

Разведочный анализ данных в python (GoogleColab)

Матвеева Алёна Сергеевна

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

Целью исследования является построение графиков на основе данных на языке программирования Python. Для реализации использовалась свободно распространяемая платформа GoogleColab. Данное исследование может быть использовано методическим пособием в учебной деятельности, а также для пользователей, которые работают с анализом статистических данных.

Ключевые слова: Python, график, таблица.

Exploratory Data Analysis in python (Google Colab)

Matveeva Alyona Sergeevna

Sholom-Aleichem Priamursky State University

Student

Abstract

The purpose of the study is to build graphs based on data in the Python programming language. The freely distributed Google Colab platform was used for implementation. This study can be used as a methodological guide in educational activities, as well as for users who work with the analysis of statistical data.

Keywords: Python, graph, table.

1 Введение

1.1 Актуальность

Актуальностью данной темы заключается в широком использовании и представлении информации. Визуализация данных — это метод, который позволяет специалистам по анализу данных преобразовывать данные в диаграммы и графики, которые несут важную информацию. Графики уменьшают сложность данных и представляют данные более понятно для любого пользователя.

1.2 Обзор исследований

В статье А.О. Кизянова рассматривается процесс создания графиков, используя библиотеку seaborn [2]. О.А. Шутова и Д.А. Грамаков проводят обзор наиболее широко используемых библиотек для визуализации данных при программировании на языке Python [3]. А также в исследовании О.В. Кудринской рассматриваются возможности языка Python и популярные библиотеки, позволяющие визуализировать данные [4]. М.А. Насруева проводит анализ набора данных с отображением графиков и диаграмм [5].

О.А. Ивина проводит исследование визуализации статистических характеристик данных в python [6]. В англоязычной статье R. Aurachman рассматривает визуализацию данных с использованием пакета программирования seaborn python [7].

1.3 Цель исследования

Целью исследования является разведочный анализ данных на языке программирования Python.

2 Материалы и методы

В данном исследовании используется платформа Google Colab для написания кода на языке программирования Python.

Материалы с данными можно скачать по ссылке [1]. В файле содержится таблица с данными со столбцами: город, регион, контрагент, товар, дата, продажи и себестоимость.

3 Результаты

В данном исследовании рассмотрим построение графиков, используя библиотеки для визуализации данных на основе данной таблицы.

Для начала подключим библиотеки и загрузим файл «Данные.xlsx» в Google Colab (рис.1).

```
import seaborn as sns
import pandas as pd
import numpy as np
import matplotlib
df = pd.read_excel('Данные.xlsx')
```

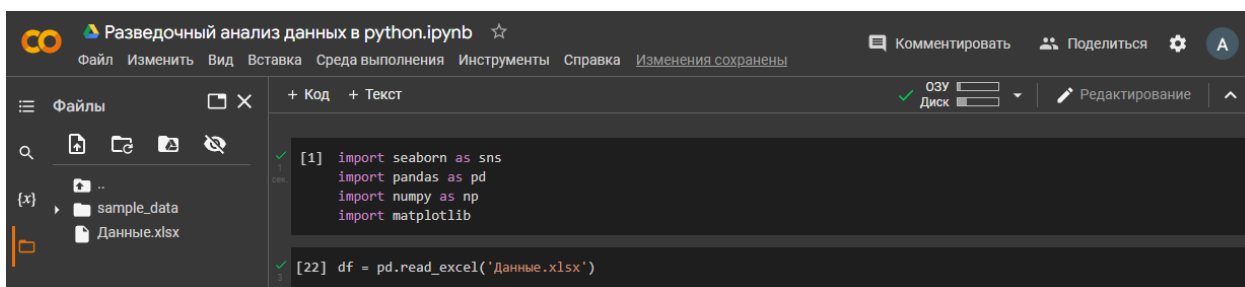
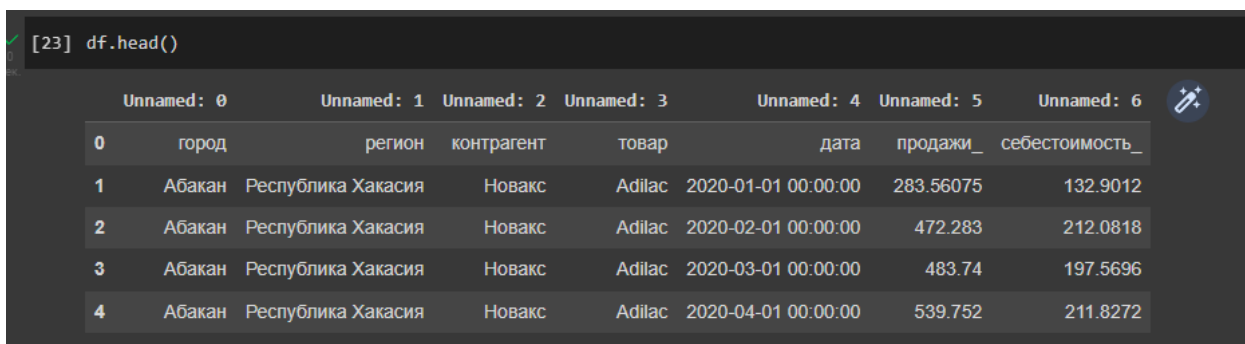


Рисунок 1 – Подключение библиотек и загрузка файла

Посмотрим первые 5 строк таблицы из файла «Данные.xlsx» (рис.2).

