Министерство науки и образования РФ

Федеральное государственное бюджетное учреждение

высшего образования

**«Тверской государственный технический университет»**

(ТвГТУ)

Кафедра программного обеспечения

**Отчет по лабораторной работе №1**

По дисциплине: «Анализ больших данных»

Тема: «Реляционные данные. Исследовательский анализ данных.

Построение визуализаций данных OLAP»

|  |
| --- |
| Выполнила:  студентка группы  Б.ПИН.РИС - 21.06  Александрова М.И. |
| Проверила:  старший преподаватель  кафедры ПО  Корнеева Е.И. |

Тверь 2025

Оглавление

[Введение 3](#_Toc188393677)

[Описание проделанной работы 4](#_Toc188393678)

[1. Создание базы данных: 4](#_Toc188393679)

[2. Подключение к базе данных: 4](#_Toc188393680)

[3. Описание данных: 4](#_Toc188393681)

[Признаки: 5](#_Toc188393682)

[4. Одномерный анализ: 5](#_Toc188393683)

[Гистограмма 1: 5](#_Toc188393684)

[Гистограмма 2: 6](#_Toc188393685)

[5. Многомерный анализ: 8](#_Toc188393686)

[График 1: 8](#_Toc188393687)

[График 2: 9](#_Toc188393688)

[Общий вывод по графикам: 11](#_Toc188393689)

[Таблица Pandas 12](#_Toc188393690)

[Краткий вывод по работе 13](#_Toc188393691)

[Ссылки на используемые материалы. Документация 14](#_Toc188393692)

[Листинг кода 15](#_Toc188393693)

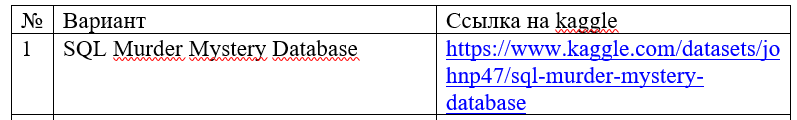
# **Введение**

Цель: Провести первичный анализ реляционной базы данных с помощью Python, pandas и библиотеки для подключения к БД. Построить визуализации данных OLAP.

Формулировка задач:

1. Создать базу из файлов или скриптов согласно варианту. Выбрать сложность задания.
2. Подключиться к базе данных из Python.
3. Сделать описание данных: из каких таблиц и полей состоят данные? Какие из данных являются признаками? К какому типу данных и к какой шкале относятся признаки?
4. Одномерный анализ. Построить гистограммы распределения количественных признаков, которые важны для задачи. Сделать вывод после построения. Какое распределение для каждого из признаков? Почему, по вашему мнению, признаки важны для задачи?
5. Многомерный анализ. Построить графики из 3-4 признаков. Выбрать категориальные (номинальные, порядковые или бинарные) признаки и количественные. Что получилось на каждом графике? Почему, по вашему мнению, признаки важны для задачи?

Вариант и условия задачи:



Сложность: Medium

**Сложность: Medium**

* Реализовать Jupiter Notebook или консольное приложение для выполнения задания.
* В качестве базы данных выбрать PostgreSQL. Создать базу из файлов или скриптов согласно варианту.
* Подключиться к базе данных из python.
* Сделать описание данных и выводы по заданию.
* Соединить признаки в 1 таблицу pandas для анализа.
* Одномерный анализ. Построить 2 гистограммы распределения количественных признаков, которые важны для задачи и сделать их описание по заданию.
* Многомерный анализ. Построить 2 графика из 3-4 признаков и сделать их описание по заданию.

**Ссылка на репозиторий с программной реализацией**:   
<https://github.com/milana-cat/BigData>

**Описание проделанной работы**

1. **Создание базы данных:**

* База данных создана в PostgreSQL из предоставленных файлов.
* Таблица Nashville\_Housing содержит данные по недвижимости.

1. **Подключение к базе данных:**

* Подключение реализовано через библиотеку SQLAlchemy.
* Данные из таблицы загружены в DataFrame для дальнейшего анализа.

