

Лабораторная работа 7

Типы визуализаций. Создание визуализаций с использованием Python

Цель работы: Освоить базовые навыки по визуализации данных.

Визуализация данных - это графическое представление информации. Визуализация данных переводит сложные наборы данных в визуальные форматы, которые легче понять человеческому мозгу. Основная цель визуализации данных заключается в том, чтобы сделать данные более доступными и легче интерпретируемыми.

Инструмент, который будем использовать для визуализации данных:

- Google Collab: <https://colab.research.google.com/>

Набор данных для использования в лабораторной работе в качестве примера:

- Набор данных – population.csv, данный набор загружается в Google Collab для визуализации данных.

Все наборы данных для работы будут загружены на курс!

Основной файл для работы в качестве примера: **population.csv**

Как создать хорошие визуализации?

- Выбрать диаграмму, правильно отображающую конкретный тип данных;
- Использовать цвета так, чтобы они помогли пользователю быстрее оценить ключевые показатели;
- Расставить акценты в формах, размерах, с помощью насыщенности цветов, которые подчеркнут самое важное;
- Выбрать такой дизайн визуализаций, который не будет препятствовать пониманию связи в данных или исказить их.

Для этого необходимо:

1. Импорт библиотеки. Для визуализации нам понадобится не одна библиотека. Библиотеки визуализации данных Python (Matplotlib, Seaborn) используемые в этом модуле:

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

- import seaborn as sns

- import numpy as np

2. Считывание csv файла - `df = pd.read_csv('population.csv', thousands='r',')`

Используя Google Collab загрузите библиотеки и прочитайте файл, для этого войдите в Google Collab – Назовите новый файл, нажмите на Содержание, а затем на Папку, чтобы загрузить файл `population.csv`. Когда файл загружен, теперь где галочка напишем код, для загрузки библиотеки и считывания данных. Весь код для работы, вы можете копировать с лабораторной работы. Каждый раз кода вы хотите написать код, нажимайте, на слово КОД в Меню.

The screenshot shows the Google Colab interface. In the left sidebar, the 'Files' section shows a file named 'population.csv' uploaded from the local disk. The main code area contains the following code:

```
[1] import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

df = pd.read_csv('population.csv', thousands='r',)

[3] df
```

The output of the code is a DataFrame with the following structure:

	Районы	Общее население	мужчины	женщины	городское население	городское население мужчины	городское население женщины	сельское население	сельское население мужчины	сельское население женщины
0	Semei c.a.	311694	149017	162677	311604.0	148967.0	162637.0	90.0	50.0	40.0
1	Kurchatov c.a.	10217	4987	5230	10217.0	4987.0	5230.0	NaN	NaN	NaN
2	Abai district	13756	6906	6850	NaN	NaN	NaN	13756.0	6906.0	6850.0
3	Aksuat district	19128	9826	9302	NaN	NaN	NaN	19128.0	9826.0	9302.0
4	Ayagoz district	64968	32699	32269	41820.0	20717.0	21103.0	23148.0	11982.0	11166.0
...	...	...	...	...	...	...	...	...	...	...
222	Abay district	230468	111427	119041	230468.0	111427.0	119041.0	NaN	NaN	NaN
223	AlFarabi district	268684	127103	141581	268684.0	127103.0	141581.0	NaN	NaN	NaN

Типы визуализаций

Теперь визуализируем данные:

1. Гистограмма распределения Общего населения

```
plt.figure(figsize=(10, 5))
```

```
sns.histplot(df["Общее население"], bins=20, kde=True, color='skyblue')
```

```
plt.xlabel("Общее население")
```

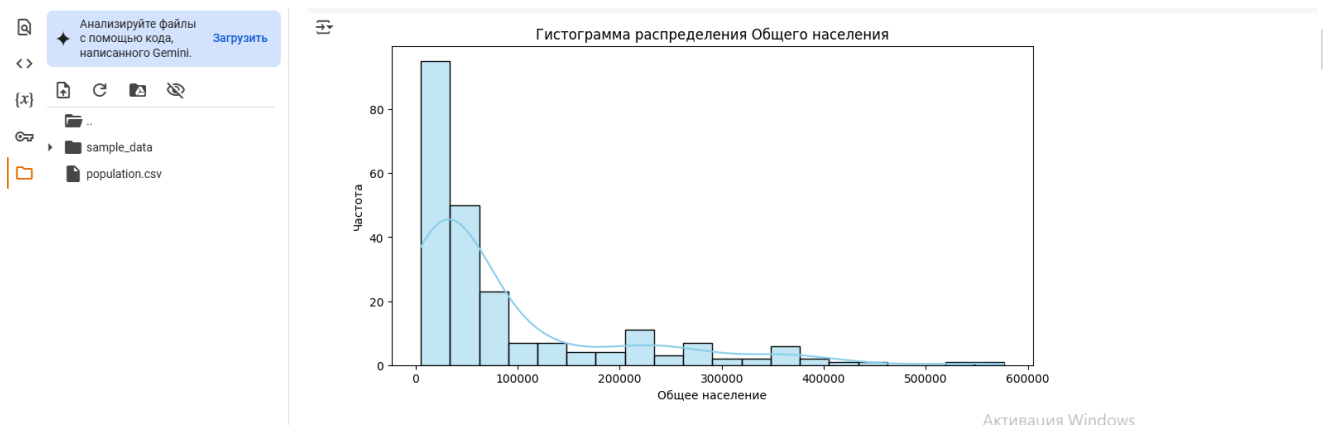
```
plt.ylabel("Частота")
```

```
plt.title("Гистограмма распределения Общего населения")
```

```
plt.show()
```

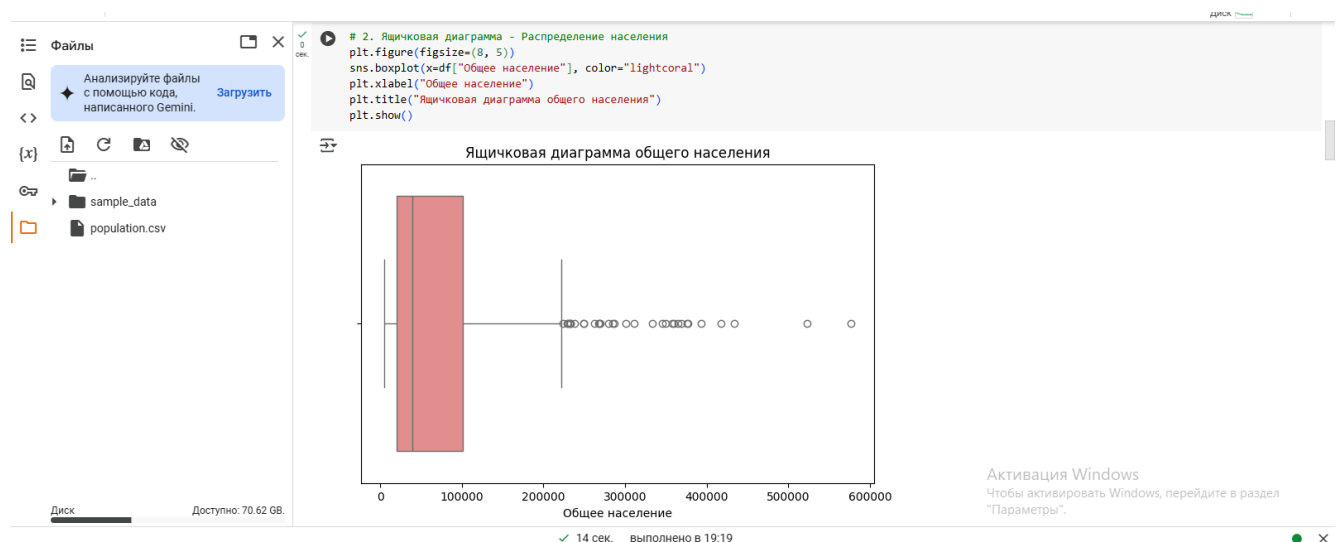
The screenshot shows the Google Colab interface with the following code in the main area:

```
# 1. Гистограмма распределения Общего населения
plt.figure(figsize=(10, 5))
sns.histplot(df["Общее население"], bins=20, kde=True, color='skyblue')
plt.xlabel("Общее население")
plt.ylabel("Частота")
plt.title("Гистограмма распределения Общего населения")
plt.show()
```



