Московский Государственный Технический Университет
им. Н.Э. Баумана

Отчет по лабораторной работе №2

по курсу
Технологии Машинного Обучения

Выполнил:
Муравьев О.М.
ИУ5-62

_____

Проверил:
Гапанюк Ю.Е.

_____

Москва, 2019

Задание

Часть 1.
Выполните первое демонстрационное задание "demo assignment" под названием "Exploratory data analysis with Pandas" со страницы курса https://mlcourse.ai/assignments
Задание заключается в том, чтобы ответить на вопросы, по повожу датасета, используя запросы pandas.

Часть 2.

Выполните следующие запросы с использованием двух различных библиотек
- Pandas и PandaSQL:

- один произвольный запрос на соединение двух наборов данных
- один произвольный запрос на группировку набора данных с использованием функций агрегирования
Сравните время выполнения каждого запроса в Pandas и PandaSQL.

Код и результаты выполнения

1. Подключаем библиотеки:

```
import numpy as np
import pandas as pd
pd.set_option('display.max.columns', 100)

%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')
```

2. Подключаем набор данных

```
data = pd.read_csv('adult.data.txt')
data.head()
```

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	salary
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K

3. Запросы

1. Как много мужчин и женщин представлено в этом наборе данных?

```
data['sex'].value_counts()
```

```
 Male      21790
 Female    10771
Name: sex, dtype: int64
```

2. Какой средний возраст женщин?

```
data.loc[data['sex'] == ' Female', 'age'].mean()
```

```
36.85823043357163
```

3. Какой процент жителей Германии?

```
(float((data['native-country'] == ' Germany').sum()) / data.shape[0])*100
```

```
0.42074874850281013
```

4,5. Среднее значение и стандартное отклонение в возрасте для тех, кто зарабатывает больше 50. тыс в год и тех, кто получает меньше 50 тысяч в год?

Зарплата больше 50 тысяч в год

Среднее значение

```
data.loc[data['salary'] == ' >50K', 'age'].mean()
```

```
44.24984058155847
```

Стандартное отклонение

```
data.loc[data['salary'] == ' >50K', 'age'].std()
```

```
10.519027719851826
```

Зарплата 50 тысяч и меньше

Среднее значение

```
data.loc[data['salary'] == ' <=50K', 'age'].mean()
```

```
36.78373786407767
```

Стандартное отклонение

```
data.loc[data['salary'] == ' <=50K', 'age'].std()
```

```
14.02008849082488
```

6. Правда ли что люди которые получают больше 50 тысяч имеют хотя бы школьное образование?

```
HighSC = {' Bachelors', ' Prof-school', ' Assoc-acdm', ' Assoc-voc', ' Masters', ' Doctorate'}
```

```
for i in data.loc[data['salary'] == ' <=50K', 'education'].unique():
    if i not in HighSC:
        print('Не правда')
        break
```

```
Не правда
```

7. Отобразите статистику вораста для каждой расы и каждого пола. Используйте groupby() и describe(). Найдите максимальный возраст мужчин Американской-инди-эскимосской расы.

```
for (race, sex), sub in data.groupby(['race', 'sex']):
    print(f"Race: {race}, sex: {sex}")
    print(sub['age'].describe())
```

```
Race:  Amer-Indian-Eskimo, sex:  Female
count    119.000000
mean      37.117647
std       13.114991
min       17.000000
25%       27.000000
50%       36.000000
75%       46.000000
max       80.000000
Name: age, dtype: float64
Race:  Amer-Indian-Eskimo, sex:  Male
count    192.000000
mean      37.208333
std       12.049563
min       17.000000
25%       28.000000
50%       35.000000
75%       45.000000
max       82.000000
Name: age, dtype: float64
Race:  Asian-Pac-Islander, sex:  Female
count    346.000000
mean      35.089595
std       12.300845
min       17.000000
25%       25.000000
50%       33.000000
75%       43.750000
max       75.000000
Name: age, dtype: float64
Race:  Asian-Pac-Islander, sex:  Male
count    693.000000
mean      39.073593
std       12.883944
min       18.000000
25%       29.000000
50%       37.000000
75%       46.000000
max       90.000000
Name: age, dtype: float64
Race:  Black, sex:  Female
count    1555.000000
mean       37.854019
std        12.637197
min        17.000000
25%        28.000000
50%        37.000000
75%        46.000000
max        90.000000
Name: age, dtype: float64
Race:  Black, sex:  Male
count    1569.000000
mean       37.682600
std        12.882612
min        17.000000
25%        27.000000
50%        36.000000
75%        46.000000
max        90.000000
Name: age, dtype: float64
Race:  Other, sex:  Female
count    109.000000
mean      31.678899
std       11.631599
min       17.000000
25%       23.000000
50%       29.000000
75%       39.000000
max       74.000000
Name: age, dtype: float64
Race:  Other, sex:  Male
count    162.000000
mean      34.654321
std       11.355531
min       17.000000
25%       26.000000
50%       32.000000
75%       42.000000
max       77.000000
Name: age, dtype: float64
Race:  White, sex:  Female
count    8642.000000
mean       36.811618
std        14.329093
min        17.000000
25%        25.000000
50%        35.000000
75%        46.000000
max        90.000000
Name: age, dtype: float64
Race:  White, sex:  Male
count    19174.000000
mean        39.652498
std         13.436029
min         17.000000
25%         29.000000
50%         38.000000
75%         49.000000
max         90.000000
Name: age, dtype: float64
```

Определим самый большой возраст среди мужчин расы АмерканскоИндийскихЭскимо

```
cake=data.loc[data['race'] == ' Amer-Indian-Eskimo']
cake.loc[cake['sex'] == ' Male', 'age'].max()
```

```
82
```

8. Доля каких мужчин больше среди тех, кто зарабатывает больше 50 тысяч, женатых или холостяков?

```
not_married_men = data.loc[(data['sex'] == ' Male') &
     (data['marital-status'].isin([' Never-married',
                                   ' Separated',
                                   ' Divorced',
                                   ' Widowed']))]

married_men = data.loc[(data['sex'] == ' Male') &
     (data['marital-status'].isin([' Married-civ-spouse',
                                   ' Married-spouse-absent',
                                   ' Married-AF-spouse']))]
print (f"Доля неженатых мужчин {(not_married_men['salary'] == ' >50K').sum()}")
print (f"Доля женатых мужчин {(married_men['salary'] == ' >50K').sum()}\n")

if ((not_married_men['salary'] == ' >50K').sum() > (married_men['salary'] == ' >50K').sum()):
       print('Доля неженатых мужчин больше')
elif ((married_men['salary'] == ' >50K').sum() > (not_married_men['salary'] == ' >50K').sum()):
       print('Доля женатых мужчин больше')
else:
       print('Доли женатых и неженатых мужчин равны')
```

```
Доля неженатых мужчин 697
Доля женатых мужчин 5965

Доля женатых мужчин больше
```

9. Какое максимальное количество часов человек работает в неделю? Как много людей работают столько часов и каков процент тех кто зарабатывает больше 50 тысяч среди них?

```
maxxi = (data['hours-per-week']).max()
print (f"Максимальное количество часов в неделю: {maxxi}")

coun = data.loc[data['hours-per-week'] == 99]
countn = coun.shape[0]
print (f"Количество работающих :столько времени {countn}")

perc = float(coun.loc[data['salary'] == ' >50K'].shape[0]) / countn * 100
print (f"Процент тех, кто зарабатывает более 50 тысяч {perc}")
```

```
Максимальное количество часов в неделю: 99
Количество работающих :столько времени 85
Процент тех, кто зарабатывает более 50 тысяч 29.411764705882355
```

