**ЛАБОРАТОРНАЯ РАБОТА**

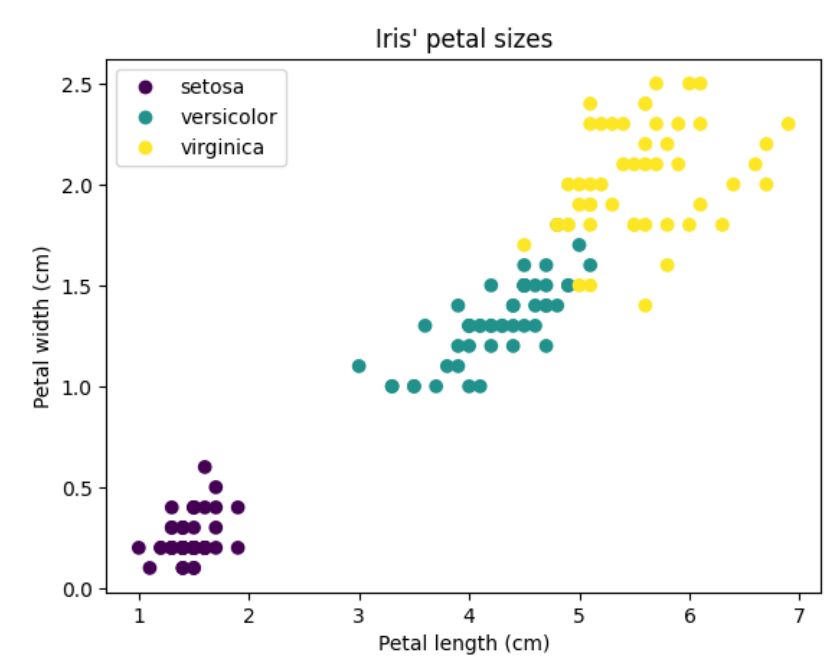
**№3 ПРЕДСТАВЛЕНИЕ ДАННЫХ В PYTHON**

**Условие задачи:**

**Задание 1. Построение диаграммы рассеяния:**

Заданы некоторые табличные данные. В них надо выбрать два столбца (фактора), на их основе построить диаграмму рассеяния, причём маркеры должны иметь различные цвета в зависимости от принадлежности к определённому классу. Сами данные, нужные факторы и разбиение на классы приведены в приложении А.

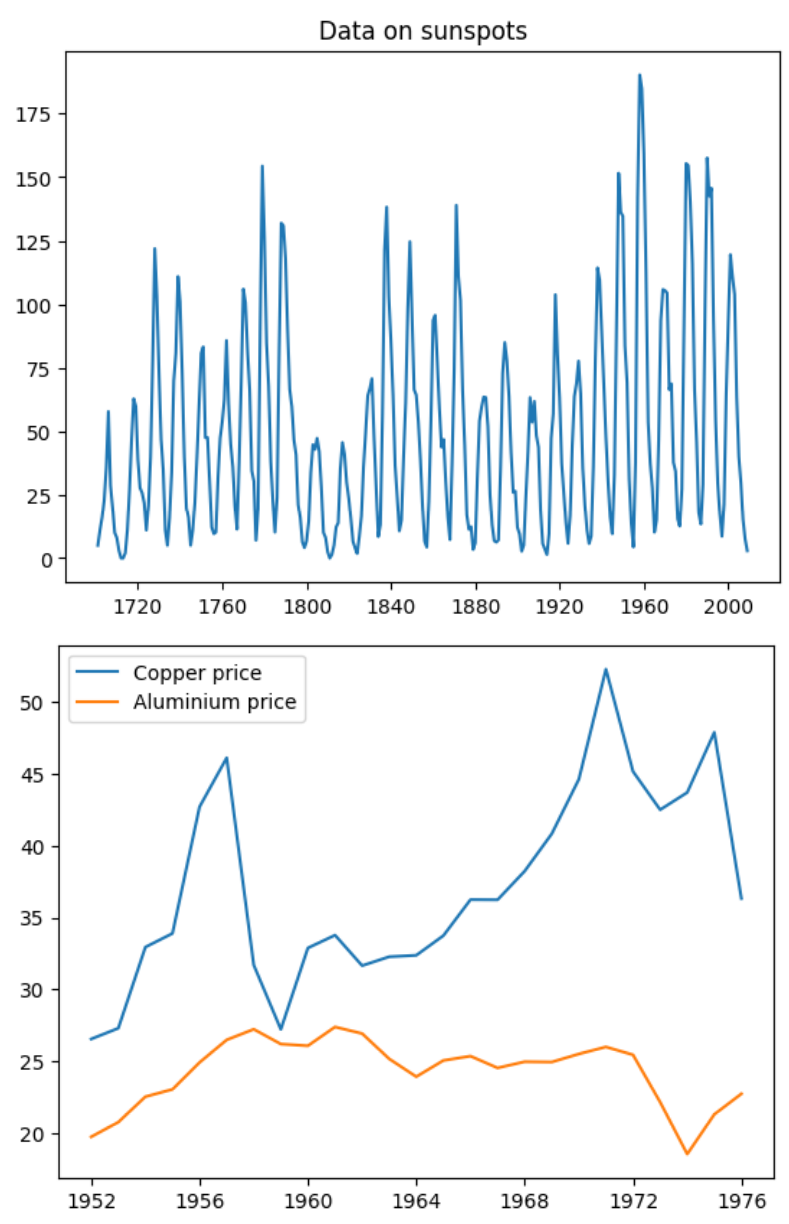
В результате должен получиться график наподобие того, что приведён на рисунке ниже.



