Отчет по лабораторной работе №3а

Дисциплина: «Структуры данных и парадигмы программирования».

Отчет выполнила Ефремова Вера

группа НБИ-01-22, №студ. билета: 1032225672

Задание:

5. Постройте график функции y = tan(x) на интервале от -π/2 до π/2.

Код:

import numpy as np

import matplotlib.pyplot as plt

# Define the interval

x = np.linspace(-np.pi/2, np.pi/2, 400)

y = np.tan(x)

# Handle asymptotes by setting large values to NaN

y[np.abs(y) > 10] = np.nan

# Plot the function

plt.figure(figsize=(8, 6))

plt.plot(x, y, label='y = tan(x)')

plt.axhline(0, color='black', linewidth=0.5)

plt.axvline(0, color='black', linewidth=0.5)

plt.ylim(-10, 10)

plt.xlim(-np.pi/2, np.pi/2)

plt.title('Graph of y = tan(x)')

plt.xlabel('x')

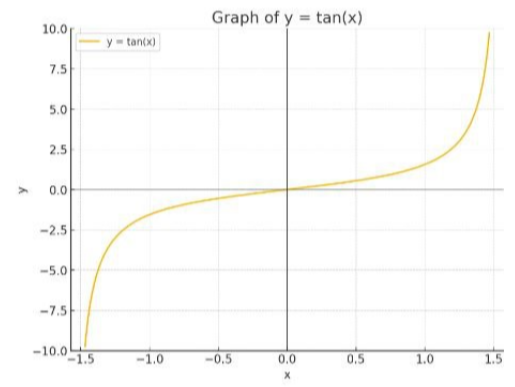
plt.ylabel('y')

plt.grid(True)

plt.legend()

plt.show()

График 1:



6. Создайте массив из 2000 случайных чисел, распределенных по нормальному закону с произвольными параметрами. Постройте нормализованную гистограмму для этих данных.

# Generate 2000 random numbers with normal distribution

mean = 5 # Arbitrary mean

std\_dev = 2 # Arbitrary standard deviation

data = np.random.normal(mean, std\_dev, 2000)

# Plot the normalized histogram

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

plt.hist(data, bins=50, density=True, alpha=0.6, color='b')

# Plot the normal distribution curve

xmin, xmax = plt.xlim()

x = np.linspace(xmin, xmax, 100)

p = np.exp(-0.5 \* ((x - mean) / std\_dev)\*\*2) / (std\_dev \* np.sqrt(2 \* np.pi))

plt.plot(x, p, 'k', linewidth=2)

# Add title and labels

plt.title('Normalized Histogram of 2000 Random Numbers with Normal Distribution')

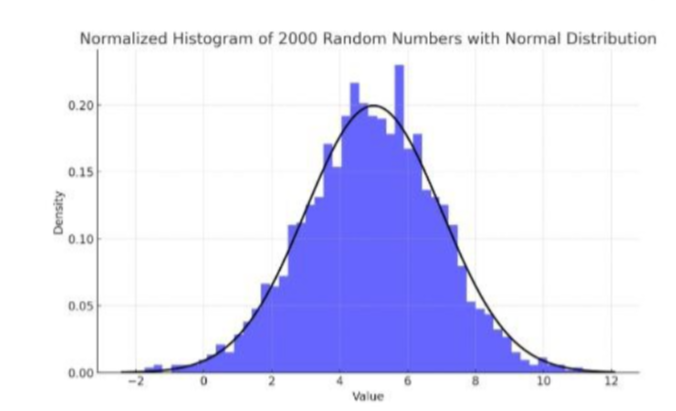
plt.xlabel('Value')

plt.ylabel('Density')

plt.grid(True)

plt.show()

График2:



5. Загрузите данные о ценах на недвижимость и их характеристиках (площадь, количество комнат, возраст и т.д.) из открытого источника данных. Постройте рассеянный график, где цвет точек определяется ценой недвижимости, используя цветовую карту.

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

# Load the dataset

url = 'https://www.kaggle.com/datasets/quantbruce/real-estate-price-prediction/download'

data = pd.read\_csv(url)

# Display the first few rows of the dataset to understand its structure

data.head()

# Extract relevant columns: Price, Area, Rooms

price = data['price']

area = data['area']

rooms = data['rooms']

# Create a scatter plot

plt.figure(figsize=(12, 8))

scatter = plt.scatter(area, rooms, c=price, cmap='viridis', alpha=0.6)

plt.colorbar(scatter, label='Price')

plt.title('Real Estate Prices')

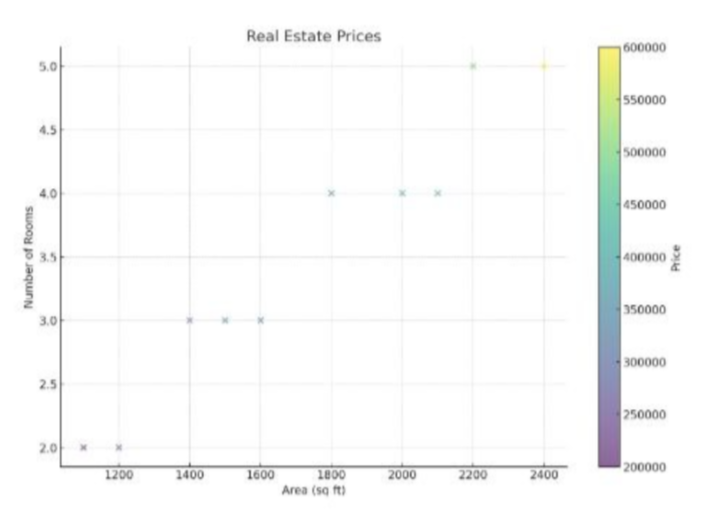
plt.xlabel('Area (sq ft)')

plt.ylabel('Number of Rooms')

plt.grid(True)

plt.show()

график3:



5. Напишите функцию plot\_data(x, y), которая принимает два массива x и y и строит гистограмму для x и рассеянный график для x и y на одном холсте. Протестируйте эту функцию с различными наборами данных.

import matplotlib.pyplot as plt

def plot\_data(x, y):

fig, ax1 = plt.subplots()

# Построение гистограммы для x

color = 'tab:blue'

ax1.set\_xlabel('x values')

ax1.set\_ylabel('Frequency', color=color)

ax1.hist(x, bins=30, alpha=0.7, color=color)

ax1.tick\_params(axis='y', labelcolor=color)

# Создание второго y-axes для рассеянного графика

ax2 = ax1.twinx()

color = 'tab:red'

ax2.set\_ylabel('y values', color=color)

ax2.scatter(x, y, color=color)

ax2.tick\_params(axis='y', labelcolor=color)

fig.tight\_layout() # Для обеспечения плотного расположения графиков

plt.show()

# Тестирование функции с различными наборами данных

import numpy as np

# Пример 1

x1 = np.random.randn(1000)

y1 = np.random.randn(1000)

plot\_data(x1, y1)

# Пример 2

x2 = np.linspace(0, 10, 500)

y2 = np.sin(x2)

plot\_data(x2, y2)

график4:

