# 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 | F |
|---|---|---|---|---|---|---|---|---|---|
| **1** | 0 | 3 | Braund, Mr. Owen Harris | male | 22.0 | 1 | 0 | A/5 21171 | 7 |
| **2** | 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Th... | female | 38.0 | 1 | 0 | PC 17599 | 71 |
| **3** | 1 | 3 | Heikkinen, Miss. Laina | female | 26.0 | 0 | 0 | STON/O2. 3101282 | 7 |
| **4** | 1 | 1 | Futrelle, Mrs. Jacques Heath (Lily May Peel) | female | 35.0 | 1 | 0 | 113803 | 53 |
| **5** | 0 | 3 | Allen, Mr. William Henry | male | 35.0 | 0 | 0 | 373450 | 8 |

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 | |
|---|---|---|---|---|---|---|---|---|---|
| **119** | 0 | 1 | Baxter, Mr. Quigg Edmond | male | 24.0 | 0 | 1 | PC 17558 | 2 |
| **259** | 1 | 1 | Ward, Miss. Anna | female | 35.0 | 0 | 0 | PC 17755 | 5 |
| **300** | 1 | 1 | Baxter, Mrs. James (Helene DeLaudeniere Chaput) | female | 50.0 | 0 | 1 | PC 17558 | 2 |
| **312** | 1 | 1 | Ryerson, Miss. Emily Borie | female | 18.0 | 2 | 2 | PC 17608 | 2 |
| **378** | 0 | 1 | Widener, Mr. Harry Elkins | male | 27.0 | 0 | 2 | 113503 | 2 |

**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 |
|---|---|---|---|---|---|---|---|---|---|
| **259** | 1 | 1 | Ward, Miss. Anna | female | 35.0 | 0 | 0 | PC 17755 | 512.33 |
| **680** | 1 | 1 | Cardeza, Mr. Thomas Drake Martinez | male | 36.0 | 0 | 1 | PC 17755 | 512.33 |
| **738** | 1 | 1 | Lesurer, Mr. Gustave J | male | 35.0 | 0 | 0 | PC 17755 | 512.33 |
| **312** | 1 | 1 | Ryerson, Miss. Emily Borie | female | 18.0 | 2 | 2 | PC 17608 | 262.38 |
| **743** | 1 | 1 | Ryerson, Miss. Susan Parker "Suzette" | female | 21.0 | 2 | 2 | PC 17608 | 262.38 |

**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 [10]:
```
data["Sex"].value_counts()

#Answer: 577 men and 314 women
```

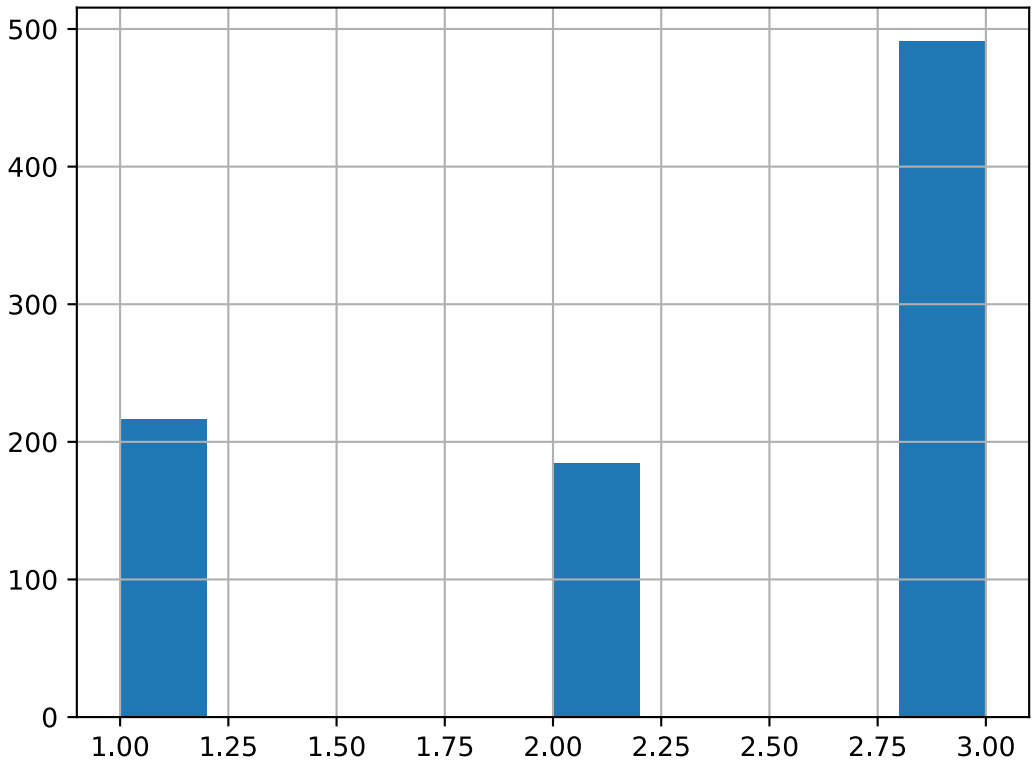
Out[10]:
```
Sex
male      577
female    314
Name: count, dtype: int64
```

