

```
In [56]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

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

```
In [57]: # Check Python Library Version
import pandas as pd
import numpy as np
import seaborn as sns

print('numpy:{}'.format(np.__version__))
print('pandas:{}'.format(pd.__version__))
print('seaborn:{}'.format(sns.__version__))
```

```
numpy:1.24.3
pandas:2.0.3
seaborn:0.12.2
```

```
In [58]: # Read CSV File
import pandas as pd
da = pd.read_csv(r"C:\Users\jasja\Downloads\archive\Titanic-Dataset.csv")
da.head().style.set_properties(**{'background-color': 'Black',
                                   'color': 'white',
                                   'border-color': 'darkblack'})
```

Out[58]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
0	1	0	3	Braund, Mr. Owen Harris	male	22.000000	1	0	A/5 21171	7
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38.000000	1	0	PC 17599	71
2	3	1	3	Heikkinen, Miss. Laina	female	26.000000	0	0	STON/O2. 3101282	7
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.000000	1	0	113803	53
4	5	0	3	Allen, Mr. William Henry	male	35.000000	0	0	373450	8

```
In [59]: # Drop Columns from the dataframe
da_drop_col = da.drop(['PassengerId', 'Name', 'Ticket'], axis=1)
da_drop_col.head()
```

Out[59]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	0	3	male	22.0	1	0	7.2500	NaN	S
1	1	1	female	38.0	1	0	71.2833	C85	C
2	1	3	female	26.0	0	0	7.9250	NaN	S
3	1	1	female	35.0	1	0	53.1000	C123	S
4	0	3	male	35.0	0	0	8.0500	NaN	S

```
In [60]: # Drop Rows from the dataframe
da_drop_rows = da.drop(labels=[0,1,3,6], axis=0)
da_drop_rows.head()
```

Out[60]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333

In [61]: `da.describe()`

Out[61]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [62]: `da[da['Survived']==0].describe().T.style.background_gradient \`
`(subset=['mean', 'std', '50%', 'count'], cmap='RdPu')`

Out[62]:

	count	mean	std	min	25%	50%	75%
PassengerId	549.000000	447.016393	260.640469	1.000000	211.000000	455.000000	675.0000
Survived	549.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0000
Pclass	549.000000	2.531876	0.735805	1.000000	2.000000	3.000000	3.0000
Age	424.000000	30.626179	14.172110	1.000000	21.000000	28.000000	39.0000
SibSp	549.000000	0.553734	1.288399	0.000000	0.000000	0.000000	1.0000
Parch	549.000000	0.329690	0.823166	0.000000	0.000000	0.000000	0.0000
Fare	549.000000	22.117887	31.388207	0.000000	7.854200	10.500000	26.0000

In [63]: `da.describe(percentiles=[0.05,0.25,0.35,0.5,0.75,0.85,0.95,0.995,0.999])`

Out[63]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
5%	45.500000	0.000000	1.000000	4.000000	0.000000	0.000000	7.225000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
35%	312.500000	0.000000	2.000000	24.000000	0.000000	0.000000	9.000000
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
85%	757.500000	1.000000	3.000000	45.000000	1.000000	1.000000	56.495800
95%	846.500000	1.000000	3.000000	56.000000	3.000000	2.000000	112.079150
99.5%	886.550000	1.000000	3.000000	70.717500	8.000000	5.000000	263.000000
99.9%	890.110000	1.000000	3.000000	75.722000	8.000000	5.110000	512.329200
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [64]: `da[['Survived', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare']].corr()`

Out[64]:

	Survived	Pclass	Age	SibSp	Parch	Fare
Survived	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307
Pclass	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549500
Age	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067
SibSp	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651
Parch	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225
Fare	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000

```
In [65]: da[da['Survived']==0].describe().T.style.background_gradient \
(subset=['mean','std','50%','count'], cmap='RdPu')
```

Out[65]:

	count	mean	std	min	25%	50%	75%
PassengerId	549.000000	447.016393	260.640469	1.000000	211.000000	455.000000	675.0000
Survived	549.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0000
Pclass	549.000000	2.531876	0.735805	1.000000	2.000000	3.000000	3.0000
Age	424.000000	30.626179	14.172110	1.000000	21.000000	28.000000	39.0000
SibSp	549.000000	0.553734	1.288399	0.000000	0.000000	0.000000	1.0000
Parch	549.000000	0.329690	0.823166	0.000000	0.000000	0.000000	0.0000
Fare	549.000000	22.117887	31.388207	0.000000	7.854200	10.500000	26.0000

