МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ЛИПЕЦКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Институт компьютерных наук
Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №1

по МАШИННОМУ ОБУЧЕНИЮ

на тему «Основы работы в R»

Студент М-РИТ-25-1 _____ Киселев М.С.
(подпись, дата)

Студент М-РИТ-25-1 _____ Красиков И.А.
(подпись, дата)

Руководитель

Профессор _____ Сараев П.В.
(подпись, дата)

Липецк 2025 г.

Цель работы

Изучение основ работы с языком R и базовых методов работы с данными

Порядок выполнения работы

1. Подключите таблицу данных состояния качества воздуха в г. Нью-Йорк

airquality следующимм образом:

>library (datasets)

>head (airquality)

2. Посчитайте с помощью средств языка R (без использования циклов):

-Число строк в таблице.

-Число столбцов в таблице.

-Число строк, не имеющих пропусков (NA).

-Число строк, имеющих пропусков одновременно по столбцам Ozone и Solar.R.

-Диапазоны варьирования (минимальное и максимальное значения), а также средние значения по столбцам Ozone, Solar.R, Wind и Temp (без учета пропущенных значений).

-Среднее значение по столбцу Solar.R для 5-го месяца (без учета пропущенных значений).

3. Напишите функцию и сохраните ее в файл meanAir.R(factor, tMin, tMax), которая рассчитывает среднее значение по одному из столбцов factor (Ozone, Solar.R, Wind), когда Temp принимает значения от tMin до t.Max включительно (без учета пропущенных значений). Значения по умолчанию: tMin = 60, tMax = 80.

4. Найдите и выведите на экран с помощью средств языка R средние значения по столбцу Solar.R для каждого месяца (без учета пропущенных значений), используя функции split и sapply/lapply.

5. Реализуйте функцию maxTemp(days = 1), которая возвращает требуемое количество пар месяц/день с максимальной температурой. Например, если параметр days = 3, то должны быть выведены ровно 3 пары значений «месяц/день». Параметр days - количество дней - некоторое натуральное число.

6. Реализуйте функцию testSet(perc = 20), которая возвращает тестовое множество, состоящее из заданного процента строк. Строки должны выбираться случайным образом. Параметр perc - вещественное число от 0 до 100. Если переданное значение параметра выходит за пределы [0; 100], необходимо выдать сообщение об ошибке с помощью функции stop.

7. Сделайте выводы о проделанной работе.

Ход работы

Для выполнения лабораторной работы был использован язык программирования Python и библиотека pandas

1. Подключение таблицы данных состояния качества воздуха в г. Нью-Йорк

```
dataset_url = "https://vincentarelbundock.github.io/Rdatasets/csv/datasets/airquality.csv"
dataset = pd.read_csv(dataset_url)
```

2. Посчитайте с помощью средств языка Python (без использования циклов):

-Число строк в таблице.

```
print('Число строк в таблица:', end=' ')
print(dataset.shape[0])
```

```
Число строк в таблица: 153
```

-Число столбцов в таблице.

```
print('Число столбцов в таблице:', end=' ')
print(dataset.shape[1])
```

```
Число столбцов в таблице: 7
```

-Число строк, не имеющих пропусков (NA).

```
print('Число строк, не имеющих пропусков:', end=' ')
print(dataset.dropna().shape[0])
```

```
Число строк, не имеющих пропусков: 111
```

-Число строк, имеющих пропусков одновременно по столбцам Ozone и Solar.R.

```
print('Число строк, имеющих пропуски одновременно в столбцах Ozone и Solar.R:', end=' ')
print((dataset['Solar.R'].isnull() & dataset['Ozone'].isnull()).sum())
```

```
Число строк, имеющих пропуски одновременно в столбцах Ozone и Solar.R: 2
```

-Диапазоны варьирования (минимальное и максимальное значения), а также средние значения по столбцам Ozone, Solar.R, Wind и Temp (без учета пропущенных значений).

```
print('Минимальные значения в столбцах Ozone, Solar.R, Wind, Temp:')
print(dataset[['Ozone', 'Solar.R', 'Wind', 'Temp']].min())
```

```
Минимальные значения в столбцах Ozone, Solar.R, Wind, Temp:
Ozone       1.0
Solar.R     7.0
Wind        1.7
Temp       56.0
dtype: float64
```

[19]:

```
print('Максимальные значения в столбцах Ozone, Solar.R, Wind, Temp:')
print(dataset[['Ozone', 'Solar.R', 'Wind', 'Temp']].max())
```

```
Максимальные значения в столбцах Ozone, Solar.R, Wind, Temp:
Ozone      168.0
Solar.R    334.0
Wind        20.7
Temp        97.0
dtype: float64
```

[20]:

```
print('Средние значения в столбцах Ozone, Solar.R, Wind, Temp:')
print(dataset[['Ozone', 'Solar.R', 'Wind', 'Temp']].mean())
```

```
Средние значения в столбцах Ozone, Solar.R, Wind, Temp:
Ozone       42.129310
Solar.R    185.931507
Wind         9.957516
Temp        77.882353
dtype: float64
```

-Среднее значение по столбцу Solar.R для 5-го месяца (без учета пропущенных значений).

```
print("Среднее значение Solar.R для 5-го месяца:", end=' ')
print(dataset[dataset['Month'] == 5]['Solar.R'].mean())
```

```
Среднее значение Solar.R для 5-го месяца: 181.2962962962963
```

3. Напишите функцию и сохраните ее в файл meanAir.R(factor, tMin, tMax), которая рассчитывает среднее значение по одному из столбцов factor (Ozone, Solar.R, Wind), когда Temp принимает значения от tMin до t.Max включительно (без учета пропущенных значений). Значения по умолчанию: tMin = 60, tMax = 80.

```
def mean_air(df, factor, t_min = 60, t_max = 80):
    filtered_df = df[(df['Temp'] >= t_min) & (df['Temp'] <= t_max)]
    return filtered_df[factor].dropna().mean()

print('mean_air для Ozone:', mean_air(dataset, 'Ozone'))
print('mean_air для Solar.R:', mean_air(dataset, 'Solar.R'))
print('mean_air для Wind:', mean_air(dataset, 'Wind'))
print()
print('mean_ait для Ozone в пределах температуры [80, 130]:', mean_air(dataset, 'Ozone', t_min = 80, t_max = 130))
print('mean_ait для Solar.R в пределах температуры [60, 100]:', mean_air(dataset, 'Solar.R', t_max = 100))
print('mean_ait для Wind в пределах температуры [40, 100]:', mean_air(dataset, 'Wind', t_min = 40, t_max = 100))
```

```
mean_air для Ozone: 23.775862068965516
mean_air для Solar.R: 173.97333333333333
mean_air для Wind: 10.812987012987014

mean_ait для Ozone в пределах температуры [80, 130]: 62.3859649122807
mean_ait для Solar.R в пределах температуры [60, 100]: 189.74285714285713
mean_ait для Wind в пределах температуры [40, 100]: 9.957516339869281
```

4. Найдите и выведите на экран с помощью средств языка R средние значения по столбцу Solar.R для каждого месяца (без учета пропущенных значений), используя функции split и sapply/lapply.

```
for month in dataset['Month'].dropna().unique():
    print(f"Среднее значение Solar.R для месяца {month}:", dataset[dataset['Month'] == month]['Solar.R'].mean())
```

```
Среднее значение Solar.R для месяца 5: 181.2962962962963
Среднее значение Solar.R для месяца 6: 190.16666666666666
Среднее значение Solar.R для месяца 7: 216.48387096774192
Среднее значение Solar.R для месяца 8: 171.85714285714286
Среднее значение Solar.R для месяца 9: 167.43333333333334
```

5. Реализуйте функцию maxTemp(days = 1), которая возвращает требуемое количество пар месяц/день с максимальной температурой. Например, если параметр days = 3, то должны быть выведены ровно 3 пары значений «месяц/день». Параметр days - количество дней - некоторое натуральное число.

```
def max_temp(df, days = 1):
    result_df = df.sort_values(by='Temp', ascending=False).head(days)[['Month', 'Day']]
    return (result_df['Month'].astype(str) + '/' + result_df['Day'].astype(str)).tolist()

print('max_temp для 1 days:', max_temp(dataset))
print('max_temp для 2 days:', max_temp(dataset, 2))
print('max_temp для 3 days:', max_temp(dataset, 3))
```

```
max_temp для 1 days: ['8/28']
max_temp для 2 days: ['8/28', '8/30']
max_temp для 3 days: ['8/28', '8/30', '8/29']
```

6. Реализуйте функцию testSet(perc = 20), которая возвращает тестовое множество, состоящее из заданного процента строк. Строки должны выбираться случайным образом. Параметр perc - вещественное число от 0 до 100. Если переданное значение параметра выходит за пределы [0; 100], необходимо выдать сообщение об ошибке с помощью функции stop.

```
def test_set(df, perc = 20):
    if not 0 <= perc <= 100:
        raise ValueError('Параметр perc должен быть в диапозоне [0; 100]')
    set_rate = perc / 100.0
    return df.sample(frac=set_rate, random_state=random.randint(1, 100))
```

```
# Тестовая выборка пример 1
test_set1 = test_set(dataset)
print("Количество строк:", test_set1.shape[0])
print(test_set1.head())
```

```
Количество строк: 31
    rownames  Ozone  Solar.R  Wind  Temp  Month  Day
56        57    NaN    127.0   8.0    78      6   26
86        87   20.0     81.0   8.6    82      7   26
73        74   27.0    175.0  14.9    81      7   13
97        98   66.0      NaN   4.6    87      8    6
76        77   48.0    260.0   6.9    81      7   16
```

```
# Тестовая выборка пример 2
test_set1 = test_set(dataset, 50)
print("Количество строк:", test_set1.shape[0])
print(test_set1.head())
```

```
Количество строк: 76
     rownames  Ozone  Solar.R  Wind  Temp  Month  Day
54         55    NaN    250.0   6.3    76      6   24
103       104   44.0    192.0  11.5    86      8   12
37         38   29.0    127.0   9.7    82      6    7
126       127   91.0    189.0   4.6    93      9    4
82         83    NaN    258.0   9.7    81      7   22
```

```
# Проверка ошибки в функции test_set
test_set1 = test_set(dataset, -1)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[28], line 2
      1 # Проверка ошибки в функции test_set
----> 2 test_set1 = test_set(dataset, -1)

Cell In[25], line 3, in test_set(df, perc)
      1 def test_set(df, perc = 20):
      2     if not 0 <= perc <= 100:
----> 3         raise ValueError('Параметр perc должен быть в диапозоне [0; 100]')
```

Вывод

Мы изучили основы работы с языком Python и базовыми методами работы с данными. Провели работу над датасетом загрязнений воздуха в городе Нью-Йорк.