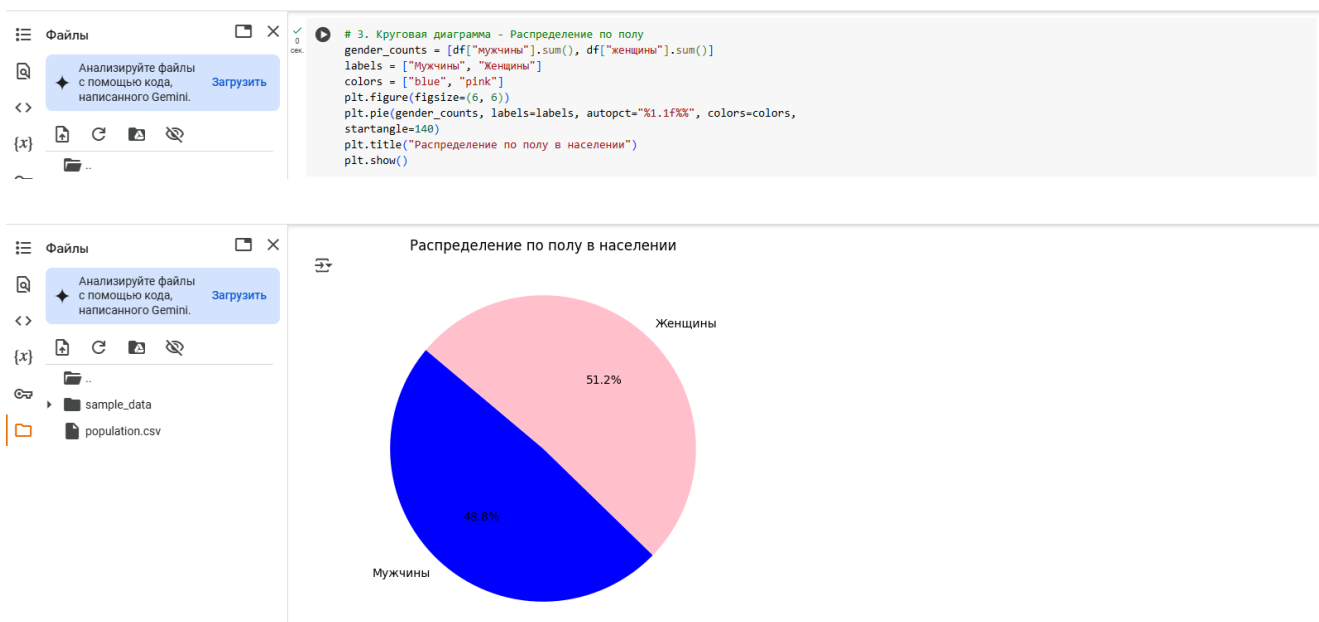
2. Ящичковая диаграмма - Распределение населения

```
plt.figure(figsize=(8, 5))
sns.boxplot(x=df["Общее население"], color="lightcoral")
plt.xlabel("Общее население")
plt.title("Ящичковая диаграмма общего населения")
plt.show()
```



3. Круговая диаграмма - Распределение по полу

```
gender_counts = [df["мужчины"].sum(), df["женщины"].sum()]
labels = ["Мужчины", "Женщины"]
colors = ["blue", "pink"]
plt.figure(figsize=(6, 6))
plt.pie(gender_counts, labels=labels, autopct="%1.1f%%", colors=colors,
startangle=140)
plt.title("Распределение по полу в населении")
plt.show()
```



4. Столбчатая диаграмма - Сравнение городского и сельского населения

```
urban_population = df["городское население"].sum()
```

```
rural_population = df["сельское население"].sum()
```

```
plt.figure(figsize=(8, 5))
```

```
sns.barplot(x=["Городское население", "Сельское население"],
```

```
y=[urban_population, rural_population], palette=["green", "brown"])
```

```
plt.ylabel("Общее население")
```

