Лабораторная работа №2. Шанаурин Антон Александрович, ЛБ-20 НЕТ

Вариант 26

Задача 1.

Средствами библиотеки Pandas создать серию или фрейм данных (в зависимости от варианта). В серии должно быть не менее 20 элементов, во фрейме не менее 15 строк и не менее двух колонок. Набор данных должен иметь индексы, колонки фрейма – осмысленные названия. Полученную серию (фрейм) вывести целиком, показать длину, размерность, число элементов. Покажите тип элементов. Подсчитайте число уникальных значений каждого столбца.

> Создайте DataFrame с колонками – название страны, площадь, материк, где расположена

In [2]:
```
import pandas as pd

# Данные о странах
data = {
    "Страна": [
        "Россия", "Канада", "США", "Китай", "Бразилия",
        "Австралия", "Индия", "Аргентина", "Каскос", "Афганистан",
        "Мексика", "Индонезия", "Судан", "Ливия", "Тайланд", "Тайланд"
    ],
    "Площадь (км²)": [
        17098242, 9984670, 9372610, 9596961, 8515767,
        7692024, 3287263, 2780400, 637657, 652230,
        1943950, 1904569, 1861484, 1759540, 513120,
        513120
    ],
    "Материк": [
        "Евразия", "Северная Америка", "Северная Америка", "Азия", "Южная Америк
        "Океания", "Азия", "Южная Америка", "Азия", "Азия",
        "Северная Америка", "Азия", "Африка", "Африка", "Азия", "Азия"
    ]
}

# Создаем DataFrame
df = pd.DataFrame(data)

# Выводим фрейм данных
print("Фрейм данных:\n", df)

# Длина, размерность и число элементов
length = len(df)
shape = df.shape
size = df.size
data_types = df.dtypes
```

```python
# Подсчет уникальных значений для каждого столбца
unique_values = df.nunique()

# Вывод результатов
print("\nДлина фрейма данных:", length)
print("Размерность фрейма данных:", shape)
print("Число элементов в фрейме данных:", size)
print("Типы элементов в каждом столбце:\n", data_types)
print("\nЧисло уникальных значений в каждом столбце:\n", unique_values)
```

```
Фрейм данных:
         Страна  Площадь (км²)           Материк
0        Россия       17098242           Евразия
1        Канада        9984670  Северная Америка
2           США        9372610  Северная Америка
3         Китай        9596961             Азия
4      Бразилия        8515767     Южная Америка
5     Австралия        7692024           Океания
6         Индия        3287263             Азия
7     Аргентина        2780400     Южная Америка
8        Каскос         637657             Азия
9    Афганистан         652230             Азия
10      Мексика        1943950  Северная Америка
11    Индонезия        1904569             Азия
12        Судан        1861484           Африка
13        Ливия        1759540           Африка
14     Тайланд         513120             Азия
15     Тайланд         513120             Азия

Длина фрейма данных: 16
Размерность фрейма данных: (16, 3)
Число элементов в фрейме данных: 48
Типы элементов в каждом столбце:
 Страна           object
Площадь (км²)     int64
Материк          object
dtype: object

Число уникальных значений в каждом столбце:
 Страна           15
Площадь (км²)    15
Материк           6
dtype: int64
```

Задача 2.

На просторах интернета найти файл с расширением csv, представляющий собой датасет реальных (а не выдуманных) данных. Для поиска можно загуглить фразы типа «Датасеты для машинного обучения и анализа данных», «20 лучших датасетов». Запрещено брать датасеты «ирисы», «преступность в Чикаго», «пингвины», «недвижимость Бостона», «пассажиры Титаника», а также те, что рассматривались на парах. Датасет должен иметь не менее 100 строчек (записей) и не менее четырёх столбцов, как минимум два из которых числовые, один категориальный уникальный (без повторения значений, например, номера телефонов) и один категориальный с повторением значений (например, национальность).

Файл загрузить во фрейм df (переменную типа DataFrame). Если в исходном фрейме много колонок, то удалить лишние, оставив не более 6, удовлетворяющих условиям из предыдущего абзаца. Вывести на экран список колонок, число строк фрейма, размерность, общее число элементов, проверить, есть ли пустые элементы. Вывести первые и последние три строки фрейма.

Для каждой колонки указать тип данных и число уникальных элементов. А для каждой числовой колонки дополнительно вывести значения максимального, минимального элементов и среднее арифметическое.

In [1]:
```
import pandas as pd

# Загрузка данных из CSV файла
df = pd.read_csv('шанаурин estate.csv', sep=';')

# Удаление лишних колонок, оставив не более 6
df = df.iloc[:, :6]  # Оставить только первые 6 колонок

# Вывод информации о фрейме
print("Список колонок:", df.columns.tolist())
print("Число строк фрейма:", df.shape[0])
print("Размерность фрейма:", df.shape)
print("Общее число элементов:", df.size)
print("Есть ли пустые элементы:", df.isnull().values.any())

# Вывод первых и последних трех строк фрейма
print("\nПервые три строки:\n", df.head(3))
print("\nПоследние три строки:\n", df.tail(3))

# Вывод типов данных и количества уникальных элементов для каждой колонки
for column in df.columns:
    print(f"\nКолонка: {column}")
    print(f"Тип данных: {df[column].dtype}")
    print(f"Число уникальных элементов: {df[column].nunique()}")

    # Для числовых колонок выводим дополнительные статистики
    if pd.api.types.is_numeric_dtype(df[column]):
        print(f"Минимальное значение: {df[column].min()}")
        print(f"Максимальное значение: {df[column].max()}")
        print(f"Среднее арифметическое: {df[column].mean()}")
```

```
Список колонок: ['address', 'adm_district', 'city_district', 'lattitude', 'longit
ude']
Число строк фрейма: 15366
Размерность фрейма: (15366, 5)
Общее число элементов: 76830
Есть ли пустые элементы: True

Первые три строки:
                                              address      adm_district  \
0         город Москва, улица Егора Абакумова, дом 9  Северо-восточный
1  город Москва, улица Талалихина, дом 2/1, корпус 1       Центральный
2          город Москва, Абельмановская улица, дом 6       Центральный

  city_district  lattitude  longitude
0   Ярославский  55.878996  37.714462
1     Таганский  55.738298  37.673337
2     Таганский  55.735528  37.669516

Последние три строки:
                                                 address    adm_district  \
15363           город Москва, улица Земляной Вал, дом 33     Центральный
15364  город Москва, поселение Московский, Киевское ш...  Новомосковский
15365             город Москва, Ходынский бульвар, дом 4        Северный

        city_district  lattitude  longitude
15363       Басманный  55.757339  37.659236
15364  Новомосковский  55.634242  37.441395
15365     Хорошевский  55.790387  37.530400

Колонка: address
Тип данных: object
Число уникальных элементов: 9085

Колонка: adm_district
Тип данных: object
Число уникальных элементов: 12

Колонка: city_district
Тип данных: object
Число уникальных элементов: 127

Колонка: lattitude
Тип данных: float64
Число уникальных элементов: 9002
Минимальное значение: 55.1241231
Максимальное значение: 56.0162572
Среднее арифметическое: 55.742271755826216

Колонка: longitude
Тип данных: float64
Число уникальных элементов: 8996
Минимальное значение: 36.8676238
Максимальное значение: 38.4622104
Среднее арифметическое: 37.59385917992575
```

Задача 3.

По данным фрейма из второй задачи построить два наиболее показательных графика (диаграммы) разных типов. Обосновать, почему выбраны именно эти два

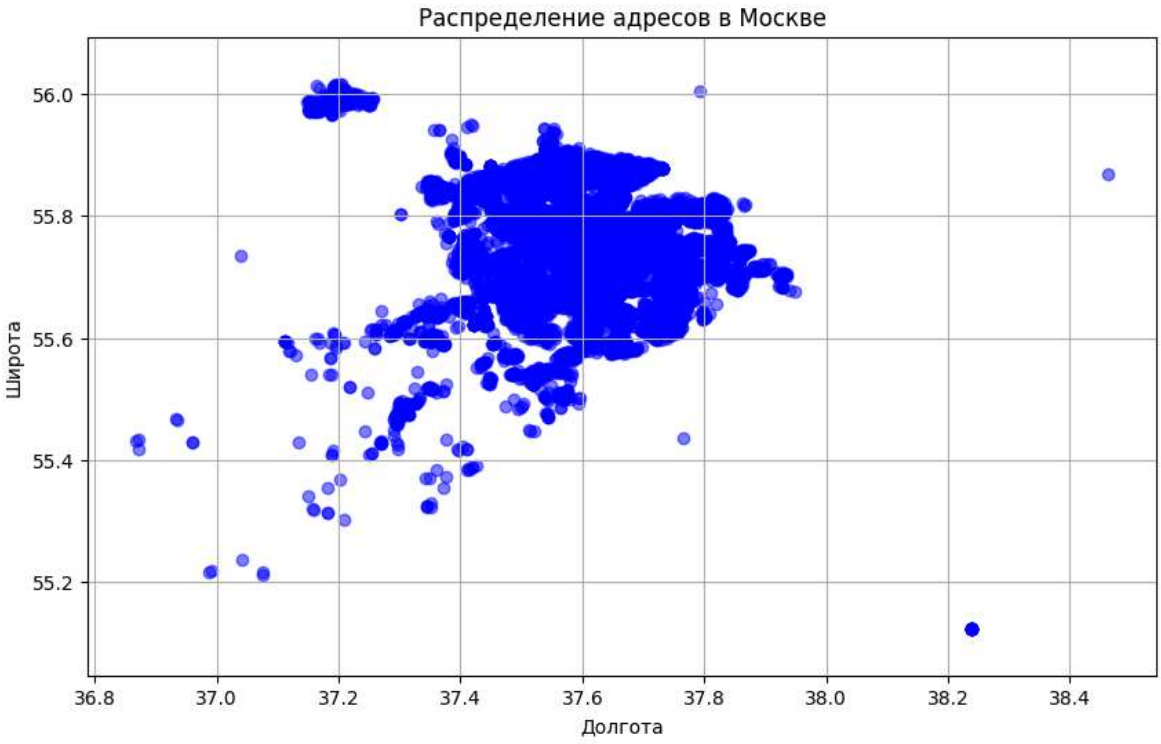
типа. Графики (диаграммы) оформить максимально информативно.

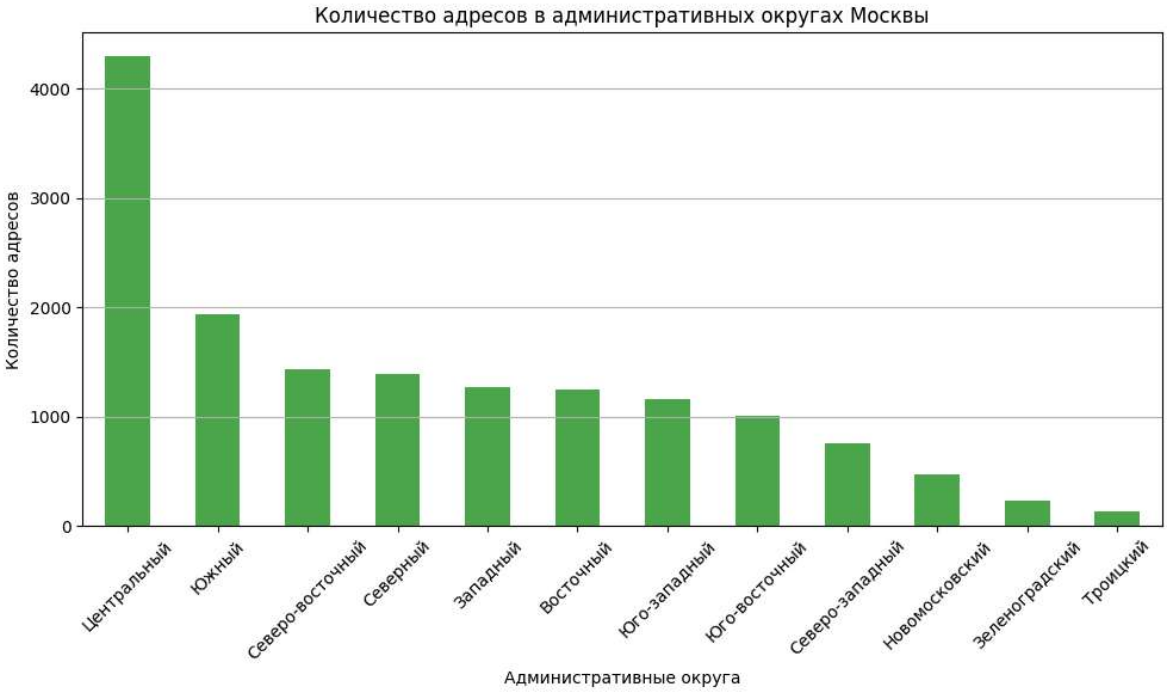
In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Загрузка данных из CSV файла
df = pd.read_csv('шанаурин estate.csv', sep=';')

# Построение диаграммы рассеяния
plt.figure(figsize=(10, 6))
plt.scatter(df['longitude'], df['lattitude'], c='blue', alpha=0.5)
plt.title('Распределение адресов в Москве')
plt.xlabel('Долгота')
plt.ylabel('Широта')
plt.grid(True)
plt.axis('equal')  # Для правильного соотношения осей
plt.show()

# Построение гистограммы
plt.figure(figsize=(10, 6))
df['adm_district'].value_counts().plot(kind='bar', color='green', alpha=0.7)
plt.title('Количество адресов в административных округах Москвы')
plt.xlabel('Административные округа')
plt.ylabel('Количество адресов')
plt.xticks(rotation=45)
plt.grid(axis='y')
plt.tight_layout()  # Чтобы уникорректно отобразить график
plt.show()
```





Описание результата

Первый график — диаграмма рассеяния — показывает, как адреса распределены по широте и долготе. Это позволяет выделить наиболее плотно застроенные районы.

Второй график — гистограмма — иллюстрирует распределение количества адресов по административным округам, что позволяет увидеть, в каких округах наибольшее и наименьшее количество адресов.

Оба графика вместе дают полное представление о пространственном распределении адресов в Москве и помогают проводить дальнейший анализ