The image shows a screenshot of the Google Colab interface displaying the output of the `df.head()` command. The output is a table with 7 columns and 5 rows. The columns are labeled 'Unnamed: 0' through 'Unnamed: 6'. The data rows show information about sales, including city, region, counterparty, goods, date, sales volume, and cost.

| | Unnamed: 0 | Unnamed: 1 | Unnamed: 2 | Unnamed: 3 | Unnamed: 4 | Unnamed: 5 | Unnamed: 6 |
|---|------------|--------------------|------------|------------|---------------------|------------|----------------|
| 0 | город | регион | контрагент | товар | дата | продажи | себестоимость_ |
| 1 | Абакан | Республика Хакасия | Новакс | Adilac | 2020-01-01 00:00:00 | 283.56075 | 132.9012 |
| 2 | Абакан | Республика Хакасия | Новакс | Adilac | 2020-02-01 00:00:00 | 472.283 | 212.0818 |
| 3 | Абакан | Республика Хакасия | Новакс | Adilac | 2020-03-01 00:00:00 | 483.74 | 197.5696 |
| 4 | Абакан | Республика Хакасия | Новакс | Adilac | 2020-04-01 00:00:00 | 539.752 | 211.8272 |

Рисунок 2 – Вывод первых 5 строк файла

Чтобы первая строка стала названием столбцов, используем функцию `set_index`, затем проверяем Index данной таблицы и с помощью команды `df.columns` изменяем названия у столбцов (рис. 3).

```
df = df.T.set_index(0).T
df.head()
df.columns.str.lower()
df.columns = ['город', 'регион', 'контрагент', 'товар', 'дата', 'продажи',
              'себестоимость']
```

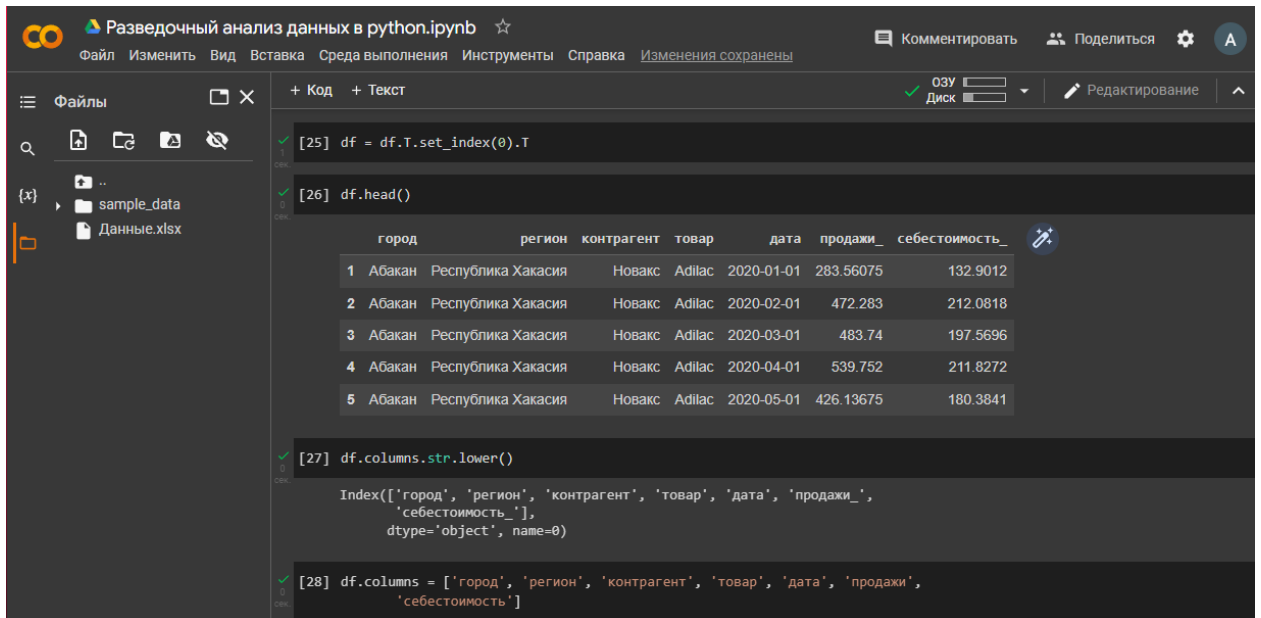


Рисунок 3 – Устанавливаем название столбцов таблицы

Проверим типы данных в таблице (рис.4).

