

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт перспективной инженерии
Департамент цифровых, робототехнических систем и электроники

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №5
дисциплины
«Искусственный интеллект и машинное обучение»

Выполнил:
Митряшкина Дарина Сергеевна
2 курс, группа ИТС-б-о-23-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные системы и
сети», очная форма обучения

(подпись)

Проверил:
Доцент департамента цифровых,
робототехнических систем и
электроники Воронкин Р.А.

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2025 г.

Тема: Введение в pandas: изучение структуры DataFrame и базовых операций

Цель: познакомить с основами работы с библиотекой pandas, в частности, со структурой данных DataFrame.

Порядок выполнения работы:

1. Подключение библиотек для выполнения заданий

✓ Лабораторная работа №5

```
[10]
import pandas as pd
import numpy as np
import json
import io
from IPython.display import display
```

Рисунок 1. Подключение библиотек

2. Выполнение задания №1.

Задание 1 — Создание DataFrame

```
# 1.1
data1 = {'Имя': ['Анна', 'Борис', 'Мария'], 'Возраст': [25, 30, 28]}
df1 = pd.DataFrame(data1)
display(df1)

# 1.2
data2 = [{'Имя': 'Анна', 'Возраст': 25}, {'Имя': 'Борис', 'Возраст': 30}, {'Имя': 'Мария', 'Возраст': 28}]
df2 = pd.DataFrame(data2)
display(df2)

# 1.3
arr = np.array([[1, 2], [3, 4]])
df3 = pd.DataFrame(arr, columns=['A', 'B'])
display(df3)

# 1.4
display(df1.dtypes)
```

	Имя	Возраст
0	Анна	25
1	Борис	30
2	Мария	28

Рисунок 2. Задание №1

3. Выполнение задания №2.

Задание 2 — Загрузка данных

```
[12]
csv_data = '''Имя;Возраст;Город
Анна;25;Москва
Борис;30;СПб
Мария;28;Казань'''
df_csv = pd.read_csv(io.StringIO(csv_data), sep=';')
display(df_csv)

# "Excel"
excel_data = {'Имя': ['Анна', 'Борис'], 'Возраст': [25, 30]}
df_excel = pd.DataFrame(excel_data)
display(df_excel)

# JSON
json_data = '[{"Имя": "Анна", "Возраст": 25}, {"Имя": "Борис", "Возраст": 30}]'
df_json = pd.read_json(io.StringIO(json_data))
display(df_json)
```



Рисунок 3. Задание №2

4. Выполнение задания №3

Задание 3 — Фильтрация и доступ

```
]
df_filter = df_csv.copy()

val_iloc = df_filter.iloc[2, 1]
display(val_iloc)

val_loc = df_filter[df_filter['Имя'] == 'Мария']
display(val_loc)

val_at = df_filter.at[0, 'Имя']
display(val_at)

val_iat = df_filter.iat[0, 0]
display(val_iat)
```

np.int64(28)



Рисунок 4. Задание №3

5. Выполнение задания №4

Задание 4 – Добавление и удаление

```
[4] df_add = df_csv.copy()
df_add['Категория'] = 'Неизвестно'
df_add.loc[len(df_add)] = {'Имя': 'Олег', 'Возраст': 22, 'Город': 'Томск', 'Категория': 'Новая'}
df_add.drop('Возраст', axis=1, inplace=True)
display(df_add)
```

	Имя	Город	Категория
0	Анна	Москва	Неизвестно
1	Борис	СПб	Неизвестно
2	Мария	Казань	Неизвестно
3	Олег	Томск	Новая

Рисунок 5. Задание №4

6. Выполнение задания №5.

Задание 5 – Пропущенные значения

```
df_nan = pd.DataFrame({'A': [1, np.nan, 3], 'B': [4, 5, np.nan]})
display(df_nan)

df_drop_rows = df_nan.dropna()
display(df_drop_rows)

df_drop_cols = df_nan.dropna(axis=1)
display(df_drop_cols)

df_na_sum = df_nan.isna().sum()
display(df_na_sum)
```

	A	B
0	1.0	4.0
1	NaN	5.0
2	3.0	NaN

	A	B
0	1.0	4.0

	A	B
0	1.0	4.0

Рисунок 6. Задание №5

7. Выполнение задания №6.

Задание 6 — Подсчёт

```
df_count = df_csv.copy()
display(df_count.count())
```

	0
Имя	3
Возраст	3
Город	3

dtype: int64

Рисунок 7. Задание №6

8. Выполнение задания №7.

Задание 7 — Значения и категории

```
vc = df_csv['Город'].value_counts()
display(vc)

nuniq = df_csv['Город'].nunique()
display(nuniq)
```


	count
Город	
Москва	1
СПб	1
Казань	1

dtype: int64
3

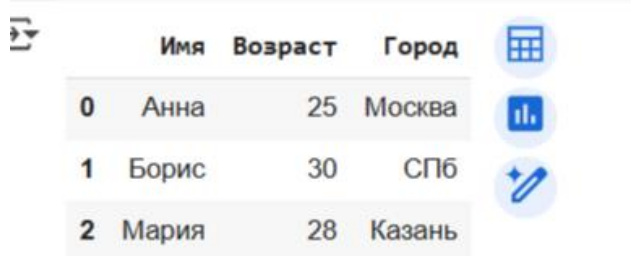
Рисунок 8. Задание №7

9. Выполнение задания №8.

Задание 8 – Отображение



```
pd.set_option('display.max_rows', 100)
display(df_csv)
```




	Имя	Возраст	Город
0	Анна	25	Москва
1	Борис	30	СПб
2	Мария	28	Казань

Рисунок 9. Задание №8

10. Выполнение индивидуального задания

Индивидуальное задание №10 – Маршруты

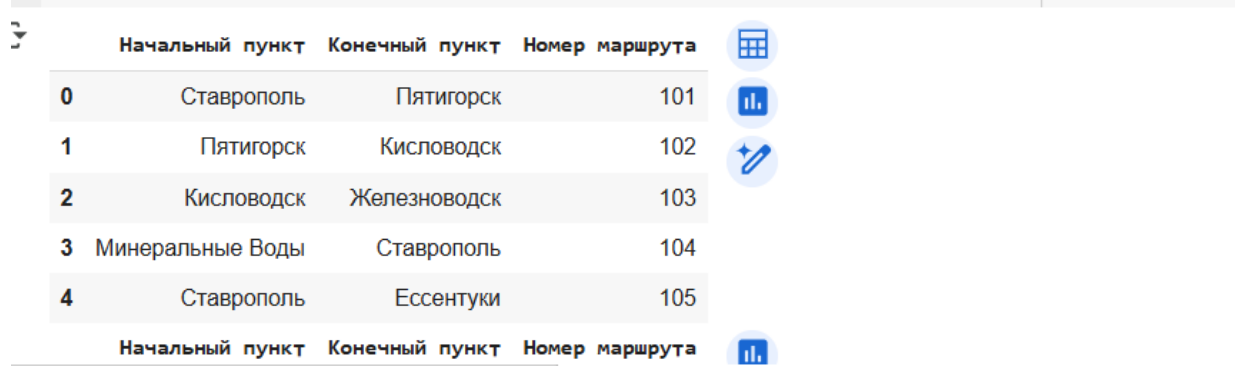


```
columns = ['Начальный пункт', 'Конечный пункт', 'Номер маршрута']
df_routes = pd.DataFrame(columns=columns)

df_routes.loc[len(df_routes)] = ['Ставрополь', 'Пятигорск', 101]
df_routes.loc[len(df_routes)] = ['Минеральные Воды', 'Ставрополь', 104]
df_routes.loc[len(df_routes)] = ['Кисловодск', 'Железноводск', 103]
df_routes.loc[len(df_routes)] = ['Пятигорск', 'Кисловодск', 102]
df_routes.loc[len(df_routes)] = ['Ставрополь', 'Ессентуки', 105]

df_routes = df_routes.sort_values(by='Номер маршрута').reset_index(drop=True)
display(df_routes)

пункт = "Ставрополь"
результат = df_routes[(df_routes['Начальный пункт'] == пункт) | (df_routes['Конечный пункт'] == пункт)]
display(результат)
```



	Начальный пункт	Конечный пункт	Номер маршрута
0	Ставрополь	Пятигорск	101
1	Пятигорск	Кисловодск	102
2	Кисловодск	Железноводск	103
3	Минеральные Воды	Ставрополь	104
4	Ставрополь	Ессентуки	105

Рисунок 14. Индивидуальное задание

Ответы на контрольные вопросы:

1. Как создать pandas.DataFrame из словаря списков?

```
import pandas as pd  
data = {'Имя': ['Анна', 'Борис', 'Мария'], 'Возраст': [25, 30, 28]}  
df = pd.DataFrame(data)
```

2. В чем отличие создания DataFrame из списка словарей и словаря списков?

