L09-18-11-7-P1-Bayes-Theorem

November 12, 2018

1 Titanic Data

VARIABLE DESCRIPTIONS

- survival: Survival (0 = No; 1 = Yes) |
- pclass: Passenger Class (1 = 1st; 2 = 2nd; 3 = 3rd)
- name: Name
- sex: Sex
- age: Age

VARIABLE DESCRIPTIONS

- sibsp: Number of Siblings/Spouses Aboard
- parch: Number of Parents/Children Aboard
- ticket: Ticket Number
- fare: Passenger Fare
- cabin: Cabin
- embarked: Port of Embarkation (C = Cherbourg; Q = Queenstown; S = Southampton)

SPECIAL NOTES
Pclass is a proxy for socio-economic status (SES) 1st ~ Upper; 2nd ~ Middle; 3rd ~ Lower
Age is in Years; Fractional if Age less than One (1) If the Age is Estimated, it is in the form xx.5
Family Notes
With respect to the family relation variables (i.e. sibsp and parch) some relations were ignored. The following are the definitions used for sibsp and parch.

- Sibling: Brother, Sister, Stepbrother, or Stepsister of Passenger Aboard Titanic
- Spouse: Husband or Wife of Passenger Aboard Titanic (Mistresses and Fiances Ignored)
- Parent: Mother or Father of Passenger Aboard Titanic
- Child: Son, Daughter, Stepson, or Stepdaughter of Passenger Aboard Titanic

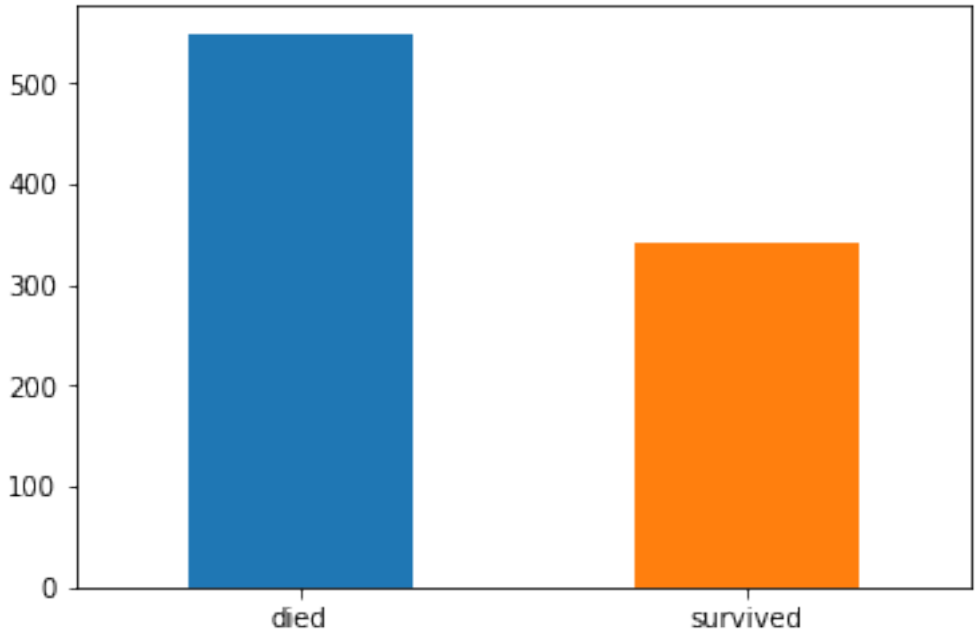
Other family relatives excluded from this study include cousins, nephews/nieces, aunts/uncles, and in-laws. Some children travelled only with a nanny, therefore parch=0 for them. As well, some travelled with very close friends or neighbors in a village, however, the definitions do not support such relations.

```
In [5]: import pandas as pd
        from matplotlib import pyplot as plt
```

```
In [2]: df = pd.read_csv("http://bit.ly/tscv17")

In [3]: df.head()

Out[3]:    PassengerId  Survived  Pclass  \
        0            1         0       3
        1            2         1       1
        2            3         1       3
        3            4         1       1
        4            5         0       3

                                                    Name     Sex   Age  SibSp  \
        0                            Braund, Mr. Owen Harris    male  22.0      1
        1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0      1
        2                             Heikkinen, Miss. Laina  female  26.0      0
        3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
        4                           Allen, Mr. William Henry    male  35.0      0

           Parch            Ticket     Fare Cabin Embarked
        0      0         A/5 21171   7.2500   NaN        S
        1      0          PC 17599  71.2833   C85        C
        2      0  STON/O2. 3101282   7.9250   NaN        S
        3      0            113803  53.1000  C123        S
        4      0            373450   8.0500   NaN        S
```

First, let's look at those survival rates, and see what's going on there?

```
In [6]: %matplotlib inline
        fig, ax = plt.subplots()
        _ = df['Survived'].value_counts().plot.bar(ax=ax)
        _ = ax.set_xticklabels(["died", "survived"], rotation=0)
```



$$\text{Probability of an Event} = \frac{\text{Number of Favorable Outcomes}}{\text{Total Number of Possible Outcomes}}$$

```
In [7]: sp = (df['Survived']==1).sum()/df['Survived'].count()
        print(f"Survival probabality is {sp}")

Survival probabality is 0.3838383838383838
```

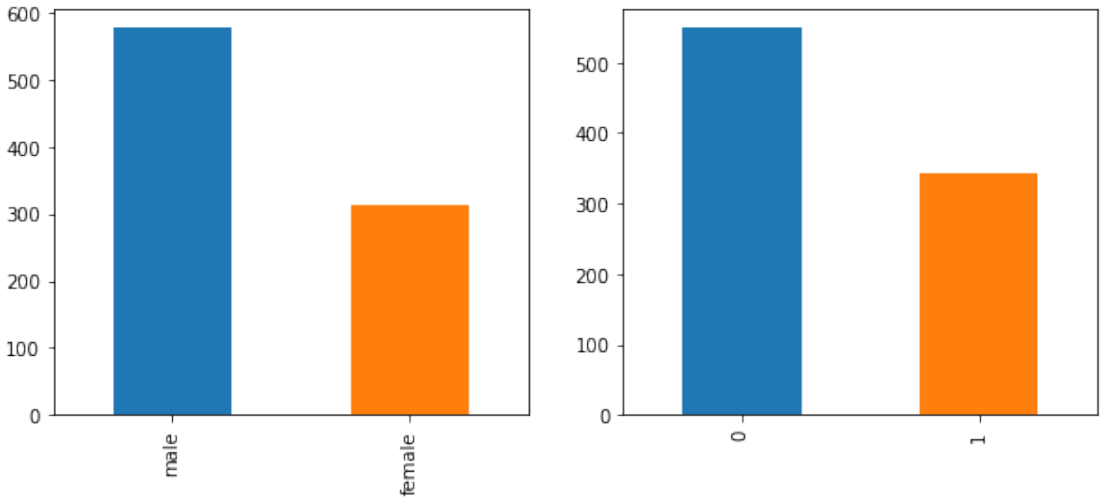
2 Lets see a break down by survival & gender

```
In [8]: subset = df[["PassengerId","Survived", "Sex", "Age", "Pclass", "SibSp", "Parch"]]
        subset.head()
```

Out[8]:

	PassengerId	Survived	Sex	Age	Pclass	SibSp	Parch
0	1	0	male	22.0	3	1	0
1	2	1	female	38.0	1	1	0
2	3	1	female	26.0	3	0	0
3	4	1	female	35.0	1	1	0
4	5	0	male	35.0	3	0	0

```
In [9]: fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(10,4))
        _ = subset["Sex"].value_counts().plot.bar(ax=ax1)
        _ = subset["Survived"].value_counts().plot.bar(ax=ax2)
```



3 How do we compute probability of surviving given gender?

```
In [16]: gsub = subset[['Sex', 'Survived','PassengerId']].groupby(['Sex', "Survived"]).count()
         gsub.unstack()
```

```
Out[16]:           PassengerId
         Survived           0    1
         Sex
         female            81  233
         male             468  109
```

```
In [21]: _ = gsub['PassengerId'].unstack().plot.bar(stacked='True')
```