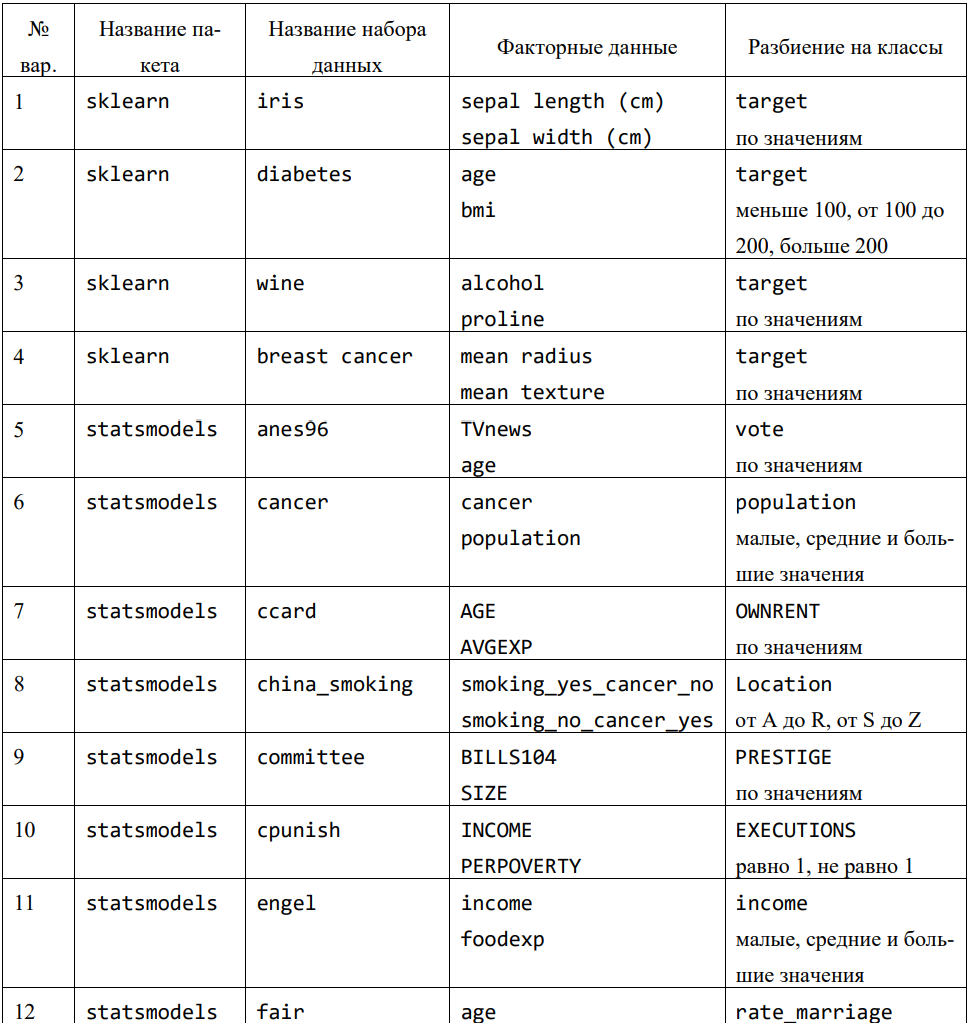
**Задание 2. Построение графика динамики временных рядов:**

