

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
Зырянов Арсений Вадимович

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

Вариант 8

Ход работы:

1. Создал свой общедоступный репозиторий на Github, выбрав лицензию MIT и язык программирования Python.

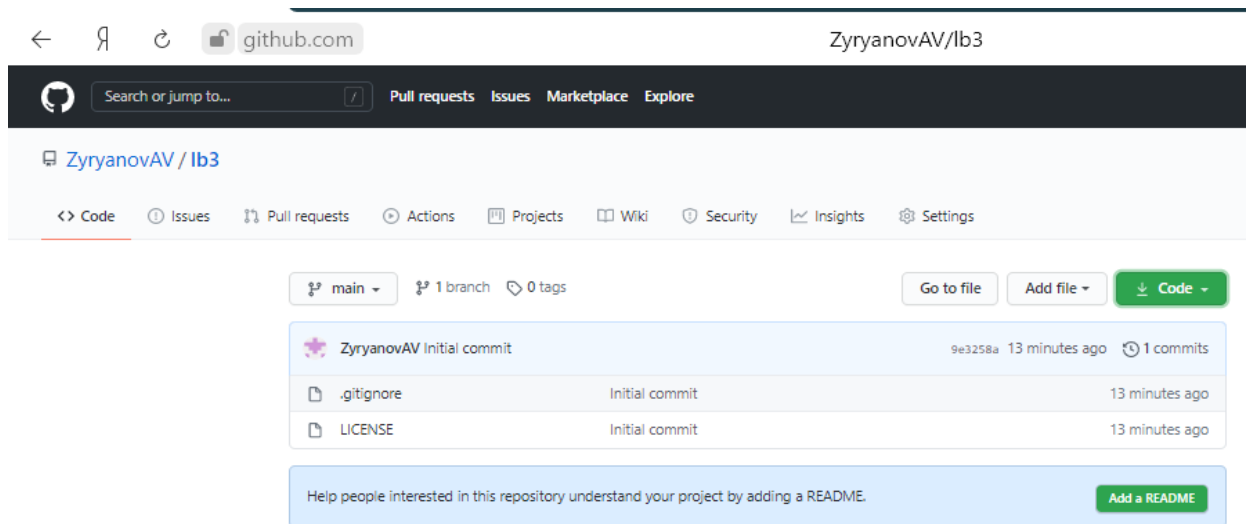


Рисунок 1. Создание нового репозитория

2. Дополнил файл .gitignore необходимыми правилами для работы с IDE PyCharm:



Рисунок 2. Файл .gitignore

3. Организовал свой репозиторий в соответствии с моделью ветвления git-flow

```
C:\Users\Orion>git config --global user.email ar200103zeranov@yandex.ru
C:\Users\Orion>git status
fatal: not a git repository (or any of the parent directories): .git
C:\Users\Orion>cd lb3
C:\Users\Orion\lb3>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   .gitignore

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        arithmetic.py
        individual.py
        numbers.py
        user.py

no changes added to commit (use "git add" and/or "git commit -a")
C:\Users\Orion\lb3>git checkout -b develop
Switched to a new branch 'develop'
```

Рисунок 3. Создал модель ветвления git-flow

4. Создал проект PyCharm в папке репозитория

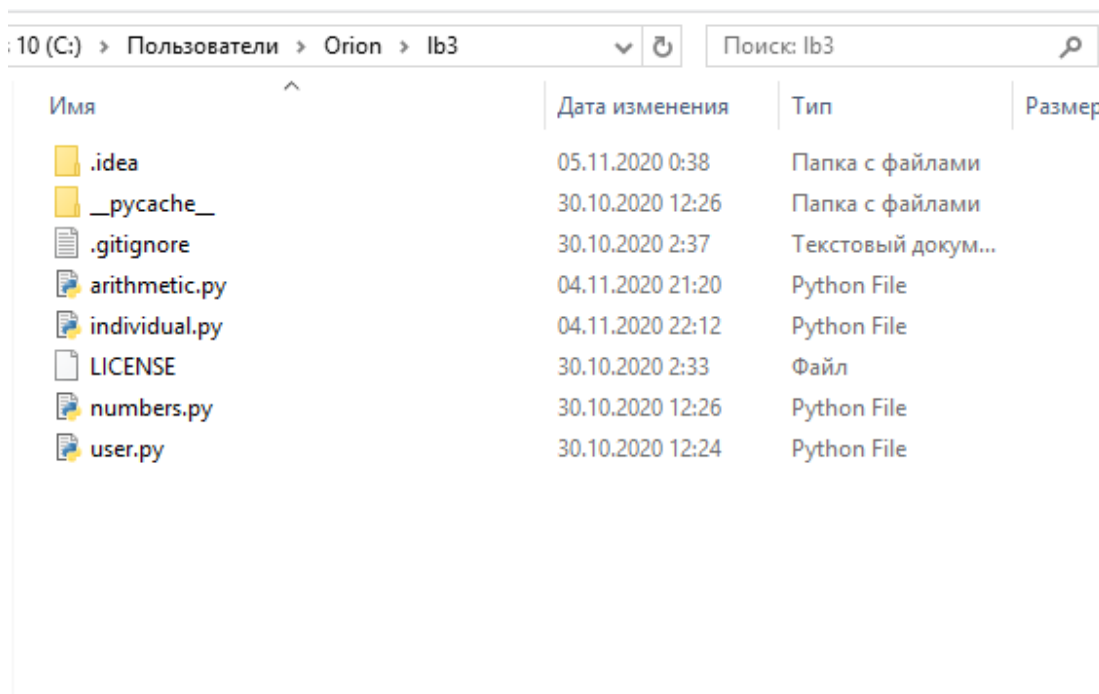


Рисунок 4. Папка репозитория, проект PyCharm

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
```

7. С помощью команды select вывел следующую информацию:

- 1) У каких городов в названии есть слово «Красный»?

Узнали самое длинное название города в таблице