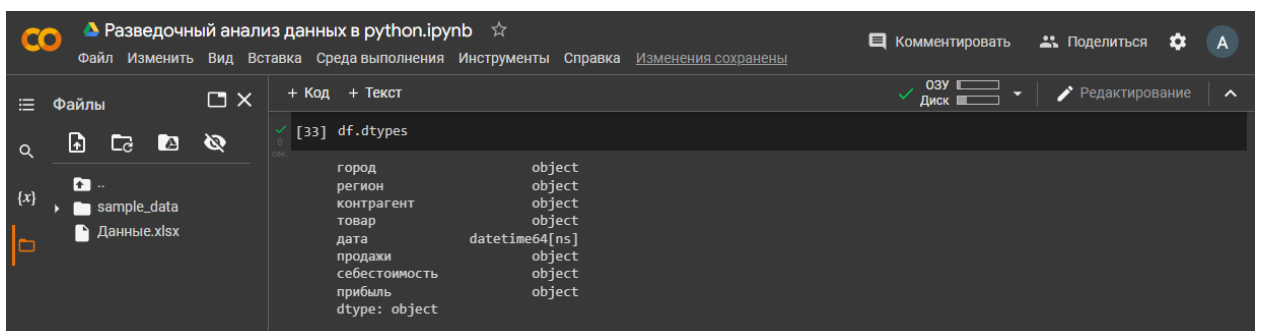


Рисунок 4 – Типы данных

Из этих данных видно, что типы всех данных object, а у столбца «Дата» тип данных `datetime64[ns]`, для подсчёта данных нужно колонки «продажи» и «себестоимость» преобразовать в численный тип.

Преобразуем «продажи» и «себестоимость» в вещественный тип данных. Затем рассчитаем прибыль, для этого требуется от продаж отнять себестоимость, и выведем таблицу для просмотра результата (рис.5).

```
df['продажи'] = df.продажи.astype(float)
df['себестоимость'] = df.себестоимость.astype(float)
df['прибыль'] = df['продажи'] - df['себестоимость']
df.dtypes
df
```

The screenshot shows a Jupyter Notebook environment with the following code and output:

```
[42] df['продажи'] = df.продажи.astype(float)
df['себестоимость'] = df.себестоимость.astype(float)

[43] df['прибыль'] = df['продажи'] - df['себестоимость']

df.dtypes
```

The output of `df.dtypes` is:

```
город          object
регион         object
контрагент     object
товар          object
дата          datetime64[ns]
продажи        float64
себестоимость  float64
прибыль        float64
dtype: object
```

The output of `df` is a DataFrame with the following data:

| | город | регион | контрагент | товар | дата | продажи | себестоимость | прибыль |
|-------|-------------|---------------------|------------|---------|------------|-----------|---------------|-----------|
| 1 | Абакан | Республика Хакасия | Новакс | Adilac | 2020-01-01 | 283.56075 | 132.9012 | 150.65955 |
| 2 | Абакан | Республика Хакасия | Новакс | Adilac | 2020-02-01 | 472.283 | 212.0818 | 260.2012 |
| 3 | Абакан | Республика Хакасия | Новакс | Adilac | 2020-03-01 | 483.74 | 197.5696 | 286.1704 |
| 4 | Абакан | Республика Хакасия | Новакс | Adilac | 2020-04-01 | 539.752 | 211.8272 | 327.9248 |
| 5 | Абакан | Республика Хакасия | Новакс | Adilac | 2020-05-01 | 426.13675 | 180.3841 | 245.75265 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 23324 | Новокузнецк | Кемеровская область | Свет | Galosha | 2022-08-01 | 0.6372 | 0.1845 | 0.4527 |
| 23325 | Новокузнецк | Кемеровская область | Свет | Galosha | 2022-09-01 | 0.654 | 0.3 | 0.354 |
| 23326 | Новокузнецк | Кемеровская область | Свет | Galosha | 2022-10-01 | 0.756 | 0.336 | 0.42 |
| 23327 | Новокузнецк | Кемеровская область | Свет | Galosha | 2022-11-01 | 0.9048 | 0.26325 | 0.64155 |

Рисунок 5 – Изменение типов данных и их расчет

Подключаем `matplotlib.pyplot`, прописываем функцию `plt.subplots` с параметрами для изменения размера рисунка графика. Затем прописываем функцию для построения гистограммы. В данном случае, чтобы сделать график «Продажи по городам», назначаем осям X и Y колонки "город" и "продажи" (рис.6).

```
import matplotlib.pyplot as plt
ax = plt.subplots(figsize=(15, 6))
sns.barplot(data=df, x="город", y="продажи", estimator=np.sum)
```