```
In [66]: da.describe(percentiles=[0.05,0.25,0.35,0.5,0.75,0.85,0.95,0.995,0.999])
```

Out[66]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
5%	45.500000	0.000000	1.000000	4.000000	0.000000	0.000000	7.225000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
35%	312.500000	0.000000	2.000000	24.000000	0.000000	0.000000	9.000000
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
85%	757.500000	1.000000	3.000000	45.000000	1.000000	1.000000	56.495800
95%	846.500000	1.000000	3.000000	56.000000	3.000000	2.000000	112.079150
99.5%	886.550000	1.000000	3.000000	70.717500	8.000000	5.000000	263.000000
99.9%	890.110000	1.000000	3.000000	75.722000	8.000000	5.110000	512.329200
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

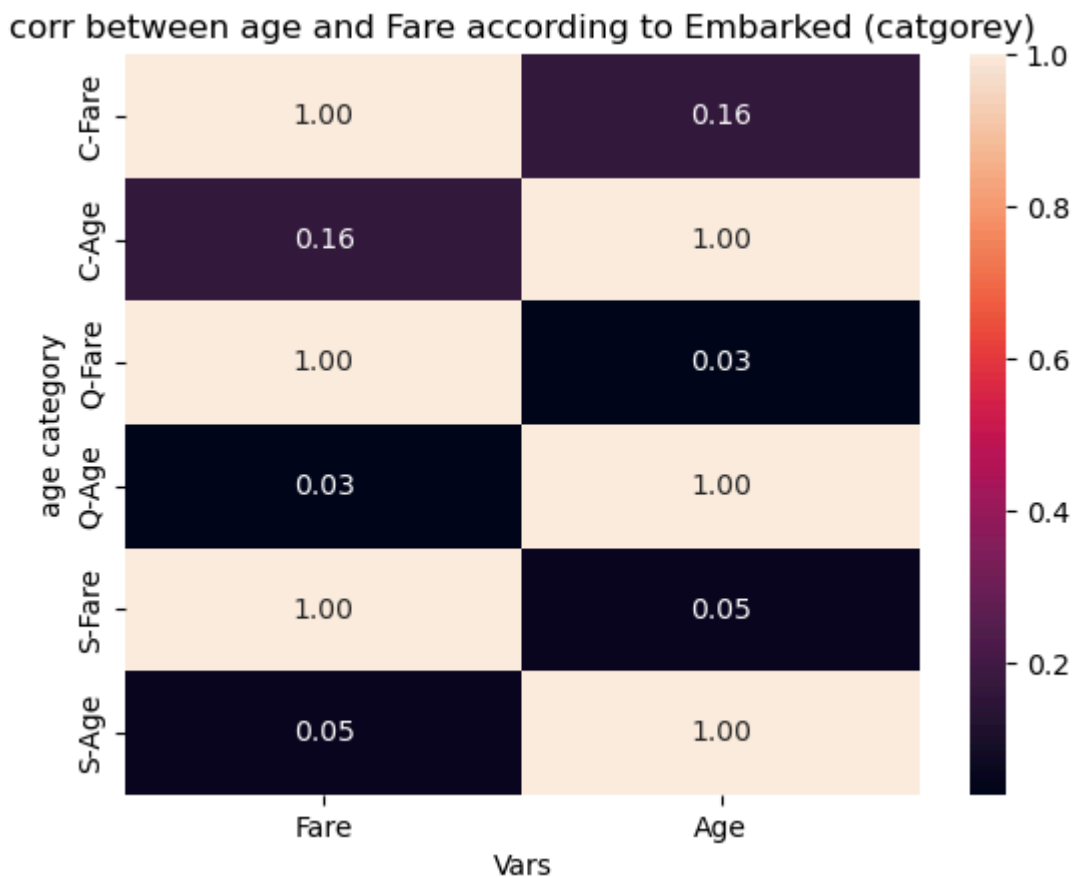
```
In [67]: da[['Survived', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare']].corr()
```

Out[67]:

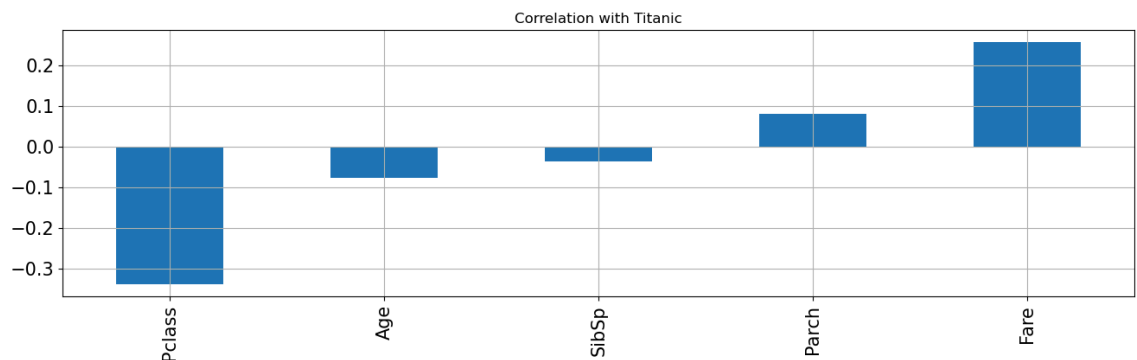
	Survived	Pclass	Age	SibSp	Parch	Fare
Survived	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307
Pclass	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549500
Age	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067
SibSp	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651
Parch	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225
Fare	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000

```
In [68]: corr = da.groupby(["Embarked"])[["Fare", "Age"]].corr()
corr.head()

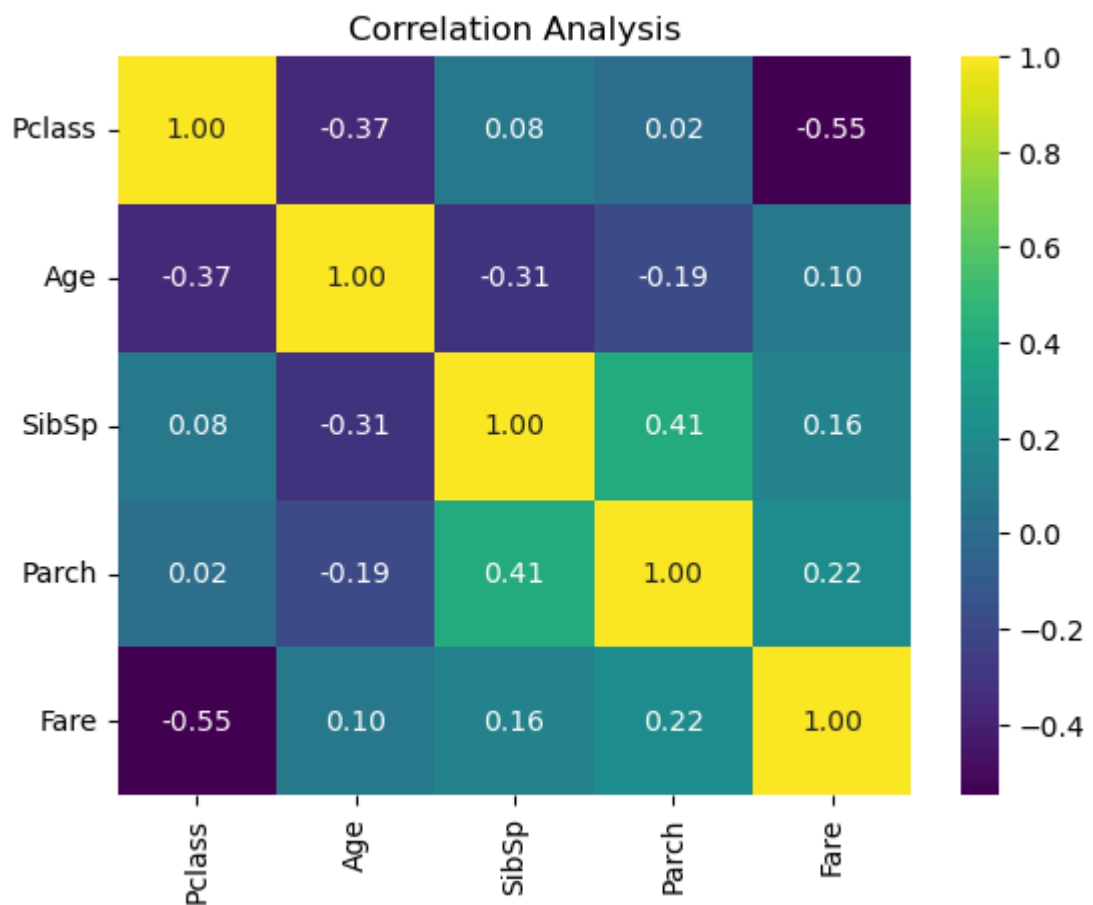
sns.heatmap(corr, annot=True, fmt=".2f")
plt.xlabel("Vars")
plt.ylabel("age category")
plt.title("corr between age and Fare according to Embarked (catgorey)")
plt.show()
```



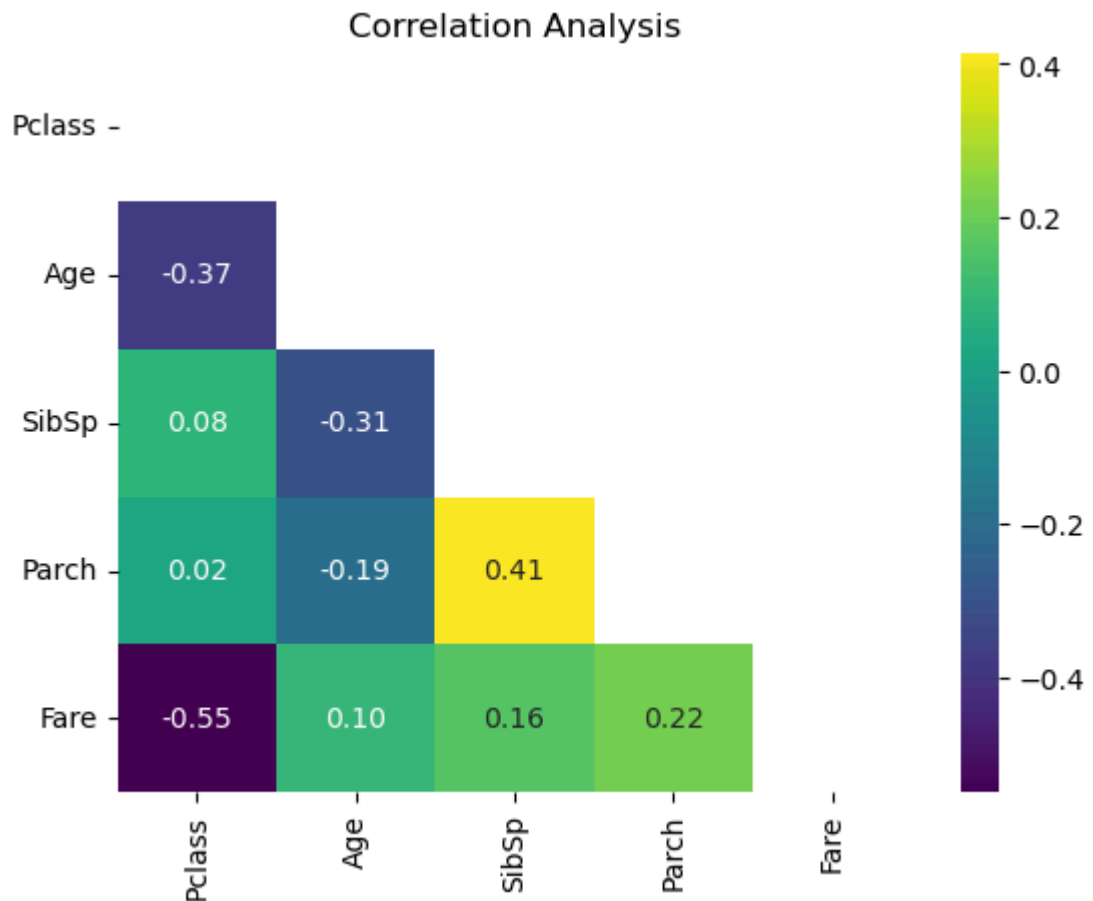
```
In [69]: X = da[['Pclass', 'Age', 'SibSp', 'Parch', 'Fare']]
y = da['Survived']
X.corrwith(y).plot.bar(
    figsize = (16, 4), title = "Correlation with Titanic", fontsize = 1
    rot = 90, grid = True)
plt.show()
```



```
In [70]: corr = da[['Pclass', 'Age', 'SibSp', 'Parch', 'Fare']].corr()
plt.figure(dpi=100)
plt.title('Correlation Analysis')
sns.heatmap(corr, annot=True, lw=0, linecolor='white', cmap='viridis', fmt = "0.
plt.xticks(rotation=90)
plt.yticks(rotation = 0)
plt.show()
```



