Topic 1. Exploratory data analysis with Pandas

Practice. Analyzing "Titanic" passengers

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

In [1]:
```
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 [2]:
```
data = pd.read_csv("titanic_train.csv", index_col="PassengerId")
```

First 5 rows

In [3]:
```
data.head(5)
```

Out[3]:

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25	NaN	S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.28	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.92	NaN	S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.10	C123	S
5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.05	NaN	S

In [4]:
```
data.describe()
```

Out[4]:

	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 [5]:
```
data[(data["Embarked"] == "C") & (data.Fare > 200)].head()
```

Out[5]:

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
119	0	1	Baxter, Mr. Quigg Edmond	male	24.0	0	1	PC 17558	247.52	B58 B60	C
259	1	1	Ward, Miss. Anna	female	35.0	0	0	PC 17755	512.33	NaN	C
300	1	1	Baxter, Mrs. James (Helene DeLaudeniere Chaput)	female	50.0	0	1	PC 17558	247.52	B58 B60	C
312	1	1	Ryerson, Miss. Emily Borie	female	18.0	2	2	PC 17608	262.38	B57 B59 B63 B66	C
378	0	1	Widener, Mr. Harry Elkins	male	27.0	0	2	113503	211.50	C82	C

We can sort these people by Fare in descending order.

In [6]:
```
data[(data["Embarked"] == "C") & (data["Fare"] > 200)].sort_values(
    by="Fare", ascending=False
).head()
```

Out[6]:

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
259	1	1	Ward, Miss. Anna	female	35.0	0	0	PC 17755	512.33	NaN	C
680	1	1	Cardeza, Mr. Thomas Drake Martinez	male	36.0	0	1	PC 17755	512.33	B51 B53 B55	C
738	1	1	Lesurer, Mr. Gustave J	male	35.0	0	0	PC 17755	512.33	B101	C
312	1	1	Ryerson, Miss. Emily Borie	female	18.0	2	2	PC 17608	262.38	B57 B59 B63 B66	C
743	1	1	Ryerson, Miss. Susan Parker "Suzette"	female	21.0	2	2	PC 17608	262.38	B57 B59 B63 B66	C

Let's create a new feature.

In [7]:
```
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 [8]:
```
age_categories = [age_category(age) for age in data.Age]
data["Age_category"] = age_categories
```

Another way is to do it with `apply`.

In [9]:
```
data["Age_category"] = data["Age"].apply(age_category)
```

1. How many men/women were there onboard?

- 412 men and 479 women
- 314 men and 577 women
- 479 men and 412 women
- 577 men and 314 women

In [18]:
```
data["Sex"].value_counts()
```

Out[18]:
```
Sex
male      577
female    314
Name: count, dtype: int64
```

Answer: 577 men and 314 women

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?

- 104
- 108
- 112
- 125

In [22]:
```
# Pclass distribution
data["Pclass"].value_counts()
```

Out[22]:
```
Pclass
3    491
1    216
2    184
Name: count, dtype: int64
```

In [42]:
```
# Women Pclass distribution
data[data["Sex"] == "female"]["Pclass"].value_counts()
```

Out[42]:
```
Pclass
3    144
1     94
2     76
Name: count, dtype: int64
```

In [43]:
```
# Men Pclass distribution
data[data["Sex"] == "male"]["Pclass"].value_counts()
```

Out[43]:
```
Pclass
3    347
1    122
2    108
Name: count, dtype: int64
```

Answer: 108

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

- median is 14.45, standard deviation is 49.69
- median is 15.1, standard deviation is 12.15
- median is 13.15, standard deviation is 35.3
- median is 17.43, standard deviation is 39.1

In [48]:
```
# Median
np.round(data["Fare"].median(), 2)
```

Out[48]:
```
np.float64(14.45)
```

In [49]:
```
# Standart Deviation
np.round(data["Fare"].std(), 2)
```

Out[49]:
```
np.float64(49.69)
```

Answer: median is 14.45, standard deviation is 49.69

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

- Yes
- No

In [122…
```
survived_mean_age = data[data["Survived"] == 1]["Age"].mean()
died_mean_age = data[data["Survived"] == 0]["Age"].mean()

survived_mean_age > died_mean_age
```

Out[122…
```
np.False_
```

Answer: No

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?

- 22.7% among young and 40.6% among old
- 40.6% among young and 22.7% among old
- 35.3% among young and 27.4% among old
- 27.4% among young and 35.3% among old

In [121…
```
younger_thirty = data[data["Age"] < 30]
older_thirty = data[data["Age"] > 60]

younder_thirty_survive_freq = np.round(younger_thirty["Survived"].mean(), 3)
older_thirty_survive_freq = np.round(older_thirty["Survived"].mean(), 3)

print(f"{younder_thirty_survive_freq:.1%} among young and {older_thirty_survive_freq:.1%} among old")
```

```
40.6% among young and 22.7% among old
```

Answer: 40.6% among young and 22.7% among old

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

- 30.2% among men and 46.2% among women
- 35.7% among men and 74.2% among women
- 21.1% among men and 46.2% among women
- 18.9% among men and 74.2% among women

In [80]:
```
men = data[data["Sex"] == "male"]
women = data[data["Sex"] == "female"]

men_survive_freq = np.round(men["Survived"].mean(), 3)
women_survive_freq = np.round(women["Survived"].mean(), 3)

print(f"{men_survive_freq:.1%} among men and {women_survive_freq:.1%} among women")
```

```
18.9% among men and 74.2% among women
```

Answer: 18.9% among men and 74.2% among women

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

- Charles
- Thomas
- William
- John

In [94]:
```
men = data[data["Sex"] == "male"]
men_names = men["Name"].apply(lambda full_name: full_name.split(",")[1].split()[1]) # Get only first name of every
men_names.value_counts().head(1)
```

Out[94]:
```
Name
William    35
Name: count, dtype: int64
```

Answer: William

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
- On average, women 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 [108…
```
data["Pclass"].value_counts()
```

Out[108…
```
Pclass
3    491
1    216
2    184
Name: count, dtype: int64
```

In [114…
```
# Get passengers from every class
class_passengers_data = []
```

```
for i in range(3):
    class_passengers_data.append(data[data["Pclass"] == i+1])
```

In [115…
```
# Men average age depending on the class
for i in range(3):
    class_data = class_passengers_data[i]
    man_average_age = np.round(class_data[class_data["Sex"] == "male"]["Age"].mean(), 2)
    print(f"For {i+1} class: average Men age is {man_average_age}")
```

```
For 1 class average Men age is 41.28
For 2 class average Men age is 30.74
For 3 class average Men age is 26.51
```

In [119…
```
# Women average age depending on the class
for i in range(3):
    class_data = class_passengers_data[i]
    man_average_age = np.round(class_data[class_data["Sex"] == "female"]["Age"].mean(), 2)
    print(f"For {i+1} class: average Women age is {man_average_age}")
```

```
For 1 class average Women age is 34.61
For 2 class average Women age is 28.72
For 3 class average Women age is 21.75
```

In [120…
```
# All passengers average age depending on the class
for i in range(3):
    class_data = class_passengers_data[i]
    man_average_age = np.round(class_data["Age"].mean(), 2)
    print(f"For {i+1} class: average age is {man_average_age}")
```

```
For 1 class: average age is 38.23
For 2 class: average age is 29.88
For 3 class: average age is 25.14
```

Answer

- 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 of the first class are older than those of the 2nd class who are older than passengers of the 3rd class