представлен код для реализации графика:

*рисунок 2.1.1 – Код гистограммы*

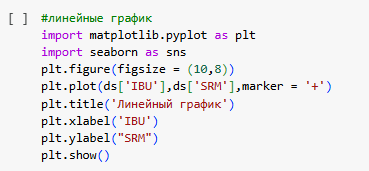
**

Далее посмотрим на получившийся график:

*рисунок 2.1.2 – Гистограмма пива*

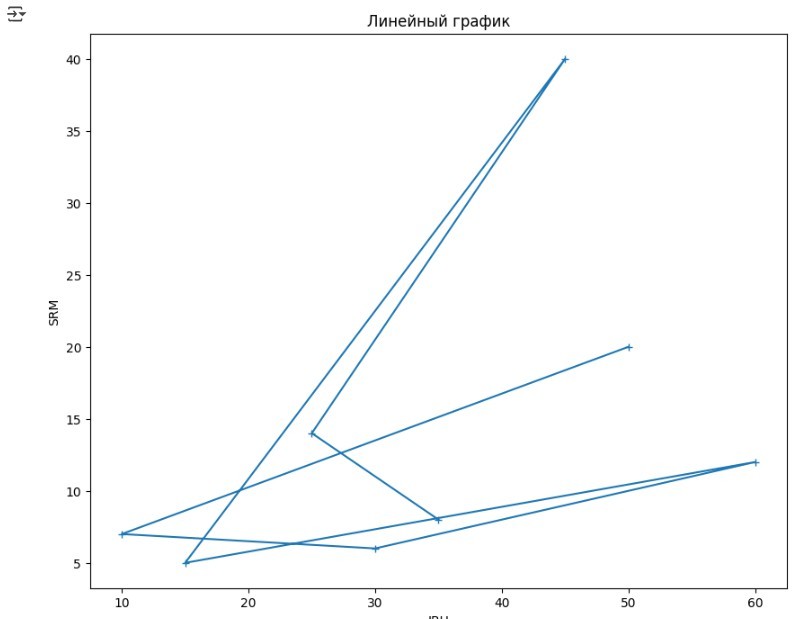
Далее построим график линейной зависимости в Python и напишем код:

*рисунок 2.1.3 – код графика линейной зависимости*

**

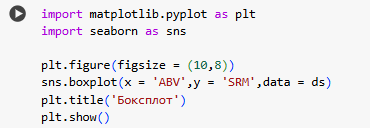
Вот график, который нам выводит Python:

*рисунок 2.1.4 – График линейной зависимости*

**

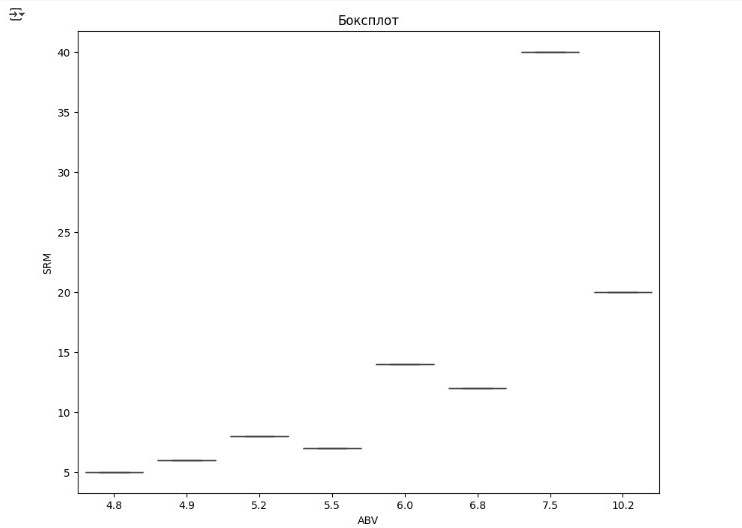
Далее расмотрим последний график из практической работы «бокс- плот». Напишем код его реализации :

*рисунок 2.1.5 – Код реализации боксплота*

**

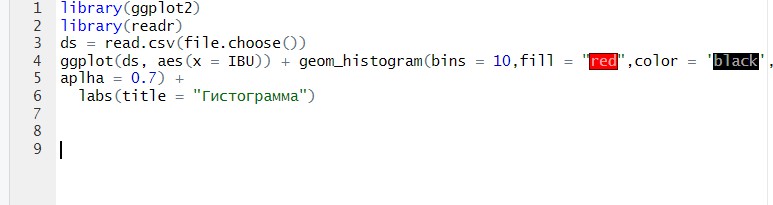
Посмотрим что он нам выводит:

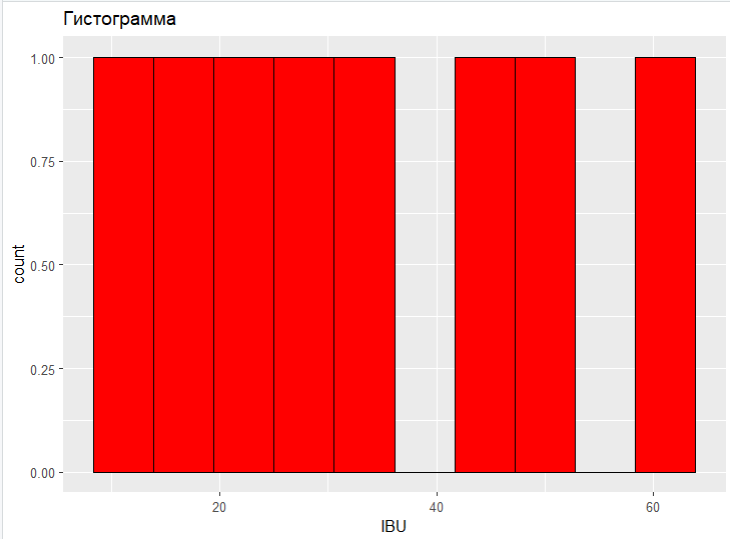
*рисунок 2.1.6 – график блоксплот*а



## Графическое представление в Rstudio

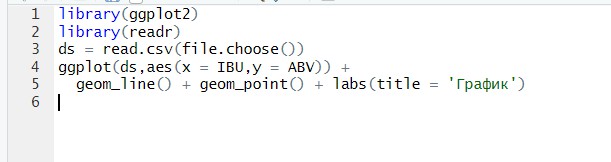
Теперь рассмотрим реализацию тех же график с помощью языка R.

*Рисунок 2.2.1 – код Гистограммы.*

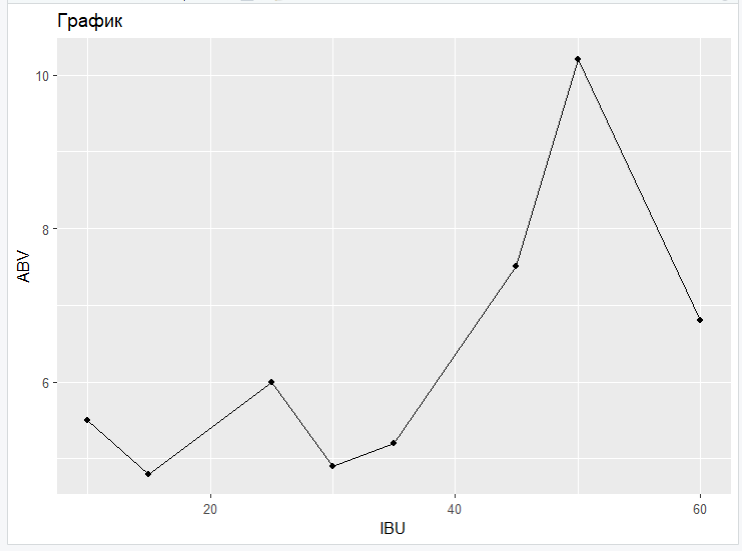
*Рисунок 2.2.2 – Гистограмма.*

Теперь посмотрим на реализацию графика линейной зависимости:

*Рисунок 2.2.3 – код линейной зависимости.*

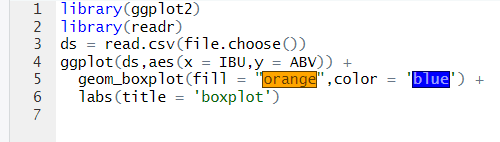
**

*Рисунок 2.2.4 – график линейной зависимости*

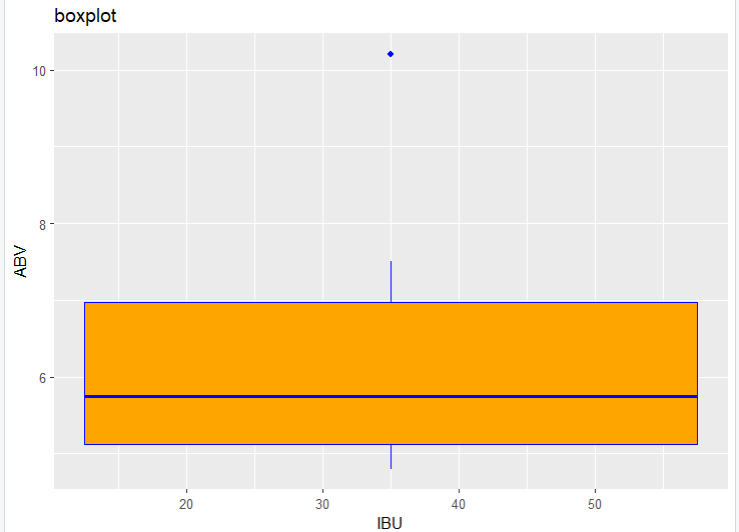
**

Рассмотрим «блоксплот» на языке R:

*Рисунок 2.2.5 – код блоксплота*

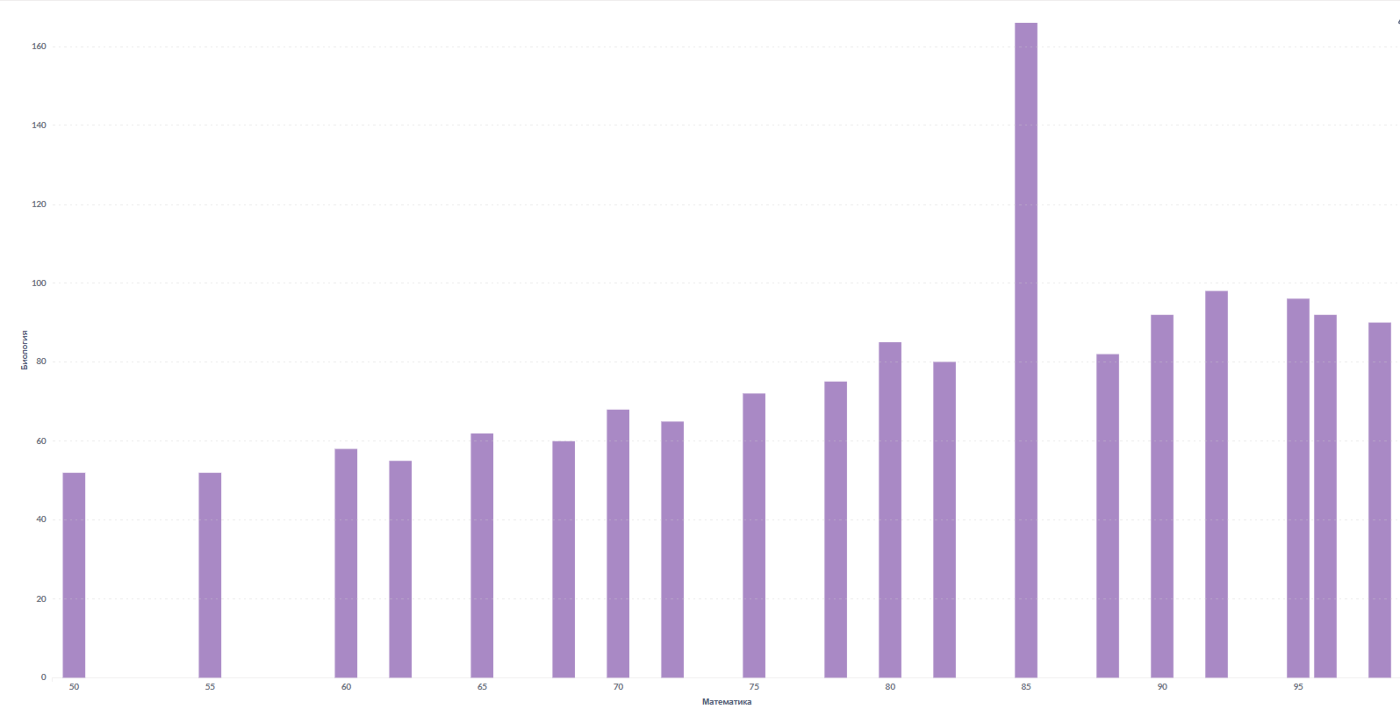
**

*Рисунок 2.2.6 – Блоксплот*

**

## Графическое представление в GLARUS BI.

Нажав одну кнопку в пункте визуализации данных в GLARUS BI мож- но построить много различных диаграмм, например гистограмму снизу.

*Рисунок 2.3.1 – Гистограмма GLARUS BI*

# ИТОГИ И ВЫВОДЫ:

Рассмотрев различные способы построения графиков и диаграмм с по- мощью двух языков программирования и программы для анализа данных, можно сделать пару выводов. Например, работа с графиками и таблица в R куда удобнее засчет того, что код выглядит и читается более понятно, Python же показывает самые «красивые» диаграммы для глаза, по ним делать выво- ды показалось делать более проще, но а Glarus BI, как всегда бесполезен, и является инструментом, который полезен будет только D2-шнику.