In [2]:
```
print("Hello Jupyter!")
```

```
Hello Jupyter!
```

In [3]:
```
#Definition of radius in km
r = 192500
#import radians function of math package
from math import radians
dist = r * radians(12)
print(dist)
```

```
40317.10572106901
```

In [4]:
```
import pandas as pd
train = pd.read_csv('datasets/train.csv')
test = pd.read_csv('datasets/test.csv')
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-4-942a4ac91985> in <module>
      1 import pandas as pd
----> 2 train = pd.read_csv('datasets/train.csv')
      3 test = pd.read_csv('datasets/test.csv')

D:\Anaconda3\lib\site-packages\pandas\io\parsers.py in parser_f(filepath_or_buf
fer, sep, delimiter, header, names, index_col, usecols, squeeze, prefix, mangle
_dupe_cols, dtype, engine, converters, true_values, false_values, skipinitialsp
ace, skiprows, skipfooter, nrows, na_values, keep_default_na, na_filter, verbos
e, skip_blank_lines, parse_dates, infer_datetime_format, keep_date_col, date_pa
rser, dayfirst, iterator, chunksize, compression, thousands, decimal, linetermi
nator, quotechar, quoting, doublequote, escapechar, comment, encoding, dialect,
tupleize_cols, error_bad_lines, warn_bad_lines, delim_whitespace, low_memory, m
emory_map, float_precision)
    700                     skip_blank_lines=skip_blank_lines)
    701
--> 702         return _read(filepath_or_buffer, kwds)
    703
    704     parser_f.__name__ = name

D:\Anaconda3\lib\site-packages\pandas\io\parsers.py in _read(filepath_or_buffe
r, kwds)
    427
    428     # Create the parser.
--> 429     parser = TextFileReader(filepath_or_buffer, **kwds)
    430
    431     if chunksize or iterator:

D:\Anaconda3\lib\site-packages\pandas\io\parsers.py in __init__(self, f, engin
e, **kwds)
    893             self.options['has_index_names'] = kwds['has_index_names']
    894
--> 895         self._make_engine(self.engine)
    896
    897     def close(self):

D:\Anaconda3\lib\site-packages\pandas\io\parsers.py in _make_engine(self, engin
e)
   1120     def _make_engine(self, engine='c'):
   1121         if engine == 'c':
-> 1122             self._engine = CParserWrapper(self.f, **self.options)
   1123         else:
   1124             if engine == 'python':

D:\Anaconda3\lib\site-packages\pandas\io\parsers.py in __init__(self, src, **kw
ds)
   1851         kwds['usecols'] = self.usecols
   1852
-> 1853         self._reader = parsers.TextReader(src, **kwds)
   1854         self.unnamed_cols = self._reader.unnamed_cols
```

```
   1855

pandas/_libs/parsers.pyx in pandas._libs.parsers.TextReader.__cinit__()

pandas/_libs/parsers.pyx in pandas._libs.parsers.TextReader._setup_parser_sourc
e()

FileNotFoundError: [Errno 2] File b'datasets/train.csv' does not exist: b'datas
ets/train.csv'
```