10. Посчитаете среднее время работы в неделю для тех кто получает много и мало, для каждой страны. Какими они будут для Японии?

```
rich = data.loc[data['salary'] == ' >50K']
poor = data.loc[data['salary'] == ' <=50K']

print ("Среднее поличество часов работы в неделю \n")
for country in data['native-country'].unique():
    print(country)
    print(f"Зарплата больше 50 тысяч: {rich.loc[rich['native-country'] == country, 'hours-per-week'].mean()}")
    print(f"Зарплата меньше 50 тысяч: {poor.loc[poor['native-country'] == country, 'hours-per-week'].mean()}\n")
```

```
Среднее поличество часов работы в неделю

 United-States
Зарплата больше 50 тысяч: 45.50536884674383
Зарплата меньше 50 тысяч: 38.79912723305605

 Cuba
Зарплата больше 50 тысяч: 42.44
Зарплата меньше 50 тысяч: 37.98571428571429

 Jamaica
Зарплата больше 50 тысяч: 41.1
Зарплата меньше 50 тысяч: 38.23943661971831

 India
Зарплата больше 50 тысяч: 46.475
Зарплата меньше 50 тысяч: 38.233333333333334

 ?
Зарплата больше 50 тысяч: 45.54794520547945
Зарплата меньше 50 тысяч: 40.16475972540046

 Poland
Зарплата больше 50 тысяч: 39.0
Зарплата меньше 50 тысяч: 38.166666666666664

 Columbia
Зарплата больше 50 тысяч: 50.0
Зарплата меньше 50 тысяч: 38.68421052631579

 Cambodia
Зарплата больше 50 тысяч: 40.0
Зарплата меньше 50 тысяч: 41.416666666666664

 Thailand
Зарплата больше 50 тысяч: 58.333333333333336
Зарплата меньше 50 тысяч: 42.86666666666667

 Ecuador
Зарплата больше 50 тысяч: 48.75
Зарплата меньше 50 тысяч: 38.041666666666664

 Mexico
Зарплата больше 50 тысяч: 46.57575757575758
Зарплата меньше 50 тысяч: 40.00327868852459

 South
Зарплата больше 50 тысяч: 51.4375
Зарплата меньше 50 тысяч: 40.15625

 Puerto-Rico
Зарплата больше 50 тысяч: 39.416666666666664
Зарплата меньше 50 тысяч: 38.470588235294116

 Honduras
Зарплата больше 50 тысяч: 60.0
Зарплата меньше 50 тысяч: 34.333333333333336

 England
Зарплата больше 50 тысяч: 44.53333333333333
Зарплата меньше 50 тысяч: 40.483333333333334

 Laos
Зарплата больше 50 тысяч: 40.0
Зарплата меньше 50 тысяч: 40.375

 Taiwan
Зарплата больше 50 тысяч: 46.8
Зарплата меньше 50 тысяч: 33.774193548387096

 Haiti
Зарплата больше 50 тысяч: 42.75
Зарплата меньше 50 тысяч: 36.325

 Portugal
Зарплата больше 50 тысяч: 41.5
Зарплата меньше 50 тысяч: 41.93939393939394

 Dominican-Republic
Зарплата больше 50 тысяч: 47.0
Зарплата меньше 50 тысяч: 42.338235294117645

 Canada
Зарплата больше 50 тысяч: 45.64102564102564
Зарплата меньше 50 тысяч: 37.91463414634146

 Germany
Зарплата больше 50 тысяч: 44.97727272727273
Зарплата меньше 50 тысяч: 39.13978494623656

 Iran
Зарплата больше 50 тысяч: 47.5
Зарплата меньше 50 тысяч: 41.44

 Philippines
Зарплата больше 50 тысяч: 43.032786885245905
Зарплата меньше 50 тысяч: 38.065693430656935

 Italy
Зарплата больше 50 тысяч: 45.4
Зарплата меньше 50 тысяч: 39.625

 El-Salvador
Зарплата больше 50 тысяч: 45.0
Зарплата меньше 50 тысяч: 36.03092783505155

 France
Зарплата больше 50 тысяч: 50.75
Зарплата меньше 50 тысяч: 41.05882352941177

 Guatemala
Зарплата больше 50 тысяч: 36.666666666666664
Зарплата меньше 50 тысяч: 39.36065573770492

 China
Зарплата больше 50 тысяч: 38.9
Зарплата меньше 50 тысяч: 37.38181818181818

 Japan
Зарплата больше 50 тысяч: 47.958333333333336
Зарплата меньше 50 тысяч: 41.0
```

```
 Yugoslavia
Зарплата больше 50 тысяч: 49.5
Зарплата меньше 50 тысяч: 41.6

 Peru
Зарплата больше 50 тысяч: 40.0
Зарплата меньше 50 тысяч: 35.06896551724138

 Outlying-US(Guam-USVI-etc)
Зарплата больше 50 тысяч: nan
Зарплата меньше 50 тысяч: 41.857142857142854

 Scotland
Зарплата больше 50 тысяч: 46.666666666666664
Зарплата меньше 50 тысяч: 39.44444444444444

 Trinadad&Tobago
Зарплата больше 50 тысяч: 40.0
Зарплата меньше 50 тысяч: 37.05882352941177

 Greece
Зарплата больше 50 тысяч: 50.625
Зарплата меньше 50 тысяч: 41.80952380952381

 Nicaragua
Зарплата больше 50 тысяч: 37.5
Зарплата меньше 50 тысяч: 36.09375

 Vietnam
Зарплата больше 50 тысяч: 39.2
Зарплата меньше 50 тысяч: 37.193548387096776

 Hong
Зарплата больше 50 тысяч: 45.0
Зарплата меньше 50 тысяч: 39.142857142857146

 Ireland
Зарплата больше 50 тысяч: 48.0
Зарплата меньше 50 тысяч: 40.94736842105263

 Hungary
Зарплата больше 50 тысяч: 50.0
Зарплата меньше 50 тысяч: 31.3

 Holand-Netherlands
Зарплата больше 50 тысяч: nan
Зарплата меньше 50 тысяч: 40.0
```

```
print("Среднее количество часов для Японии")
print(f"Зарплата больше 50 тысяч: {rich.loc[rich['native-country'] == ' Japan', 'hours-per-week'].mean()}")
print(f"Зарплата меньше 50 тысяч: {poor.loc[poor['native-country'] == ' Japan', 'hours-per-week'].mean()}\n")
```

```
Среднее количество часов для Японии
Зарплата больше 50 тысяч: 47.958333333333336
Зарплата меньше 50 тысяч: 41.0
```

