Министерство науки и образования РФ

Федеральное государственное бюджетное учреждение

высшего образования

**«Тверской государственный технический университет»**

(ТвГТУ)

Кафедра программного обеспечения

**Отчет по лабораторной работе №4**

По дисциплине: «Новые технологии в РПС»

Тема: «Анализ и визуализация данных»

|  |
| --- |
| Выполнил:  студент группы  М.ПИН.РИС - 23.08  Тимофеев Александр Сергеевич |
| Проверила:  старший преподаватель  кафедры ПО  Корнеева Е.И. |

Тверь 2023

**Ссылка на GoogleNotebook:**

**https://colab.research.google.com/drive/1L8GM0V83I-1W-wcp4G6OZdL88fZGJvfa?usp=sharing**

**Постановка задачи:**

*Вариант 1*

1. С использованием средств библиотеки Pandas прочитать все данные из csv-файла и сформировать из прочитанных данных DataFrame (в DataFrame будет несколько столбцов, в зависимости от варианта, например, у варианта 1 - Дата, Курс).
2. Произвести именование колонок сформированного DataFrame следующим образом: нижний регистр, слова соединены через "\_", английский язык. Если изначально было сделано так, пункт пропускается.
3. Произвести проверку на наличие невалидных значений в колонках (NaN, None и т.п.). В случае нахождения выполнить обработку таких значений (isnan().mean(), isnull().sum()).
4. Добавить в DataFrame столбцы, в которых будет содержаться информация об отклонении от медианы и среднего значения курса (сначала медиана, среднее, затем рассчетный столбец)
5. С использованием Pandas вычислить статистическую информацию (describe, \* распределения данных на графике boxplot) для столбцов, содержащих информацию о курсе и отклонениях (из пункта 4). Учесть выбросы (отклонения).
6. Написать функцию, которая на вход принимает DataFrame и значение отклонения от среднего значения курса, а возвращает отфильтрованный по значению отклонения от среднего значения курса DataFrame. Условие фильтрации - в новый DataFrame включаются те строки, для которых значение отклонения от курса ≥ заданного значения.
7. Написать функцию, которая на вход принимает DataFrame, начальную и конечную дату, а возвращает отфильтрованный по датам DataFrame. Условие фильтрации - в новый DataFrame включаются те строки, для которых дата удовлетворяет следующему условию: начальная дата ≤ дата ≤ конечная дата.
8. Выполнить группировку DataFrame по месяцу с вычислением среднего значения курса. Groupby
9. С использованием средств библиотеки matplotlib или seaborn нарисовать графики изменения курса за весь период. Графики и оси должны иметь соответствующие подписи, подписать значения.
10. Написать функцию, которая на вход принимает DataFrame и месяц. С использованием средств библиотеки matplotlib или seaborn нарисовать графики изменения курса, а также медиану и среднее значение за указанный месяц (дополнительно отметить на графике).

**Код программы (Python):**

Библиотеки главной формы:

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from google.colab import drive

Привинчивание Гугл Диска:

drive.mount('/content/drive')

Константы:

CURR\_DIR = '/content/drive/MyDrive'

CURRENCY\_FIELDS = ['date', 'nominal', 'value', 'vunitRate']

Функция чтения датасета из файла:

def create\_dataset\_from\_files(files: list) -> pd.DataFrame:

    df = pd.DataFrame()

    for file in files:

        data = pd.read\_csv(file)

        data['date'] = pd.to\_datetime(data['date'])

        df = df.append(data, ignore\_index=True)

    return df

Чтение и вывод датасета:

df = create\_dataset\_from\_files([CURR\_DIR + '/outputs/USD\_19910101\_20231231.csv'])

print(df)

Вывод информации о датасете:

df.info(memory\_usage='deep')

Поиск NaN, Null:

print(df.isnull().sum())

Замена NaN, Null значений:

df.fillna(df.mean(), inplace=True)

Расчет и добавление медианы и среднего значения:

median\_value = df['vunit\_rate'].median()

mean\_value = df['vunit\_rate'].mean()

df['deviation\_from\_median'] = df['vunit\_rate'] - median\_value

df['deviation\_from\_mean'] = df['vunit\_rate'] - mean\_value

print(df.head())

Статистическая информация:

print(df[['vunit\_rate', 'deviation\_from\_median', 'deviation\_from\_mean']].describe())

График отклонений:

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

plt.boxplot([df['vunit\_rate'], df['deviation\_from\_median'], df['deviation\_from\_mean']], labels=['vunit\_rate', 'deviation\_from\_median', 'deviation\_from\_mean'])

plt.title('График vunit\_rate и отклонений')

plt.show()

Отфильтрованные данные по отклонение:

def filter\_by\_deviation(df: pd.DataFrame, deviation\_value: float) -> pd.DataFrame:

    filtered\_df = df[df['deviation\_from\_mean'] >= deviation\_value]

    return filtered\_df

Отфильрованные данные по дате:

def filter\_by\_date(df: pd.DataFrame, start\_date: str, end\_date: str) -> pd.DataFrame:

    df['date'] = pd.to\_datetime(df['date'])

    filtered\_df = df[(df['date'] >= start\_date) & (df['date'] <= end\_date)]

    return filtered\_df

Группировка данных по месяцам:

df['date'] = pd.to\_datetime(df['date'])

df['month'] = df['date'].dt.month

grouped\_df = df.groupby('month')['vunit\_rate'].mean()

График изменения курса по дате:

df['date'] = pd.to\_datetime(df['date'])

plt.figure(figsize=(10, 6))

plt.plot(df['date'], df['vunit\_rate'])

plt.title('Изменение курса за весь период')

plt.xlabel('Дата')

plt.ylabel('Курс')

plt.show()

График изменение курса по месяцам:

def plot\_rate(df: pd.DataFrame, month: int) -> None:

    df\_month = df[df['date'].dt.month == month]

    median\_value = df\_month['vunit\_rate'].median()

    mean\_value = df\_month['vunit\_rate'].mean()

    plt.figure(figsize=(10, 6))

    plt.plot(df\_month['date'], df\_month['vunit\_rate'], label='Курс')

    plt.axhline(median\_value, color='r', linestyle='--', label='Медиана')

    plt.axhline(mean\_value, color='g', linestyle=':', label='Среднее значение')

    plt.title('Изменение курса за месяц')

    plt.xlabel('Дата')

    plt.ylabel('Курс')

    plt.legend()

    plt.show()

График изменения курса по месяцам в году:

def plot\_rate\_year\_month(df: pd.DataFrame, year: int, month: int) -> pd.DataFrame:

    df\_year = df[(df['date'].dt.year == year) & (df['date'].dt.month == month)]

    median\_value = df\_year['vunit\_rate'].median()

    mean\_value = df\_year['vunit\_rate'].mean()

    plt.figure(figsize=(10, 6))

    plt.plot(df\_year['date'], df\_year['vunit\_rate'], label='Курс')

    plt.axhline(median\_value, color='r', linestyle='--', label='Медиана')

    plt.axhline(mean\_value, color='g', linestyle=':', label='Среднее значение')

    plt.title('Изменение курса за месяц')

    plt.xlabel('Дата')

    plt.ylabel('Курс')

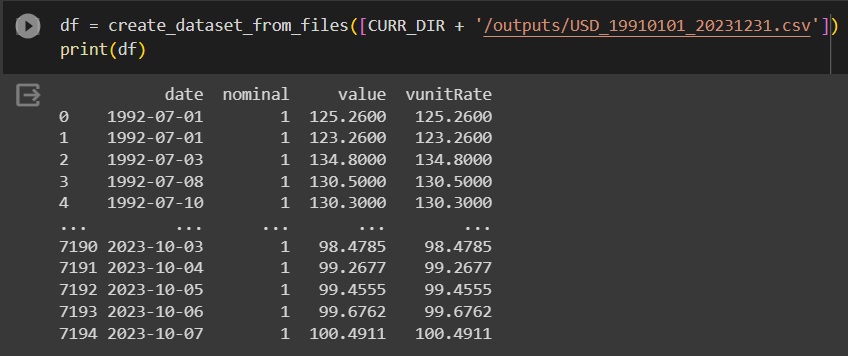
    plt.legend()

    plt.show()