Задано несколько временных рядов в виде таблицы, в которой одно либо два поля указывают на определённый момент времени. Нужно построить графики динамики (единый для всех либо отдельные) для заданных временных рядов за определённый промежуток времени. Данные и заданный промежуток приведены в приложении Б.

Примеры результатов приведены ниже.

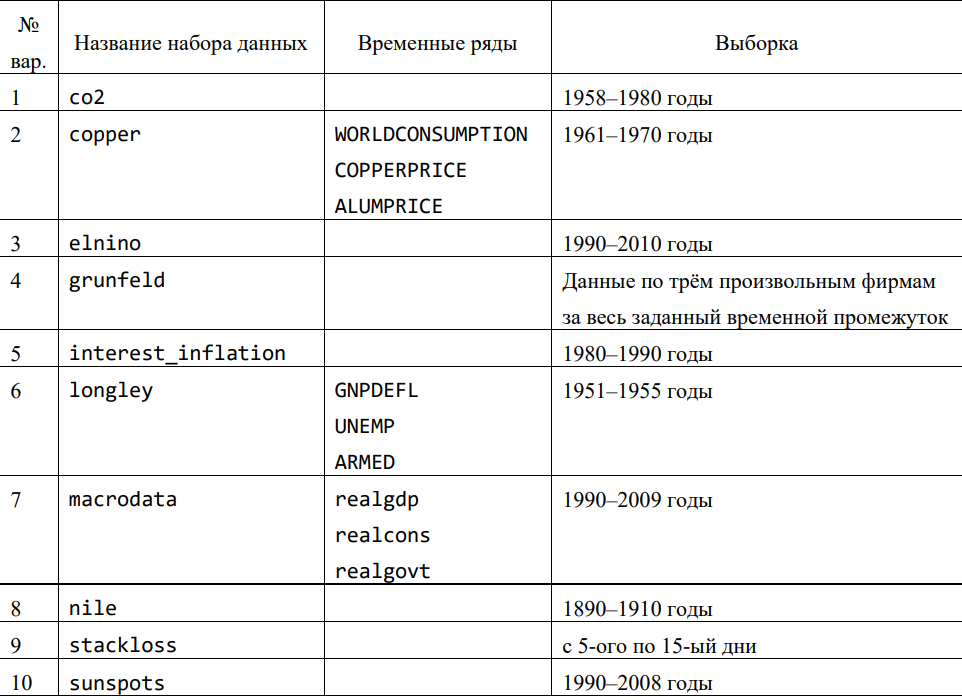


**ВАРИАНТЫ ДЛЯ ЗАДАНИЯ 1**



Мой выбор пал на 2ю строку таблицы.

**ВАРИАНТЫ ДЛЯ ЗАДАНИЯ 2**

****

Мой выбор пал на 1ю строку таблицы.

**Анализ задачи:**

**Задание** **1**. Задача состоит в визуализации табличных данных с использованием диаграммы рассеяния. Необходимо выбрать два столбца данных (факторы) в качестве координат X и Y на графике. Каждая точка на диаграмме должна быть окрашена в цвет, соответствующий классу, к которому принадлежит эта точка.

**Задание** **2**. Задача состоит в визуализации временных рядов. Данные представлены в виде таблицы, где один или два столбца идентифицируют моменты времени. Требуется создать графики, отображающие динамику изменения значений временных рядов в заданном временном интервале. Может потребоваться построение как единого графика для всех рядов, так и отдельных графиков для каждого ряда.

**Написание кода**

1. Для начала скачем все необходимые нам библиотеки. Для выполнения этого задания нам потребуется библиотеки numpy, statsmodels, sklearn, matplotlib. Всех их можно сказать через терминал введя команду pip install «название библиотеки». После этого импортируем все библиотеки:

import matplotlib.pyplot as plt  
import numpy as np  
from sklearn.datasets import load\_diabetes  
import statsmodels.api as sm

1. Создадим переменную **data** для загрузки данных о диабете. Далее создадим переменные **age, bmi** и **target** для того, чтобы извлечь данные о возрасте, индексе массы тела и целевой переменной из загруженного набора данных.
2. Для дальнейшей удобности работы с данными, создадим функцию **group**, которая классифицирует значения целевой переменной **target** на три группы: "Меньше 100", "100–200" и "Больше 200".
3. Создадим массив **groups** содержащий метки групп для каждой точки данных. Так же применяет функцию **group** к каждому значению в **target.**
4. Создадим переменную **colors** , которая сопоставляет каждой группе цвет для отображения на графике.

Приступим к созданию графиков:

1. Создает новое полотно для графика размером 8x6 дюймов с помощью функции plt.figure
2. Дальше нам нужно перебрать уникальные группы в массиве. Для этого хорошо будет воспользоваться циклом for. 1м делом создадим «маску»(логический массив), которая выделяет точки, принадлежащие текущей группе g. mask = groups == g
3. Построим диаграмму рассеяния, используя age в качестве оси X и bmi в качестве оси Y. Точки, принадлежащие одной группе, имеют одинаковый цвет, заданный через словарь colors. alpha=0.6 делает точки полупрозрачными. label=g задает метку для группы.

Далее обзовём оси, график, добавим легенду, объясняющую, какие цвета соответствуют каким группам, включим отображение сетки на графике, и предотвратим перекрытие графика. Итоговый код выглядит следующим образом:



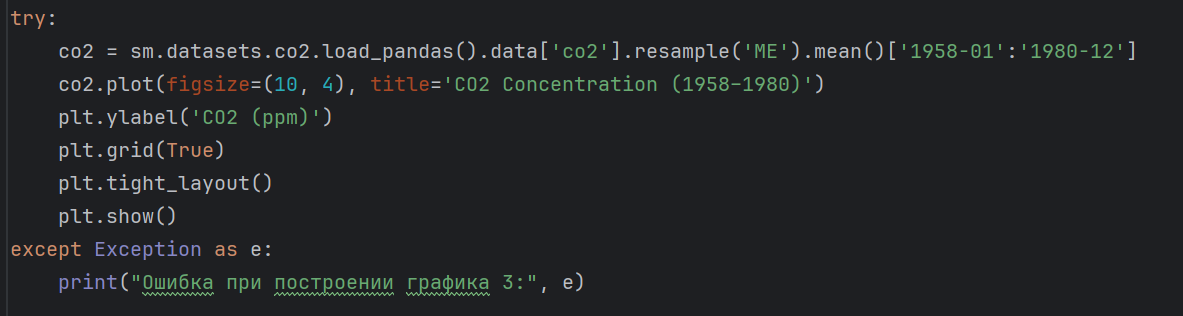
import matplotlib.pyplot as plt  
import numpy as np  
from sklearn.datasets import load\_diabetes  
import statsmodels.api as sm  
  
data = load\_diabetes()  
age, bmi, target = data.data[:, 0], data.data[:, 2], data.target  
  
def group(val): return 'Меньше 100' if val < 100 else '100–200' if val <= 200 else 'Больше 200'  
groups = np.array([group(t) for t in target])  
colors = {'Меньше 100': 'blue', '100–200': 'orange', 'Больше 200': 'red'}  
  
plt.figure(figsize=(8, 6))  
for g in np.unique(groups):  
 mask = groups == g  
 plt.scatter(age[mask], bmi[mask], label=g, color=colors[g], alpha=0.6)  
  
plt.xlabel('Age (норм.)')  
plt.ylabel('BMI')  
plt.title('Diabetes: Age vs BMI (по target)')  
plt.legend(title='Группа target')  
plt.grid(True)  
plt.tight\_layout()  
plt.show()

Теперь задание 2:

1. Загрузим данные о концентрации CO2 из набора данных co2 библиотеки statsmodels. resample('ME').mean() выполняет пересчет данных на месячные средние значения, а ['1958-01':'1980-12'] выбирает данные за период с января 1958 года по декабрь 1980 года.
2. Строим график временного ряда для данных CO2. **figsize**=(10, 4) зададим размер графика.

