МГТУ им. Н. Э. Баумана, кафедра ИУ5

курс “Технологии машинного обучения”

Лабораторная работа №2

«Изучение библиотек обработки данных»

ВЫПОЛНИЛ:

Ерохин И.А.

Группа: ИУ5-61Б

ПРОВЕРИЛ:

Гапанюк Ю.Е.

Москва 2020

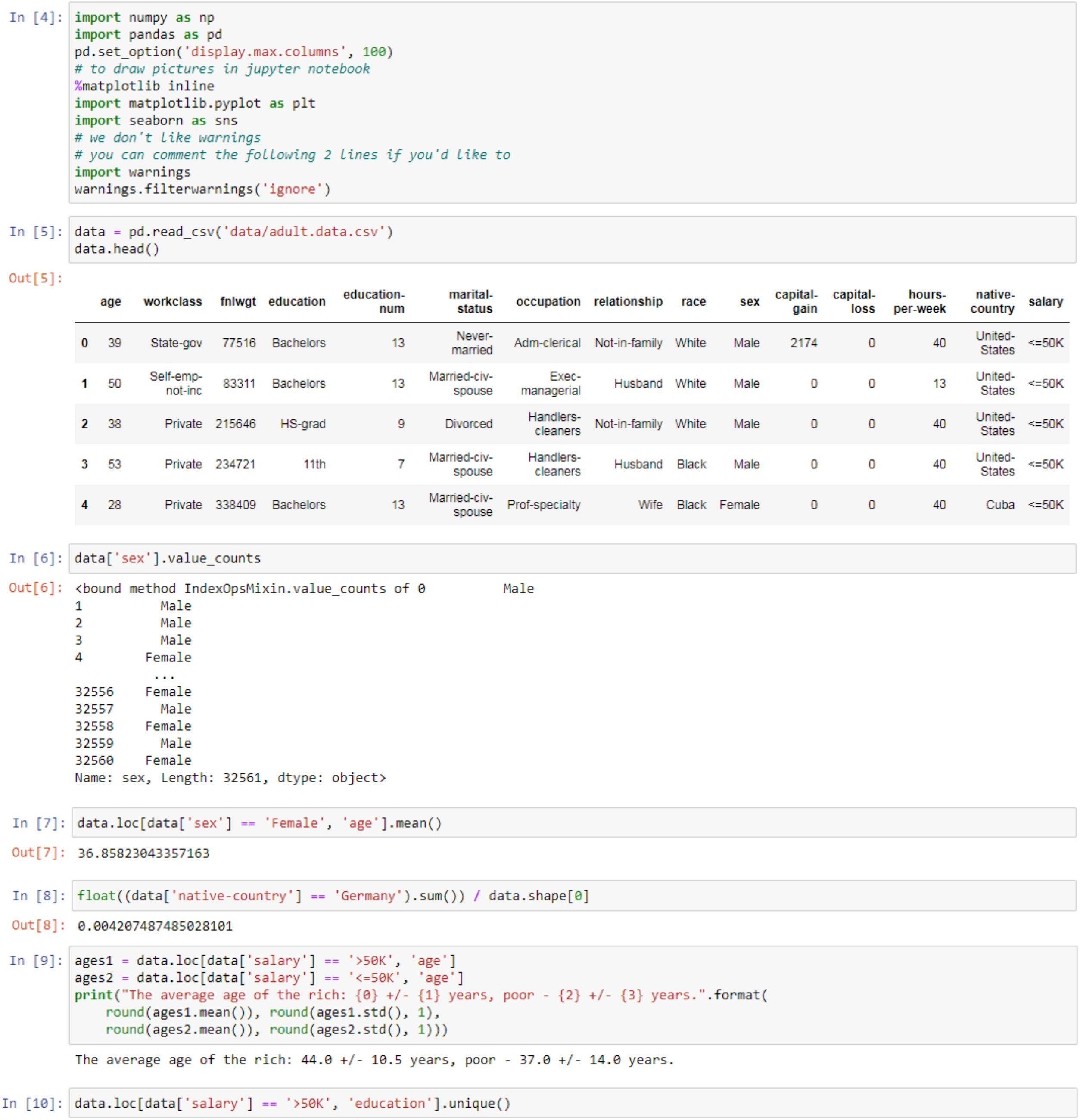
Цель лабораторной работы: изучение библиотеки обработки данных Pandas.