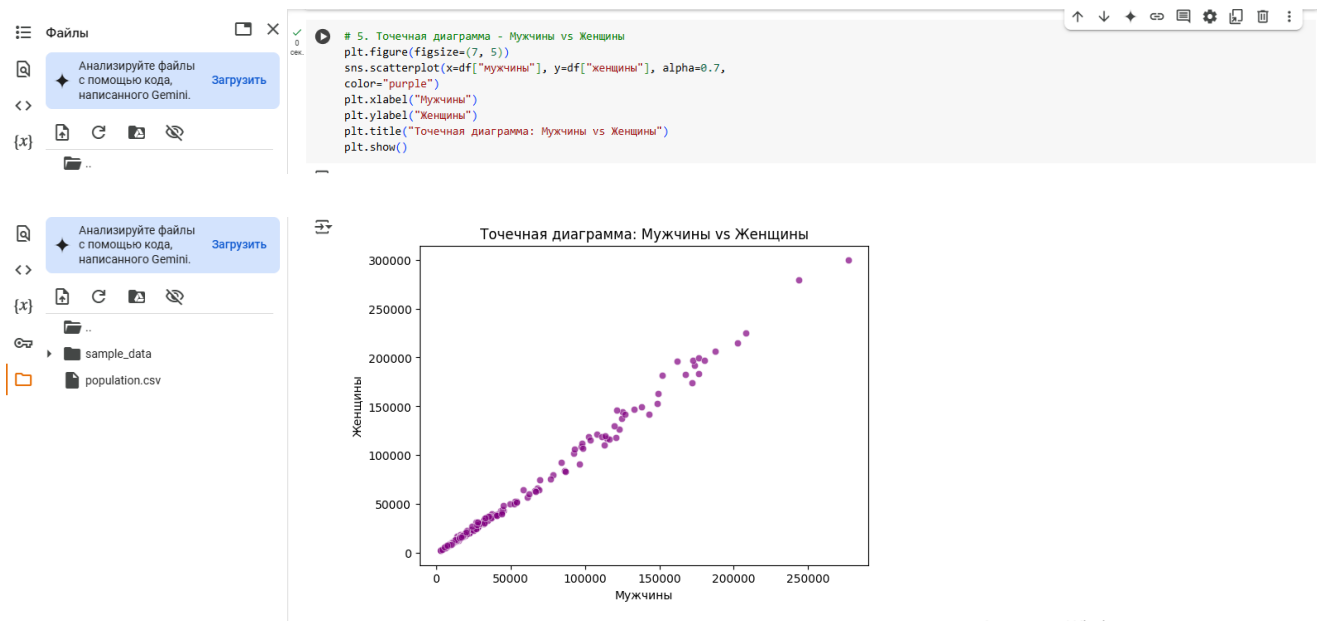
```
plt.title("Сравнение городского и сельского населения")
```

```
plt.show()
```



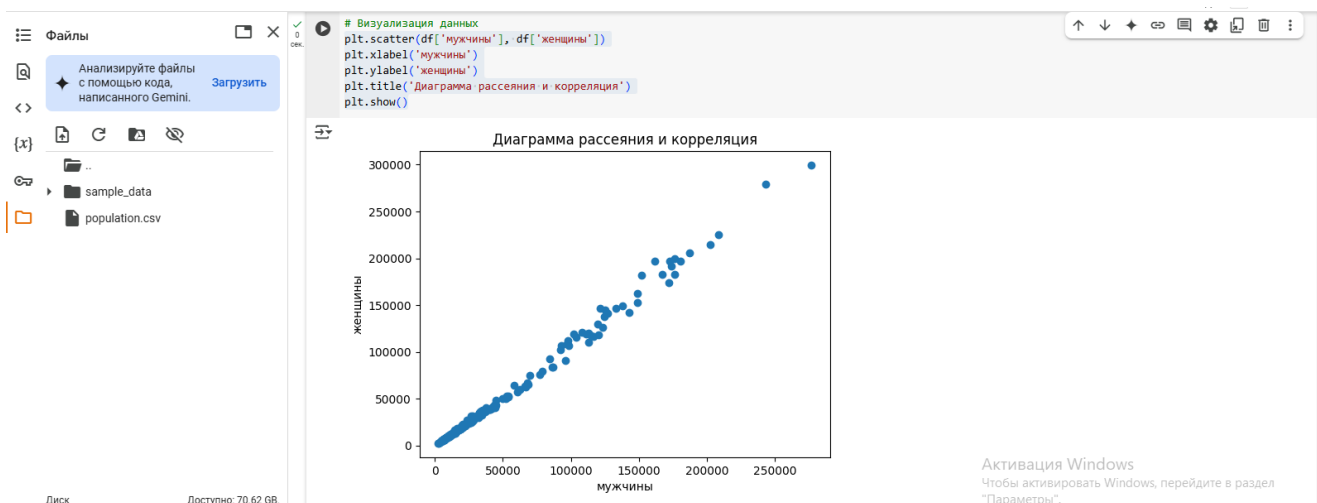
5. Точечная диаграмма - Мужчины vs Женщины

```
plt.figure(figsize=(7, 5))
sns.scatterplot(x=df["мужчины"], y=df["женщины"], alpha=0.7,
color="purple")
plt.xlabel("Мужчины")
plt.ylabel("Женщины")
plt.title("Точечная диаграмма: Мужчины vs Женщины")
plt.show()
```



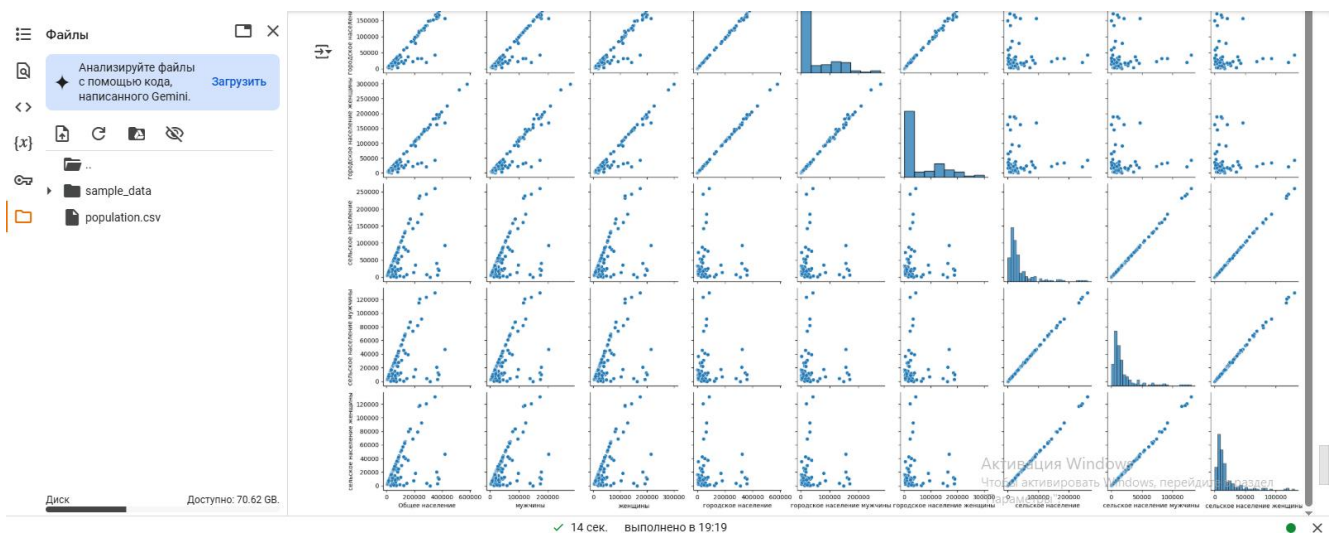
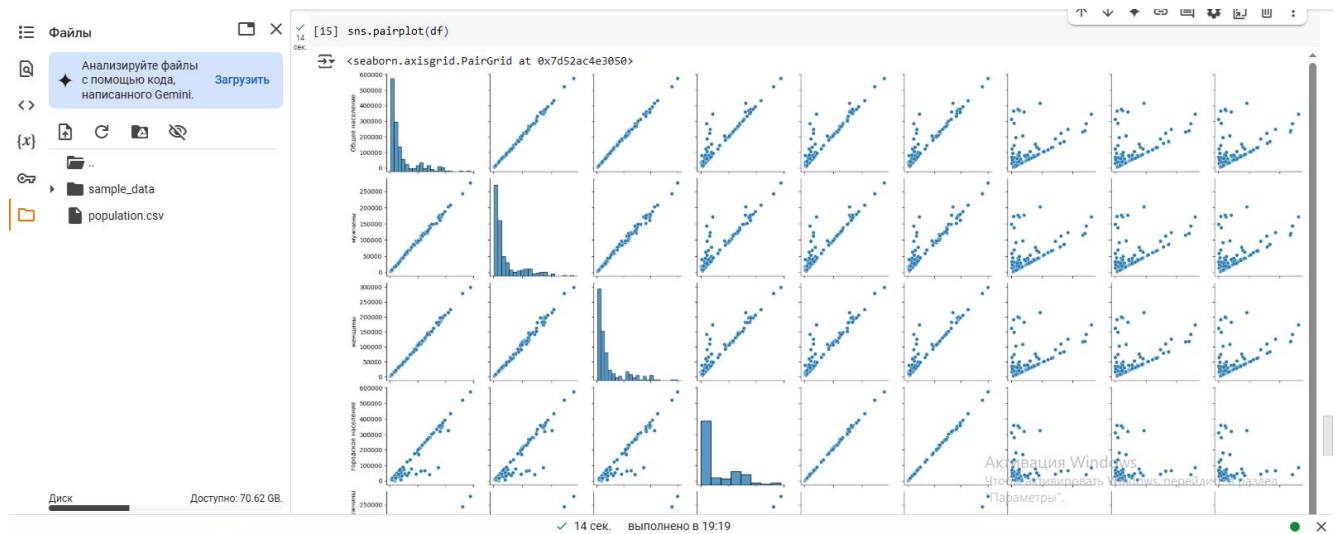
6. Визуализация данных. Корреляционный анализ

```
plt.scatter(df['мужчины'], df['женщины'])
plt.xlabel('мужчины')
plt.ylabel('женщины')
plt.title('Диаграмма рассеяния и корреляция')
plt.show()
```



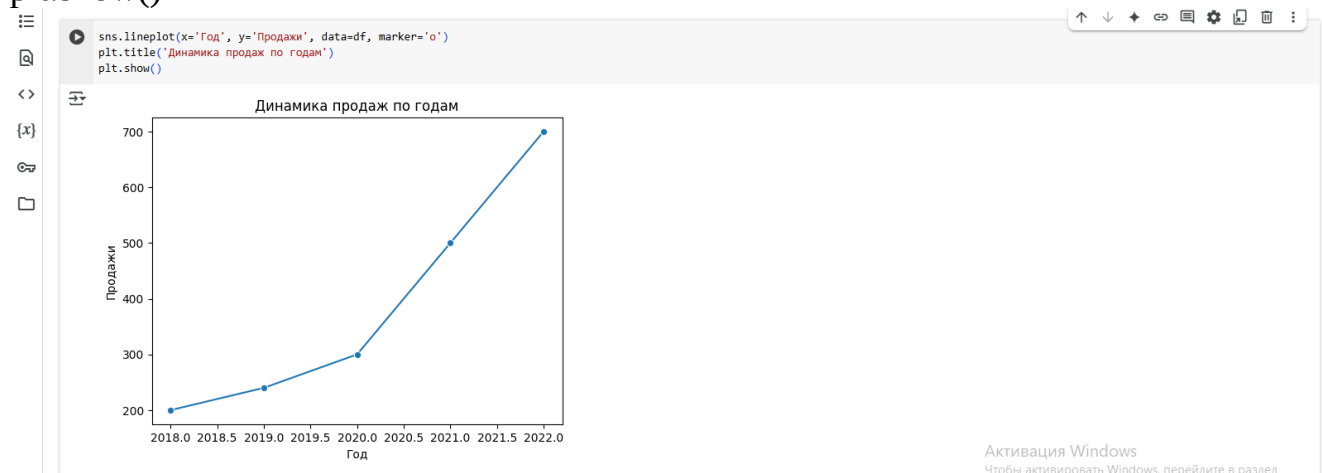
7. Визуализация данных. Визуализация зависимостей всех переменных.

sns.pairplot(df) — используя этот код вы получите зависимости по всем показателям в одной визуализации.

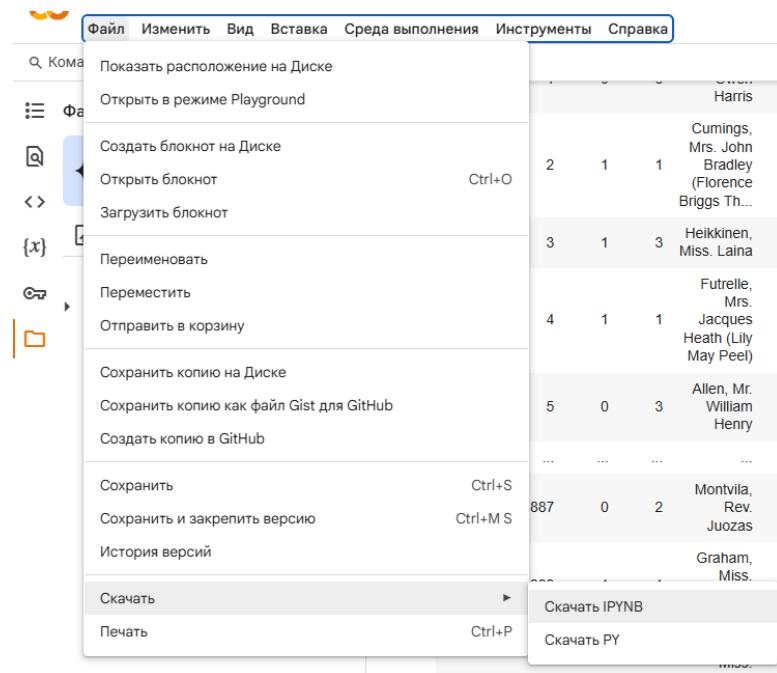


8. Линейный график. Линейный график построен с другими данными для примера.

```
sns.lineplot(x='Год', y='Продажи', data=df, marker='o')  
plt.title('Динамика продаж по годам')  
plt.show()
```



В конце, когда все визуализации выполнены, необходимо сохранить файл. Для этого в Меню выберите Файл-Скачать-Скачать IPYNB. Дайте имя файлу и сохраните. Именно этот файл необходимо отправлять на проверку.



P.S.: Обратите внимание, например, когда вы работаете в Google Collab, а затем прервали работу и решили продолжить на следующий день, открыв свой код (если, что весь код также автоматически сохраняется на Гугл Диске в папке Google Collab), вам снова сначала надо загрузить ваш файл population.csv, ну или тот файл с которым вы работаете, название может меняться. И после загрузки надо опять прогнать весь код Выбрав в Меню-Среда выполнения-Выполнить все или вы вручную должны прогнать весь код нажимая на него и получить галочки. Как удобнее выбираете сами.

Самостоятельное практическое задание.

Цель: Научиться строить различные виды графиков с использованием библиотек Matplotlib и Seaborn для анализа данных.

ЗАДАНИЯ

Используйте библиотеку pandas для загрузки набора данных Titanic (файл titanic.csv). Это уже знакомый для вас файл, с которым вы ранее работали на лабораторной 6. Файл загружен на платформу ДОТ под этим заданием.

1. Подготовка данных

1.1. Загрузите данные о пассажирах Титаника:

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

# Загрузка данных
df = pd.read_csv("titanic.csv")
df.head()
```

1.2. Проверьте данные на пропущенные значения и обработайте их (например, удалите или замените).

Обратитесь к предыдущей работе, где вы уже убрали пропущенные значения и сделайте их обработку.

2. Построение визуализаций

Создайте визуализации, отвечающие на следующие вопросы.

2.1. Гистограмма распределения возрастов пассажиров.

Используйте `plt.hist()` или `sns.histplot()`

2.2. Ящикковая диаграмма (boxplot) для сравнения возрастов пассажиров по классам кают.

Используйте `sns.boxplot(x='класс', y='Возраст', data=df)`

2.3 Круговая диаграмма для визуализации доли выживших и погибших.

Используйте метод `plt.pie()`

2.4 Линейный график зависимости количества пассажиров от возраста.

Используйте `sns.lineplot()`

3. Дополнительный анализ по визуализации

3.1. Попробуйте создать парную диаграмму `sns.pairplot()` для нескольких числовых столбцов.

3.2. Постройте точечную диаграмму `scatterplot` с трендовой линией для анализа взаимосвязи любых двух переменных.

4. Требования к оформлению графиков

4.1. Добавьте заголовки и подписи осей `plt.title()`, `plt.xlabel()`, `plt.ylabel()`

В каком виде сдавать работу: 1. Выполненное задание рекомендуется выполнить в Google Colab. После завершения работы необходимо скачать блокнот (блокнот .ipynb) с Google Colab и сдать его. 2. Также необходимо сделать отчет со скринами, как вы делали, с комментариями и ответами по заданию по каждому пункту. Скриншоты должны соответствовать требованиям на курсе, быть полными с системным временем выполнения и датой. Если скриншоты не соответствуют требованиям, оценка снижается на 50%.

В итоге вы должны прикрепить два файла, один файл в формате *.docx, один файл в формате *.ipynb, не прикреплять файлы в архиве, т.к. не всегда архивы открываются.