```
# This Python 3 environment comes with many helpful analytics
libraries installed
# It is defined by the kaggle/python Docker image:
https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/"
directory
# For example, running this (by clicking run or pressing Shift+Enter)
will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/)
that gets preserved as output when you create a version using "Save &
Run All"
# You can also write temporary files to /kaggle/temp/, but they won't
be saved outside of the current session
```

```
/kaggle/input/titanic/train.csv
/kaggle/input/titanic/test.csv
/kaggle/input/titanic/gender_submission.csv
```

```
train_data = pd.read_csv("/kaggle/input/titanic/train.csv")
```

```
train_data
```

```
     PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3
..           ...       ...     ...
886          887         0       2
887          888         1       1
888          889         0       3
889          890         1       1
890          891         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
..                                                 ...     ...   ...
...
886                              Montvila, Rev. Juozas    male  27.0
0
887                       Graham, Miss. Margaret Edith  female  19.0
0
888           Johnston, Miss. Catherine Helen "Carrie"  female   NaN
1
889                              Behr, Mr. Karl Howell    male  26.0
0
890                                Dooley, Mr. Patrick    male  32.0
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
..     ...               ...      ...   ...      ...
886      0            211536  13.0000   NaN        S
887      0            112053  30.0000   B42        S
888      2        W./C. 6607  23.4500   NaN        S
889      0            111369  30.0000  C148        C
890      0            370376   7.7500   NaN        Q

[891 rows x 12 columns]
```

```
train_data.head()
```

```
   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
```

```
train_data.shape
```

```
(891, 12)
```

```
train_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

```
train_data.columns
```

```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age',
'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

```
train_data.describe(include = "all")
```

```
        PassengerId    Survived      Pclass                     Name
Sex  \
count    891.000000  891.000000  891.000000                      891
```

```
891
unique          NaN         NaN         NaN                      891
2
top             NaN         NaN         NaN  Braund, Mr. Owen Harris
male
freq            NaN         NaN         NaN                        1
577
mean     446.000000    0.383838    2.308642                      NaN
NaN
std      257.353842    0.486592    0.836071                      NaN
NaN
min        1.000000    0.000000    1.000000                      NaN
NaN
25%      223.500000    0.000000    2.000000                      NaN
NaN
50%      446.000000    0.000000    3.000000                      NaN
NaN
75%      668.500000    1.000000    3.000000                      NaN
NaN
max      891.000000    1.000000    3.000000                      NaN
NaN

               Age       SibSp       Parch  Ticket        Fare
Cabin  \
count   714.000000  891.000000  891.000000     891  891.000000
204
unique         NaN         NaN         NaN     681         NaN
147
top            NaN         NaN         NaN  347082         NaN  B96
B98
freq           NaN         NaN         NaN       7         NaN
4
mean     29.699118    0.523008    0.381594     NaN   32.204208
NaN
std      14.526497    1.102743    0.806057     NaN   49.693429
NaN
min       0.420000    0.000000    0.000000     NaN    0.000000
NaN
25%      20.125000    0.000000    0.000000     NaN    7.910400
NaN
50%      28.000000    0.000000    0.000000     NaN   14.454200
NaN
75%      38.000000    1.000000    0.000000     NaN   31.000000
NaN
max      80.000000    8.000000    6.000000     NaN  512.329200
NaN

       Embarked
count       889
unique        3
```

```
top           S
freq        644
mean        NaN
std         NaN
min         NaN
25%         NaN
50%         NaN
75%         NaN
max         NaN
```

Libraries

```
import seaborn as sns
import matplotlib.pyplot as plt

import warnings
warnings.filterwarnings("ignore")
```

```
train_data.nunique()
```

```
PassengerId    891
Survived         2
Pclass           3
Name           891
Sex              2
Age             88
SibSp            7
Parch            7
Ticket         681
Fare           248
Cabin          147
Embarked         3
dtype: int64
```

Percentage of Null value in eacn columns

```
train_data.isnull().sum()/len(train_data)*100
```

```
PassengerId     0.000000
Survived        0.000000
Pclass          0.000000
Name            0.000000
Sex             0.000000
Age            19.865320
SibSp           0.000000
Parch           0.000000
Ticket          0.000000
Fare            0.000000
Cabin          77.104377
```

```
Embarked        0.224467
dtype: float64
```

Percentage of men survived

```
men = train_data[train_data.Sex == "male"].Survived
rate_men = sum(men)/len(men)

print("% of men who survived : ", rate_men)
```

```
% of men who survived :  0.18890814558058924
```

Percentage of women survived

```
women = train_data[train_data.Sex == "female"].Survived
rate_women = sum(women)/len(women)

print("% of women who survived : " , rate_women)
```

```
% of women who survived :  0.7420382165605095
```

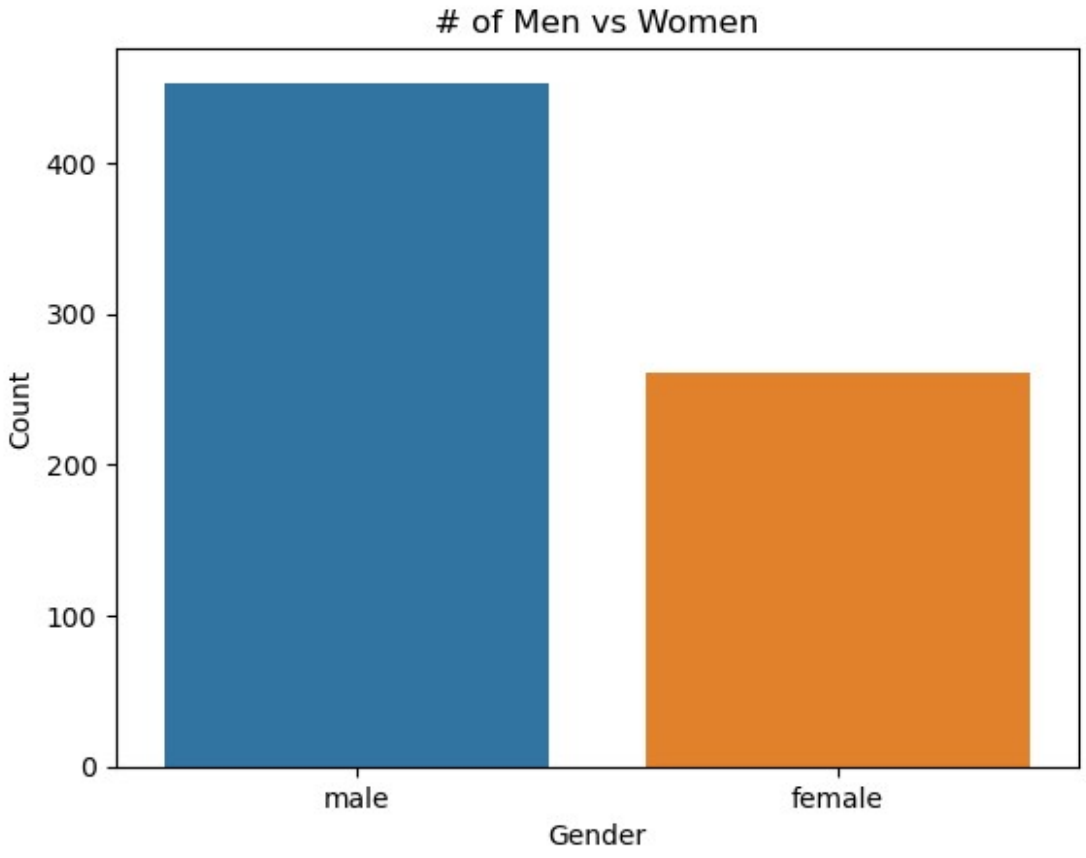
Total Missing Value

```
total_null = train_data.isnull().sum().sort_values(ascending = False)
percent =
((train_data.isnull().sum()/train_data.isnull().count())*100).sort_val
ues(ascending = False)
print("Total records = ", train_data.shape[0])

missing_data =
pd.concat([total_null,percent.round(2)],axis=1,keys=['Total
Missing','In Percent'])
missing_data.head(10)
```

```
Total records =  891

             Total Missing  In Percent
Cabin                  687       77.10
Age                    177       19.87
Embarked                 2        0.22
PassengerId              0        0.00
Survived                 0        0.00
Pclass                   0        0.00
Name                     0        0.00
Sex                      0        0.00
SibSp                    0        0.00
Parch                    0        0.00
```

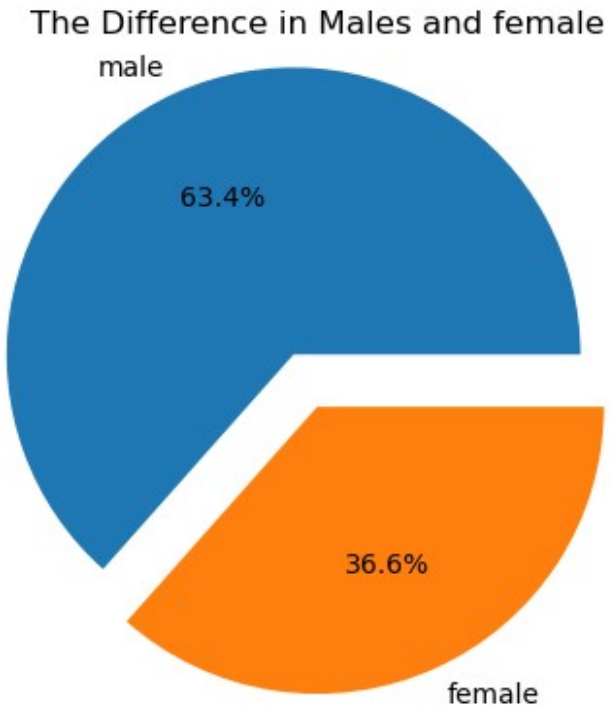
Dealing with Null values

```
train_data = train_data.dropna(subset = ["Age"])

train_data.shape

(714, 12)

gender = train_data['Sex'].value_counts()
gender

male      453
female    261
Name: Sex, dtype: int64

sns.barplot(x=gender.index, y=gender.values)  # Also you can use
`sns.countplot`
plt.xlabel('Gender')
plt.ylabel('Count')
plt.title('# of Men vs Women')
plt.savefig('No. of Men vs Women.pdf')
plt.show()
```




```
plt.pie(x=gender.values, labels=gender.index, autopct='%.1f%%',
explode=[.2, 0])
```

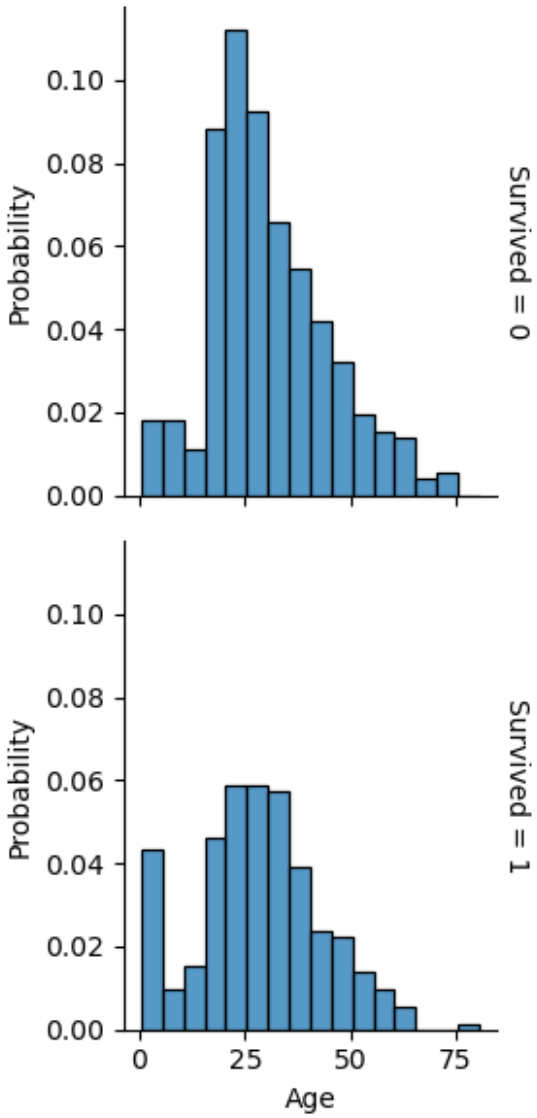
```
plt.title('The Difference in Males and female')
plt.show()
```



Graph Between Age and Probability of Surviving

```
sns.displot(
    train_data, x="Age", row="Survived",
    binwidth=5, height=3, stat='probability',
facet_kws=dict(margin_titles=True),
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7fa02ed7f5d0>
```


Probability
0.10
0.08
0.06
0.04
0.02
0.00
Survived = 0
Probability
0.10
0.08
0.06
0.04
0.02
0.00
Survived = 1
0
25
50
75
Age