Лабораторна робота № 0.
Використання основних функцій бібліотеки Pandas

In [6]:
```
import numpy as np
import pandas as pd
```

In [7]:
```
np.set_printoptions(precision=2)
```

Доступ до даних на google drive, якщо ви відкриваєте блокнот в **google colab**, а не на PC, можна отримати шляхом монтування google drive

In [2]:
```
from google.colab import drive
drive.mount('/content/gdrive')
```

```
Mounted at /content/gdrive
```

In [3]:
```
!ls gdrive/'My Drive'/TEACHING/IntroDataScience/intro_to_data_science/Lab_1_2/da
```

```
adult.data.csv  beauty.csv  titanic_test.csv  titanic_train.csv
```

In [4]:
```
# шлях до папки з даними на моєму google drive, відредагуйте згідно вашого випадк
data_folder = "gdrive/My Drive/TEACHING/IntroDataScience/intro_to_data_science/La
```

Зчитуємо дані з файлу

In [8]:
```
#data = pd.read_csv('data/beauty.csv', sep=';')

data = pd.read_csv(data_folder+'/beauty.csv', sep=';')
data.head()
```

Out[8]:

	wage	exper	union	goodhlth	black	female	married	service	educ	looks
0	5.73	30	0	1	0	1	1	1	14	4
1	4.28	28	0	1	0	1	1	0	12	3
2	7.96	35	0	1	0	1	0	0	10	4
3	11.57	38	0	1	0	0	1	1	16	3
4	11.42	27	0	1	0	0	1	0	16	3

In [ ]:
```
type(data)
```

Out[4]:
```
pandas.core.frame.DataFrame
```

Дивимося на перші 5 рядків

In [ ]:

```
data.head()
```

Out[5]:

	wage	exper	union	goodhlth	black	female	married	service	educ	looks
0	5.73	30	0	1	0	1	1	1	14	4
1	4.28	28	0	1	0	1	1	0	12	3
2	7.96	35	0	1	0	1	0	0	10	4
3	11.57	38	0	1	0	0	1	1	16	3
4	11.42	27	0	1	0	0	1	0	16	3

In [ ]:

```
data.shape
```

Out[6]:

```
(1260, 10)
```

Коротка статистика – info і describe

In [ ]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1260 entries, 0 to 1259
Data columns (total 10 columns):
wage        1260 non-null float64
exper       1260 non-null int64
union       1260 non-null int64
goodhlth    1260 non-null int64
black       1260 non-null int64
female      1260 non-null int64
married     1260 non-null int64
service     1260 non-null int64
educ        1260 non-null int64
looks       1260 non-null int64
dtypes: float64(1), int64(9)
memory usage: 98.5 KB
```

In [ ]:

```
data.describe()
```

Out[8]:

	wage	exper	union	goodhlth	black	female	married
count	1260.000000	1260.000000	1260.000000	1260.000000	1260.000000	1260.000000	1260.000000
mean	6.306690	18.206349	0.272222	0.933333	0.073810	0.346032	0.691270
std	4.660639	11.963485	0.445280	0.249543	0.261564	0.475892	0.462153
min	1.020000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	3.707500	8.000000	0.000000	1.000000	0.000000	0.000000	0.000000
50%	5.300000	15.000000	0.000000	1.000000	0.000000	0.000000	1.000000
75%	7.695000	27.000000	1.000000	1.000000	0.000000	1.000000	1.000000
max	77.720000	48.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Индексация

In [ ]:

```
data['exper'].head()
```

Out[9]:

```
0    30
1    28
2    35
3    38
4    27
Name: exper, dtype: int64
```

loc та iloc

In [ ]:

```
data.loc[0:5, ['wage', 'female']]
```

Out[10]:

	wage	female
0	5.73	1
1	4.28	1
2	7.96	1
3	11.57	0
4	11.42	0
5	3.91	1

In [ ]:

```
data.iloc[:,2:4].head()
```

Out[11]:

	union	goodhlth
0	0	1
1	0	1
2	0	1
3	0	1
4	0	1

Логічна індексація

In [ ]:

```
data[data['female'] == 1]['wage'].mean(), \
data[data['female'] == 0]['wage'].mean()
```

Out[12]:

```
(4.299357798165136, 7.3688228155339734)
```

In [ ]:

```
data[(data['female'] == 0) & (data['married'] == 1)]['wage'].median(), \
data[(data['female'] == 0) & (data['married'] == 0)]['wage'].median()
```

Out[13]:

```
(6.710000000000001, 5.0649999999999995)
```

Groupby

In [ ]:

```
for look, sub_df in data.groupby('looks'):
    print(look)

    # що заодно
    print(sub_df['goodhlth'].mean())
```

```
1
0.8461538461538461
2
0.9366197183098591
3
0.9210526315789473
4
0.9560439560439561
5
1.0
```

In [ ]:

```
data.groupby('looks')[['wage', 'exper']].agg(np.median)
```

Out[15]:

	wage	exper
looks		
1	3.460	32.0
2	4.595	18.0
3	5.635	18.0
4	5.240	12.5
5	4.810	8.0

Сводная таблица

In [ ]:

```
pd.crosstab(data['female'], data['married'])
```

Out[16]:

married	0	1
female		
0	166	658
1	223	213

In [ ]:

```
pd.crosstab(data['female'], data['looks'])
```

Out[17]:

looks	1	2	3	4	5
female					
0	8	88	489	228	11
1	5	54	233	136	8

Добавление столбцов (построение признаков)

In [ ]:

```
data['is_rich'] = (data['wage'] > 
                   data['wage'].quantile(.75)).astype('int64')
```

In [ ]:

```
data.head()
```

Out[19]:

	wage	exper	union	goodhlth	black	female	married	service	educ	looks	is_rich
0	5.73	30	0	1	0	1	1	1	14	4	0
1	4.28	28	0	1	0	1	1	0	12	3	0
2	7.96	35	0	1	0	1	0	0	10	4	1
3	11.57	38	0	1	0	0	1	1	16	3	1
4	11.42	27	0	1	0	0	1	0	16	3	1

In [ ]:

```
data['rubbish'] = .56 * data['wage'] + 0.32 * data['exper']
```

map и apply

In [ ]:

```
def string_gender(female):
    return 'female' if female else 'male'
```

In [ ]:

```
d =  {1: 'union', 0: 'non-union'}
```

In [ ]:

```
data['union'].map(d).head()
```

Out[23]:

```
0    non-union
1    non-union
2    non-union
3    non-union
4    non-union
Name: union, dtype: object
```

In [ ]:

```
data['female'].apply(lambda female: 'female' if female else 'male').head()
```

Out[24]:

```
0    female
1    female
2    female
3      male
4      male
Name: female, dtype: object
```