```
In [128]: q=[]
for i in city["city"]:
    i = str(i)
    q.append(i)
print(f'Самое длинное название города в таблице имеет {len(max(list, key=len))} символов')
```

Самое длинное название города в таблице имеет 25 символов

Количество городов для каждого часового пояса в Сибирском и Приволжском федеральных округах
 Ответ: 58

- 2) Какие города появились за последние 30 лет?

```
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

- 3) Сколько городов в Приволжском и Уральском округах?

```
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

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

```
In [116]: weather = pd.read_csv('aus_weather.csv')
weather
```

Out[116]:

	Year	Month	Day	rainfall_mm	min_temp_C	max_temp_C	daily_avg	daily_range	uv_MJ_m*m	Season
0	1944	5	1	0.0	NaN	NaN	0.00	0.0	NaN	3
1	1944	5	2	0.0	NaN	NaN	0.00	0.0	NaN	3
2	1944	5	3	0.0	NaN	NaN	0.00	0.0	NaN	3
3	1944	5	4	4.3	NaN	NaN	0.00	0.0	NaN	3
4	1944	5	5	0.0	NaN	NaN	0.00	0.0	NaN	3
...
26538	2016	12	27	0.4	16.3	23.2	19.75	6.9	26.8	2
26539	2016	12	28	0.6	8.9	23.5	16.20	14.6	30.3	2
26540	2016	12	29	0.0	9.5	24.8	17.15	15.3	31.6	2
26541	2016	12	30	0.0	10.6	24.2	17.40	13.6	30.6	2
26542	2016	12	31	0.0	12.1	28.9	20.50	16.8	31.5	2

26543 rows x 10 columns

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

время года и средняя температура за дни когда осадки были от 3.5 мм до 4 мм

```
In [132]: q = weather[weather['rainfall_mm'].between(3.5, 4.0)]
          q[['daily_avg', 'Season']].rename({'daily_avg': 'Средняя температура по Цельсию'}, axis=1)
```

```
Out[132]:
```

	Средняя температура по Цельсию	Season
73	14.25	4
148	16.65	1
374	16.35	3
409	11.10	4
418	15.65	4
...
26380	10.90	4
26393	13.80	4
26394	12.15	4
26454	16.25	1
26465	14.50	1

301 rows x 2 columns

10. Познакомился с настройками экспорта в CSV файл(для наглядности выводил на экран):

```
In [118]: q = weather[['Year', 'max_temp_C']].groupby('Year')
          c = q.max()
          c.rename({'max_temp_C': 'Максимальная температура за год'}, axis=1)
```

```
Out[118]:
```

	Максимальная температура за год
Year	
1944	38.7
1945	41.8
1946	38.8
1947	38.7
1948	42.8
...	...
2012	42.2
2013	41.1
2014	43.5
2015	44.2
2016	43.0

73 rows x 1 columns

11. Вставил данные из таблицы city в другую БД:

Случаи, когда первого числа каждого месяца, когда температура опускалась ниже чем 20 градусов. Сохранено в csv файл

```
In [134]: k = weather[weather['Day'].isin(['1'])]
          k = k[k['min_temp_C'] < 20]
          k.to_csv('1.csv', encoding='utf-8', index=False, sep='|')
```

12. Экспортировал данные таблицы в формат html:

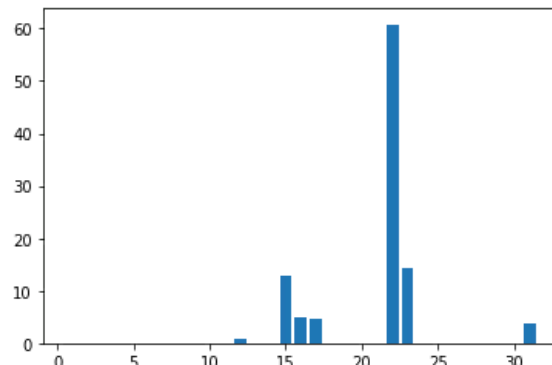
Показатели за 2016 годб представленные в формат json

```
In [ ]: weather_2016 = weather[weather['Year'].isin(['2016'])]
          weather_2016 = weather_2016[['rainfall_mm', 'min_temp_C', 'max_temp_C', 'daily_avg']]
          result = weather_2016.to_json(orient="records")
          parsed = json.loads(result)
          parse
```

13. Импортировал данные из файла samara.csv в новую таблицу samara:

```
In [147]: q_1 = weather[weather['Year'] == 2000]
q_2 = q_1[q_1['Month'] == 1]
x = q_2['Day']
y = q_2['rainfall_mm']
plt.bar(x, y)
```

Out[147]: <BarContainer object of 31 artists>



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
```