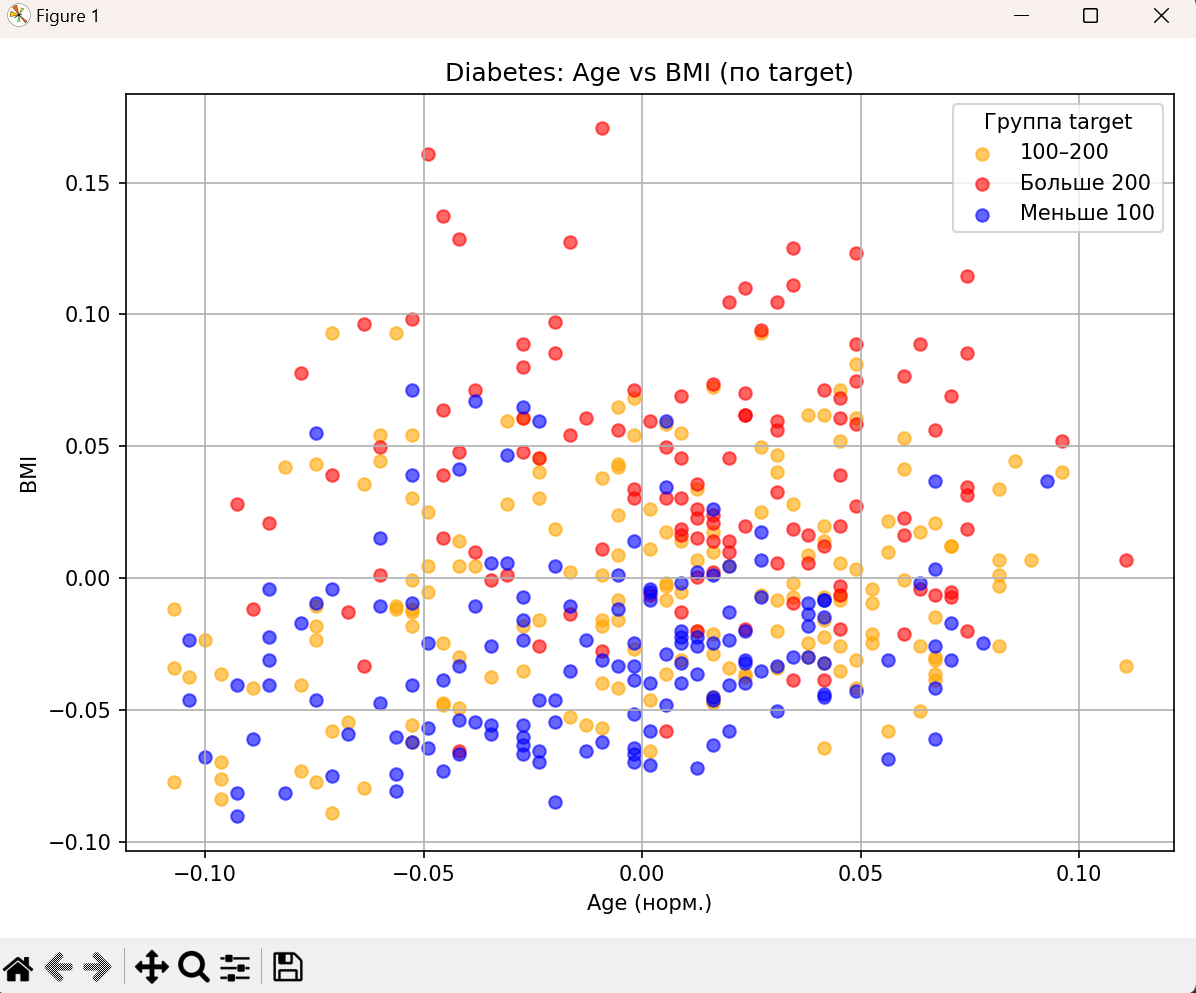
Назовём оси, включим сетку, скорректируем элементы и отображение графика, и напишем условие,на тот случай если при выполнении кода в блоке try возникает ошибка. Она будет поймана блоком except, и будет выведено сообщение об ошибке.

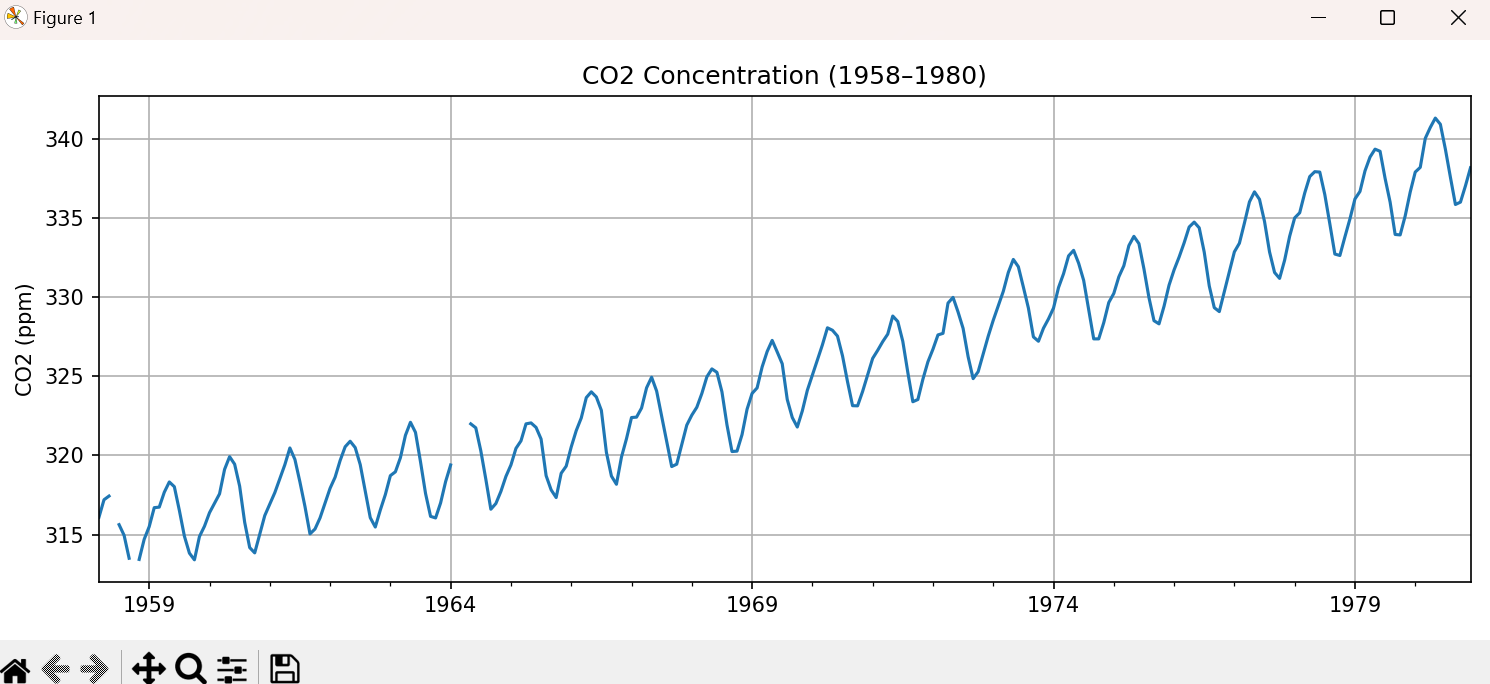
Итоговый код:



try:  
 co2 = sm.datasets.co2.load\_pandas().data['co2'].resample('ME').mean()['1958-01':'1980-12']  
 co2.plot(figsize=(10, 4), title='CO2 Concentration (1958–1980)')  
 plt.ylabel('CO2 (ppm)')  
 plt.grid(True)  
 plt.tight\_layout()  
 plt.show()  
except Exception as e:  
 print("Ошибка при построении графика 3:", e)

**Проверка кода**

****

****

**Всё готово!**