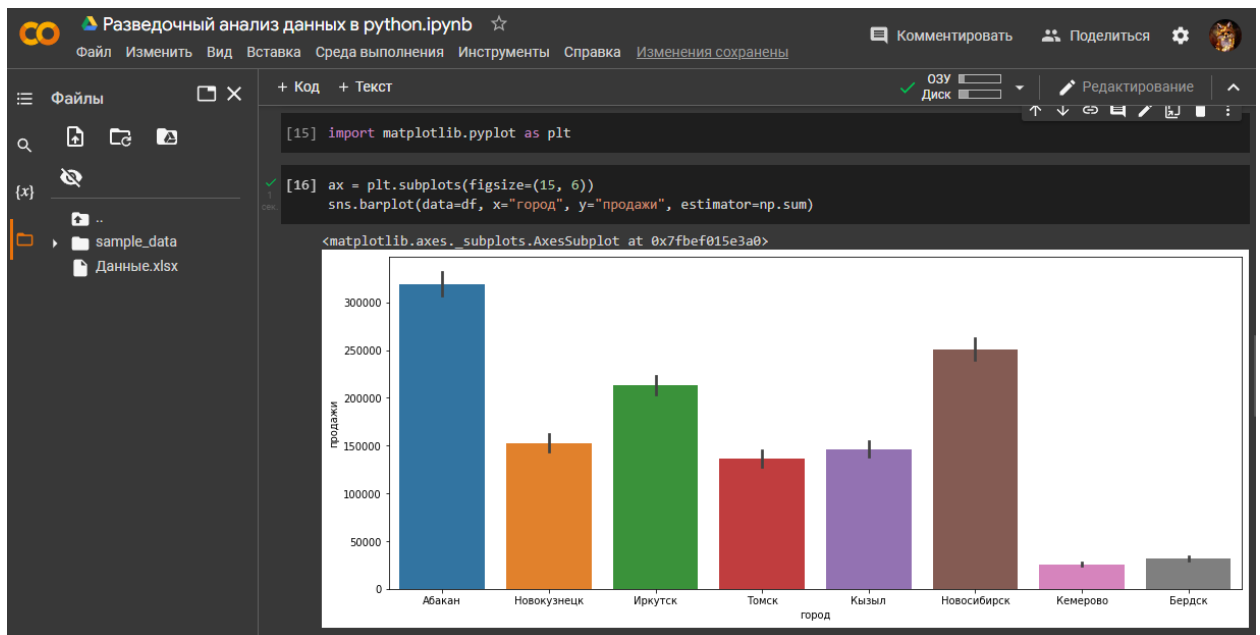


Рисунок 6 – График «Продажи по городам»

Чтобы построить график «Продажа по брендам», используем функцию `df.groupby` и просуммируем столбец «продажи» по товарам, назовем таблицу «`товар`» (рис.7).

```
товар = df.groupby("товар")["продажи"].sum().sort_values()
товар
```

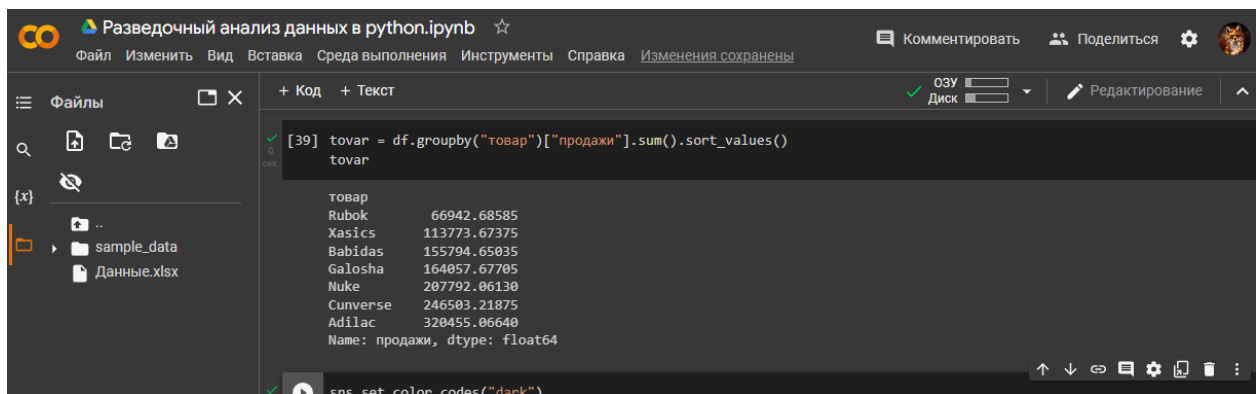


Рисунок 7 – Суммирования столбца «продажи» по товарам

Пропишем функцию `sns.set_color_codes("dark")`, чтобы цвет графика был темным оттенком. К таблице «`товар`» прописываем функцию для построения линейного графика, и указываем параметры размера шрифта и самого графика, а также цвет (рис. 8).

```
sns.set_color_codes("dark")
товар.plot(kind="barh", fontsize=10, figsize=(12, 9), color="m")
```