**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?**

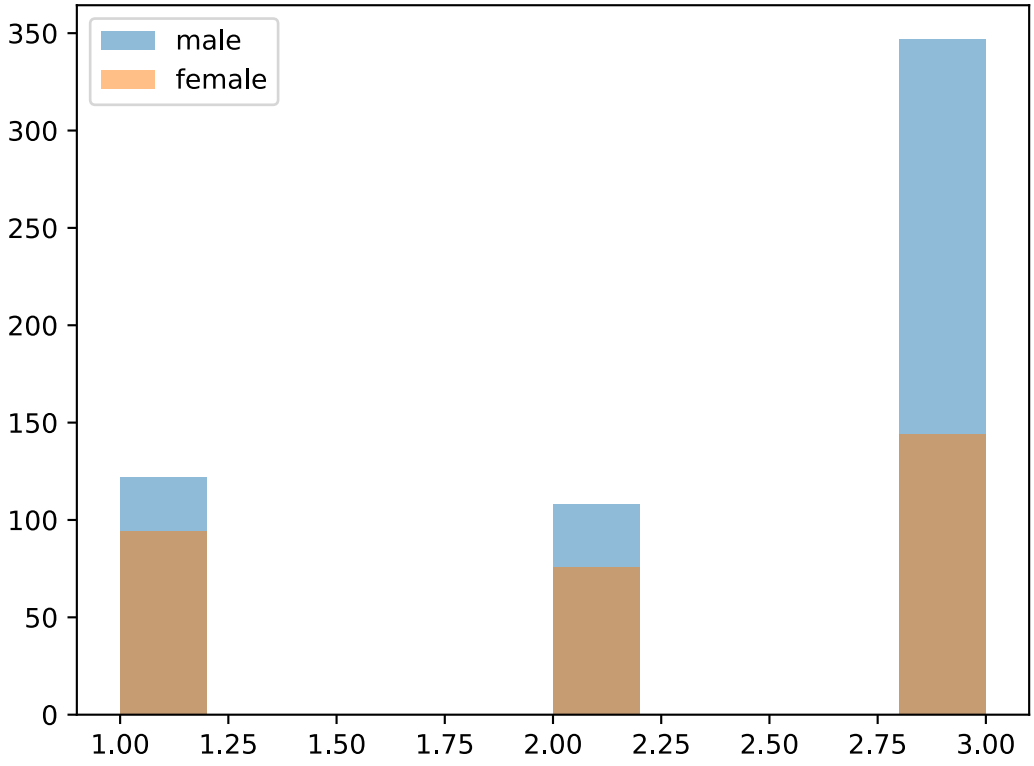
- 104
- 108
- 112
- 125

In [11]:
```
data['Pclass'].hist()
```

Out[11]: <Axes: >




In [12]:
```
for gender in data["Sex"].unique():
    plt.hist(data[data['Sex'] == gender]['Pclass'], label=gender, alpha=0
    plt.legend()
```