```
In [71]: corr = da[['Pclass', 'Age', 'SibSp', 'Parch', 'Fare']].corr()
mask = np.triu(np.ones_like(corr, dtype = bool))
plt.figure(dpi=100)
plt.title('Correlation Analysis')
sns.heatmap(corr, mask=mask, annot=True, lw=0, linecolor='white', cmap='viridis')
plt.xticks(rotation=90)
plt.yticks(rotation = 0)
plt.show()
```



```
In [72]: da[['Age', 'Fare', 'Pclass']].agg(['sum', 'max', 'mean', 'std', 'skew', 'kurt'])
```

Out[72]:

	Age	Fare	Pclass
sum	21205.170000	28693.949300	2057.000000
max	80.000000	512.329200	3.000000
mean	29.699118	32.204208	2.308642
std	14.526497	49.693429	0.836071
skew	0.389108	4.787317	-0.630548
kurt	0.178274	33.398141	-1.280015


```
In [73]: da.groupby("Embarked").agg({"Fare": np.mean, "Sex": np.size})
```

Out[73]:

	Fare	Sex
Embarked		
C	59.954144	168
Q	13.276030	77
S	27.079812	644

```
In [74]: da.groupby(["Embarked", "Pclass"]).agg({"Fare": [np.size, np.mean]})
```

Out[74]:

			Fare	
			size	mean
Embarked	Pclass			
C	1	85	104.718529	
	2	17	25.358335	
	3	66	11.214083	
Q	1	2	90.000000	
	2	3	12.350000	
	3	72	11.183393	
S	1	127	70.364862	
	2	164	20.327439	
	3	353	14.644083	

```
In [75]: da.pivot_table(index='Pclass', values='Age', aggfunc=np.median)
```

Out[75]:

	Age
Pclass	
1	37.0
2	29.0
3	24.0

```
In [76]: x=pd.DataFrame(pd.pivot_table(da,index=['Sex','Embarked'],aggfunc='count')[x
```

Out[76]:

		Fare	
Sex	Embarked		
female	C	73	
	Q	36	
	S	203	
male	C	95	
	Q	41	
	S	441	

```
In [77]: da.groupby(["Embarked", "Pclass"]).agg({"Fare": [np.size, np.mean]})
```

Out[77]:

		Fare	
		size	mean
Embarked	Pclass		
C	1	85	104.718529
	2	17	25.358335
	3	66	11.214083
Q	1	2	90.000000
	2	3	12.350000
	3	72	11.183393
S	1	127	70.364862
	2	164	20.327439
	3	353	14.644083

```
In [78]: da.groupby("Embarked").agg({"Fare": np.mean, "Sex": np.size})
```

Out[78]:

		Fare	Sex
Embarked			
C	59.954144	168	
Q	13.276030	77	
S	27.079812	644	

```
In [79]: da.groupby('Embarked').Fare.agg(['count', 'mean', 'sum']).sort_values('sum')
```

Out[79]:

	count	mean	sum
Embarked			
Q	77	13.276030	1022.2543
C	168	59.954144	10072.2962
S	644	27.079812	17439.3988

```
In [80]: da['Embarked'].value_counts().to_frame()
```

Out[80]:

	count
Embarked	
S	644
C	168
Q	77

```
In [81]: da['Embarked'].value_counts().tolist()
```

Out[81]: [644, 168, 77]

```
In [82]: males = da[da['Sex'] == 'male']
males['Survived'].value_counts()/len(males)
```

Out[82]: Survived
0 0.811092
1 0.188908
Name: count, dtype: float64

```
In [83]: da['Embarked'][da['Sex']=='female'].value_counts(normalize=True)*100
```

Out[83]: Embarked
S 65.064103
C 23.397436
Q 11.538462
Name: proportion, dtype: float64

```
In [85]: titanic = da.copy('deep')
titanic.fillna(method="ffill", inplace=True)
titanic["Age"] = titanic["Age"].fillna(titanic["Age"].mean())
titanic['Embarked'] = titanic['Embarked'].fillna(titanic['Embarked'] == 'Q'
titanic['Embarked'].fillna("C", inplace=True)
titanic.fillna({'Embarked' : 'Z'},inplace=True)
titanic['Embarked'] = titanic['Embarked'].fillna('S')
titanic = titanic['Embarked'].fillna(value='No Passenger')
titanic = da.copy('deep')
titanic.head()
```

Out[85]:

	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



In []: