PROJECT 1: EXPLORATORY DATA ANALYSIS

In this project, we explore a data set, formulate three hypotheses about relations between its features, and perform a statistical test on those hypotheses.

Description of data set
We use the Titanic data set from Kaggle

In [4]:
```
print(df.shape)
df.head()
```

```
(891, 12)
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

The data has 891 entries and the following 12 features

- PassengerID, a unique ID given to each passenger
- Survived, a binary value indicating whether the passenger survived
- Pclass, the passenger class
- Name
- Sex
- Age
- SibSp, the number of siblings or spouses
- Parch, the number of parents or children
- Ticket, a serial number
- Fare, the amount the passenger paid for their ticket
- Cabin, the type of cabin in which the passenger stayed
- Embarked, the location at which the passenger embarked

Actions taken for data cleaning and feature engineering
1. We observe that PassengerID and Name are unique for each passenger, and thus can be dropped

```
print(len(df['PassengerId'].unique()))
print(len(df['Name'].unique()))
```

```
891
891
```

2. Ticket and Cabin have significant variance, so we can probably also drop them

```
print(len(df['Ticket'].unique()))
print(len(df['Cabin'].unique()))
```

```
681
148
```

3. We observe that there is a large number of missing data for Age. For simplicity, one option is to impute the mean of ages to replace the missing values. Here, we divide the dataset into classes according to Sex and impute the mean of each class, which hopefully will afford better precision in future projects.

```
print('Number of missing ages: {}'.format(sum(df['Age'].isna())))
```

```
Number of missing ages: 177
```

```
meanM = df[df['Sex']=='male']['Age'].mean()
meanF = df[df['Sex']=='female']['Age'].mean()

for i in range(len(df)):
    if math.isnan(df['Age'][i]):
        if (df['Sex'][i] == 'male'): df['Age'].iloc[i] = meanM
        if (df['Sex'][i] == 'female'): df['Age'].iloc[i] = meanF

df['Age']
```

```
0      22.000000
1      38.000000
2      26.000000
3      35.000000
4      35.000000
         ...
886    27.000000
887    19.000000
888    27.915709
889    26.000000
890    32.000000
Name: Age, Length: 891, dtype: float64
```

4. For future projects, where categorical features potentially cause difficulties, we treat the categorical features Pclass, Embarked, and Sex by encoding them as ordinal values. There is a small number of missing values for Embarked, which we mask. For this project, we skip this step because here we do not require our features to have numeric values.

```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder

print(sum(df['Embarked'].isna()))
df['Embarked'] = df['Embarked'].fillna('Z')

le = LabelEncoder()
df[['Pclass','Embarked','Sex']] = df[['Pclass','Embarked','Sex']].apply(le.fit_transform)
```

The resulting dataset:

```
df.head()
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	2	1	22.0	1	0	7.2500	2
1	1	0	0	38.0	1	0	71.2833	0
2	1	2	0	26.0	0	0	7.9250	2
3	1	0	0	35.0	1	0	53.1000	2
4	0	2	1	35.0	0	0	8.0500	2

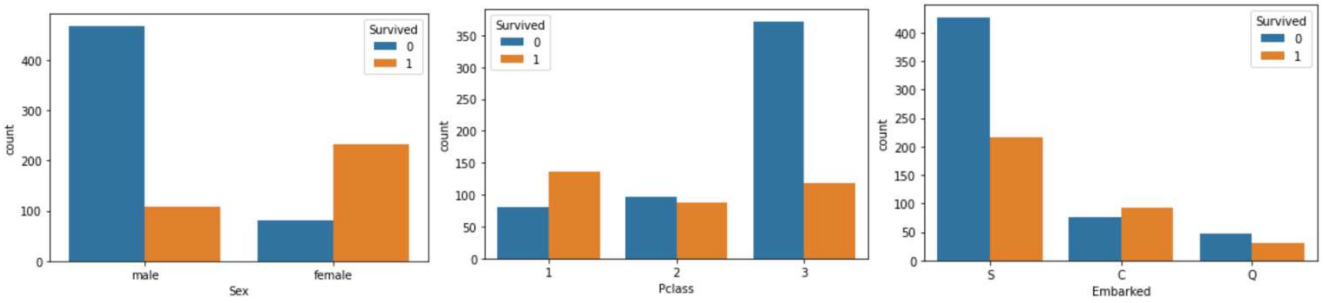
Data exploration

1. We assume that the feature of interest is Survived
2. Hence, we look for correlations between Survived and other numeric columns