Задание:

- Выполните первое демонстрационное задание "demo assignment" под названием "Exploratory data analysis with Pandas" со страницы курса - https://mlcourse.ai/assignments

- Условие задания - https://nbviewer.jupyter.org/github/Yorko/mlcourse_open/blob/master/jupyter_english/assignments_demo/assignment01_pandas_uci_adult.ipynb?flush_cache=true

Выполненная работа:



In [4]:
```
import numpy as np
import pandas as pd
pd.set_option('display.max.columns', 100)
# to draw pictures in jupyter notebook
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
# we don't like warnings
# you can comment the following 2 lines if you'd like to
import warnings
warnings.filterwarnings('ignore')
```

In [5]:
```
data = pd.read_csv('data/adult.data.csv')
data.head()
```

Out[5]:

	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

In [6]:
```
data['sex'].value_counts
```

Out[6]:
```
<bound method IndexOpsMixin.value_counts of 0          Male
1          Male
2          Male
3          Male
4        Female
          ...
32556    Female
32557      Male
32558    Female
32559      Male
32560    Female
Name: sex, Length: 32561, dtype: object>
```

In [7]:
```
data.loc[data['sex'] == 'Female', 'age'].mean()
```

Out[7]:
```
36.85823043357163
```

In [8]:
```
float((data['native-country'] == 'Germany').sum()) / data.shape[0]
```

Out[8]:
```
0.004207487485028101
```

In [9]:
```
ages1 = data.loc[data['salary'] == '>50K', 'age']
ages2 = data.loc[data['salary'] == '<=50K', 'age']
print("The average age of the rich: {0} +/- {1} years, poor - {2} +/- {3} years.".format(
    round(ages1.mean()), round(ages1.std(), 1),
    round(ages2.mean()), round(ages2.std(), 1)))
```

```
The average age of the rich: 44.0 +/- 10.5 years, poor - 37.0 +/- 14.0 years.
```

In [10]:
```
data.loc[data['salary'] == '>50K', 'education'].unique()
```

```
Out[10]: array(['HS-grad', 'Masters', 'Bachelors', 'Some-college', 'Assoc-voc',
                'Doctorate', 'Prof-school', 'Assoc-acdm', '7th-8th', '12th',
                '10th', '11th', '9th', '5th-6th', '1st-4th'], dtype=object)
```

```
In [11]: for (race, sex), sub_df in data.groupby(['race', 'sex']):
             print("Race: {0}, sex: {1}".format(race, sex))
             print(sub_df['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
```

```
In [12]: data.loc[(data['sex'] == 'Male') &
              (data['marital-status'].isin(['Never-married',
                                            'Separated',
                                            'Divorced',
                                            'Widowed'])), 'salary'].value_counts()
```

```
Out[12]: <=50K    7552
         >50K      697
         Name: salary, dtype: int64
```

```
In [13]: data.loc[(data['sex'] == 'Male') &
              (data['marital-status'].str.startswith('Married')), 'salary'].value_counts()
```

```
Out[13]: <=50K    7576
         >50K     5965
         Name: salary, dtype: int64
```

```
In [14]: data['marital-status'].value_counts()
```

```
Out[14]: Married-civ-spouse       14976
         Never-married            10683
         Divorced                  4443
         Separated                 1025
         Widowed                    993
         Married-spouse-absent      418
         Married-AF-spouse           23
         Name: marital-status, dtype: int64
```

```
In [15]: max_load = data['hours-per-week'].max()
         print("Max time - {0} hours./week.".format(max_load))

         num_workaholics = data[data['hours-per-week'] == max_load].shape[0]
         print("Total number of such hard workers {0}".format(num_workaholics))

         rich_share = float(data[(data['hours-per-week'] == max_load)
                          & (data['salary'] == '>50K')].shape[0]) / num_workaholics
         print("Percentage of rich among them {0}%".format(int(100 * rich_share)))
```

```
Max time - 99 hours./week.
Total number of such hard workers 85
Percentage of rich among them 29%
```

```
In [16]: for (country, salary), sub_df in data.groupby(['native-country', 'salary']):
             print(country, salary, round(sub_df['hours-per-week'].mean(), 2))
```

```
? <=50K 40.16
? >50K 45.55
Cambodia <=50K 41.42
Cambodia >50K 40.0
Canada <=50K 37.91
Canada >50K 45.64
China <=50K 37.38
China >50K 38.9
Columbia <=50K 38.68
```

```
In [17]: pd.crosstab(data['native-country'], data['salary'],
                    values=data['hours-per-week'], aggfunc=np.mean).T
```

Out[17]:

native-country	?	Cambodia	Canada	China	Columbia	Cuba	Dominican-Republic	Ecuador	El-Salvador	England	France	Germany	Greece	Gu
salary														
<=50K	40.164760	41.416667	37.914634	37.381818	38.684211	37.985714	42.338235	38.041667	36.030928	40.483333	41.058824	39.139785	41.809524	39
>50K	45.547945	40.000000	45.641026	38.900000	50.000000	42.440000	47.000000	48.750000	45.000000	44.533333	50.750000	44.977273	50.625000	36