Словарь списков: ключи - названия столбцов, значения - данные столбцов

Список словарей: каждый словарь - строка, ключи - названия столбцов

3. Как создать pandas.DataFrame из массива NumPy?

```
import numpy as np  
arr = np.array([[1, 2], [3, 4]])  
df = pd.DataFrame(arr, columns=['A', 'B'])
```

4. Как загрузить DataFrame из CSV-файла, указав разделитель ;?

```
df = pd.read_csv('file.csv', sep=';')
```

5. Как загрузить данные из Excel в pandas.DataFrame и выбрать конкретный лист?

```
df = pd.read_excel('file.xlsx', sheet_name='Лист1')
```

6. Чем отличается чтение данных из JSON и Parquet в pandas?

JSON: текстовый формат, медленнее, больше места

Parquet: бинарный, сжатый, сохраняет типы данных, быстрее

7. Как проверить типы данных в DataFrame после загрузки?

```
df.dtypes
```

8. Как определить размер DataFrame (количество строк и столбцов)?

`df.shape`

9. В чем разница между `.loc[]` и `.iloc[]`?

`.loc[]` - выбор по меткам (индексам и названиям столбцов)

`.iloc[]` - выбор по позициям (целочисленным индексам)

10. Как получить данные третьей строки и второго столбца с `.iloc[]`?

`value = df.iloc[2, 1]`

11. Как получить строку с индексом "Мария" из DataFrame?

`row = df.loc['Мария']`

12. Чем `.at[]` отличается от `.loc[]`?

`.at[]` - доступ к одному элементу, быстрее

`.loc[]` - может выбирать несколько элементов

13. В каких случаях `.iat[]` работает быстрее, чем `.iloc[]`?

`.iat[]` быстрее при доступе к одному элементу по позиции

14. Как выбрать все строки, где "Город" равен "Москва" или "СПб", используя `.isin()`?

`df[df['Город'].isin(['Москва', 'СПб'])]`

15. Как отфильтровать DataFrame, оставив только строки, где "Возраст" от 25 до 35 лет, используя `.between()`?

`df[df['Возраст'].between(25, 35)]`

16. В чем разница между `.query()` и `.loc[]` для фильтрации данных?

`.query()` - синтаксис как в SQL, удобен для сложных условий

`.loc[]` - стандартный pandas-синтаксис

17. Как использовать переменные Python внутри `.query()`?

```
min_age = 25
```

```
df.query('Возраст > @min_age')
```

18. Как узнать, сколько пропущенных значений в каждом столбце DataFrame?

```
df.isna().sum()
```

19. В чем разница между `.isna()` и `.notna()`?

`.isna()` - True для пропущенных значений

`.notna()` - True для не пропущенных

20. Как вывести только строки, где нет пропущенных значений?

```
df.dropna()
```

21. Как добавить новый столбец "Категория" в DataFrame, заполнив его фиксированным значением "Неизвестно"?

```
df['Категория'] = 'Неизвестно'
```

22. Как добавить новую строку в DataFrame, используя `.loc[]`?

```
df.loc[новый_индекс] = {'Колонка1': знач1, 'Колонка2': знач2}
```

23. Как удалить столбец "Возраст" из DataFrame?

```
df.drop('Возраст', axis=1, inplace=True)
```

24. Как удалить все строки, содержащие хотя бы один NaN, из DataFrame?

```
df.dropna(how='any')
```

25. Как удалить столбцы, содержащие хотя бы один NaN, из DataFrame?

```
df.dropna(axis=1)
```

26. Как посчитать количество непустых значений в каждом столбце DataFrame?

```
df.count()
```

27. Чем `.value_counts()` отличается от `.nunique()`?

`.value_counts()` - частоты всех значений

`.nunique()` - количество уникальных значений

28. Как определить сколько раз встречается каждое значение в столбце "Город"?

```
df['Город'].value_counts()
```

29. Почему `display(df)` лучше, чем `print(df)`, в Jupyter Notebook?

`display()` в Jupyter показывает красивое форматирование с возможностью сортировки

30. Как изменить максимальное количество строк, отображаемых в DataFrame в Jupyter Notebook?

```
pd.set_option('display.max_rows', 100)
```

Вывод: в ходе лабораторной работы были изучены основные методы работы с pandas DataFrame, включая создание, фильтрацию, модификацию и анализ данных. Освоены различные способы доступа к данным (`loc`, `iloc`, `at`, `iat`), а также методы обработки пропущенных значений и работы с разными форматами данных (CSV, Excel, JSON, Parquet). Полученные навыки позволяют эффективно работать с табличными данными в Python для решения практических задач анализа данных.

Ссылка на Google Colab:

https://colab.research.google.com/drive/1WuBWQJXjaye_hTFUHtYgz98PuKtp-ty?usp=sharing

Ссылка на Git Hub: <https://github.com/darina-rtm/ai5labaa.git>