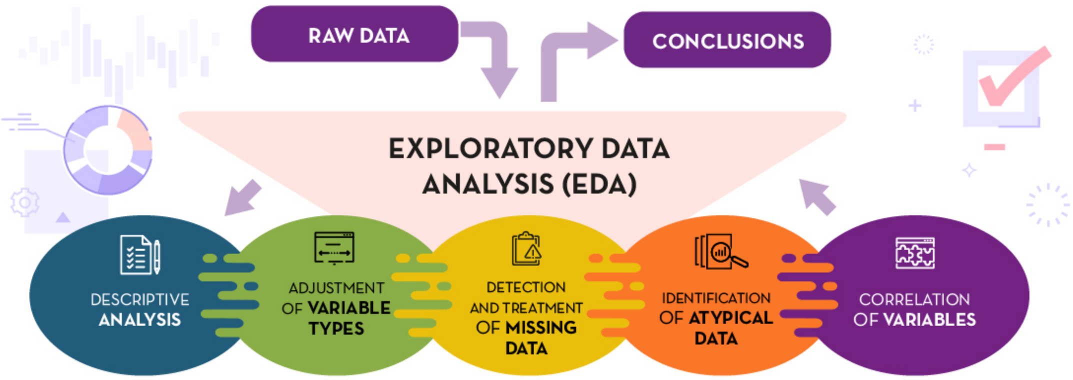
Exploratory data analysis

To analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods.


RAW DATA
CONCLUSIONS
EXPLORATORY DATA ANALYSIS (EDA)
DESCRIPTIVE ANALYSIS
ADJUSTMENT OF VARIABLE TYPES
DETECTION AND TREATMENT OF MISSING DATA
IDENTIFICATION OF ATYPICAL DATA
CORRELATION OF VARIABLES

Why do EDA

- Model building
- Analysis and reporting
- Validate assumptions
- Handling missing values
- feature engineering
- detecting outliers

```
In [1]: import pandas as pd
        import numpy as np
        import seaborn as sns
        import matplotlib.pyplot as plt

        plt.style.use('ggplot')
```

Remember it is an iterative process

```
In [2]: df =pd.read_csv("D:\\datascience\\Nitish sir\\Data Wrangling\\EDA\\train.csv")
```

In [3]:
```
df.head()
```

Out[3]:

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

Column Types

- **Numerical** - Age,Fare,PassengerId
- **Categorical** - Survived, Pclass, Sex, SibSp, Parch,Embarked
- **Mixed** - Name, Ticket, Cabin

Univariate Analysis

Univariate analysis focuses on analyzing each feature in the dataset independently.

- **Distribution analysis**: The distribution of each feature is examined to identify its shape, central tendency, and dispersion.
- **Identifying potential issues**: Univariate analysis helps in identifying potential problems with the data such as outliers, skewness, and missing values

The shape of a data distribution refers to its overall pattern or form as it is represented on a graph. Some common shapes of data distributions include:

- **Normal Distribution**: A symmetrical and bell-shaped distribution where the mean, median, and mode are equal and the majority of the data falls in the middle of the distribution with gradually decreasing frequencies towards the tails.

- **Skewed Distribution**: A distribution that is not symmetrical, with one tail being longer than the other. It can be either positively skewed (right-skewed) or negatively skewed (left-skewed).
- **Bimodal Distribution**: A distribution with two peaks or modes.
- **Uniform Distribution**: A distribution where all values have an equal chance of occurring.

Dispersion is a statistical term used to describe the spread or variability of a set of data. It measures how far the values in a data set are spread out from the central tendency (mean, median, or mode) of the data.

There are several measures of dispersion, including:

- **Range**: The difference between the largest and smallest values in a data set.
- **Variance**: The average of the squared deviations of each value from the mean of the data set.
- **Standard Deviation**: The square root of the variance. It provides a measure of the spread of the data that is in the same units as the original data.
- **Interquartile range (IQR)**: The range between the first quartile (25th percentile) and the third quartile (75th percentile) of the data.

Dispersion helps to describe the spread of the data, which can help to identify the presence of outliers and skewness in the data.

1. Steps of doing Univariate Analysis on Numerical columns

- **Descriptive Statistics**: Compute basic summary statistics for the column, such as mean, median, mode, standard deviation, range, and quartiles. These statistics give a general understanding of the distribution of the data and can help identify skewness or outliers.
- **Visualizations**: Create visualizations to explore the distribution of the data. Some common visualizations for numerical data include histograms, box plots, and density plots. These visualizations provide a visual representation of the distribution of the data and can help identify skewness an outliers.
- **Identifying Outliers**: Identify and examine any outliers in the data. Outliers can be identified using visualizations. It is important to determine whether the outliers are due to measurement errors, data entry errors, or legitimate differences in the data, and to decide whether to include or exclude them from the analysis.
- **Skewness**: Check for skewness in the data and consider transforming the data or using robust statistical methods that are less sensitive to skewness, if necessary.
- **Conclusion**: Summarize the findings of the EDA and make decisions about how to proceed with further analysis.

- **Numerical** - Age,Fare,PassengerId

```
In [4]: df
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500
...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500

891 rows × 12 columns

Age (Numerical Data)

conclusions

- Age is normally(almost) distributed
- 20% of the values are missing

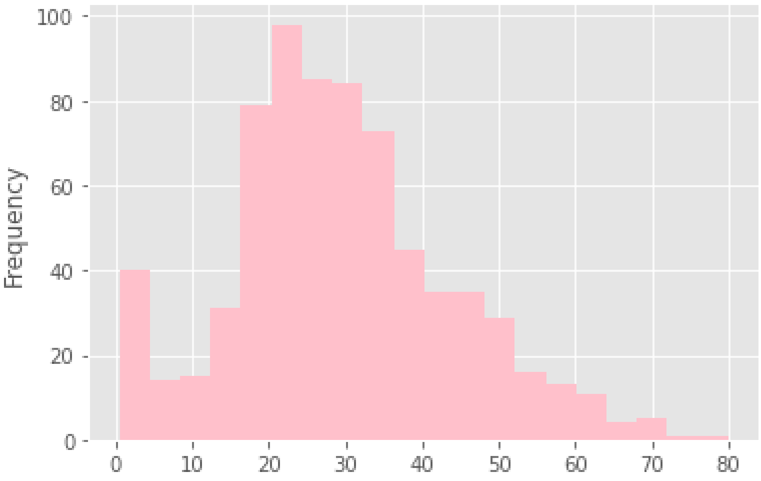
- There are some outliers

In [5]:
```
df['Age'].describe()
```

Out[5]:
```
count    714.000000
mean      29.699118
std       14.526497
min        0.420000
25%       20.125000
50%       28.000000
75%       38.000000
max       80.000000
Name: Age, dtype: float64
```

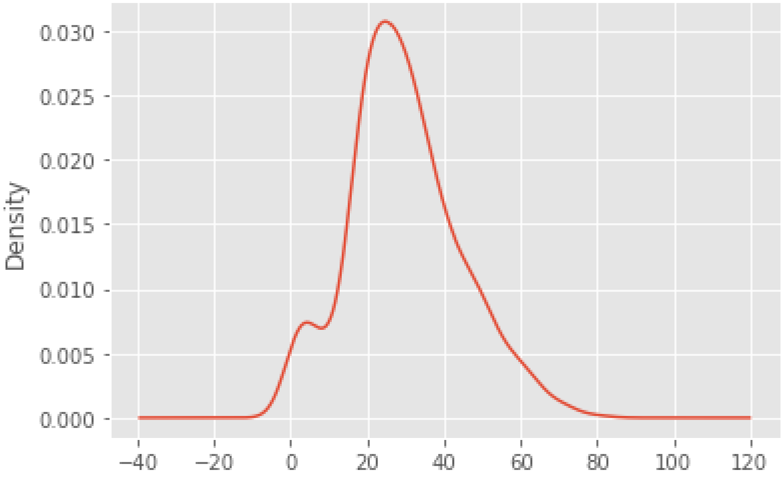
In [6]:
```
df['Age'].plot(kind ='hist',bins = 20, color= 'pink')
```

Out[6]:
```
<AxesSubplot:ylabel='Frequency'>
```



In [7]:
```
df['Age'].plot(kind ='kde') # distribution plot
```

Out[7]:
```
<AxesSubplot:ylabel='Density'>
```



In [8]:

```
df['Age'].skew() # skewness
```

Out[8]:

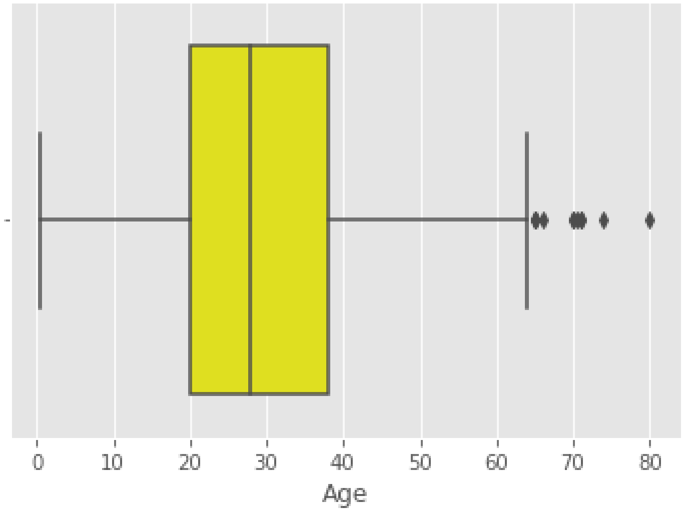
```
0.38910778230082704
```

In [9]:

```
# df['Age'].plot(kind ='box')
sns.boxplot(x = df['Age'] ,color ='yellow')
```

Out[9]:

```
<AxesSubplot:xlabel='Age'>
```



In [10]:
```
df[df['Age']>65] # no weird data (outliers) in age column
```

Out[10]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Ca
33	34	0	2	Wheadon, Mr. Edward H	male	66.0	0	0	C.A. 24579	10.5000	
96	97	0	1	Goldschmidt, Mr. George B	male	71.0	0	0	PC 17754	34.6542	
116	117	0	3	Connors, Mr. Patrick	male	70.5	0	0	370369	7.7500	
493	494	0	1	Artagaveytia, Mr. Ramon	male	71.0	0	0	PC 17609	49.5042	
630	631	1	1	Barkworth, Mr. Algernon Henry Wilson	male	80.0	0	0	27042	30.0000	
672	673	0	2	Mitchell, Mr. Henry Michael	male	70.0	0	0	C.A. 24580	10.5000	
745	746	0	1	Crosby, Capt. Edward Gifford	male	70.0	1	1	WE/P 5735	71.0000	
851	852	0	3	Svensson, Mr. Johan	male	74.0	0	0	347060	7.7750	

In [11]:
```
df['Age'].isnull().sum()
```

Out[11]:
```
177
```

In [12]:
```
df['Age'].isnull().sum()/len(df['Age'])  # 19 % missing values
```

Out[12]:
```
0.19865319865319866
```

Fare (Numerical Data)

conclusions

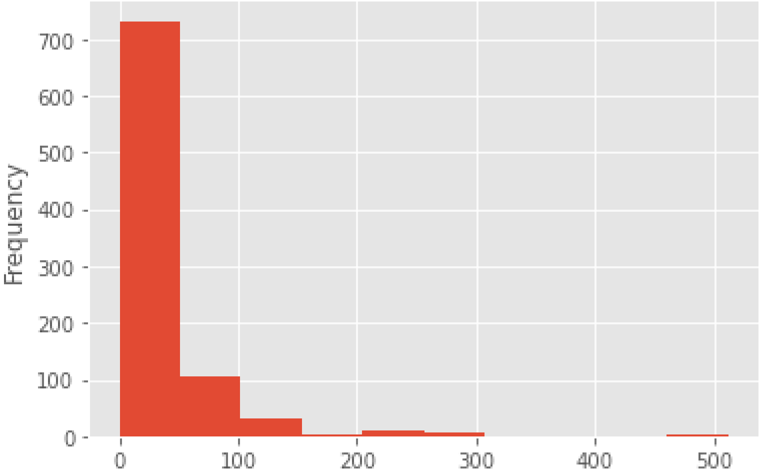
- The data is highly(positively) skewed
- Fare col actually contains the group fare and not the individual fare(This migth be and issue)
- We need to create a new col called individual fare

In [13]:
```
df['Fare'].describe()
```

Out[13]:
```
count    891.000000
mean      32.204208
std       49.693429
min        0.000000
25%        7.910400
50%       14.454200
75%       31.000000
max      512.329200
Name: Fare, dtype: float64
```

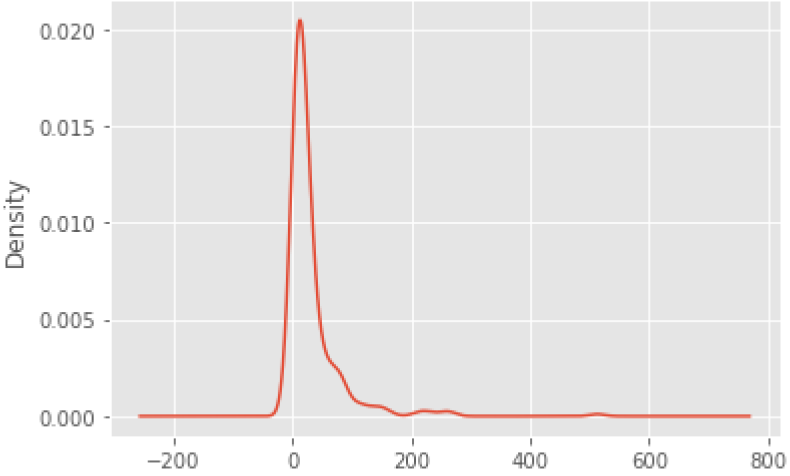
In [14]:
```
df['Fare'].plot(kind ='hist')
```

Out[14]:
```
<AxesSubplot:ylabel='Frequency'>
```



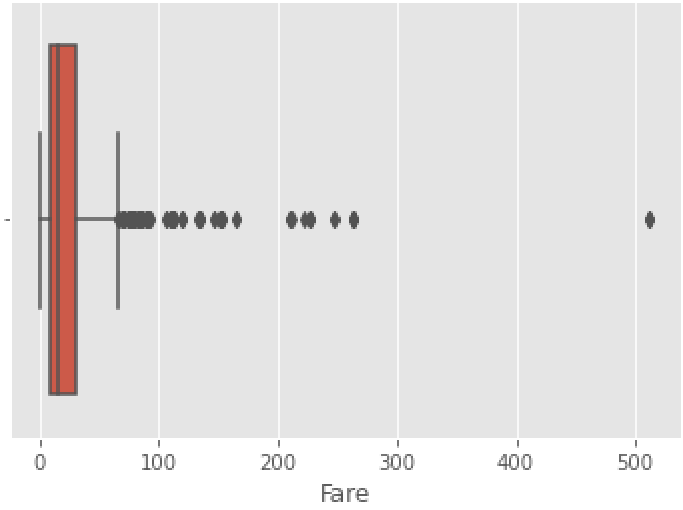
In [15]:
```
df['Fare'].plot(kind ='kde') # Right skew
```

Out[15]:
```
<AxesSubplot:ylabel='Density'>
```



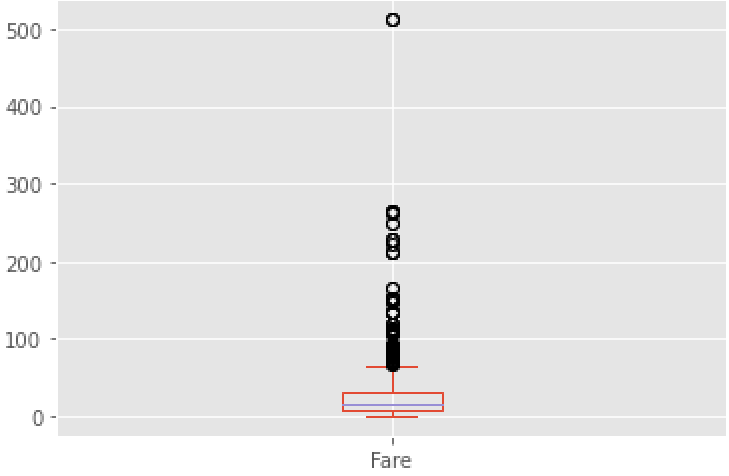
```
In [16]: sns.boxplot(x = df['Fare'])
```

```
Out[16]: <AxesSubplot:xlabel='Fare'>
```



```
In [17]: df['Fare'].plot(kind ='box')
```

```
Out[17]: <AxesSubplot:>
```



```
In [18]: df['Fare'].skew() # positively Skewed
```

```
Out[18]: 4.787316519674893
```

In [19]:
```
df[df['Fare']> 250]
```

Out[19]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Ca
27	28	0	1	Fortune, Mr. Charles Alexander	male	19.0	3	2	19950	263.0000	C C C
88	89	1	1	Fortune, Miss. Mabel Helen	female	23.0	3	2	19950	263.0000	C C C
258	259	1	1	Ward, Miss. Anna	female	35.0	0	0	PC 17755	512.3292	N
311	312	1	1	Ryerson, Miss. Emily Borie	female	18.0	2	2	PC 17608	262.3750	B B B B
341	342	1	1	Fortune, Miss. Alice Elizabeth	female	24.0	3	2	19950	263.0000	C C C
438	439	0	1	Fortune, Mr. Mark	male	64.0	1	4	19950	263.0000	C C C
679	680	1	1	Cardeza, Mr. Thomas Drake Martinez	male	36.0	0	1	PC 17755	512.3292	B B B
737	738	1	1	Lesurer, Mr. Gustave J	male	35.0	0	0	PC 17755	512.3292	B
742	743	1	1	Ryerson, Miss. Susan Parker "Suzette"	female	21.0	2	2	PC 17608	262.3750	B B B B

In [20]:
```
df['Fare'].isnull().sum() # No missing Values
```

Out[20]:
```
0
```

2. Steps of doing Univariate Analysis on Categorical columns

Descriptive Statistics: Compute the frequency distribution of the categories in the column. This will give a general understanding of the distribution of the categories and their relative frequencies.

Visualizations: Create visualizations to explore the distribution of the categories. Some common visualizations for categorical data include count plots and pie charts. These visualizations provide a visual representation of the distribution of the categories and can help identify any patterns or anomalies in the data.

Missing Values: Check for missing values in the data and decide how to handle them. Missing values can be imputed or excluded from the analysis, depending on the research question and the data set.

- **Categorical** - Survived, Pclass, Sex, SibSp, Parch,Embarked

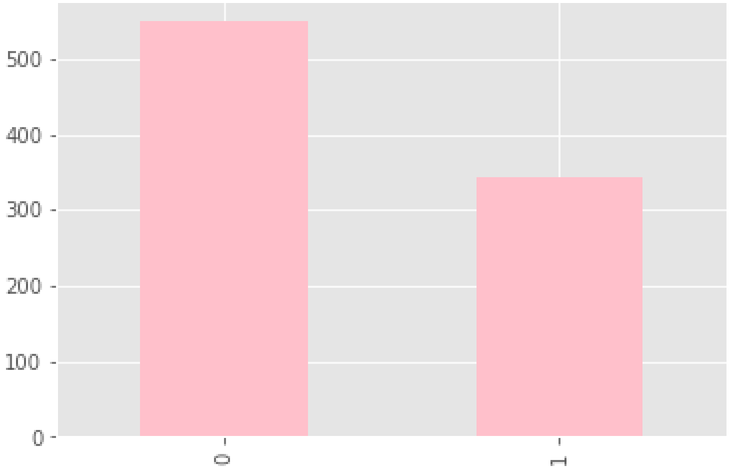
Survived

```
In [21]: df['Survived'].value_counts() # 0 = died
```

```
Out[21]: 0    549
         1    342
         Name: Survived, dtype: int64
```

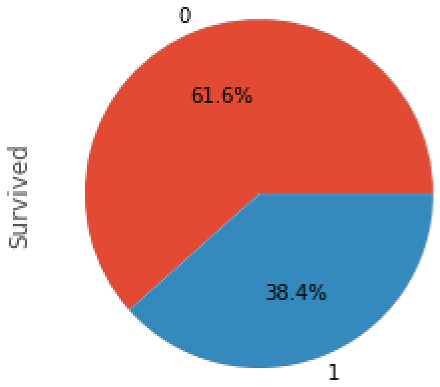
```
In [22]: df['Survived'].value_counts().plot(kind='bar', color ='pink')
```

```
Out[22]: <AxesSubplot:>
```



In [23]:
```
df['Survived'].value_counts().plot(kind='pie',autopct = '%0.1f%%')
```

Out[23]:
```
<AxesSubplot:ylabel='Survived'>
```



In [24]:
```
df['Survived'].isnull().sum()
```

Out[24]:
```
0
```

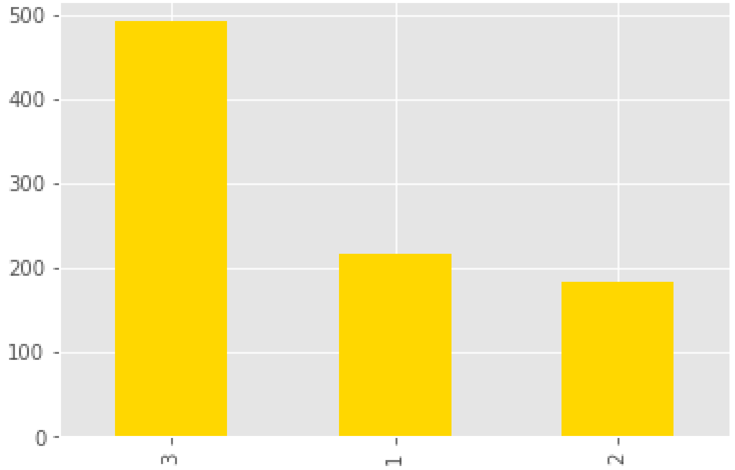
Pclass

In [25]:
```
df['Pclass'].value_counts().sort_values(ascending=True)
```

Out[25]:
```
2    184
1    216
3    491
Name: Pclass, dtype: int64
```

In [26]:
```
df['Pclass'].value_counts().plot(kind ='bar',color ='gold')
```

Out[26]:
```
<AxesSubplot:>
```

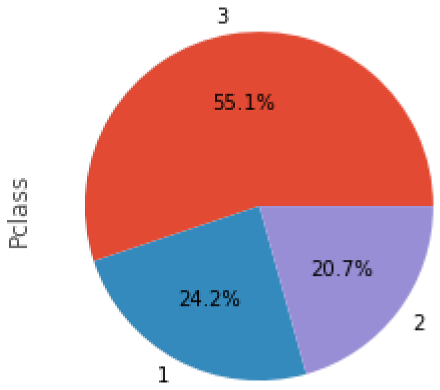


In [27]:

```
df['Pclass'].value_counts().plot(kind ='pie',autopct='%0.1f%%')
```

Out[27]:

```
<AxesSubplot:ylabel='Pclass'>
```



In [28]:

```
df['Pclass'].isnull().sum()
```

Out[28]:

```
0
```

Sex

In [29]:

```
df['Sex'].value_counts()
```

Out[29]:

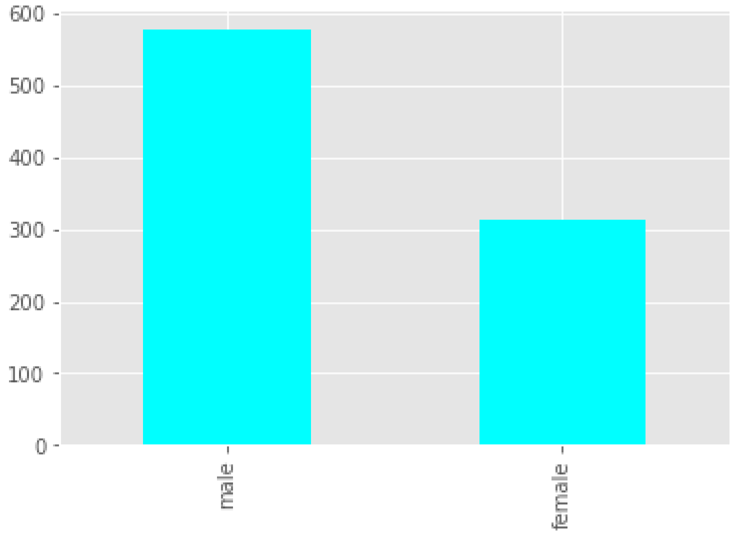
```
male      577
female    314
Name: Sex, dtype: int64
```

In [30]:

```
df['Sex'].value_counts().plot(kind ='bar', color ='cyan')
```

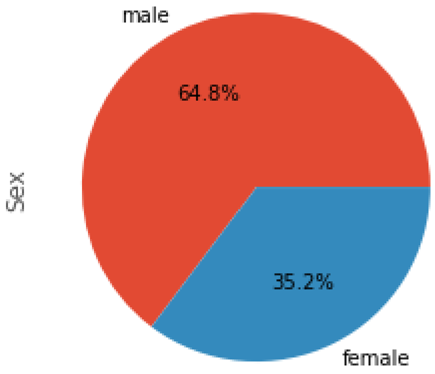
Out[30]:

```
<AxesSubplot:>
```



In [31]:
```python
df['Sex'].value_counts().plot(kind ='pie',autopct ='%0.1f%%')
```

Out[31]:
```
<AxesSubplot:ylabel='Sex'>
```



In [32]:
```python
df['Sex'].isnull().sum()
```

Out[32]:
```
0
```

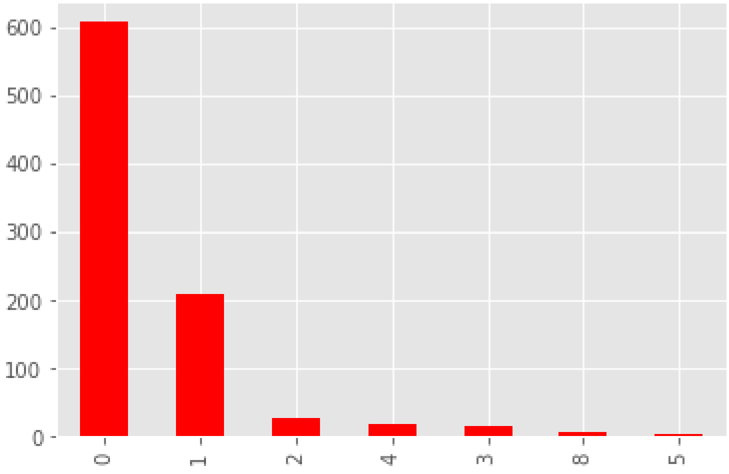
SibSp

In [33]:
```python
df['SibSp'].value_counts()
```

Out[33]:
```
0    608
1    209
2     28
4     18
3     16
8      7
5      5
Name: SibSp, dtype: int64
```

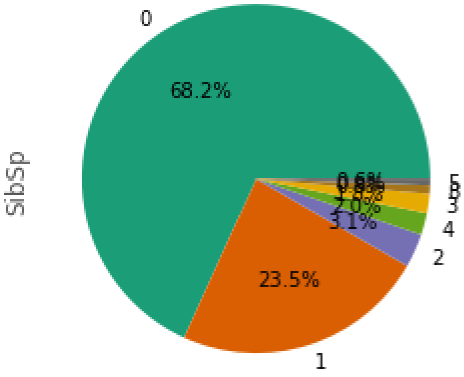
In [34]:
```python
df['SibSp'].value_counts().plot(kind = 'bar' ,color ='red')
```

Out[34]:
```
<AxesSubplot:>
```



In [35]:
```
df['SibSp'].value_counts().plot(kind = 'pie', autopct ='%0.1f%%',cmap ='Dark2'
```

Out[35]:
```
<AxesSubplot:ylabel='SibSp'>
```



In [36]:
```
df['SibSp'].isnull().sum()
```

Out[36]:
```
0
```

Parch

conclusions

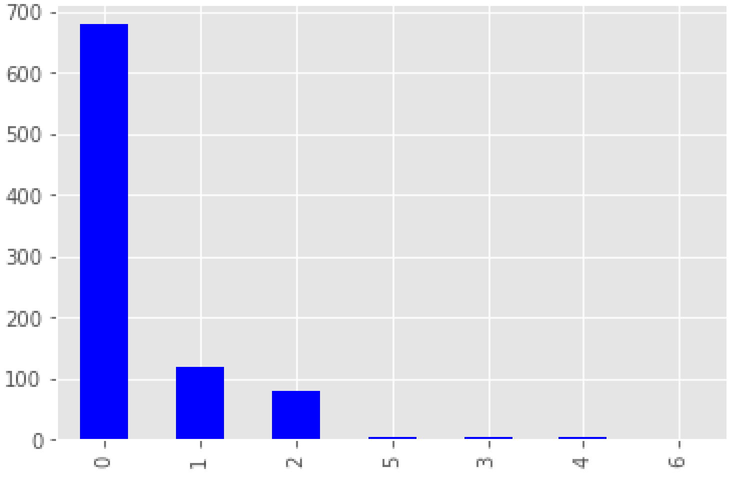
- Parch and SibSp cols can be merged to form a new col call family_size
- Create a new col called is_alone

In [37]:
```
df['Parch'].value_counts()
```

Out[37]:
```
0    678
1    118
2     80
5      5
3      5
4      4
6      1
Name: Parch, dtype: int64
```

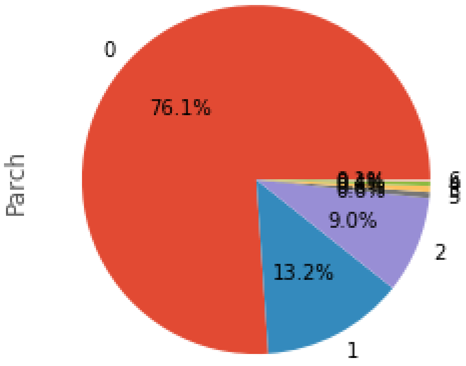
In [38]:
```
df['Parch'].value_counts().plot(kind ='bar',color ='blue')
```

Out[38]:
```
<AxesSubplot:>
```



In [39]:
```
df['Parch'].value_counts().plot(kind ='pie' , autopct='%0.1f%%')
```

Out[39]:
```
<AxesSubplot:ylabel='Parch'>
```



In [40]:
```
df['Parch'].isnull().sum()
```

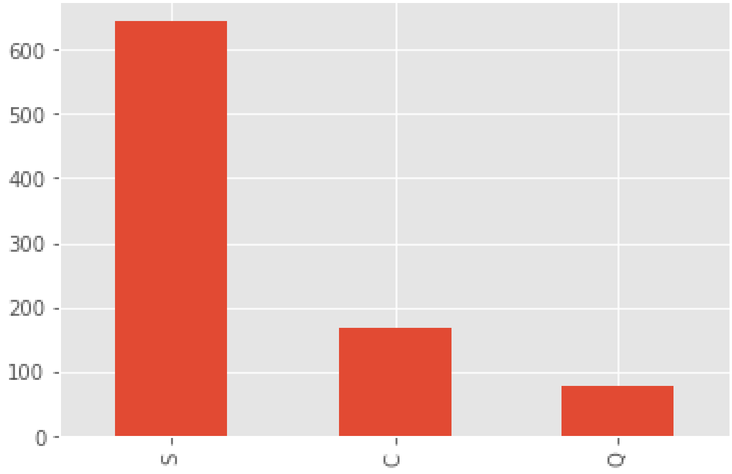
Out[40]:
```
0
```

Embarked

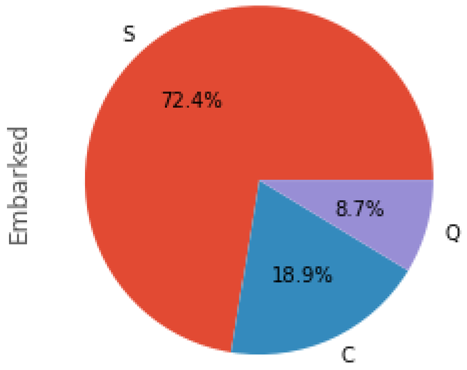
In [41]:
```
df['Embarked'].value_counts()
```

Out[41]:
```
S    644
C    168
Q     77
Name: Embarked, dtype: int64
```

```
In [42]: df['Embarked'].value_counts().plot(kind='bar')

Out[42]: <AxesSubplot:>
```



```
In [43]: df['Embarked'].value_counts().plot(kind='pie',autopct='%0.1f%%')

Out[43]: <AxesSubplot:ylabel='Embarked'>
```



```
In [44]: df['Sex'].isnull().sum()

Out[44]: 0
```

Need More Feature Engineer to Analyse 'Mixed Columns'

```
In [ ]:
```

Steps of doing Bivariate Analysis

- Select 2 cols
- Understand type of relationship

1. **Numerical - Numerical**
   a. You can plot graphs like scatterplot(regression plots), 2D histplot, 2D KDEplots
   b. Check correlation coefficent to check linear relationship
2. **Numerical - Categorical** - create visualizations that compare the distribution of the numerical data across different categories of the categorical data.
   a. You can plot graphs like barplot, boxplot, kdeplot violinplot even scatterplots
3. **Categorical - Categorical**
   a. You can create cross-tabulations or contingency tables that show the distribution of values in one categorical column, grouped by the values in the other categorical column.
   b. You can plots like heatmap, stacked barplots, treemaps

In [45]:

```
df.head()
```

Out[45]:

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

catgorical + categorical = Contingency tables

In [46]:

```
pd.crosstab(df['Survived'],df['Pclass'])
```

Out[46]:

Pclass	1	2	3
Survived			
0	80	97	372
1	136	87	119

In [47]:

```
# noramalize on column wise

pd.crosstab(df['Survived'],df['Pclass'],normalize='columns')* 100
```

Out[47]:

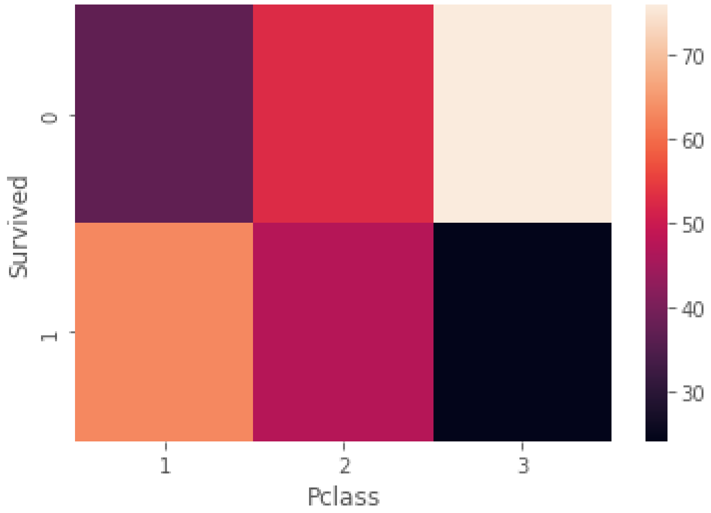
Pclass	1	2	3
Survived			
0	37.037037	52.717391	75.763747
1	62.962963	47.282609	24.236253

In [48]:

```
# heatmap

sns.heatmap(pd.crosstab(df['Survived'],df['Pclass'],normalize='columns')* 100)
```

Out[48]:

```
<AxesSubplot:xlabel='Pclass', ylabel='Survived'>
```



Survived + Sex

In [49]:

```
pd.crosstab(df['Survived'],df['Sex'])
```

Out[49]:

Sex	female	male
Survived		
0	81	468
1	233	109

In [50]:

```
# Normalize

pd.crosstab(df['Survived'],df['Sex'],normalize='columns')*100
```

Out[50]:

Sex	female	male
Survived		
0	25.796178	81.109185
1	74.203822	18.890815

Survived + Embarked

In [51]:

```
pd.crosstab(df['Survived'],df['Embarked'])
```

Out[51]:

Embarked	C	Q	S
Survived			
0	75	47	427
1	93	30	217

In [52]:

```
pd.crosstab(df['Survived'],df['Embarked'],normalize='columns')*100
```

Out[52]:

Embarked	C	Q	S
Survived			
0	44.642857	61.038961	66.304348
1	55.357143	38.961039	33.695652

In [53]:

```
pd.crosstab(df['Sex'],df['Embarked'],normalize='columns')*100
```

Out[53]:

Embarked	C	Q	S
Sex			
female	43.452381	46.753247	31.521739
male	56.547619	53.246753	68.478261

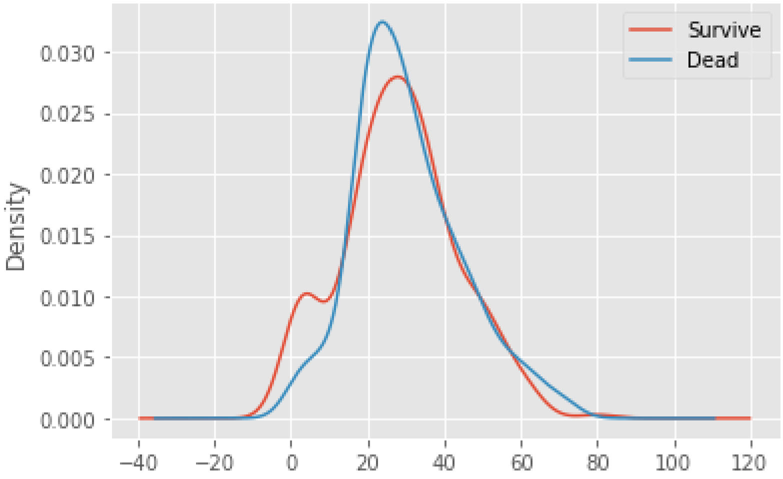
In [54]:
```
pd.crosstab(df['Pclass'],df['Embarked'],normalize='columns')*100
```

Out[54]:

Embarked	C	Q	S
Pclass			
1	50.595238	2.597403	19.720497
2	10.119048	3.896104	25.465839
3	39.285714	93.506494	54.813665

Categorical + Numerical

In [55]:
```
# Survived + Age

df[df['Survived']==1] ['Age'].plot(kind ='kde', label ='Survive')
df[df['Survived']==0] ['Age'].plot(kind ='kde',label ='Dead')

plt.legend()
plt.show()
```

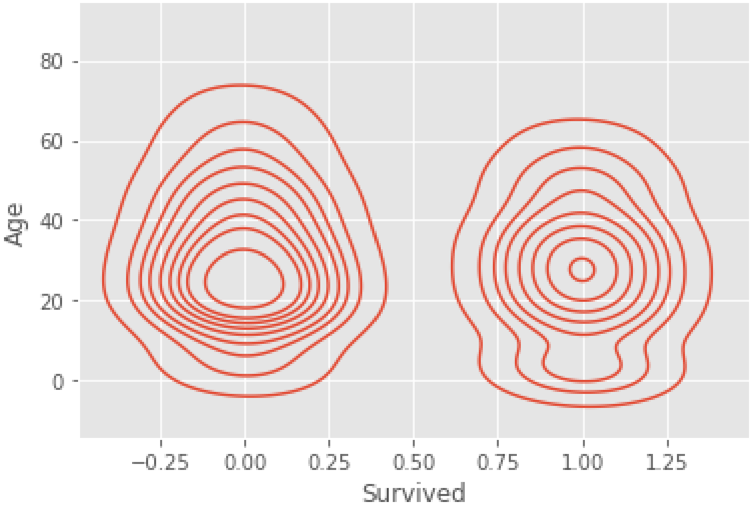


In [56]:
```
df[df['Pclass']==1]['Age'].mean()
```

Out[56]:
```
38.233440860215055
```

```
In [57]: sns.kdeplot(data =df,x='Survived',y='Age')
```

```
Out[57]: <AxesSubplot:xlabel='Survived', ylabel='Age'>
```



In [58]:

```
## Feature Engineering on Fare col
df
```

Out[58]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500
...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500

891 rows × 12 columns

In [59]:

```
#siblingsSpouse column

df['SibSp'].value_counts()
```

Out[59]:

```
0    608
1    209
2     28
4     18
3     16
8      7
5      5
Name: SibSp, dtype: int64
```

In [60]:

```
df[df['SibSp'] == 8]

# 11 members in family , 8 siblingspouse ,2 Parent child , 1 individual Name
```

Out[60]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
159	160	0	3	Sage, Master. Thomas Henry	male	NaN	8	2	CA. 2343	69.55	NaN
180	181	0	3	Sage, Miss. Constance Gladys	female	NaN	8	2	CA. 2343	69.55	NaN
201	202	0	3	Sage, Mr. Frederick	male	NaN	8	2	CA. 2343	69.55	NaN
324	325	0	3	Sage, Mr. George John Jr	male	NaN	8	2	CA. 2343	69.55	NaN
792	793	0	3	Sage, Miss. Stella Anna	female	NaN	8	2	CA. 2343	69.55	NaN
846	847	0	3	Sage, Mr. Douglas Bullen	male	NaN	8	2	CA. 2343	69.55	NaN
863	864	0	3	Sage, Miss. Dorothy Edith "Dolly"	female	NaN	8	2	CA. 2343	69.55	NaN

In [61]:

```
69.55/11 # 3rd class (maybe fare is ok)
```

Out[61]:

```
6.322727272727272
```

In [62]:
```
df[df['Ticket'] == 'CA. 2343']
```

Out[62]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
159	160	0	3	Sage, Master. Thomas Henry	male	NaN	8	2	CA. 2343	69.55	NaN
180	181	0	3	Sage, Miss. Constance Gladys	female	NaN	8	2	CA. 2343	69.55	NaN
201	202	0	3	Sage, Mr. Frederick	male	NaN	8	2	CA. 2343	69.55	NaN
324	325	0	3	Sage, Mr. George John Jr	male	NaN	8	2	CA. 2343	69.55	NaN
792	793	0	3	Sage, Miss. Stella Anna	female	NaN	8	2	CA. 2343	69.55	NaN
846	847	0	3	Sage, Mr. Douglas Bullen	male	NaN	8	2	CA. 2343	69.55	NaN
863	864	0	3	Sage, Miss. Dorothy Edith "Dolly"	female	NaN	8	2	CA. 2343	69.55	NaN

In [63]:
```
df[df['Name'].str.contains('Sage')]
```

Out[63]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
159	160	0	3	Sage, Master. Thomas Henry	male	NaN	8	2	CA. 2343	69.55	NaN
180	181	0	3	Sage, Miss. Constance Gladys	female	NaN	8	2	CA. 2343	69.55	NaN
201	202	0	3	Sage, Mr. Frederick	male	NaN	8	2	CA. 2343	69.55	NaN
324	325	0	3	Sage, Mr. George John Jr	male	NaN	8	2	CA. 2343	69.55	NaN
641	642	1	1	Sagesser, Mlle. Emma	female	24.0	0	0	PC 17477	69.30	B35
792	793	0	3	Sage, Miss. Stella Anna	female	NaN	8	2	CA. 2343	69.55	NaN
846	847	0	3	Sage, Mr. Douglas Bullen	male	NaN	8	2	CA. 2343	69.55	NaN
863	864	0	3	Sage, Miss. Dorothy Edith "Dolly"	female	NaN	8	2	CA. 2343	69.55	NaN

In [64]:
```
# remaining data on TEST

df1 =pd.read_csv("D:\\datascience\\Nitish sir\\Data Wrangling\\EDA\\test.csv")
```

In [65]:
```
df1
```

Out[65]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN
...	...	...	...	...	...	...	...	...	...	...
413	1305	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	NaN
414	1306	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105
415	1307	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	NaN
416	1308	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	NaN
417	1309	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	NaN

418 rows × 11 columns

In [66]:
```
df = pd.concat([df,df1])
```

In [67]:
```
df
```

Out[67]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
0	1	0.0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25
1	2	1.0	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.28
2	3	1.0	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.92
3	4	1.0	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.10
4	5	0.0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.05
...	...	...	...	...	...	...	...	...	...	
413	1305	NaN	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.05
414	1306	NaN	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.90
415	1307	NaN	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.25
416	1308	NaN	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.05
417	1309	NaN	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.35

1309 rows × 12 columns

In [68]:
```
df[df['Ticket'] == 'CA. 2343']
```

Out[68]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
159	160	0.0	3	Sage, Master. Thomas Henry	male	NaN	8	2	CA. 2343	69.55	NaN
180	181	0.0	3	Sage, Miss. Constance Gladys	female	NaN	8	2	CA. 2343	69.55	NaN
201	202	0.0	3	Sage, Mr. Frederick	male	NaN	8	2	CA. 2343	69.55	NaN
324	325	0.0	3	Sage, Mr. George John Jr	male	NaN	8	2	CA. 2343	69.55	NaN
792	793	0.0	3	Sage, Miss. Stella Anna	female	NaN	8	2	CA. 2343	69.55	NaN
846	847	0.0	3	Sage, Mr. Douglas Bullen	male	NaN	8	2	CA. 2343	69.55	NaN
863	864	0.0	3	Sage, Miss. Dorothy Edith "Dolly"	female	NaN	8	2	CA. 2343	69.55	NaN
188	1080	NaN	3	Sage, Miss. Ada	female	NaN	8	2	CA. 2343	69.55	NaN
342	1234	NaN	3	Sage, Mr. John George	male	NaN	1	9	CA. 2343	69.55	NaN
360	1252	NaN	3	Sage, Master. William Henry	male	14.5	8	2	CA. 2343	69.55	NaN
365	1257	NaN	3	Sage, Mrs. John (Annie Bullen)	female	NaN	1	9	CA. 2343	69.55	NaN

In [69]:

```
df['Fare']/(df['SibSp'] + df['Parch'] + 1)
```

Out[69]:

```
0        3.625000
1       35.641650
2        7.925000
3       26.550000
4        8.050000
          ...
413      8.050000
414    108.900000
415      7.250000
416      8.050000
417      7.452767
Length: 1309, dtype: float64
```

In [70]:

```
df['individual_fare'] = df['Fare']/(df['SibSp'] + df['Parch'] + 1)
```

In [71]:
```
df
```

Out[71]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
0	1	0.0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25
1	2	1.0	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.28
2	3	1.0	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.92
3	4	1.0	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.10
4	5	0.0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.05
...	...	...	...	...	...	...	...	...	...	
413	1305	NaN	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.05
414	1306	NaN	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.90
415	1307	NaN	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.25
416	1308	NaN	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.05
417	1309	NaN	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.35

1309 rows × 13 columns

In [72]:
```
df[['individual_fare','Fare']].describe()
```

Out[72]:

	individual_fare	Fare
count	1308.000000	1308.000000
mean	20.518215	33.295479
std	35.774337	51.758668
min	0.000000	0.000000
25%	7.452767	7.895800
50%	8.512483	14.454200
75%	24.237500	31.275000
max	512.329200	512.329200

In [73]:
```
df[df['individual_fare'] == 512.329200]
```

Out[73]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabi
258	259	1.0	1	Ward, Miss. Anna	female	35.0	0	0	PC 17755	512.3292	Na
737	738	1.0	1	Lesurer, Mr. Gustave J	male	35.0	0	0	PC 17755	512.3292	B10

In [74]:
```
# combine sbisp + parch = Family size

df['family_size'] = df['SibSp'] + df['Parch'] + 1
```

In [75]:
```
df.sample(3)
```

Out[75]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	C
570	571	1.0	2	Harris, Mr. George	male	62.0	0	0	S.W./PP 752	10.500	
426	427	1.0	2	Clarke, Mrs. Charles V (Ada Maria Winfield)	female	28.0	1	0	2003	26.000	
104	105	0.0	3	Gustafsson, Mr. Anders Vilhelm	male	37.0	2	0	3101276	7.925	

In [76]:
```
# Function that returns family size

# family_type
# 1 -> alone
# 2-4 -> small
# >5 -> large

def trasform_size(num):

    if num == 1:
        return 'alone'
    elif num > 2 and num < 5:
        return 'small'
    else:
        return 'large'
```

In [77]:
```
df['family_size'].apply(trasform_size)
```

Out[77]:
```
0      large
1      large
2      alone
3      large
4      alone
       ...
413    alone
414    alone
415    alone
416    alone
417    small
Name: family_size, Length: 1309, dtype: object
```

In [78]:
```
df['family_type'] = df['family_size'].apply(trasform_size)
```

In [79]:
```
df.sample(3)
```

Out[79]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Ca
734	735	0.0	2	Troupiansky, Mr. Moses Aaron	male	23.0	0	0	233639	13.00	N
594	595	0.0	2	Chapman, Mr. John Henry	male	37.0	1	0	SC/AH 29037	26.00	N
888	889	0.0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	N

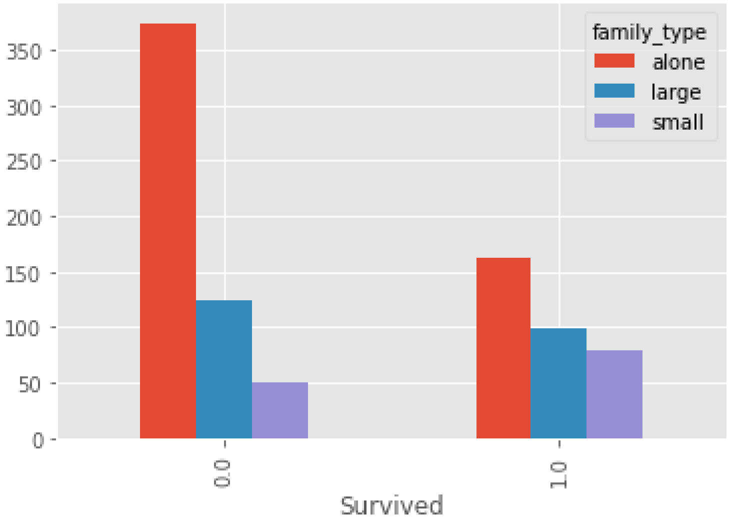
In [80]:
```
# Bivariate Analysis

pd.crosstab(df['Survived'],df['family_type'])
```

Out[80]:

family_type	alone	large	small
Survived			
0.0	374	124	51
1.0	163	99	80

In [81]:
```
pd.crosstab(df['Survived'],df['family_type']).plot(kind ='bar')
```

Out[81]:
```
<AxesSubplot:xlabel='Survived'>
```



In [82]:
```
# Normalize

pd.crosstab(df['Survived'],df['family_type'],normalize=True)*100
```

Out[82]:

family_type	alone	large	small
Survived			
0.0	41.975309	13.916947	5.723906
1.0	18.294052	11.111111	8.978676

In [83]:
```
df
```

Out[83]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
0	1	0.0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25
1	2	1.0	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.28
2	3	1.0	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.92
3	4	1.0	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.10
4	5	0.0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.05
...	...	...	...	...	...	...	...	...	...	
413	1305	NaN	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.05
414	1306	NaN	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.90
415	1307	NaN	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.25
416	1308	NaN	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.05
417	1309	NaN	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.35

1309 rows × 15 columns

In [84]:
```
# Surname

df['surname'] = df['Name'].str.split(',').str.get(0)
```

In [85]:
```
df
```

Out[85]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
0	1	0.0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25
1	2	1.0	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.28
2	3	1.0	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.92
3	4	1.0	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.10
4	5	0.0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.05
...	...	...	...	...	...	...	...	...	...	
413	1305	NaN	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.05
414	1306	NaN	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.90
415	1307	NaN	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.25
416	1308	NaN	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.05
417	1309	NaN	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.35

1309 rows × 16 columns

In [86]:
```
df['surname'].value_counts()
```

Out[86]:
```
Andersson    11
Sage         11
Goodwin       8
Asplund       8
Davies        7
             ..
Milling       1
Maisner       1
Goncalves     1
Campbell      1
Saether       1
Name: surname, Length: 875, dtype: int64
```

In [87]:
```
# titles

df ['title'] = df['Name'].str.split(',').str.get(1).str.split('.').str.get(0)
```

In [88]:
```
df
```

Out[88]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
0	1	0.0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25
1	2	1.0	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.28
2	3	1.0	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.92
3	4	1.0	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.10
4	5	0.0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.05
...	...	...	...	...	...	...	...	...	...	
413	1305	NaN	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.05
414	1306	NaN	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.90
415	1307	NaN	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.25
416	1308	NaN	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.05
417	1309	NaN	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.35

1309 rows × 17 columns

In [89]:
```
df['title'].value_counts()
```

Out[89]:
```
 Mr                 757
 Miss               260
 Mrs                197
 Master              61
 Rev                  8
 Dr                   8
 Col                  4
 Mlle                 2
 Major                2
 Ms                   2
 Lady                 1
 Sir                  1
 Mme                  1
 Don                  1
 Capt                 1
 the Countess         1
 Jonkheer             1
 Dona                 1
Name: title, dtype: int64
```

In [95]:
```
df
```

Out[95]:

ld	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embaı
1	0.0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	
2	1.0	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	
3	1.0	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
4	1.0	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	
5	0.0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	
...	...	...	...	...	...	...	...	...	...	...	
)5	NaN	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	NaN	
)6	NaN	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	
)7	NaN	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	NaN	
)8	NaN	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	NaN	
)9	NaN	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	NaN	

columns

In [103]:
```
df['Cabin'].isnull().sum()
```

Out[103]:
```
1014
```

In [106]:
```
df['Cabin'].isnull().sum() /len(df['Cabin'])
```

Out[106]:
```
0.774637127578304
```

In [108]:
```
df['Cabin'].value_counts().head(10)
```

Out[108]:
```
C23 C25 C27        6
G6                 5
B57 B59 B63 B66    5
C22 C26            4
F33                4
F2                 4
B96 B98            4
C78                4
F4                 4
D                  4
Name: Cabin, dtype: int64
```

In [113]:
```
df['Cabin'].fillna('M', inplace =True)
```

In [114]:
```
df['Cabin'].value_counts().head(10)
```

Out[114]:
```
M                  1014
C23 C25 C27           6
B57 B59 B63 B66       5
G6                    5
F33                   4
D                     4
C78                   4
B96 B98               4
F4                    4
F2                    4
Name: Cabin, dtype: int64
```

In [115]:
```
df['deck'] = df['Cabin'].str[0]
```

In [116]:
```
df
```

Out[116]:

rvived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	i
0.0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	M	S	
1.0	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C	
1.0	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	M	S	
1.0	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S	
0.0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	M	S	
...	...	...	...	...	...	...	...	...	...	...	
NaN	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	M	S	
NaN	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	C	
NaN	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	M	S	
NaN	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	M	S	
NaN	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	M	C	

ns

In [118]:
```
df['deck'].value_counts()
```

Out[118]:
```
M    1014
C      94
B      65
D      46
E      41
A      22
F      21
G       5
T       1
Name: deck, dtype: int64
```

In [122]:
```
pd.crosstab(df['deck'],df['Pclass'])
```

Out[122]:

Pclass	1	2	3
deck			
A	22	0	0
B	65	0	0
C	94	0	0
D	40	6	0
E	34	4	3
F	0	13	8
G	0	0	5
M	67	254	693
T	1	0	0

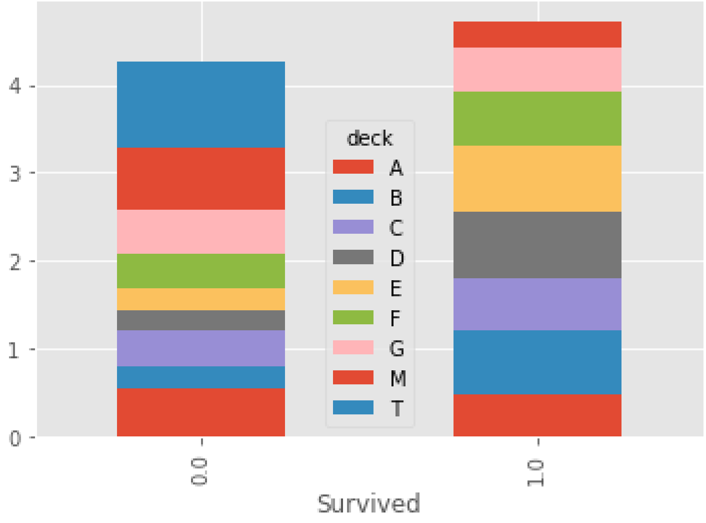
In [128]:
```
pd.crosstab(df['Survived'],df['deck'],normalize='columns')*100
```

Out[128]:

deck	A	B	C	D	E	F	G	M	T
Survived									
0.0	53.333333	25.531915	40.677966	24.242424	25.0	38.461538	50.0	70.014556	100.0
1.0	46.666667	74.468085	59.322034	75.757576	75.0	61.538462	50.0	29.985444	0.0

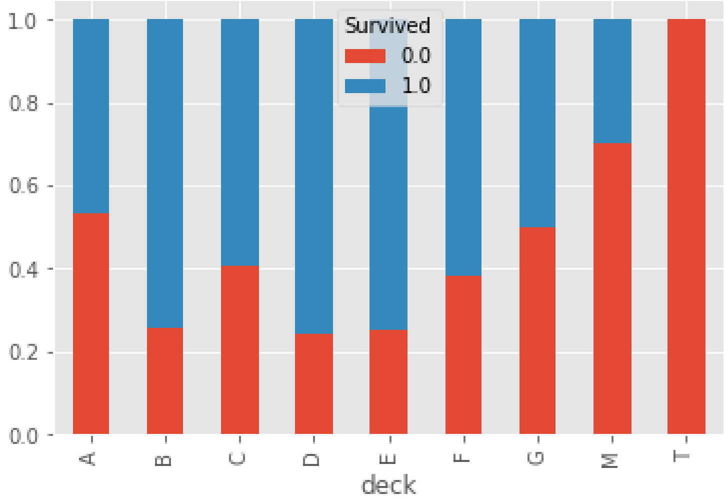
```
In [129]: pd.crosstab(df['Survived'],df['deck'],
                      normalize='columns').plot(kind ='bar',stacked =True)
```

```
Out[129]: <AxesSubplot:xlabel='Survived'>
```



```
In [130]: pd.crosstab(df['deck'],df['Survived'],
                      normalize='index').plot(kind='bar',stacked=True)
```

```
Out[130]: <AxesSubplot:xlabel='deck'>
```



In [131]:

```
# Multivariate analysis

df.corr()
```

Out[131]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	ind
PassengerId	1.000000	-0.005007	-0.038354	0.028814	-0.055224	0.008942	0.031428	
Survived	-0.005007	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307	
Pclass	-0.038354	-0.338481	1.000000	-0.408106	0.060832	0.018322	-0.558629	
Age	0.028814	-0.077221	-0.408106	1.000000	-0.243699	-0.150917	0.178740	
SibSp	-0.055224	-0.035322	0.060832	-0.243699	1.000000	0.373587	0.160238	
Parch	0.008942	0.081629	0.018322	-0.150917	0.373587	1.000000	0.221539	
Fare	0.031428	0.257307	-0.558629	0.178740	0.160238	0.221539	1.000000	
individual_fare	0.035365	0.221600	-0.504270	0.193545	-0.089807	-0.065498	0.832029	
family_size	-0.031437	0.016639	0.050027	-0.240229	0.861952	0.792296	0.226492	

In [135]:

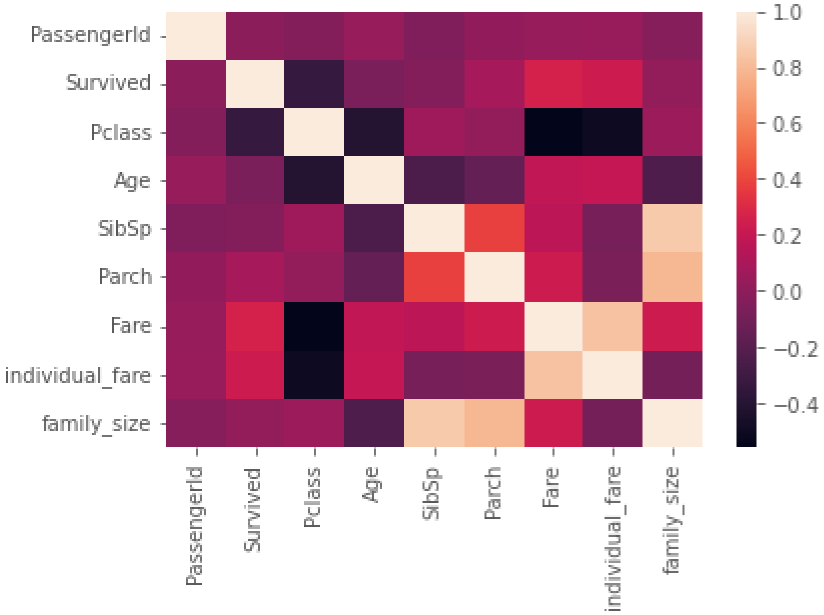
```
df.corr()['Survived']
```

Out[135]:

```
PassengerId       -0.005007
Survived           1.000000
Pclass            -0.338481
Age               -0.077221
SibSp             -0.035322
Parch              0.081629
Fare               0.257307
individual_fare    0.221600
family_size        0.016639
Name: Survived, dtype: float64
```

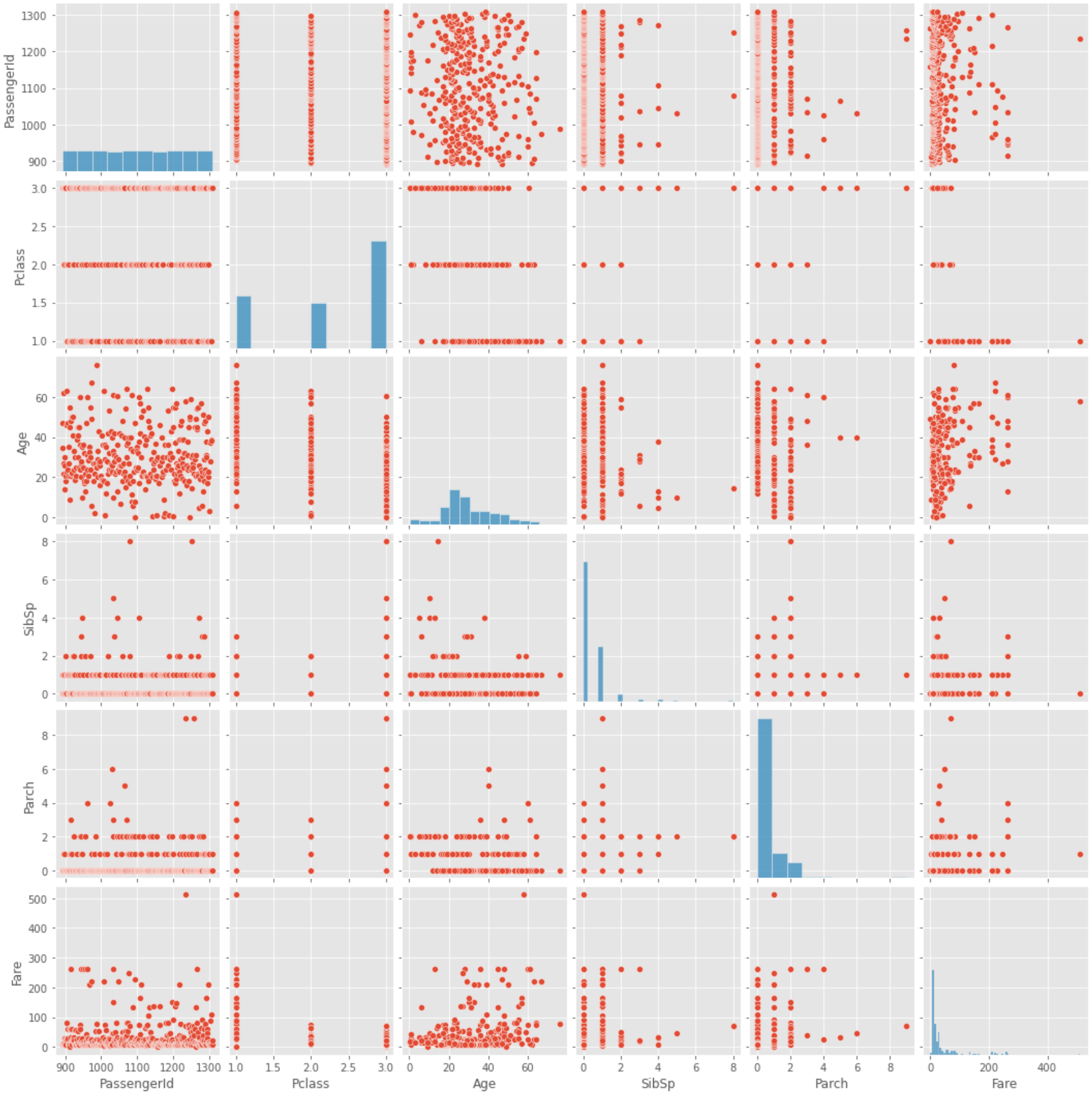
```
In [136]: sns.heatmap(df.corr())
```

```
Out[136]: <AxesSubplot:>
```



In [138]:
```
sns.pairplot(df1)
```

Out[138]:
```
<seaborn.axisgrid.PairGrid at 0x29080c2e040>
```


PassengerId
Pclass
Age
SibSp
Parch
Fare
PassengerId
Pclass
Age
SibSp
Parch
Fare