1. **Описание данных**:

* Данные состоят из 9 таблиц.
* . Признаки разделены на количественные и категориальные. Определены шкалы для каждого признака (номинальная, порядковая, количественная).
  + Person
    - Id – Уникальный идентификатор человека (числовой, номинальная шкала)
    - Name – Фамилия и имя человека (текстовый, номинальная шкала)
    - license\_id – номер водительских прав (числовой, номинальная шкала)
    - address\_number – номер дома (числовой, номинальная шкала)
    - address\_street\_name – название улицы (текстовый, номинальная шкала)
    - ssn – номер социального страхования (числовой, номинальная шкала)
  + Get\_fit\_now\_member
    - Id ‒ Уникальный идентификатор членства (числовой, номинальная шкала)
    - person\_id ‒ Уникальный идентификатор человека (числовой, номинальная шкала)
    - name – Фамилия и имя члена (текстовый, номинальная шкала)
    - membership\_start\_date – Дата получения членства (числовой, порядковая шкала)
    - membership\_status – статус членства (строковый, номинальная шкала)
  + Drivers\_license
    - Id – Номер водительских прав (числовой, номинальная шкала)
    - Age – возраст водителя (числовой, количественная шкала)
    - Height – рост водителя (числовой, количественная шкала)
    - eye\_color – цвет глаз (строковый, номинальная шкала)
    - hair\_color – цвет волос (строковый, номинальная шкала)
    - gender – пол (строковый, номинальная шкала)
    - plate\_number – номер автомобиля (строковый, номинальная шкала)
    - car\_make – марка автомобиля (строковый, номинальная шкала)
    - car\_model – модель автомобиля (строковый, номинальная шкала)
  + Income
    - Ssn – номер социального страхования (числовой, номинальная шкала)
    - annual\_income – годовой доход (числовой, количественная шкала)

### **Признаки:**

Категориальные признаки:

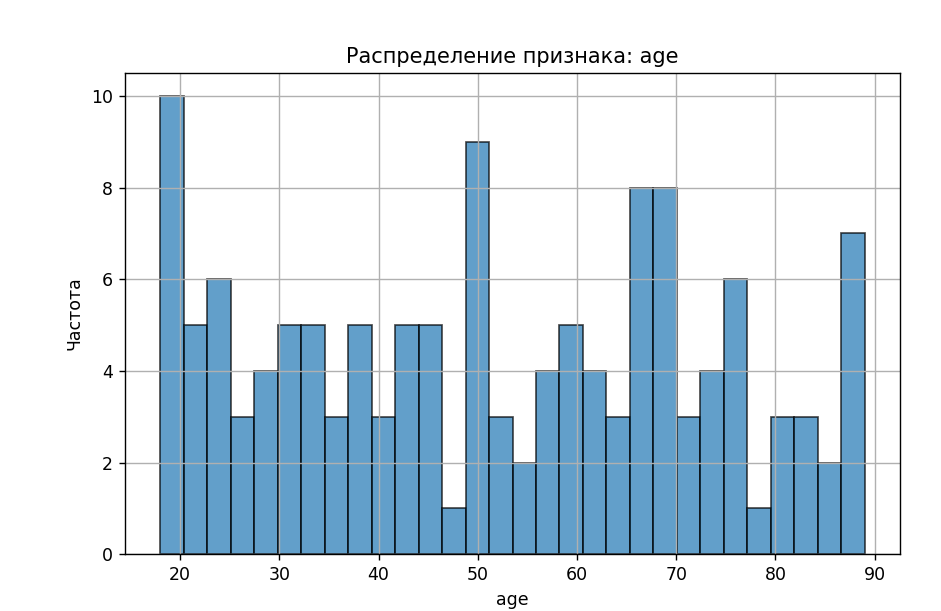
* Номинальные: Id, Name, license\_id, address\_number, address\_street\_name, ssn, person\_id, plate\_number, membership\_status, eye\_color, hair\_color, gender, car\_make, car\_model
* Порядковые: membership\_start\_date,

Количественные признаки:

* Age, Height, annual\_income

1. **Одномерный анализ:**

### Гистограмма 1:



Age: Это значения признака Age (количественный), представляющего возраст водителей автомобилей.

Частота: Это количество водителей заданного возраста, которые попадают в соответствующий интервал на оси X.