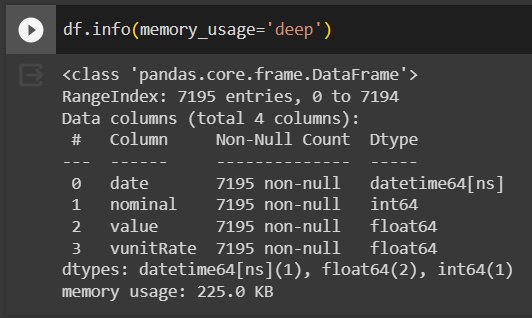
plot\_rate(df, 2012, 8)

Тестирование:

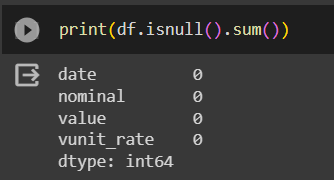
Загрузка и чтение датасета:



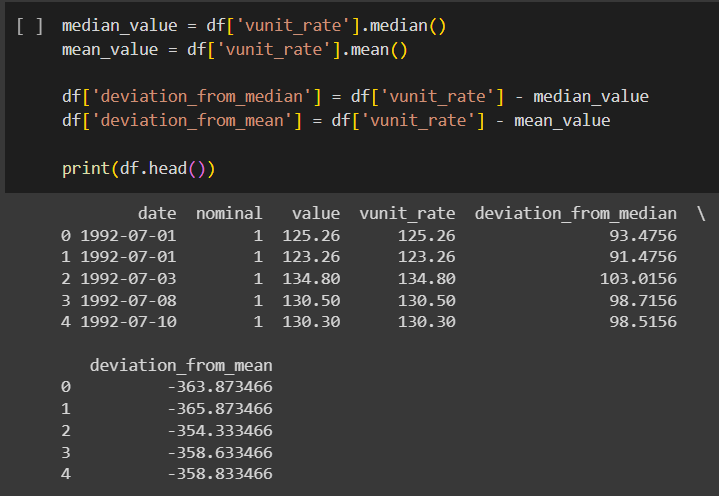
Информация о датасете:



Поиск NaN, Null значений:



Расчет и добавление медианы и среднего значения:



Статистическая информация:

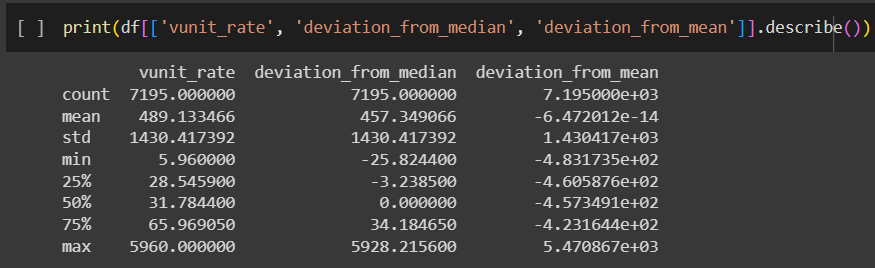
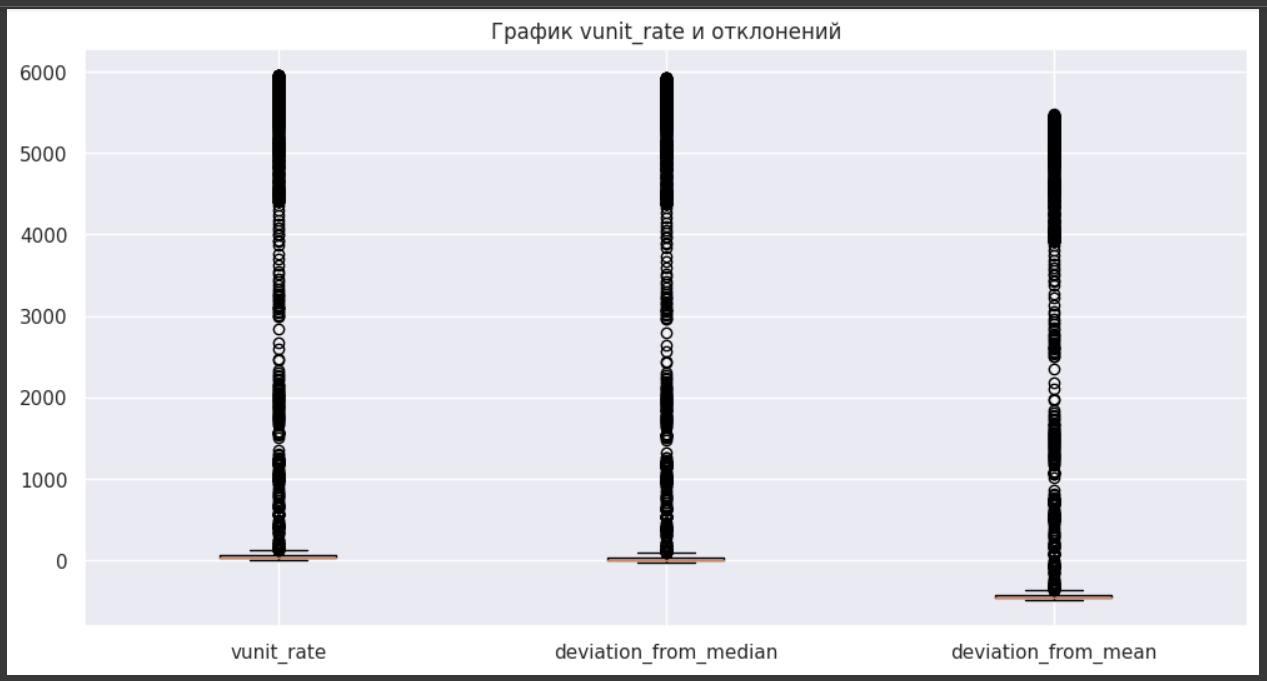
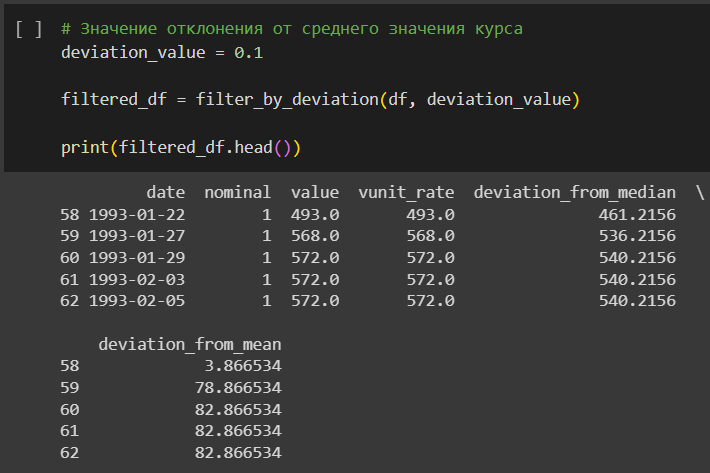


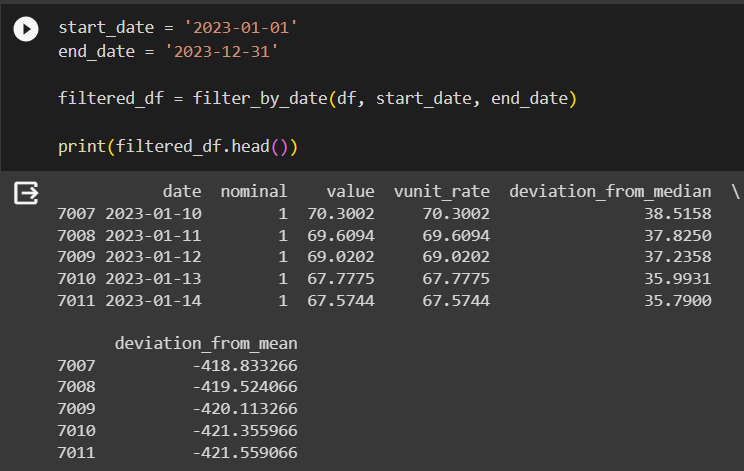
График отклонений:



Отфильтрованные данные по отклонению:



Отфильтрованные данные по дате:



Сгруппированные по месяцам:

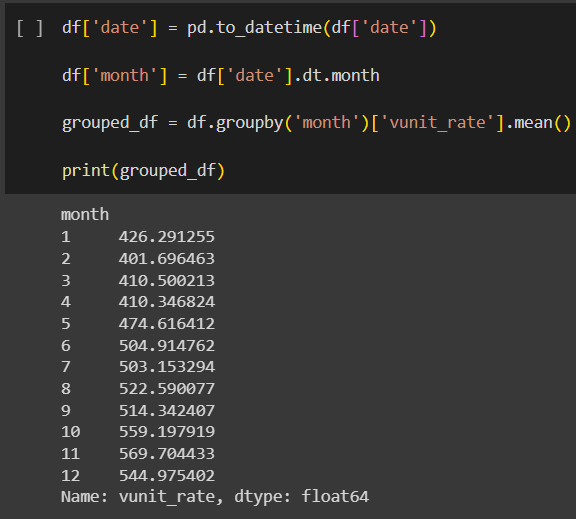


График изменения курса по дате:



График изменения курса за 12 месяц:

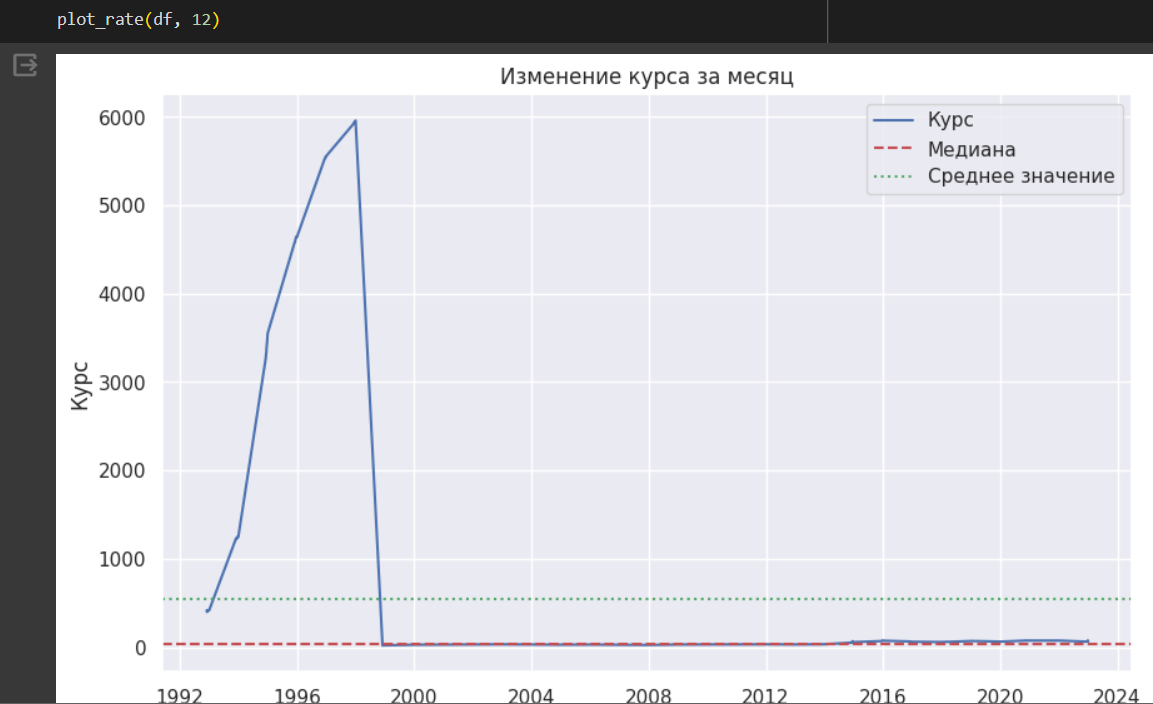


График изменения курса за 8 месяц 12 года:

