Topic 1. Exploratory data analysis with Pandas

Practice. Analyzing "Titanic" passengers

Fill in the missing code ("You code here").

In [2]:

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt

# Graphics in SVG format are more sharp and legible
%config InlineBackend.figure_format = 'svg'
pd.set_option("display.precision", 2)
```

Read data into a Pandas DataFrame

In [3]:

```
data = pd.read_csv("titanic_train.csv", index_col="PassengerId")
```

First 5 rows

In [4]:

```
data.head(5)
```

Out[4]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
PassengerId								
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599
3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803
5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450

In [5]:

```
data.describe()
```

Out[5]:

	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.00	891.00	714.00	891.00	891.00	891.00
mean	0.38	2.31	29.70	0.52	0.38	32.20
std	0.49	0.84	14.53	1.10	0.81	49.69
min	0.00	1.00	0.42	0.00	0.00	0.00
25%	0.00	2.00	20.12	0.00	0.00	7.91
50%	0.00	3.00	28.00	0.00	0.00	14.45
75%	1.00	3.00	38.00	1.00	0.00	31.00
max	1.00	3.00	80.00	8.00	6.00	512.33

Let's select those passengers who embarked in Cherbourg (Embarked=C) and paid > 200 pounds for their ticker (fare > 200).

Make sure you understand how actually this construction works.

In [6]:

```
data[(data["Embarked"] == "C") & (data.Fare > 200)].head()
```

Out[6]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticke
PassengerId								
119	0	1	Baxter, Mr. Quigg Edmond	male	24.0	0	1	P 1755
259	1	1	Ward, Miss. Anna	female	35.0	0	0	P 1775
300	1	1	Baxter, Mrs. James (Helene DeLaudeniere Chaput)	female	50.0	0	1	P 1755
312	1	1	Ryerson, Miss. Emily Borie	female	18.0	2	2	P 1760
378	0	1	Widener, Mr. Harry Elkins	male	27.0	0	2	11350

We can sort these people by Fare in descending order.

In [7]:

```
data[(data["Embarked"] == "C") & (data["Fare"] > 200)].sort_values(
    by="Fare", ascending=False
).head()
```

Out[7]:

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
259	1	1	Ward, Miss. Anna	female	35.0	0	0	PC 17755	5
680	1	1	Cardeza, Mr. Thomas Drake Martinez	male	36.0	0	1	PC 17755	5
738	1	1	Lesurer, Mr. Gustave J	male	35.0	0	0	PC 17755	5
312	1	1	Ryerson, Miss. Emily Borie	female	18.0	2	2	PC 17608	2
743	1	1	Ryerson, Miss. Susan Parker "Suzette"	female	21.0	2	2	PC 17608	2

Let's create a new feature.

In [8]:

```python
def age_category(age):
    """
    < 30 -> 1
    >= 30, <55 -> 2
    >= 55 -> 3
    """
    if age < 30:
        return 1
    elif age < 55:
        return 2
    elif age >= 55:
        return 3
```

In [9]:

```python
age_categories = [age_category(age) for age in data.Age]
data["Age_category"] = age_categories
```

Another way is to do it with `apply`.

In [10]:

```python
data["Age_category"] = data["Age"].apply(age_category)
```

1. How many men/women were there onboard?

- 577 men and 314 women

In [11]:

```
data["Sex"].value_counts()
```

Out[11]:

```
         count
    Sex
   male    577
 female    314
```

dtype: int64

2. Print the distribution of the `Pclass` feature. Then the same, but for men and women separately. How many men from second class were there onboard?

- 108

In [12]:

```
data["Pclass"].value_counts()
```

Out[12]:

```
        count
Pclass
     3    491
     1    216
     2    184
```

dtype: int64

In [13]:

```
data[data["Sex"] == "female"]["Pclass"].value_counts()
```

Out[13]:

```
        count
Pclass
     3    144
     1     94
     2     76
```

dtype: int64

In [14]:

```
data[data["Sex"] == "male"]["Pclass"].value_counts()
```

Out[14]:

Pclass	count
3	347
1	122
2	108

dtype: int64

3. What are median and standard deviation of `Fare`?. Round to two decimals.

- median is 14.45, standard deviation is 49.69

```
In [15]: data["Fare"].describe()
```

Out[15]:

	Fare
count	891.00
mean	32.20
std	49.69
min	0.00
25%	7.91
50%	14.45
75%	31.00
max	512.33

dtype: float64

4. Is that true that the mean age of survived people is higher than that of passengers who eventually died?

- Yes
- No

```
In [16]: data[data["Survived"] == 1]["Age"].mean() > data[data["Survived"] == 0]["Age
```

```
Out[16]: False
```

5. Is that true that passengers younger than 30 y.o. survived more frequently than those older than 60 y.o.? What are shares of survived people among young and old people?

Loading [MathJax]/extensions/Safe.js among young and 22.7% among old

In [25]:

```
young = data[data["Age"] < 30]
old = data[data["Age"] > 60]
print(young[young["Survived"] == 1].count()["Survived"] / young.count()["Sur
print(old[old["Survived"] == 1].count()["Survived"] / old.count()["Survived"
```

```
0.40625
0.22727272727272727
```

6. Is that true that women survived more frequently than men? What are shares of survived people among men and women?

- 18.9% among men and 74.2% among women

In [18]:

```
male = data[data["Sex"] == "male"]
female = data[data["Sex"] == "female"]
print(male[male["Survived"] == 1].count()["Survived"] / male.count()["Surviv
print(female[female["Survived"] == 1].count()["Survived"] / female.count()["
```

```
0.18890814558058924
0.7420382165605095
```

7. What's the most popular first name among male passengers?

- John

In [19]:

```
data["first_name"] = data["Name"].apply(lambda x: x.split(" ")[-1])
```

In [20]:

```
data["first_name"].head(5)
```

Out[20]:

	first_name
PassengerId	
1	Harris
2	Thayer)
3	Laina
4	Peel)
5	Henry

dtype: object

In [21]:

```
data[data["Sex"] == "male"]["first_name"].value_counts()
```

Out[21]:

	count
first_name	
John	16
William	15
Henry	15
James	14
Jr	9
...	...
"Harry"	1
Bernard	1
Adolphe	1
Mansour	1
Howell	1

375 rows × 1 columns

dtype: int64

8. How is average age for men/women dependent on `Pclass`? Choose all correct statements:

- On average, men of 1 class are older than 40
- Men of all classes are on average older than women of the same class
- On average, passengers ofthe first class are older than those of the 2nd class who are older than passengers of the 3rd class

In [22]:
```
male.groupby("Pclass")["Age"].mean()
```

Out[22]:

	Age
Pclass	
1	41.28
2	30.74
3	26.51

dtype: float64

In [23]:
```
female.groupby("Pclass")["Age"].mean()
```

Out[23]:

	Age
Pclass	
1	34.61
2	28.72
3	21.75

dtype: float64