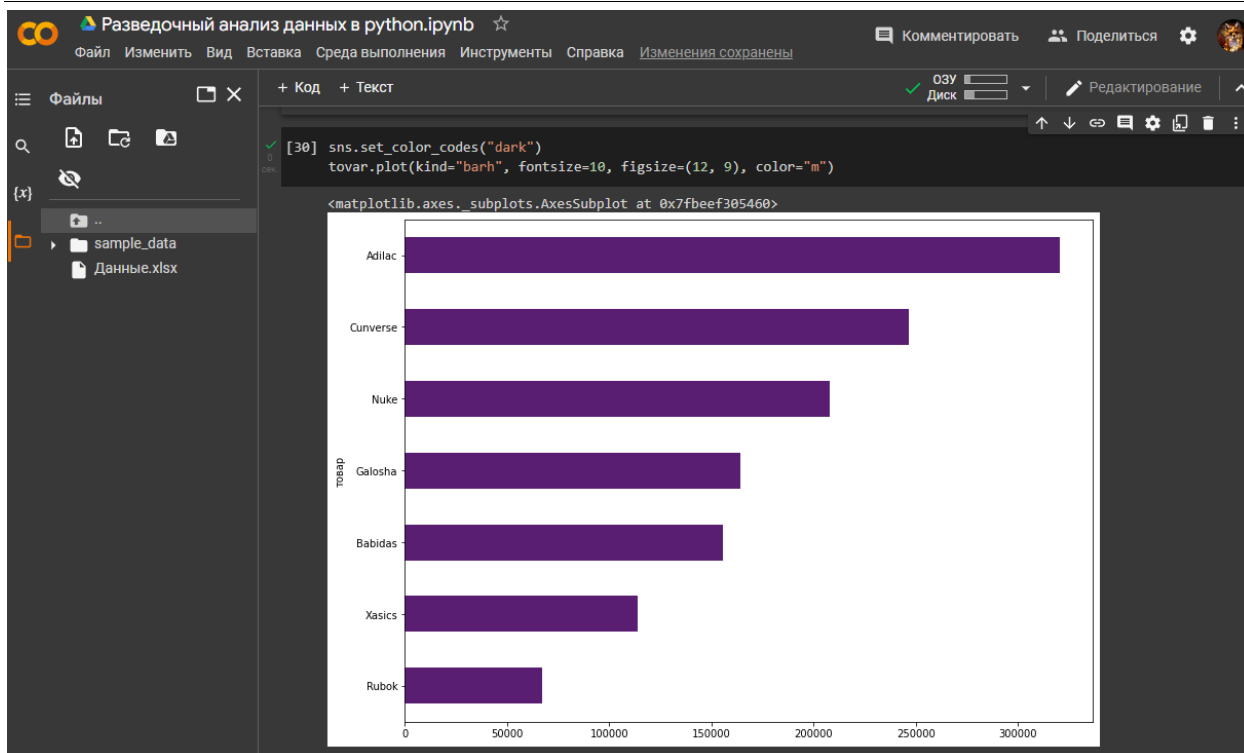


Рисунок 8 – График «Продажа по брендам»

Устанавливаем стиль `sns.set_theme()` и прописываем функцию `plt.subplots` для сложных графиков с параметром размера графика. Прописываем функцию `sns.set_color_codes("muted")`, чтобы цвет графика был приглушенным оттенком. Затем прописываем функцию `sns.barplot` для построения графиков, прописываем какие данные будут отображаться и прописываем разные параметры цвета, для того что различались данные на графике. Дополняем код функцией `ax.legend()` для добавления легенды к графическому представлению и функцией `sns.despine()`, чтобы удалить верхнюю часть изображения и справа от оси, что делает изображение графика более красивым для визуализации (рис.9).

```
sns.set_theme(style="whitegrid")
f, ax = plt.subplots(figsize=(15, 15))
sns.set_color_codes("muted")
sns.barplot(x="себестоимость", y="контрагент", data=df,
            label="Себестоимость", color="m")

sns.barplot(x="прибыль", y="контрагент", data=df,
            label="Валовая прибыль", color="c")

ax.legend(ncol=2, loc="lower right", frameon=True)
ax.set( ylabel="",
        xlabel="Продажи по клиентам")
sns.despine(left=True, bottom=True)
```

```

[62] sns.set_theme(style="whitegrid")
f, ax = plt.subplots(figsize=(15, 15))
sns.set_color_codes("muted")
sns.barplot(x="себестоимость", y="контрагент", data=df,
            label="Себестоимость", color="m")

sns.barplot(x="прибыль", y="контрагент", data=df,
            label="Валовая прибыль", color="c")

ax.legend(ncol=2, loc="lower right", frameon=True)
ax.set_ylabel="",
        xlabel="Продажи по клиентам")
sns.despine(left=True, bottom=True)
    
```

Рисунок 9 – Код для сложного графика

На графике «Продажи по клиентам» изображены данные на оси Y «контрагент», на оси X голубым цветом данные «прибыль», фиолетовым цветом данные «себестоимость» (рис. 10).

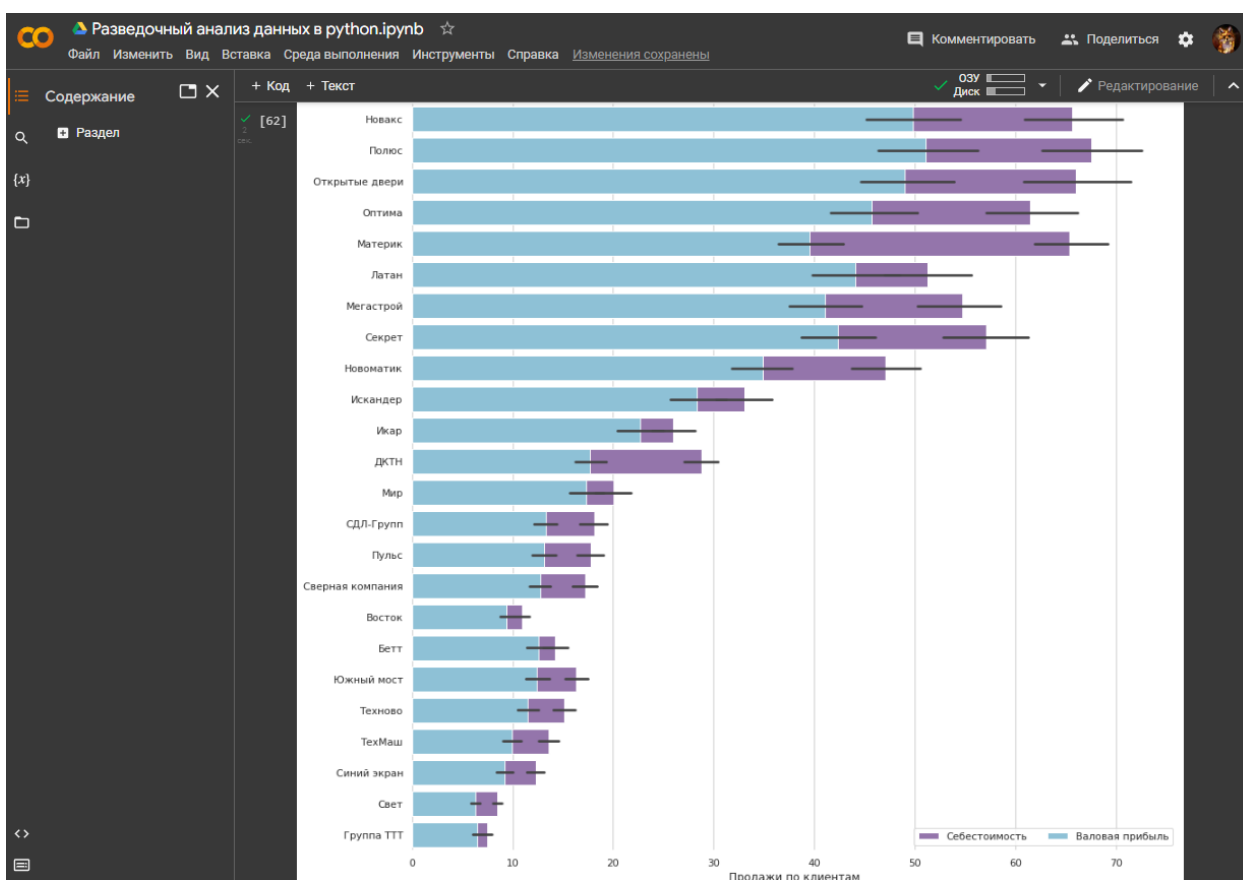
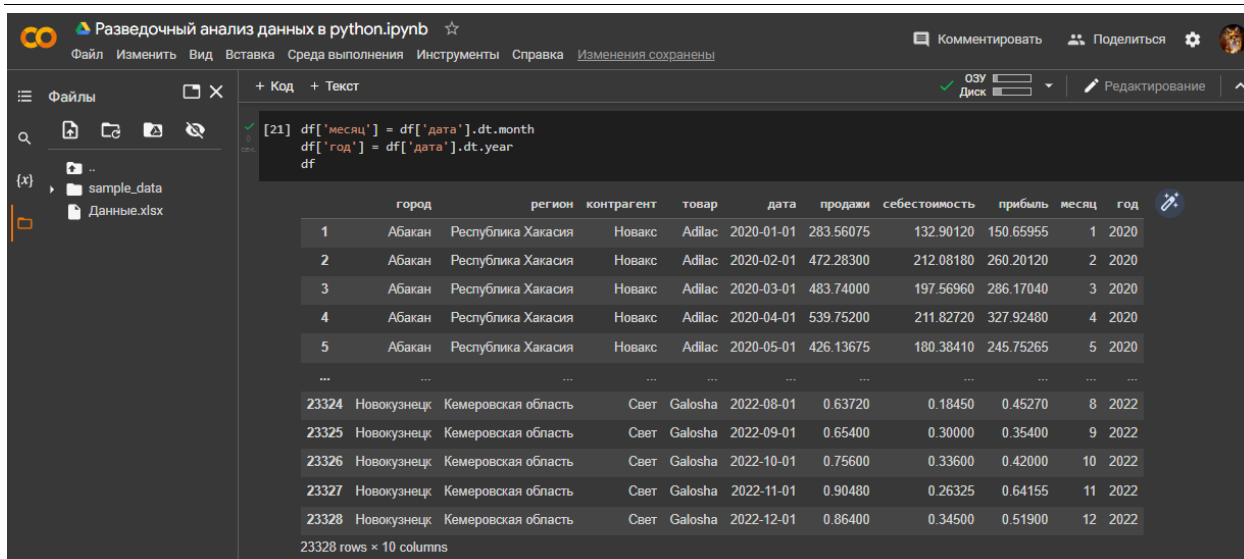


Рисунок 10 – График «Продажи по клиентам»

Прежде чем построить следующий график, требуется из столбца «дата», добавить в данную таблицу новые столбцы «месяц» и «год» (рис.11).

```

df['месяц'] = df['дата'].dt.month
df['год'] = df['дата'].dt.year
df
    
```



```
[21] df['месяц'] = df['дата'].dt.month
df['год'] = df['дата'].dt.year
df
```

| | город | регион | контрагент | товар | дата | продажи | себестоимость | прибыль | месяц | год |
|-------|-------------|---------------------|------------|---------|------------|-----------|---------------|-----------|-------|------|
| 1 | Абакан | Республика Хакасия | Новакс | Adilac | 2020-01-01 | 283.56075 | 132.90120 | 150.65955 | 1 | 2020 |
| 2 | Абакан | Республика Хакасия | Новакс | Adilac | 2020-02-01 | 472.28300 | 212.08180 | 260.20120 | 2 | 2020 |
| 3 | Абакан | Республика Хакасия | Новакс | Adilac | 2020-03-01 | 483.74000 | 197.56960 | 286.17040 | 3 | 2020 |
| 4 | Абакан | Республика Хакасия | Новакс | Adilac | 2020-04-01 | 539.75200 | 211.82720 | 327.92480 | 4 | 2020 |
| 5 | Абакан | Республика Хакасия | Новакс | Adilac | 2020-05-01 | 426.13675 | 180.38410 | 245.75265 | 5 | 2020 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 23324 | Новокузнецк | Кемеровская область | Свет | Galosha | 2022-08-01 | 0.63720 | 0.18450 | 0.45270 | 8 | 2022 |
| 23325 | Новокузнецк | Кемеровская область | Свет | Galosha | 2022-09-01 | 0.65400 | 0.30000 | 0.35400 | 9 | 2022 |
| 23326 | Новокузнецк | Кемеровская область | Свет | Galosha | 2022-10-01 | 0.75600 | 0.33600 | 0.42000 | 10 | 2022 |
| 23327 | Новокузнецк | Кемеровская область | Свет | Galosha | 2022-11-01 | 0.90480 | 0.26325 | 0.64155 | 11 | 2022 |
| 23328 | Новокузнецк | Кемеровская область | Свет | Galosha | 2022-12-01 | 0.86400 | 0.34500 | 0.51900 | 12 | 2022 |

23328 rows × 10 columns

Рисунок 11 – Добавление новых столбцов по дате

Для построения графика «Динамика продаж» прописываем функцию `plt.figure()` с параметрами для размера графика и функцию `sns.lineplot` для построения графика. По оси X колонка «месяц», по оси Y колонка «продажи» и для переменной `hue` данные из колонки «год», которая создаёт линии по данным с разными цветами (рис.12).

```
plt.figure(figsize=(15,9))
ax = sns.lineplot (data=df, x="месяц", y="продажи", hue ="год", ci= 15)
```

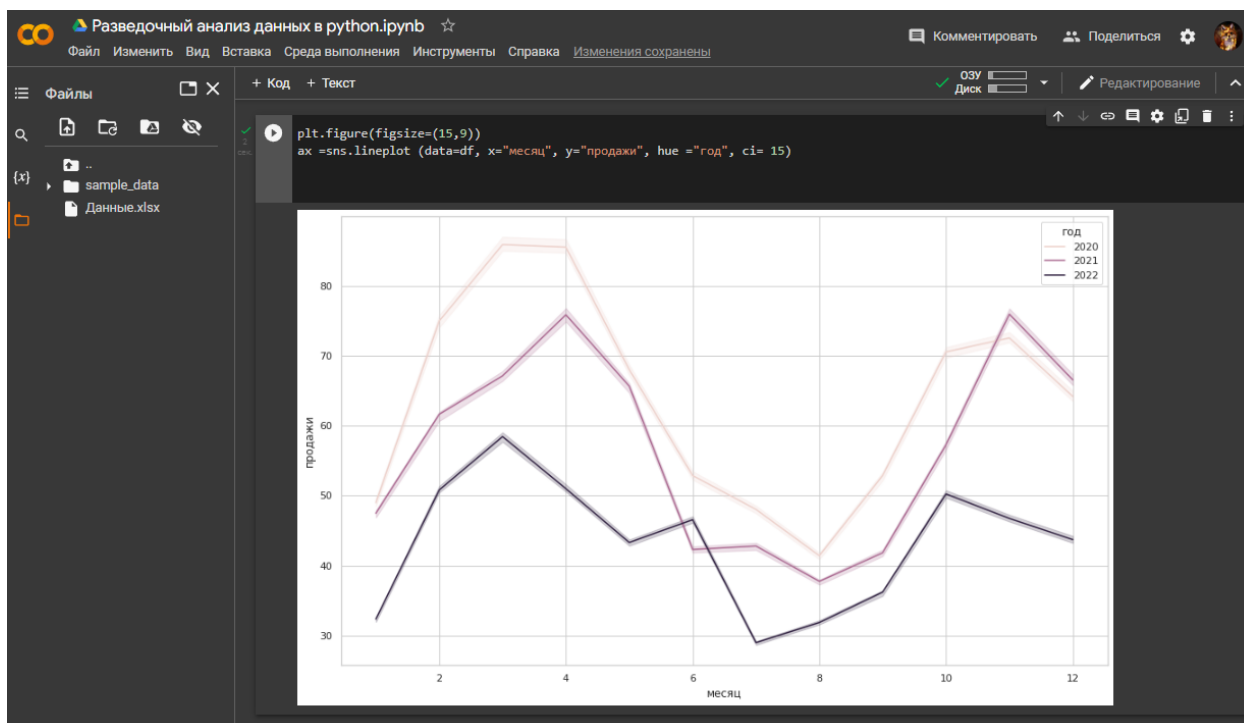


Рисунок 12 – График «Динамика продаж»

На данном графике показана динамика продаж по трем годам. Самые высокие продажи были в 2020 г. в третьем и четвертом месяце. Самые низкие продажи были в 2022 г. в июне. В 2020 г. с января по март продажи росли, с

апреля по июль продажи снижались, а с августа снова поднялись до ноября, и затем пошел спад продаж. В 2021 г. с января по апрель продажи повышались, затем снижались с апреля по август и снова стремительно росли до ноября, превысив в ноябре больше чем за 2020г. По данным графика видно, что самый низкие показатели продаж были весь 2022 г. по сравнению с предыдущими годами.

4 Выводы

Таким образом, в данной статье был рассмотрен разведочный анализ данных и процесс построение графиков на основе данных на языке программирования Python. Данное исследование может быть использовано методическим пособием в учебной деятельности, а также для пользователей, которые работают с анализом статистических данных.

Библиографический список

1. <https://drive.google.com/drive/folders/15VzLpjThE6DC1y9Z2NbcDKbrIAplnAwG?usp=sharing>
2. Кизянов А.О. Использование палитр seaborn в построении графиков на языке программирования python // Постулат. 2018. № 7 (33). С. 18.
3. Шутова О.А., Грамаков Д.А. Сравнительный анализ библиотек визуализации данных для задачи обработки образовательной информации // Материалы II Международной междисциплинарной конференции. Москва, 2021. С. 190-194.
4. Кудринская О.В. Визуализация данных с использованием возможностей языка python // Сборник научных статей ежегодной научно-практической конференции. 2021. С. 176-180.
5. Насруева М.А. Визуализация аналитических данных с использованием языка программирования python // Сборник научных статей по материалам V Международной научно-практической конференции. Уфа, 2021. С. 25-36.
6. Ивина О.А. Визуализация статистических характеристик данных в python // Инновации в информационных технологиях, машиностроении и автотранспорте. Сборник материалов V Международной научно-практической конференции. Кемерово, 2021. С. 28-31.
7. Aurachman R. Visualization of google mobility data for provinces in Indonesia using seaborn python programming package // Journal of Physics: Conference Series. 2021. Т. 1833. №. 1. С. 012002.