In [13]:
```
data[data["Sex"] == "male"]["Pclass"].value_counts()

#Answer: 108
```

Out[13]:
```
Pclass
3    347
1    122
2    108
Name: count, dtype: int64
```

**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 [14]:
```
data["Fare"].median()
```

Out[14]:
```
np.float64(14.4542)
```

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

Out[15]:
```
np.float64(49.6934285971809)
```

In [16]:
```
#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 [17]:
```
data.groupby("Survived")["Age"].mean()
```

Out[17]:
```
Survived
0    30.63
1    28.34
Name: Age, dtype: float64
```

In [18]:
```
#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 [19]:
```
cross_tab = pd.crosstab(data["Age"] < 30, data["Survived"])
print(cross_tab)
```

```
Survived    0    1
Age
False     321  186
True      228  156
```

In [20]:
```
cross_tab = pd.crosstab(data["Age"] > 60, data["Survived"])
print(cross_tab)
```

```
Survived    0    1
Age
False     532  337
True       17    5
```

In [21]:
```
print(156 / (228 + 156))
```

```
0.40625
```

In [22]:
```
print(5 / 22)
```

```
0.22727272727272727
```

In [23]:
```
#Answer: yes, 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 [24]:
```
cross_tab = pd.crosstab(data["Sex"] == "male", data["Survived"])
print(cross_tab)
```

```
Survived    0    1
Sex
False      81  233
True      468  109
```

In [25]:
```
cross_tab = pd.crosstab(data["Sex"] == "female", data["Survived"])
print(cross_tab)
```

```
Survived    0    1
Sex
False     468  109
True       81  233
```

In [26]:
```
print(109 / (468 + 109))
```

```
0.18890814558058924
```

In [27]:
```
print(233 / (233 + 81))
```

```
0.7420382165605095
```

In [28]:
```
# Answer: yes, 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 [29]:
```
from collections import Counter

male_data = data[data["Sex"] == "male"]
all_word_names = []

for male in list(male_data["Name"]):
    full = male.split(" ")
    all_word_names = all_word_names + full

counts = Counter(all_word_names)
for element, count in counts.items():
    if count > 10:
        print(f"{element}: {count}")

# Answer: William
```

```
Mr.: 517
William: 47
Henry: 28
James: 20
Master.: 40
Johan: 13
Charles: 19
Edward: 14
John: 36
Richard: 12
George: 20
Thomas: 15
Arthur: 11
Alfred: 11
```

**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 [30]:
```
female_data = data[data["Sex"] == "female"]
female_data.groupby(by=["Pclass"])["Age"].mean()
```

Out[30]:
```
Pclass
1    34.61
2    28.72
3    21.75
Name: Age, dtype: float64
```

In [31]:
```
male_data = data[data["Sex"] == "male"]
male_data.groupby(by=["Pclass"])["Age"].mean()
```

Out[31]:
```
Pclass
1    41.28
2    30.74
3    26.51
Name: Age, dtype: float64
```

In [32]:
```
print(data[(data["Sex"] == "male")]["Age"].mean())
```

```
30.72664459161148
```

In [33]:
```
data.groupby(by=["Pclass"])["Age"].mean()
```

Out[33]:
```
Pclass
1    38.23
2    29.88
3    25.14
Name: Age, dtype: float64
```

In [34]:
```
"""
Answer: On average, men of 1 class are older than 40,
        Men of all classes are on average older than women of the same cl
        On average, passengers ofthe first class are older than those of
"""
```

Out[34]:
```
'\nAnswer: On average, men of 1 class are older than 40,\n        Men of
all classes are on average older than women of the same class\n        O
n average, passengers ofthe first class are older than those of the 2nd
class who are older than passengers of the 3rd class\n'
```