타이타닉호 생존자 분석 보고서

경영학부 20(학번) 양현서(이름)

---

1. 문제 정의

특정 선착장의 생존율이 더 높은 이유를 알아보자(탑승자의 객실 등급과 성별, 나이 측면에서)

2. 데이터 임포트

In [1]:

```
import pandas as pd

train = pd.read_csv('train.csv')
test = pd.read_csv('test.csv')
```

3. 데이터 탐색

In [2]:

```
train.head()
```

Out[2]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	C
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 [3]:

```
test.head()
```

Out[3]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Emba
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	

In [4]:

```
print(train.shape, test.shape)
```

```
(891, 12) (418, 11)
```

In [5]:

```
print(train.info(), test.info())
```

```
memory usage: 88.7+ KB
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Pclass       418 non-null    int64  
 2   Name         418 non-null    object 
 3   Sex          418 non-null    object 
 4   Age          332 non-null    float64
 5   SibSp        418 non-null    int64  
 6   Parch        418 non-null    int64  
 7   Ticket       418 non-null    object 
 8   Fare         417 non-null    float64
 9   Cabin        91 non-null     object 
 10  Embarked     418 non-null    object 
dtypes: float64(2), int64(4), object(5)
memory usage: 36.0+ KB
None None
```

In [19]:

```
Embark_data = train.groupby(['Embarked'])
```

4. 데이터 시각화

4-1 데이터 시각화를 위한 라이브러리 임포트

In [6]:

```
%matplotlib inline
import matplotlib.pyplot as plt
```

4-2 그래프를 그리는 함수 생성

막대그래프

In [15]:

```
def bar_chart(feature):
    survived = train[train['Survived']==1][feature].value_counts()
    dead = train[train['Survived']==0][feature].value_counts()
    df = pd.DataFrame([survived, dead])
    df.index = ['Survived', 'Dead']
    df.plot(kind='bar', stacked=True, figsize=(10, 5))
```

파이차트

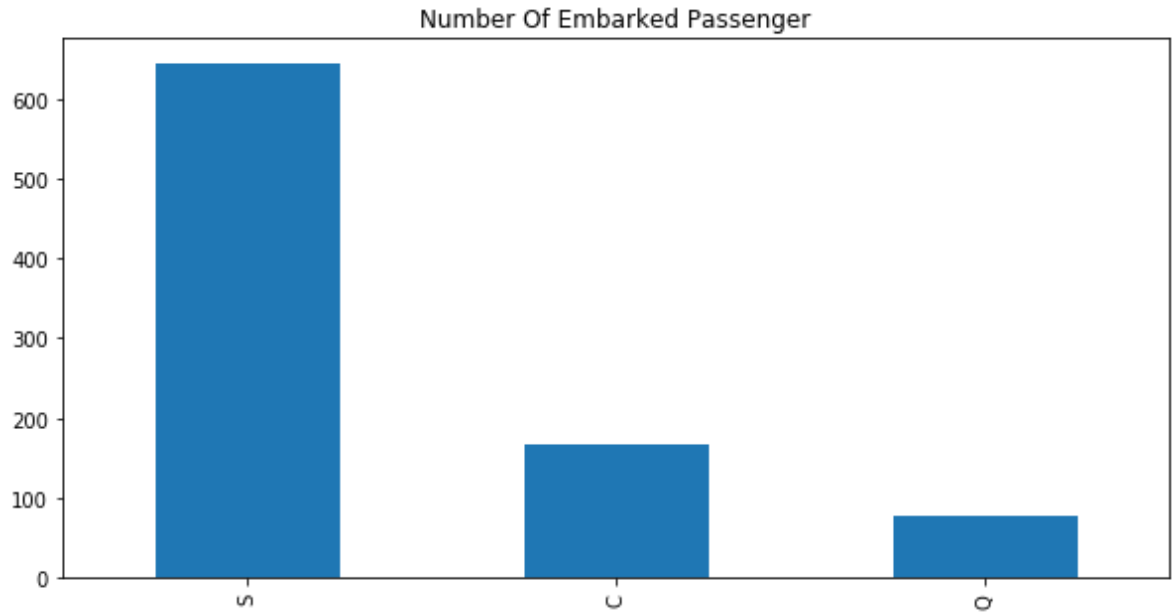
In [65]:

```
def pie_chart(feature):
    survived = train[train['Survived']==1][feature].value_counts()
    dead = train[train['Survived']==0][feature].value_counts()
    df = pd.DataFrame([survived, dead])
    df.index = ['Survived', 'Dead']
    df.plot(kind='pie', subplots=True, autopct='%.2f%%', figsize=(10, 5))
```

탑승구에 따른 탑승객 수

In [69]:

```
pclass_plt = train['Embarked'].value_counts()
pclass_plt.plot(kind='bar', figsize = (10,5))
plt.title('Number Of Embarked Passenger')
plt.show()
```

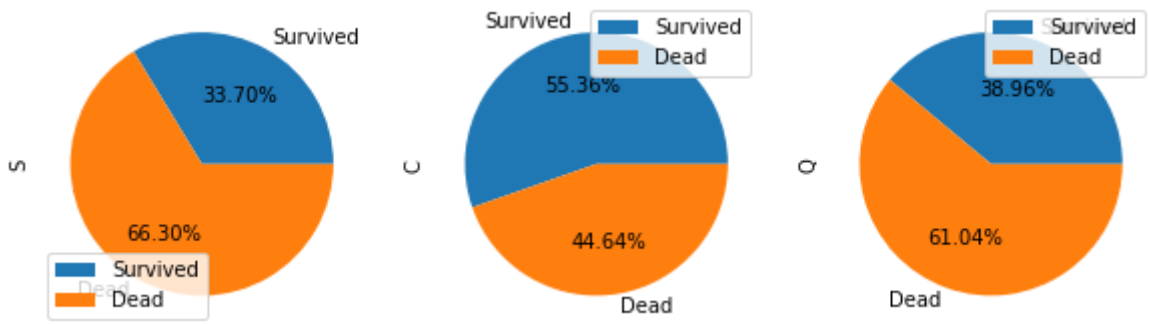


전체 탑승객의 수는 S, C, Q의 순서대로 많았다.

각 탑승구의 생존율과 사망률

In [66]:

```
pie_chart('Embarked')
```



C, Q, S순으로 생존율이 높았다. 그 이유가 무엇일까?

나의 추론 : 각 탑승구에 따라 탑승객의 수준이 달랐을 것이다. 객실별 생존율이 다른 것과 관련하여 탑승구에 따른 탑승객의 생존율이 다른 것이다. 또한, S탑승구가 탑승객이 많았던 만큼 더 혼란을 겪지 않았을까 생각한다. (더 고급진 곳에 위치한 탑승구일수록 더 높은 등급의 객실을 사용하는 사람들이 더 많이 이용할 것이고, 누구에게나 접촉이 쉬울수록 보편적으로 더 많은 사람, 그리고 객실의 등급이 낮은 사람들이 더 많이 이용했을 것이다. (높은 등급의 탑승객은 다른 탑승구를 이용했을 것이기 때문) 아마도 탑승구에 따라 객실의 위치가 정해진다면 (탑승구에서 구체적인 객실을 지정해준다면), 탑승객이 많았던 S탑승구에서 탑승한 승객들이 탈출시 더 많은 혼란을 겪었을 것이다.)

사용할 데이터인 Age, Fare를 Binning하기

In [73]:

```
# Missing Age를 각 Embarked에 대한 연령의 중간값 으로 채운다(S, C, Q)
train['Age'].fillna(train.groupby('Embarked')['Age'].transform('median'), inplace=True)
```

In [75]:

```
train_test_data = [train, test]
for dataset in train_test_data:
    dataset.loc[ dataset['Age'] <= 16, 'Age'] = 0,
    dataset.loc[(dataset['Age'] > 16) & (dataset['Age'] <= 26), 'Age'] = 1,
    dataset.loc[(dataset['Age'] > 26) & (dataset['Age'] <= 36), 'Age'] = 2,
    dataset.loc[(dataset['Age'] > 36) & (dataset['Age'] <= 62), 'Age'] = 3,
    dataset.loc[ dataset['Age'] > 62, 'Age'] = 4
```

메인 데이터인 Embarked를 객실에 따라 나눈다.

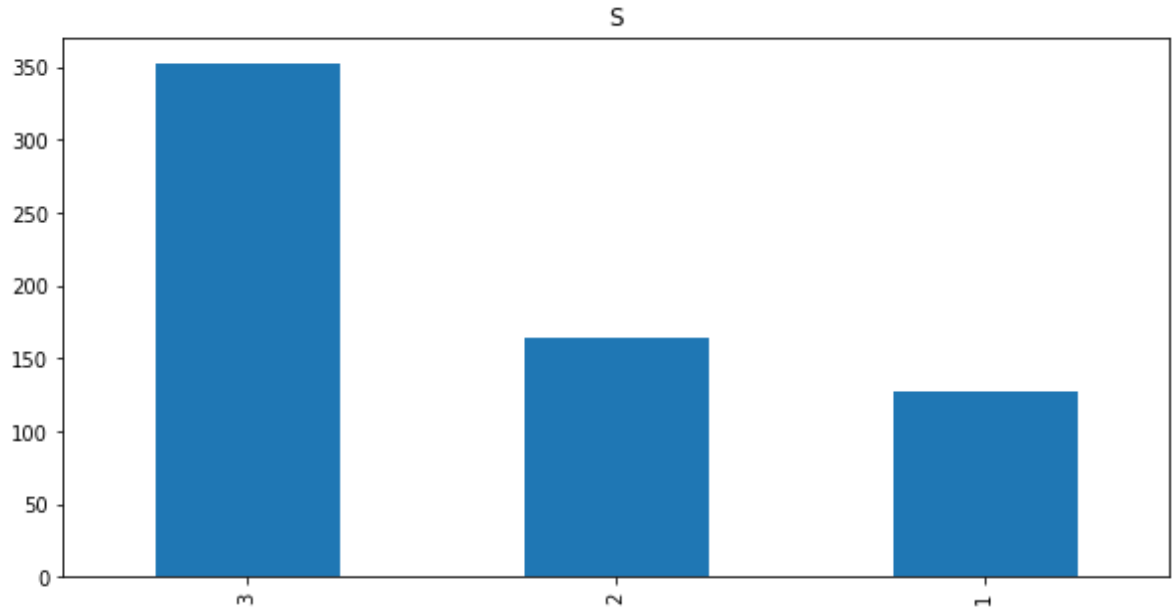
In [77]:

```
S_Embarked = train[train['Embarked']=='S']
C_Embarked = train[train['Embarked']=='C']
Q_Embarked = train[train['Embarked']=='Q']
```

S탑승구의 객실 분포

In [78]:

```
pclass_plt = S_Embarked['Pclass'].value_counts()
pclass_plt.plot(kind='bar', figsize = (10,5))
plt.title('S')
plt.show()
```

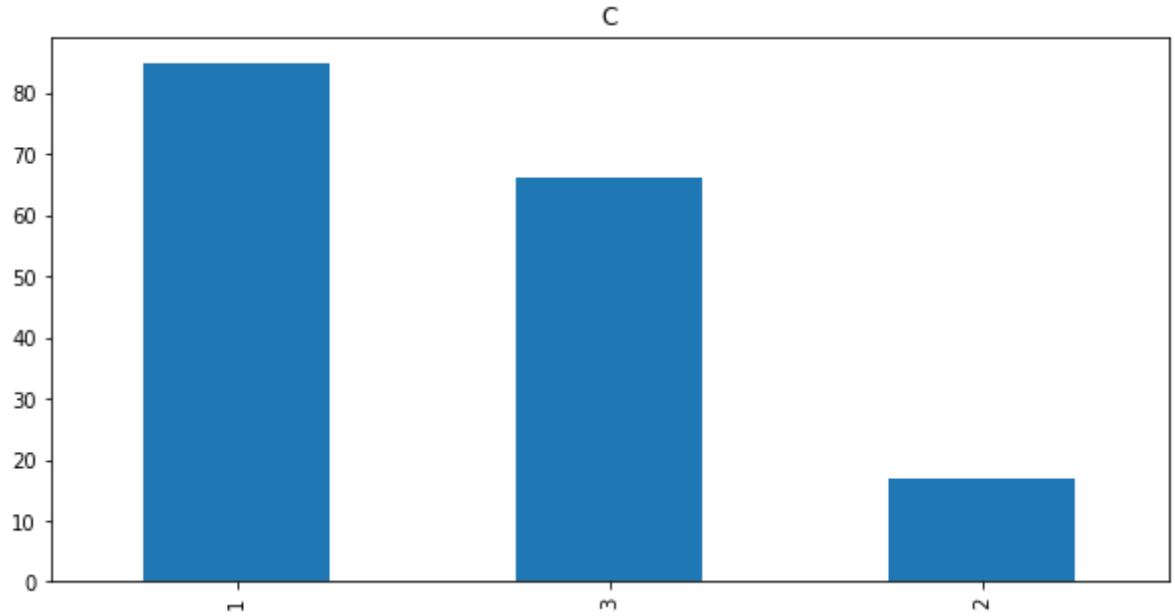


유난히 탑승객이 많은 이 탑승구는 다른 탑승구에 비해 보편적으로 접촉성이 좋았을 것으로 생각된다.

C탑승구의 객실 분포¶

In [79]:

```
pclass_plt = C_Embarked['Pclass'].value_counts()
pclass_plt.plot(kind='bar', figsize = (10,5))
plt.title('C')
plt.show()
```



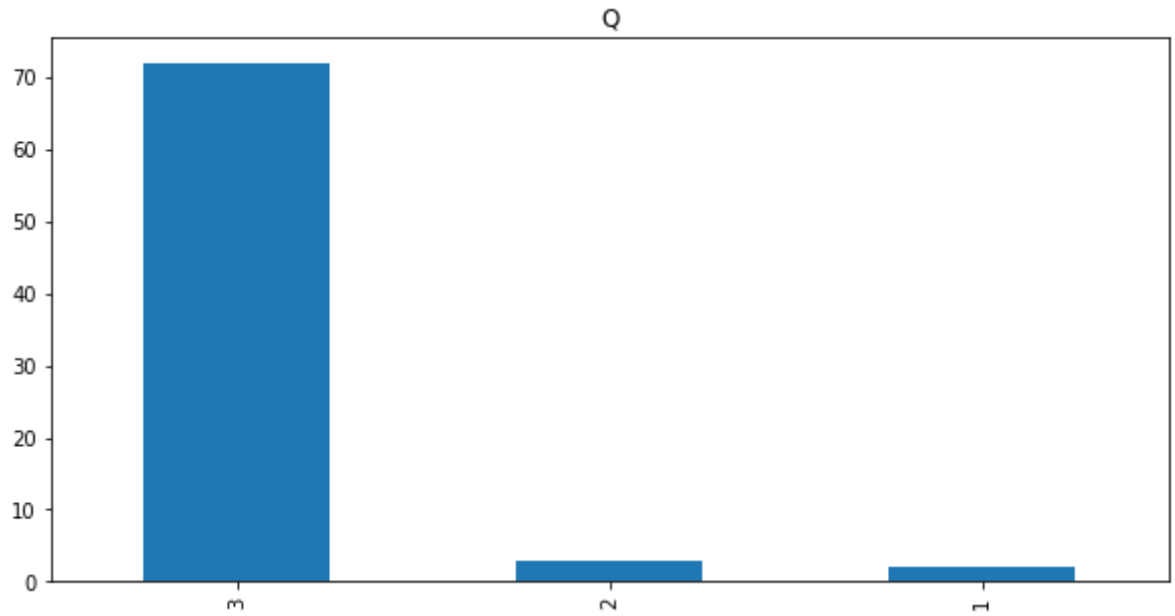
예상과 마찬가지로 특정 탑승구에서 1등급 객실의 탑승객이 다른 탑승구의 1등급 객실 탑승객보다 많았다. 이로

추론하기를 이 탑승구가 다른 탑승구보다 더 고급지고, 접촉성이 낮았을 것이다.

Q탑승구의 객실 분포¶

In [80]:

```
pclass_plt = Q_Embarked['Pclass'].value_counts()
pclass_plt.plot(kind='bar', figsize = (10,5))
plt.title('Q')
plt.show()
```



유난히 1등급 객실과 2등급 객실의 탑승객이 적고, 3등급 객실의 탑승객이 많은 것으로 봤을 때, 이 탑승구는 다른 탑승구에 비해 덜 고급진 구역, 비교적 재산이 많지 않은 사람들이 접촉하기는 더 쉽고, 전체적으로는 접촉하기 어려운 위치에 있을 것으로 예상된다.

각 탑승구별 막대 그래프 함수

In [52]:

```
def bar_chart_S(feature):
    survived = S_Embarked[S_Embarked['Survived']==1][feature].value_counts()
    dead = S_Embarked[S_Embarked['Survived']==0][feature].value_counts()
    df = pd.DataFrame([survived, dead])
    df.index = ['Survived', 'Dead']
    df.plot(kind='bar', stacked=True, figsize=(10, 5))
```

In [45]:

```
def bar_chart_C(feature):
    survived = C_Embarked[C_Embarked['Survived']==1][feature].value_counts()
    dead = C_Embarked[C_Embarked['Survived']==0][feature].value_counts()
    df = pd.DataFrame([survived, dead])
    df.index = ['Survived', 'Dead']
    df.plot(kind='bar', stacked=True, figsize=(10, 5))
```

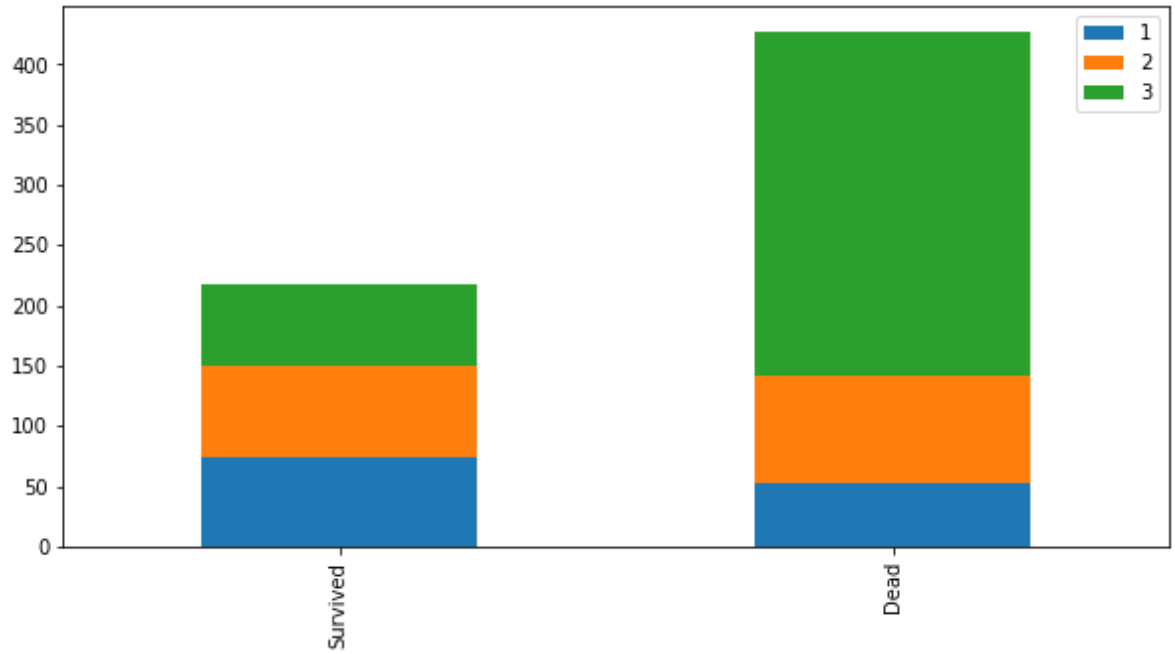
In [46]:

```
def bar_chart_Q(feature):
    survived = Q_Embarked[Q_Embarked['Survived']==1][feature].value_counts()
    dead = Q_Embarked[Q_Embarked['Survived']==0][feature].value_counts()
    df = pd.DataFrame([survived, dead])
    df.index = ['Survived', 'Dead']
    df.plot(kind='bar', stacked=True, figsize=(10, 5))
```

S선착장 탑승객의 생존율과 사망률 구체적으로 살펴보기

In [53]:

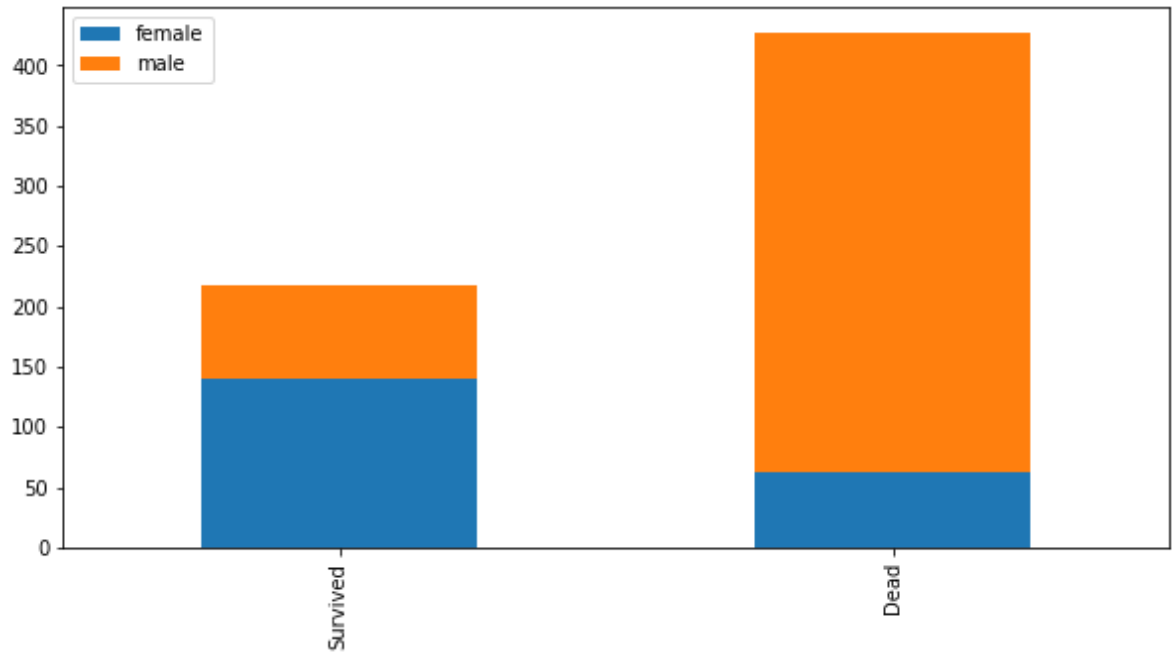
```
bar_chart_S('Pclass')
```



1등급 객실의 탑승객은 생존율이 사망률보다 높다. 3등급 객실의 탑승객의 사망률이 상당히 높다. 생존객의 수가 사망자의 수보다 월등히 낮고(약 절반정도), 생존률과 사망률의 차이의 가장 큰 원인은 3등급 객실 탑승객이다.

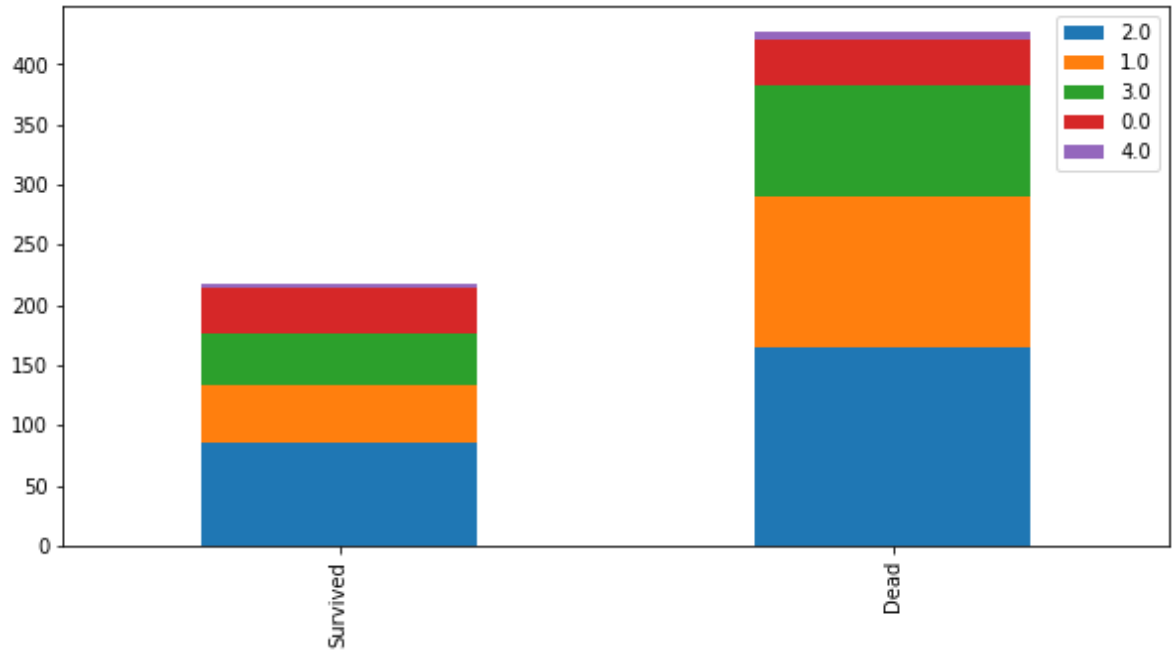
In [54]:

```
bar_chart_S('Sex')
```



In [81]:

```
bar_chart_S('Age')
```

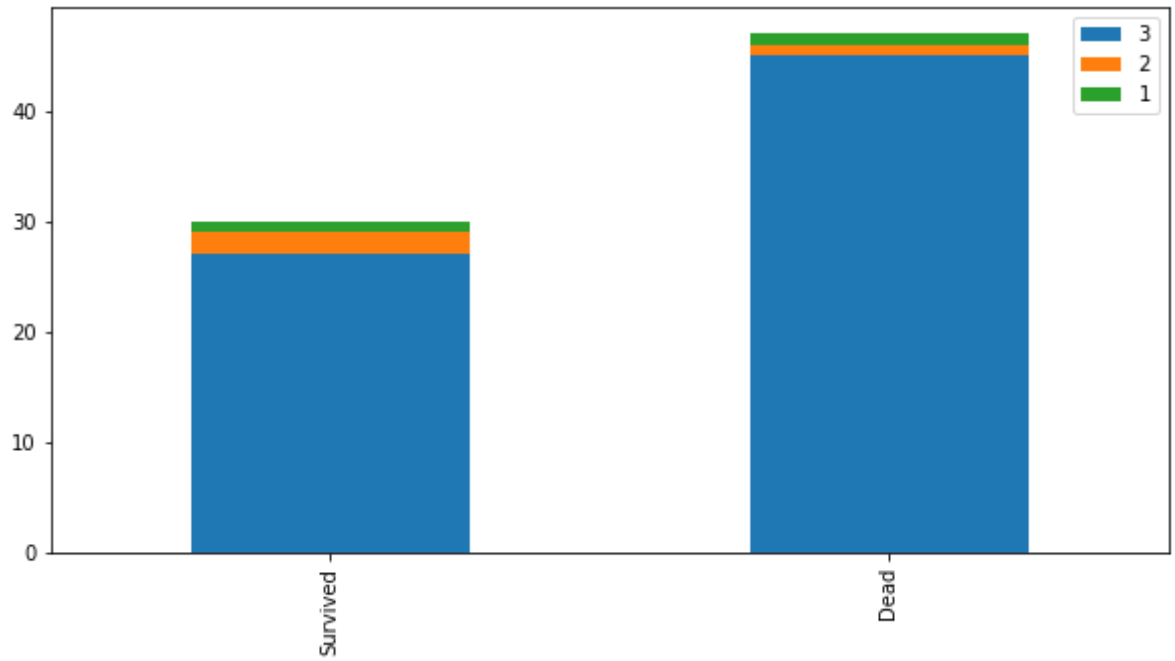


여성 생존율이 남성 생존율에 비해 상당히 높다. 또한 상대적으로 어린 탑승객의 생존율이 높다. 기존 예상과는 다르게 다른 탑승구보다 더 큰 혼란이 발생했을 것이라는 추론은 맞지 않은 것 같다

Q선착장 탑승객의 생존율과 사망률 구체적으로 살펴보기

In [61]:

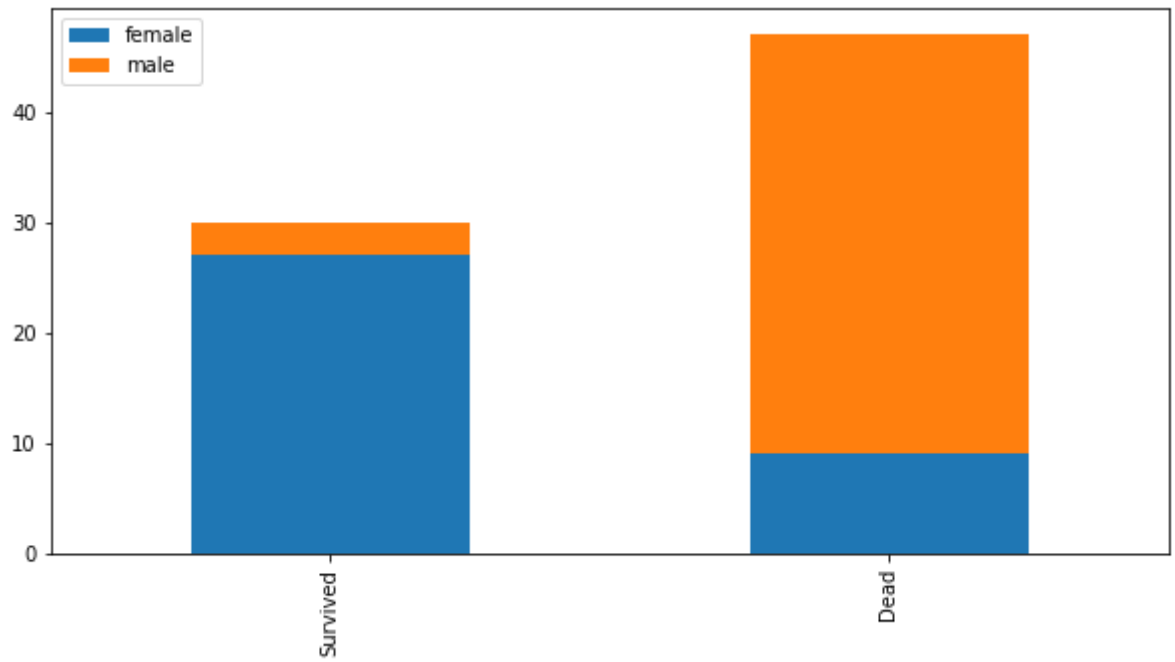
```
bar_chart_Q('Pclass')
```



상대적으로 1등급과 2등급 객실의 생존율이 더 높다. S탑승구에 비해 생존자의 비율이 사망자의 비율보다 더 높고, 여기서도 사망자와 생존자의 차이의 가장 큰 원인은 3등급 객실의 탑승객이다.

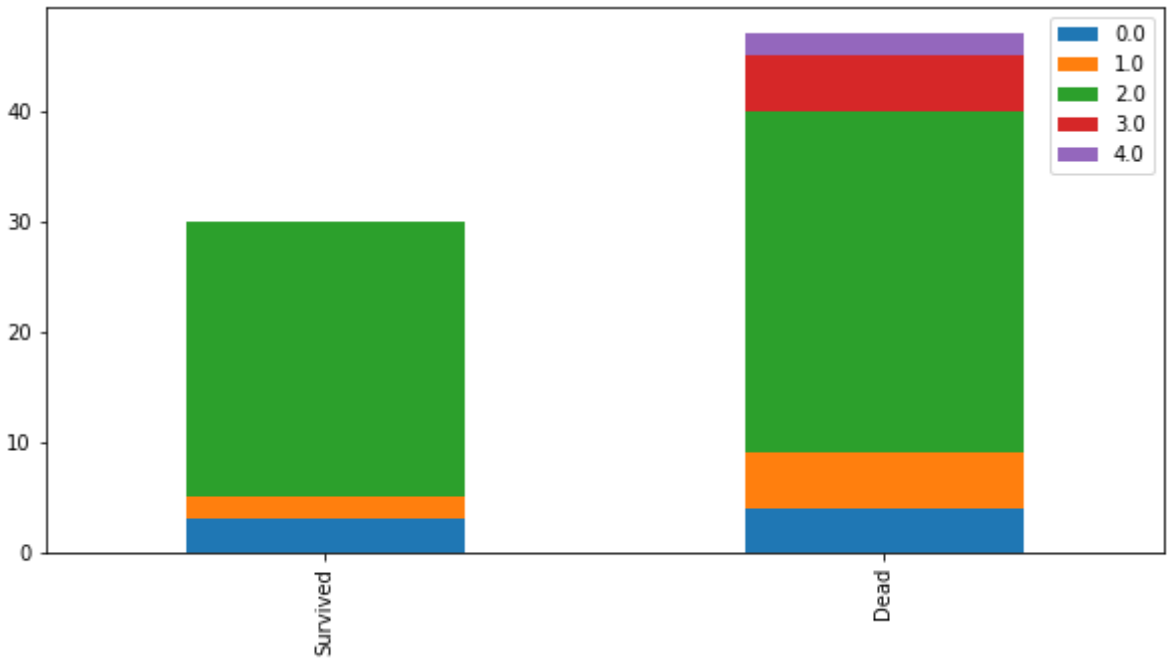
In [62]:

```
bar_chart_Q('Sex')
```



In [82]:

```
bar_chart_Q("Age")
```

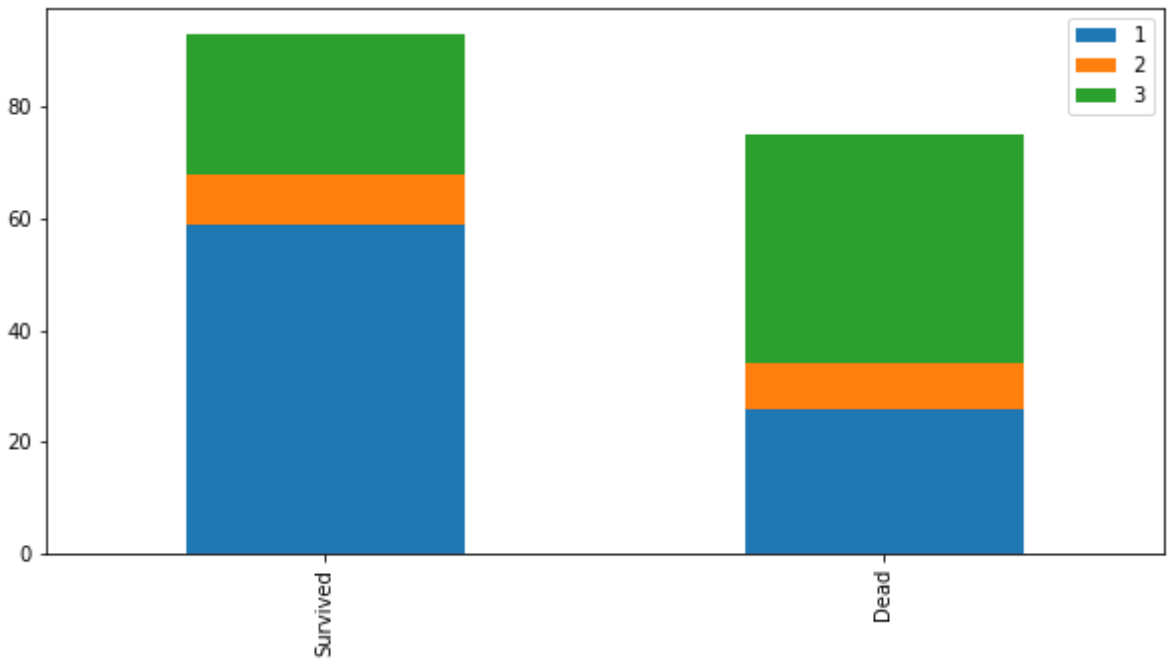


여성이 남성보다 생존율이 더 높은 것, 나이가 많은 탑승객은 거의 생존하지 못했고, 나이가 적은 탑승객의 생존율이 상대적으로 높다

C선착장 탑승객의 생존율과 사망률 구체적으로 살펴보기

In [63]:

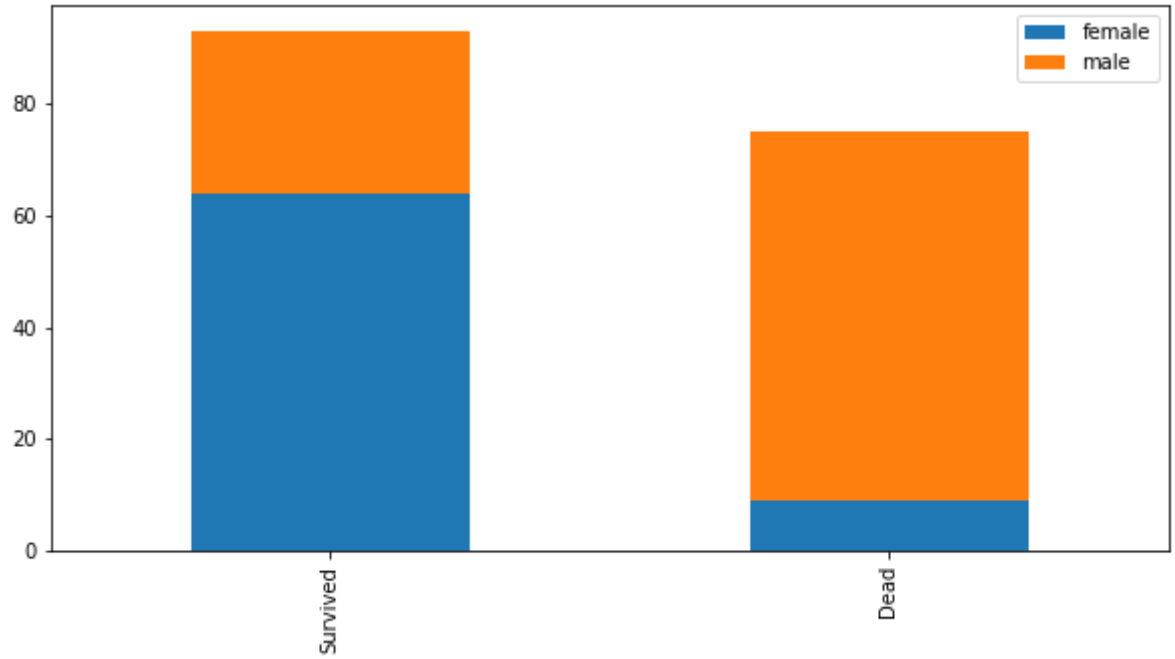
```
bar_chart_C('Pclass')
```



생존자의 수가 사망자의 수보다 더 많다. 다른 선착장에 비해 생존한 1등급 객실 탑승객의 절반 이상이 생존했다. 1등급 객실 탑승객이 많은 만큼 탈출에 더 많은 노력이 있지 않았을까 생각한다.

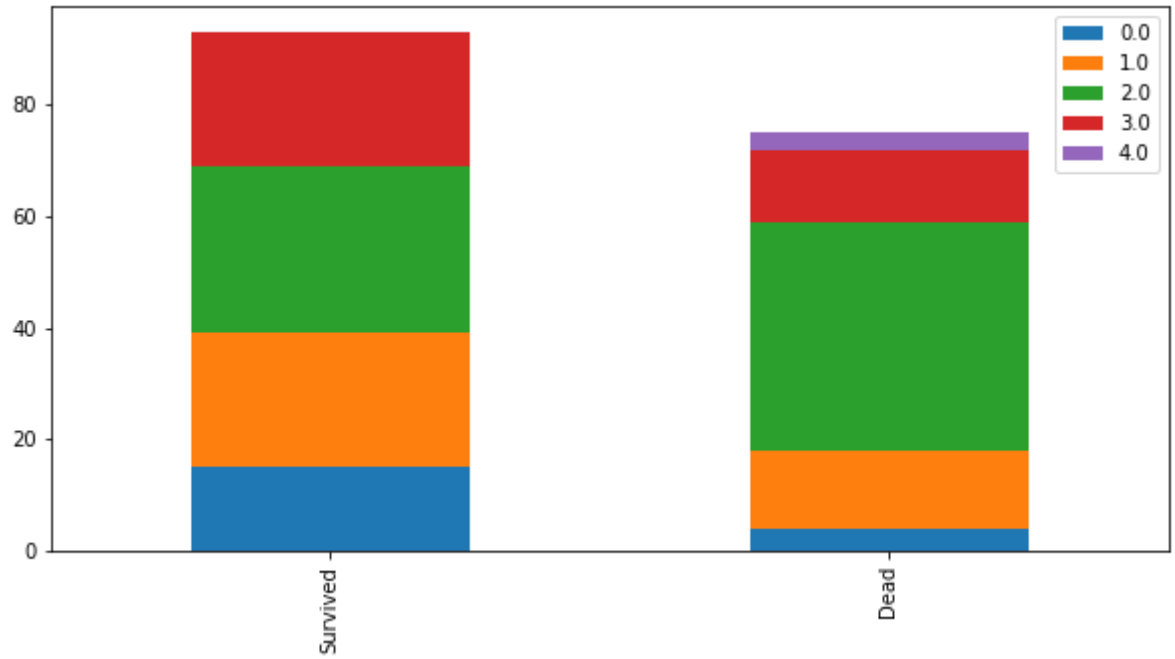
In [83]:

```
bar_chart_C("Sex")
```



In [84]:

```
bar_chart_C("Age")
```



다른 탑승구에 비해 남자 생존자의 비율이 많다. 1등급 객실의 생존자가 더 많은 것으로 생각해보건대 부유한 남성이 그렇지 않은 남성에 비해 탈출 가능성이 더 높았을것 같다

5. 결론 도출

선착장에 따라 탑승한 객실의 등급이 차이가 많이 나는 만큼 선착장에 따라 고유의 특성을 가지고 있었을 것이다. 선착장에 따른 생존율의 분명한 차이는 선착장의 특성에 따라 탑승한 탑승객의 경제적 지위의 차이로 인해 발생

했을 것이다. 여성과 어린 사람을 우선으로 하되 경제적 지위도 생존에 적지 않은 영향을 미쳤을 것이다.

In [ ]:

```
```