4. Часть 2

Импортируем библиотеки и данные

```
import pandas as pd
import pandasql as ps
import time
```

```
Data1 = pd.read_csv("user_device.csv")
Data2 = pd.read_csv("user_usage.csv")
```

```
print(f"Data1: {Data1.shape}")
print(f"Data2: {Data2.shape}")
```

```
Data1: (272, 6)
Data2: (240, 4)
```

```
Data1.head()
```

	use_id	user_id	platform	platform_version	device	use_type_id
0	22782	26980	ios	10.2	iPhone7,2	2
1	22783	29628	android	6.0	Nexus 5	3
2	22784	28473	android	5.1	SM-G903F	1
3	22785	15200	ios	10.2	iPhone7,2	3
4	22786	28239	android	6.0	ONE E1003	1

```
Data2.head()
```

	outgoing_mins_per_month	outgoing_sms_per_month	monthly_mb	use_id
0	21.97	4.82	1557.33	22787
1	1710.08	136.88	7267.55	22788
2	1710.08	136.88	7267.55	22789
3	94.46	35.17	519.12	22790
4	71.59	79.26	1557.33	22792

Объединение таблиц

```
start=time.time()
result = pd.merge(Data2, Data1[['use_id', 'platform', 'device']],
                  on='use_id', how='inner', indicator=True)
end = time.time()
print('Time: ', end-start)
```

```
Time:  0.014614105224609375
```

```
result.shape
```

```
(159, 7)
```

```
sqlqur = f'''SELECT * FROM Data1 join Data2 on Data1.use_id=Data2.use_id'''
```

```
start=time.time()
avg = ps.sqldf(sqlqur, locals())
end=time.time()
print('Time: ', end-start)
```

```
Time:  0.03294801712036133
```

```
avg.shape
```

```
(159, 10)
```

Группировка

```
start=time.time()
df=Data1.groupby(by='platform')
end=time.time()
print('Time: ', end-start)
```

```
Time:  0.0003590583801269531
```

```
k=df.groups.keys()

for key in k:
    print('Key:', key)
    print(df.get_group(key).count(),'\n')
```

```
Key: android
use_id              184
user_id             184
platform            184
platform_version    184
device              184
use_type_id         184
dtype: int64

Key: ios
use_id              88
user_id             88
platform            88
platform_version    88
device              88
use_type_id         88
dtype: int64
```

```
sqlgroup = 'SELECT platform, count(use_id) from Data1 group by platform'
start=time.time()
gr=ps.sqldf(sqlgroup, locals())
end=time.time()
print('Time: ', end-start,'\n')
print(gr)
```

```
Time:  0.011040687561035156

  platform  count(use_id)
0  android            184
1      ios             88
```