**Распределение:**

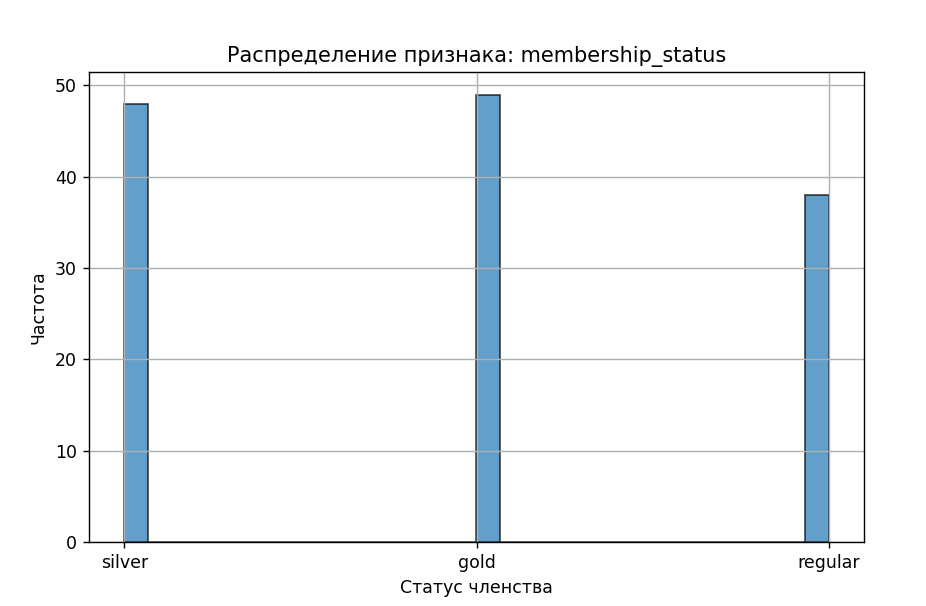
* Распределение признака Age достаточно равномерное. Наблюдается небольшая правосторонняя ассиметрия, в сторону более молодых водителей.

**Выводы по признаку:**

1. Тип распределения: Распределение возраста водителей (Age) характеризуется относительно равномерным распределением, что достаточно ожидаемо в этой области, поскольку водительские права могут получить все, кто старше 18 лет. Важность признака:

* Age позволяет оценить не только средний возраст водителей, но и проследить заинтересованность возрастных групп в получении водительских прав. Так, например, люди около 48 лет значительно реже имеют водительские права, чем 18-20-летние.
* Анализ распределения помогает выявить выбросы и аномалии, а также определить возрастной диапазон водителей, что важно для разработки моделей анализа транспортной системы.

### Гистограмма 2:



Статус членства: Это значения признака membership\_status, представляющего один из трех возможных статусов членства (regular, silver, gold).

Частота: Это количество участников с определенным статусом на оси X.

**Распределение:**

* Распределение является равномерным.
* Количество участников со статусом regular немного ниже среднего

**Выводы по признаку:**

Тип распределения:

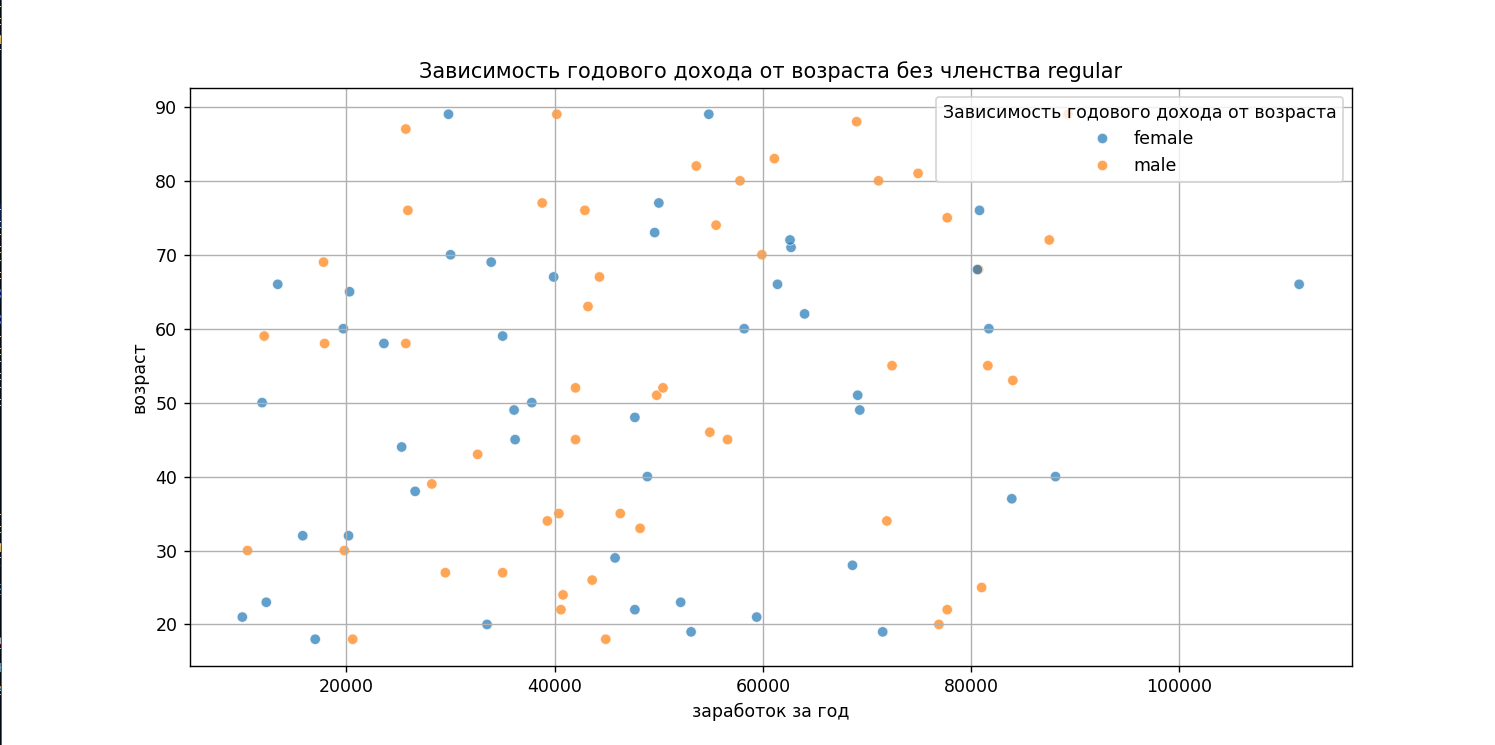
* Большинство людей в выборке имеют отличный от regular статус членства.

Важность признака:

* membership\_status — важный категориальный признак, который позволяет оценить количество членов и популярность каждой категории.
* Учет этого признака важен для определения целевой аудитории клуба и возможного продвижения более высоких статусов членства. (например, повышения стоимости gold, поскольку он наиболее популярен).

1. **Многомерный анализ:**

### График 1:



**Оси:**

Ось X (заработок за год): Это количественный признак, показывающийобщую сумму годового дохода в долларах.

Ось Y (Возраст): Это количественный признак, показывающий возраст людей.

**Цвет точек:**

* Каждая точка на графике представляет человека.
* Цвет точек отражает категориальный признак gender (пол).

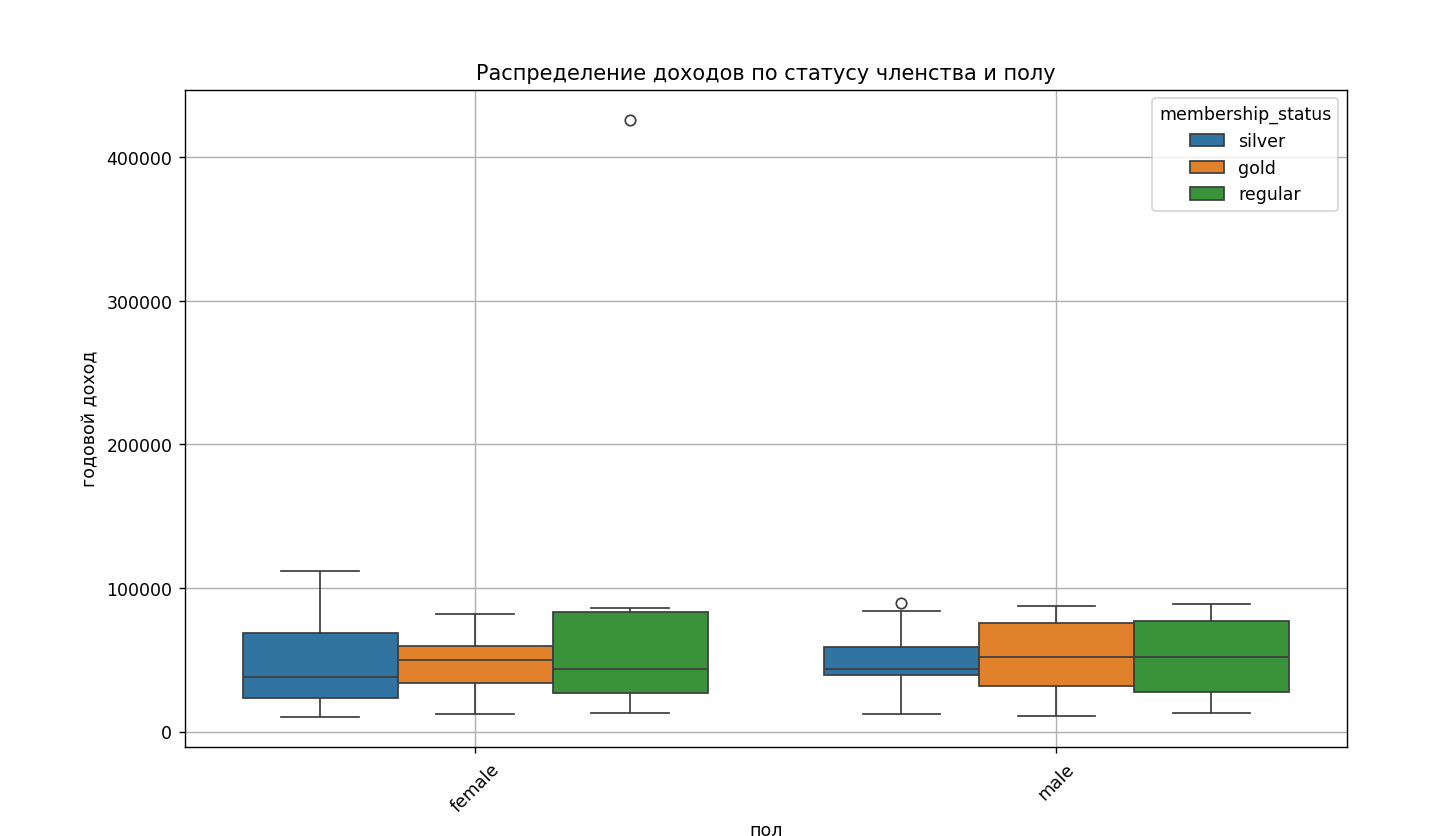
**Анализ:**

* Большинство людей со статусом членства gold или silver находятся в диапазоне от 20000 до 80000 долларов в год.
* При этом наблюдается зависимость роста дохода за год от возраста. Так, например, люди от 20 до 30 лет зарабатывают в среднем до 60 000 долларов в год, тогда как 70-80-ти летние зарабатывают в среднем от 40 000 до 80 000.
* Доходы мужчин и женщин во всех возрастных группах примерно одинаковы.

Вывод по графику:

Этот график позволяет оценить, как возраст и пол влияют на уровень доходов. Молодёжь зарабатывает в год меньше, чем люди более старшего возраста. В выборке так же можно заметить, что самыми богатыми являются в основном женщины от 35 до 80 лет. Этот график позволяет оценить не только уровень дохода людей, но и увидеть распределение доходов между мужчинами и женщинами, что, например, может быть полезно при разработке ценовой политики.

### График 2:



**Оси:**

* Ось X: Пол человека
* Ось Y: Годовой доход в долларах

Цвет боксплотов: определяет категориальный признак membership\_status (статус членства).

**Содержимое графика:**

* Боксплоты отображают распределение дохода и статусов членства между мужчинами и женщинами.
* Каждая коробка:

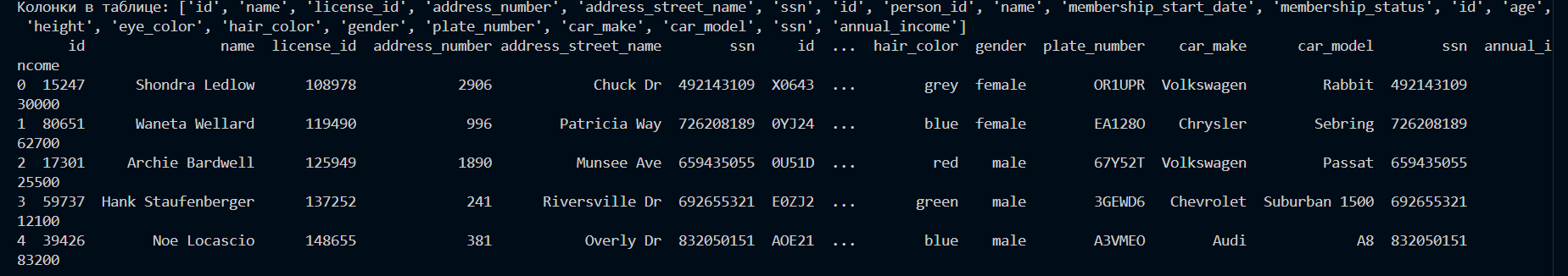
1. Верхняя и нижняя границы показывают квартильные значения (25% и 75%).
2. Горизонтальная линия внутри коробки — медиана цены.
3. «Усы» отображают диапазон данных, исключая выбросы.

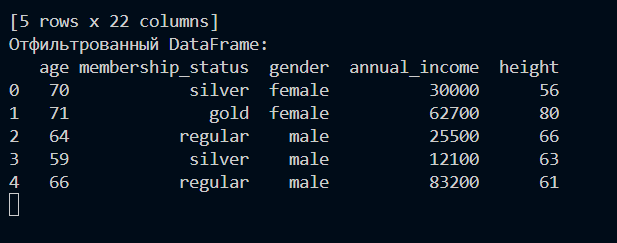
* Кружки отображают выбросы (ценовые аномалии), которые выходят за пределы статистически ожидаемого диапазона.

Вывод:

График показывает, что среди мужчин наиболее редко встречается статус regular, в то время как у женщин менее популярен статус gold. При этом, вопреки ожиданиям, что у членов статуса regular доходы будут в среднем ниже, чем у остальных, можно заметить, что они примерно на том же уровне.

# **Таблица Pandas**





# **Краткий вывод по работе**

**Описание программы:**

Реализована программа для анализа данных о недвижимости, хранящихся в базе данных PostgreSQL. Основная цель программы — провести исследовательский анализ данных (EDA), включая одномерный и многомерный анализ, с использованием Python и библиотек pandas, matplotlib, seaborn.

**Основные функции программы:**

1. Подключение к базе данных PostgreSQL:

* Программа подключается к базе данных с помощью библиотеки SQLAlchemy.
* Загружает данные из таблицы Nashville\_Housing в структуру pandas.DataFrame.

1. Одномерный анализ данных:

* Построены гистограммы для распределения количественных признаков: saleprice (цена продажи), acreage (площадь участка).
* Гистограммы позволяют понять, как распределяются данные, выявить выбросы и основные тенденции.

1. Многомерный анализ данных:

* Построены графики, отражающие взаимосвязь между несколькими признаками:

1. Зависимость saleprice (цены) от acreage (площади участка) с учётом landuse (типа использования земли) и yearbuilt (года постройки).
2. Распределение saleprice по yearbuilt и soldasvacant (продано как пустой участок).

* Графики демонстрируют, как количественные и категориальные признаки совместно влияют на целевой показатель (цена продажи).

**Вывод:**

Программа успешно выполняет задачи первичного анализа данных:

* Помогает выявить закономерности и зависимости в данных о недвижимости.
* Позволяет визуализировать ключевые признаки, влияющие на стоимость объектов.
* Может быть расширена для задач прогнозирования цен или анализа других характеристик.

Фильтровать данные для графика(для многомерного цвета). Колич. Данные оставить в столбцах, категориальные в таблицу. Код на функции раскидать

# **Ссылки на используемые материалы. Документация**

1. Клиент-серверные приложения Сокеты. Создание клиента // METANIT.COM URL:https://metanit.com/python/network/1.1.php (дата обращения: 15.01.2025).
2. Python Documentation contents // python.org URL: https://docs.python.org/3/contents.html (дата обращения: 15.01.2025).
3. Визуализация данных с matplotlib // NSU Programming URL: https://nsu-programming.github.io/textbook/python/plotting.html (дата обращения: 15.01.2025).

# **Листинг кода**

from sqlalchemy import create\_engine

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import matplotlib.ticker as mtick

def connect\_to\_db(host, port, user, password, database):

    """Подключение к базе данных PostgreSQL."""

    try:

        engine = create\_engine(f"postgresql://{user}:{password}@{host}:{port}/{database}?client\_encoding=utf8")

        print("Подключение успешно")

        return engine

    except Exception as e:

        print("Ошибка подключения:", e)

        exit()

def load\_data(engine, query):

    """Загрузка данных из базы данных в DataFrame."""

    try:

        df = pd.read\_sql\_query(query, engine)

        print("Данные успешно загружены в DataFrame")

        print("Колонки в таблице:", df.columns.tolist())

        return df

    except Exception as e:

        print("Ошибка выполнения запроса:", e)

        exit()

def filter\_columns(df, columns):

    """Фильтрация DataFrame: выбор нужных колонок."""

    filtered\_df = df[columns]

    print("Отфильтрованный DataFrame:")

    print(filtered\_df.head())

    return filtered\_df

def plot\_histograms(df, features):

    """Построение гистограмм для выбранных признаков."""

    for feature in features:

        if feature in df.columns:

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

            plt.hist(df[feature].dropna(), bins=30, edgecolor="k", alpha=0.7)

            plt.title(f"Распределение признака: {feature}")

            plt.xlabel("Площадь участка (акры)" if feature == "acreage" else feature)

            plt.ylabel("Частота")

            if feature == "saleprice":

                ax = plt.gca()

                ax.xaxis.set\_major\_formatter(mtick.StrMethodFormatter('{x:,.0f}'))  # Формат в виде тысяч

            plt.grid(True)

            plt.show()

        else:

            print(f"Признак '{feature}' не найден в данных.")

def plot\_scatter\_no\_single\_family(df, x, y, hue):

    """Построение scatter-графика без категории SINGLE FAMILY."""

    filtered\_data = df[df[hue] != "SINGLE FAMILY"]

    plt.figure(figsize=(12, 6))

    sns.scatterplot(data=filtered\_data, x=x, y=y, hue=hue, alpha=0.7)

    plt.title("Зависимость saleprice от acreage без категорий SINGLE FAMILY")

    plt.xlabel("Площадь участка (акры)")

    plt.ylabel("Цена продажи")

    ax = plt.gca()

    ax.yaxis.set\_major\_formatter(mtick.StrMethodFormatter('{x:,.0f}'))  # Форматирование оси Y

    plt.grid(True)

    plt.legend(title="Тип использования земли")

    plt.show()

def plot\_boxplot\_decades(df, x, y, hue):

    """Построение boxplot-графика по десятилетиям."""

    df["yearbuilt"] = pd.to\_numeric(df["yearbuilt"], errors="coerce")

    df["yearbuilt\_decade"] = (df["yearbuilt"] // 10) \* 10

    plt.figure(figsize=(14, 8))

    sns.boxplot(data=df, x="yearbuilt\_decade", y=y, hue=hue, palette="Set2")

    plt.title("Распределение saleprice по yearbuilt (десятилетия) и soldasvacant")

    plt.xlabel("Десятилетие постройки")

    plt.ylabel("Цена продажи")

    ax = plt.gca()

    ax.yaxis.set\_major\_formatter(mtick.StrMethodFormatter('{x:,.0f}'))  # Форматирование оси Y

    plt.xticks(rotation=45)

    plt.legend(title="Продано как пустой участок")

    plt.grid(True)

    plt.show()

# Основной поток выполнения

if \_\_name\_\_ == "\_\_main\_\_":

    # Данные для подключения

    host = "povt-cluster.tstu.tver.ru"

    port = 5432

    user = "mpi"

    password = "135a1"

    database = "db\_housing"

    # Подключение к базе данных

    engine = connect\_to\_db(host, port, user, password, database)

    # Загрузка данных

    query = 'SELECT \* FROM "Nashville\_Housing" LIMIT 1000'  # Ограничиваем выборку

    df = load\_data(engine, query)

    # Фильтрация данных

    important\_columns = ["landuse", "soldasvacant", "saleprice", "acreage", "yearbuilt"]

    filtered\_df = filter\_columns(df, important\_columns)

    # Одномерный анализ

    plot\_histograms(filtered\_df, ["saleprice", "acreage"])

    # Многомерный анализ

    plot\_scatter\_no\_single\_family(filtered\_df, "acreage", "saleprice", "landuse")

    plot\_boxplot\_decades(filtered\_df, "yearbuilt", "saleprice", "soldasvacant")