	Survived	Pclass	Age	SibSp	Parch	Fare
Survived	1.000000	-0.338481	-0.080453	-0.035322	0.081629	0.257307
Pclass	-0.338481	1.000000	-0.330391	0.083081	0.018443	-0.549500
Age	-0.080453	-0.330391	1.000000	-0.236920	-0.182556	0.089079
SibSp	-0.035322	0.083081	-0.236920	1.000000	0.414838	0.159651
Parch	0.081629	0.018443	-0.182556	0.414838	1.000000	0.216225
Fare	0.257307	-0.549500	0.089079	0.159651	0.216225	1.000000

In this table we ignore the columns for Pclass, since that is categorical feature.

3. We examine the relation between Survived and categorical features by visualising the proportion of survivors in each class.



Hypotheses

We suggest the following hypotheses about the exploration:

1. Females are more likely to survive than males. This is suggested by the disproportion between the two classes in the plot.
2. We guess that the apparent relation between Embarked and Survived—that more passengers survived whose Embarked value is C—is a coincidence. This is because, at least on the face of it, there does not seem to be a reason to expect the two variables to be connected.
3. Passengers with higher Fare are more likely to survive. This is suggested by the high correlation between Fare and Survived.

Significance test on hypothesis 1

Null hypothesis: Females are as likely to survive than males

Alternative hypothesis: Females are more likely to survive than males

We note that 342/891 passengers survived (38.4%). Then we consider the proportions for each Sex. 233/314 female passengers survived (74.2%) and 109/517 male passengers survived (21.1%).

```
In [18]: df[['Survived', 'Sex']].groupby('Sex').agg({'count','sum'})
```

Out[18]:

	Survived	
	count	sum
Sex		
female	314	233
male	577	109

If the null hypothesis is true, then the number of females who survive forms a binomial random variable, with n = 314 and p = .384. The probability of 233 or more female passengers surviving is 1 – binom.cdf(232, 314, .384) << .001. At 5% significance we reject the null hypothesis and conclude that females are more likely to survive than males.

Significance test on hypothesis 2
Null hypothesis: Passengers whose embarked value is C are as likely to survive as any other group.
Alternative hypothesis: Passengers whose embarked value is C are more likely to survive.

```
df[['Survived', 'Embarked']].groupby('Embarked').agg({'count','sum'})
```

	Survived	
	count	sum
Embarked		
C	168	93
Q	77	30
S	644	217

We note that 93/168 passengers whose Embarked value is C survived (55.4%). If the null hypothesis is true, then the number of passengers whose Embarked value is C who survive forms a binomial random variable, with n = 168 and p = .384. The probability of 93 or more such passengers surviving is 1 – binom.cdf(168, 93, .384) << .001. At 5% significance we reject the null hypothesis and conclude, against our hypothesis, that passengers whose embarked value is C are more likely to survive than those in other groups.

Significance test on hypothesis 3
Null hypothesis: Passengers with higher Fare values are not more likely to survive.
Alternative hypothesis: Passengers with higher Fare values are more likely to survive.

To test this hypothesis we find the p-value of the correlation coefficient:

```
from scipy.stats import pearsonr

pearsonr(df['Survived'],df['Fare'])
```

```
]: (0.25730652238496243, 6.120189341917992e-15)
```

We find that a correlation coefficient of .257 has a p-value << .001. At 5% significance we reject the null hypothesis and conclude that Fare and Survived as positively correlated.

Conclusion
We conclude that our hypotheses 1 and 3 are correct, while hypothesis 2 is incorrect. The way forward suggested by these results is to explore possible causal relationships between the variables in question and Survived.