

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
Институт математики и информационных технологий имени  
профессора Н.И.Червякова

Кафедра инфокоммуникаций.

Дисциплина: Технологии программирования

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

Основы Pandas

**Выполнил:** студент 3 курса  
09.03.01 «Информатика и вычислительная  
техника» группы ИВТ-б-о-19-1  
Абдуллаев Джамалудин Магомедзагирович

**Проверил:**  
Воронкин Роман Александрович

Работа защищена с оценкой:

---

Ставрополь, 2021

## Лабораторная работа №2

### Основы Pandas

**Цель работы** исследовать базовые возможности системы управления базами данных Pandas

#### Ход работы:

1. Создал свой общедоступный репозиторий на Github, выбрав лицензию MIT и язык программирования Python.
2. Дополнил файл .gitignore необходимыми правилами для работы с IDE PyCharm:
3. Организовал свой репозиторий в соответствии с моделью ветвления gitflow
4. Создал проект PyCharm в папке репозитория
5. Скачал файл и проверил содержимое таблицы:

```
In [110]: import pandas as pd
import numpy as np
import json
import matplotlib.pyplot as plt
%matplotlib inline
```

```
In [111]: city = pd.read_csv('city.csv')
```

Столбцы и количество строк в таблице city

6. С помощью команды select вывел информацию о том, сколько городов в каждом из федеральных округов:

```
In [124]: city.columns
```

```
Out[124]: Index(['address', 'postal_code', 'country', 'federal_district', 'region_type',
               'region', 'area_type', 'area', 'city_type', 'city', 'settlement_type',
               'settlement', 'kladr_id', 'fias_id', 'fias_level', 'capital_marker',
               'okato', 'oktmo', 'tax_office', 'timezone', 'geo_lat', 'geo_lon',
               'population', 'foundation_year'],
              dtype='object')
```

```
In [123]: len(city)
```

```
Out[123]: 1117
```

Узнали самое длинное название города в таблице

```
Ввод [137]: 1 q=[]
2 for i in city["city"]:
3     i = str(i)
4     q.append(i)
5 print(f'Самое длинное название города в таблице имеет {len(max(list, key=len))} символов')
```

Самое длинное название города в таблице имеет 12 символов

```
Ввод [138]: 1 time = city[city['federal_district'].isin(['Приволжский', 'Сибирский'])]
2 time[['timezone', 'address']].groupby('timezone').count().rename({'address': 'Количество городов'}, axis=1)
```

Out[138]:

Количество городов	
timezone	
UTC+3	101
UTC+4	41
UTC+5	58
UTC+6	6
UTC+7	86
UTC+8	22

Про

7. С помощью команды select вывел следующую информацию:  
Сколько городов в Приволжском и Уральском округах?

```
In [130]: time = city[city['federal_district'].isin(['Приволжский', 'Сибирский'])]
time[['timezone', 'address']].groupby('timezone').count().rename({'address': 'Количество городов'}, axis=1)
```

Out[130]:

Количество городов	
timezone	
UTC+3	101
UTC+4	41
UTC+5	58
UTC+6	6
UTC+7	86
UTC+8	22

8. Сформировал подзапрос: “Сколько городов было основано в каждом веке?”:

```
In [148]: q = city[['timezone', 'address']].groupby('timezone').count().sort_values('address', ascending=True)
q.rename({'address': 'Количество городов'}, axis=1)
```

Out[148]:

Количество городов	
timezone	
UTC+3	660
UTC+5	173
UTC+7	86
UTC+4	66
UTC+9	31
UTC+8	28
UTC+10	22
UTC+2	22
UTC+11	17

## 9. Индивидуальное задание

Индивидуальное задание

```
Ввод [140]: 1 bitcoin = pd.read_csv('btc.csv')
            2 bitcoin
```

Out[140]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2014-09-17	465.864014	468.174011	452.421997	457.334015	457.334015	2.105680e+07
1	2014-09-18	456.859985	456.859985	413.104004	424.440002	424.440002	3.448320e+07
2	2014-09-19	424.102997	427.834991	384.532013	394.795990	394.795990	3.791970e+07
3	2014-09-20	394.673004	423.295990	389.882996	408.903992	408.903992	3.686360e+07
4	2014-09-21	408.084991	412.425995	393.181000	398.821014	398.821014	2.658010e+07
...	...	...	...	...	...	...	...
2582	2021-10-12	57526.832031	57627.878906	54477.972656	56041.058594	56041.058594	4.108376e+10
2583	2021-10-13	56038.257813	57688.660156	54370.972656	57401.097656	57401.097656	4.168425e+10
2584	2021-10-14	57372.832031	58478.734375	56957.074219	57321.523438	57321.523438	3.661579e+10
2585	2021-10-15	57345.902344	62757.128906	56868.144531	61593.949219	61593.949219	5.178008e+10
2586	2021-10-16	61495.410156	62232.257813	61110.839844	61628.851563	61628.851563	4.469134e+10

2587 rows × 7 columns

```
Ввод [141]: 1 len(bitcoin)
```

Out[141]: 2587

```
Ввод [142]: 1 bitcoin.columns
```

Out[142]: Index(['Date', 'Open', 'High', 'Low', 'Close', 'Adj Close', 'Volume'], dtype='object')

В эти дни открытие торгов началось со стоимости от 55000 до 57000 долларов:

```
Ввод [143]: 1 o = bitcoin[bitcoin['Open'].between(55000, 57000)]
            2 o[['Date', 'Open']]
```

Out[143]:

	Date	Open
2348	2021-02-20	55887.335938
2349	2021-02-21	56068.566406
2367	2021-03-11	55963.179688
2372	2021-03-16	55840.785156
2373	2021-03-17	56825.828125
2383	2021-03-27	55137.566406
2384	2021-03-28	55974.941406
2385	2021-03-29	55947.898438
2395	2021-04-08	56099.914063
2406	2021-04-19	56191.585938
2407	2021-04-20	55681.792969
2408	2021-04-21	56471.128906
2415	2021-04-28	55036.636719
2420	2021-05-03	56620.273438
2424	2021-05-07	56413.953125
2428	2021-05-11	55847.242188
2429	2021-05-12	56714.531250
2577	2021-10-07	55338.625000
2583	2021-10-13	56038.257813

Наибольшая стоимость на каждые сутки начиная с 17.09.2014

```
Ввод [144]: 1 d = weather[['Date', 'High']].groupby('Date')
2 h = d.max()
3 h.rename({'High': 'Максимальная стоимость bitcoin'}, axis=1)
```

```
Out[144]:
```

Максимальная стоимость bitcoin	
Date	
2014-09-17	468.174011
2014-09-18	456.859985
2014-09-19	427.834991
2014-09-20	423.295990
2014-09-21	412.425995
...	...
2021-10-12	57627.878906
2021-10-13	57688.660156
2021-10-14	58478.734375
2021-10-15	62757.128906
2021-10-16	62232.257813

2587 rows × 1 columns

Самая большая стоимость за все время:

```
Ввод [148]: 1 list=[]
2 for c in bitcoin["High"]:
3     c = str(c)
4     list.append(c)
5 print(f'Самая большая стоимость за все время состоит из {len(max(list, key=len))} символов')
```

Самая большая стоимость за все время состоит из 12 символов

10. Выгрузил данные в CSV файл и проверил его содержимое;

Случаи когда торги закрывались на определенной стоимости. Сохранено в csv файл

```
Ввод [149]: 1 a = bitcoin[bitcoin['Close'].isin(['408.903992'])]
2 a = a[['Date', 'Close', 'Volume']]
3 a.to_csv('close.csv', encoding='utf-8', index=False, sep='|')
```

11. Экспортировал данные таблицы в формат json:

Показатели на 16.10.2021г представленные в формат json

```
Ввод [146]: 1 today = bitcoin[bitcoin['Date'].isin(['2021-10-16'])]
2 today = today[['Open', 'High', 'Low', 'Close', 'Volume']]
3 result = today.to_json(orient="records")
4 parsed = json.loads(result)
5
```

Показатели на 16.10.2021г

```
Ввод [147]: 1 today = bitcoin[bitcoin['Date'].isin(['2021-10-16'])]
2 today = today[['Open', 'High', 'Low', 'Close', 'Volume']]
3 today
```

Out[147]:

	Open	High	Low	Close	Volume
<b>2586</b>	61495.410156	62232.257813	61110.839844	61628.851563	4.469134e+10

Контрольные вопросы:

1. Какие существуют средства для импорта данных в SQLite?

```
.import --csv city.csv city
```

```
.mode csv
```

```
.import city.csv city
```

2. В чем недостатки локальных и централизованных СКВ?

Локальные СКВ

Легко запутаться в файлах и в каком конкретно ты находишься, можно изменить не тот файл или сохранить в не нужной директории

Централизованные

Централизованный сервер является уязвимым местом всей системы. Если сервер выключается на час, то в течение часа разработчики не могут взаимодействовать, и никто не может сохранить новые версии. Если же повреждается диск с центральной базой данных и нет резервной копии, вы теряете абсолютно всё - всю историю проекта, разве что за исключением

нескольких рабочих версий, сохранившихся на рабочих машинах пользователей. Локальные системы управления версиями подвержены той же проблеме: если вся история проекта хранится в одном месте, вы рискуете потерять всё.

3. Каково назначение команды `.schema` ? показывает список и структуру всех таблиц в базе

4. Как выполняется группировка и сортировка данных в запросах SQLite?

Группировка производится, когда в оператора `group` используется функция `by`

5. Каково назначение "табличных выражений" в SQLite?

Наглядная демонстрация содержимого в таблице

6. Как осуществляется экспорт данных из SQLite в форматы CSV и JSON?

```
sqlite> .mode csv
sqlite> .once samara.csv
sqlite> select kladr_id, city from city where region = 'Самарская';
sqlite> .exit
.mode json
```

```
select kladr_id, city from city where region = 'Самарская' limit 3;
```

7. Какие еще форматы для экспорта данных Вам известны?

```
.mode insert cities
```

```
.mode markdown
```

```
.mode html
```