In [5]:
```
import pandas as pd
train = pd.read_csv('datasets/titanic/train.csv')
test = pd.read_csv('datasets/titanic/test.csv')
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-5-942c60a736af> in <module>
      1 import pandas as pd
----> 2 train = pd.read_csv('datasets/titanic/train.csv')
      3 test = pd.read_csv('datasets/titanic/test.csv')

D:\Anaconda3\lib\site-packages\pandas\io\parsers.py in parser_f(filepath_or_buf
fer, sep, delimiter, header, names, index_col, usecols, squeeze, prefix, mangle
_dupe_cols, dtype, engine, converters, true_values, false_values, skipinitialsp
ace, skiprows, skipfooter, nrows, na_values, keep_default_na, na_filter, verbos
e, skip_blank_lines, parse_dates, infer_datetime_format, keep_date_col, date_pa
rser, dayfirst, iterator, chunksize, compression, thousands, decimal, linetermi
nator, quotechar, quoting, doublequote, escapechar, comment, encoding, dialect,
tupleize_cols, error_bad_lines, warn_bad_lines, delim_whitespace, low_memory, m
emory_map, float_precision)
    700                     skip_blank_lines=skip_blank_lines)
    701
--> 702         return _read(filepath_or_buffer, kwds)
    703
    704     parser_f.__name__ = name

D:\Anaconda3\lib\site-packages\pandas\io\parsers.py in _read(filepath_or_buffe
r, kwds)
    427
    428     # Create the parser.
--> 429     parser = TextFileReader(filepath_or_buffer, **kwds)
    430
    431     if chunksize or iterator:

D:\Anaconda3\lib\site-packages\pandas\io\parsers.py in __init__(self, f, engin
e, **kwds)
    893             self.options['has_index_names'] = kwds['has_index_names']
    894
--> 895         self._make_engine(self.engine)
    896
    897     def close(self):

D:\Anaconda3\lib\site-packages\pandas\io\parsers.py in _make_engine(self, engin
e)
   1120     def _make_engine(self, engine='c'):
   1121         if engine == 'c':
-> 1122             self._engine = CParserWrapper(self.f, **self.options)
   1123         else:
   1124             if engine == 'python':

D:\Anaconda3\lib\site-packages\pandas\io\parsers.py in __init__(self, src, **kw
ds)
   1851         kwds['usecols'] = self.usecols
   1852
-> 1853         self._reader = parsers.TextReader(src, **kwds)
   1854         self.unnamed_cols = self._reader.unnamed_cols
```

```
   1855

pandas/_libs/parsers.pyx in pandas._libs.parsers.TextReader.__cinit__()

pandas/_libs/parsers.pyx in pandas._libs.parsers.TextReader._setup_parser_sourc
e()

FileNotFoundError: [Errno 2] File b'datasets/titanic/train.csv' does not exist:
b'datasets/titanic/train.csv'
```

In [6]:
```
import pandas as pd
train = pd.read_csv('C:/Users/shariar_nir/Documents/datasets/titanic/train.csv')
test = pd.read_csv('C:/Users/shariar_nir/Documents/datasets/titanic/test.csv')
```

In [7]:
```
train.head(5)
```

Out[7]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabi
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	Na
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C8
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	Na
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C12
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	Na

In [9]:
```
train.shape
```

Out[9]: (891, 12)

In [10]:
```
test.shape
```

Out[10]: (418, 11)

In [11]:
```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
PassengerId    891 non-null int64
Survived       891 non-null int64
Pclass         891 non-null int64
Name           891 non-null object
Sex            891 non-null object
Age            714 non-null float64
SibSp          891 non-null int64
Parch          891 non-null int64
Ticket         891 non-null object
Fare           891 non-null float64
Cabin          204 non-null object
Embarked       889 non-null object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.6+ KB
```

In [12]:
```
test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 11 columns):
PassengerId    418 non-null int64
Pclass         418 non-null int64
Name           418 non-null object
Sex            418 non-null object
Age            332 non-null float64
SibSp          418 non-null int64
Parch          418 non-null int64
Ticket         418 non-null object
Fare           417 non-null float64
Cabin          91 non-null object
Embarked       418 non-null object
dtypes: float64(2), int64(4), object(5)
memory usage: 36.0+ KB
```

In [13]:
```
train.isnull().sum()
```

Out[13]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [14]:
```
test.isnull().sum()
```

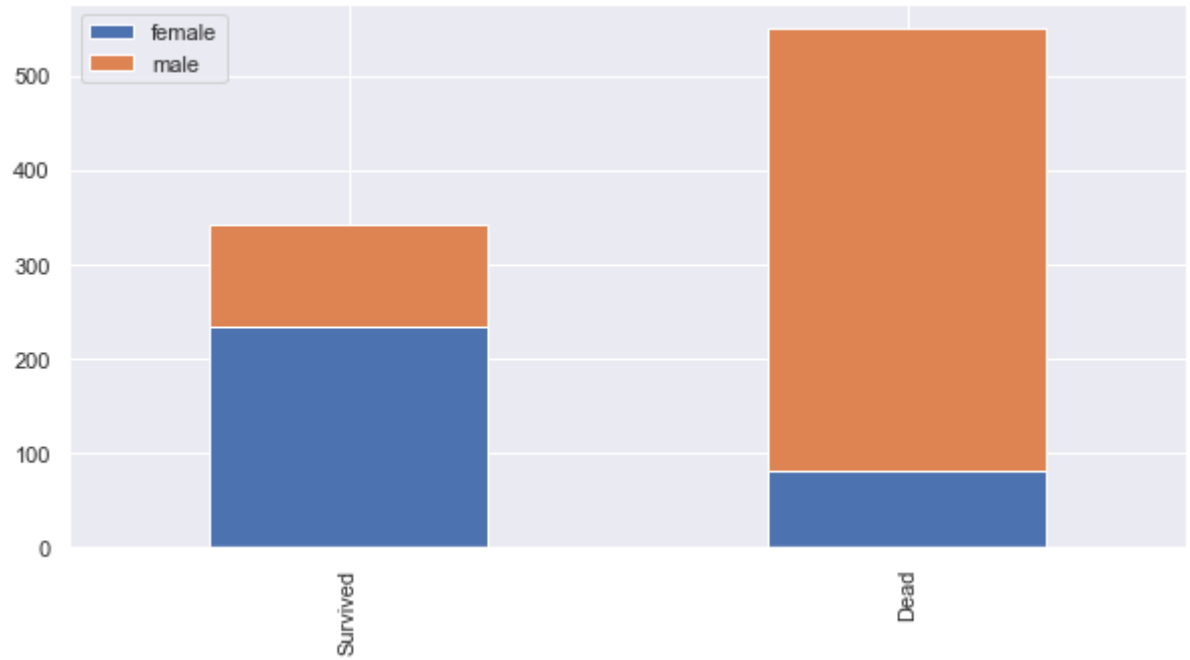
Out[14]:
```
PassengerId      0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

In [15]:
```
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
sns.set()
```

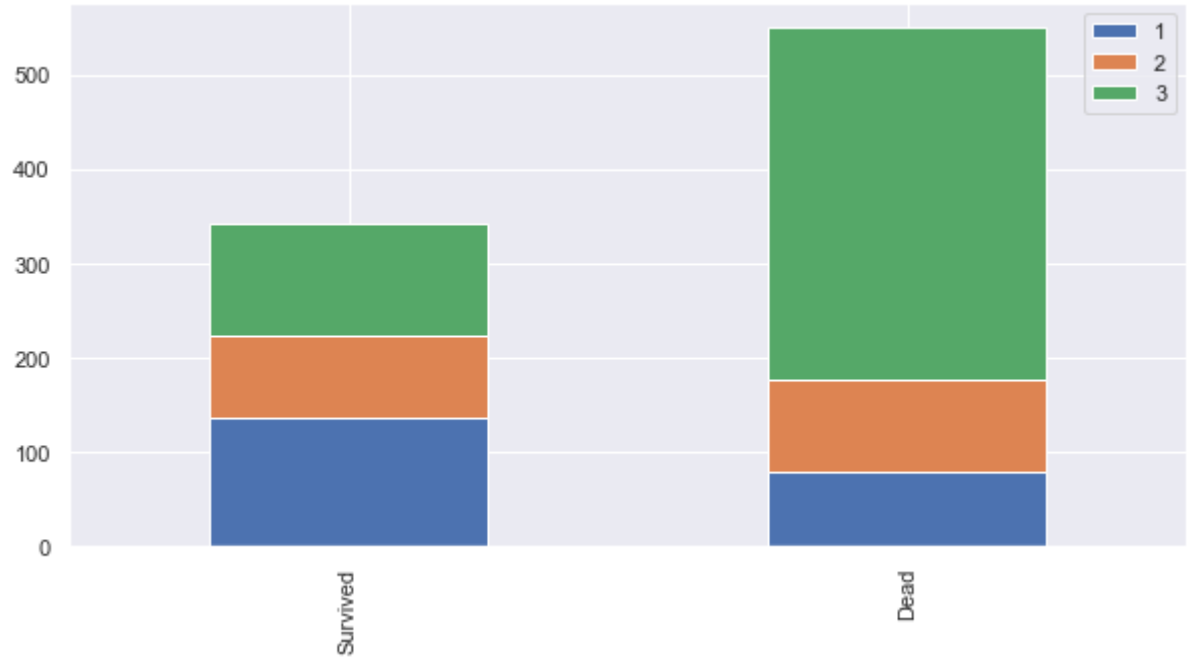
In [22]:
```
def bar_chart(feature):
    survived = train[train['Survived'] == 1][feature].value_counts()
    dead = train[train['Survived'] == 0][feature].value_counts()

    df = pd.DataFrame([survived,dead])
    df.index = ['Survived','Dead']
    df.plot(kind='bar',stacked = True, figsize = (10,5))
```

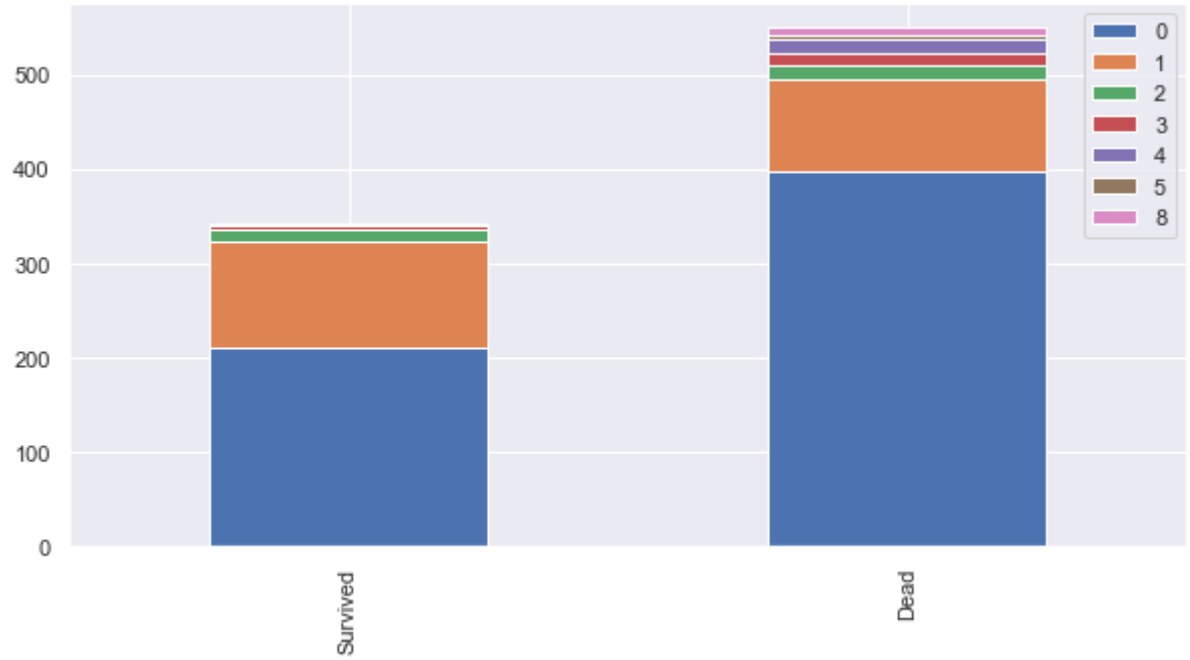
In [23]:
```
bar_chart('Sex')
```



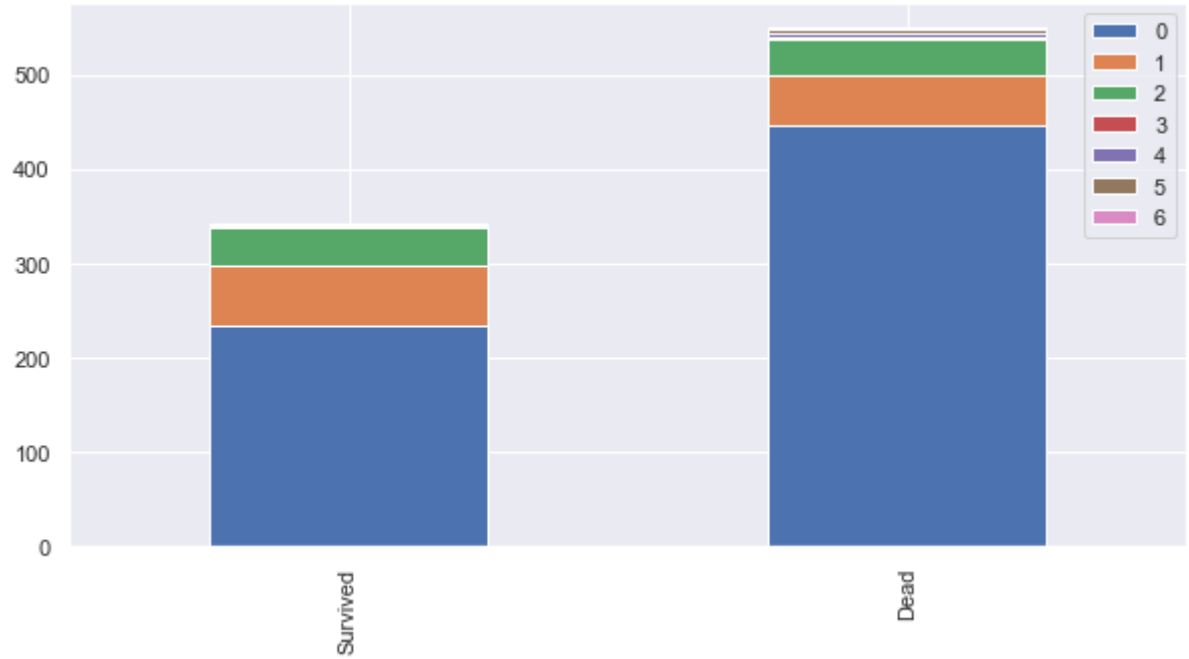
In [24]:
```
bar_chart('Pclass')
```



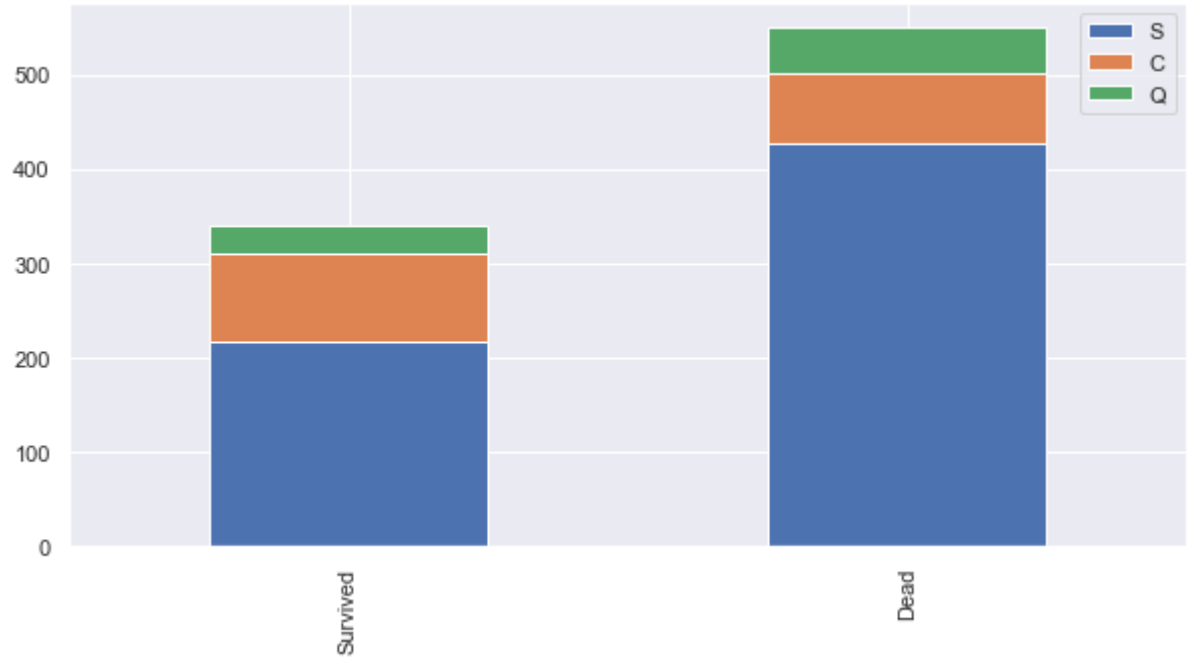
In [26]:
```
bar_chart('SibSp')
```



In [27]:
```
bar_chart('Parch')
```



In [28]:
```
bar_chart('Embarked')
```



In [31]:
```
train_test_data = [train, test]
for dataset in train_test_data:
    dataset['Title'] = dataset['Name'].str.extract('([A-Za-z]+)\.', expand = Fals
```

In [32]:
```
train['Title'].value_counts()
```

Out[32]:
```
Mr          517
Miss        182
Mrs         125
Master       40
Dr            7
Rev           6
Col           2
Mlle          2
Major         2
Ms            1
Mme           1
Countess      1
Lady          1
Capt          1
Jonkheer      1
Don           1
Sir           1
Name: Title, dtype: int64
```

In [33]:
```
test['Title'].value_counts()
```

Out[33]:
```
Mr        240
Miss       78
Mrs        72
Master     21
Rev         2
Col         2
Dr          1
Ms          1
Dona        1
Name: Title, dtype: int64
```

In [40]:
```
title_mapping = {"Mr": 0,"Miss": 1,"Mrs": 2,
                 "Master": 3,"Dr": 3,"Rev": 3,
                 "Col": 3,"Major": 3,"Mlle": 3,
                 "Countess": 3,"Ms": 3,"Lady": 3,
                 "Jonkheer": 3,"Don": 3,"Dona": 3,
                 "Mme": 3,"Capt": 3,"Sir": 3}
for dataset in train_test_data:
    dataset['Title']=dataset['Title'].map(title_mapping)
```

In [41]:
```
train.head()
```

Out[41]:

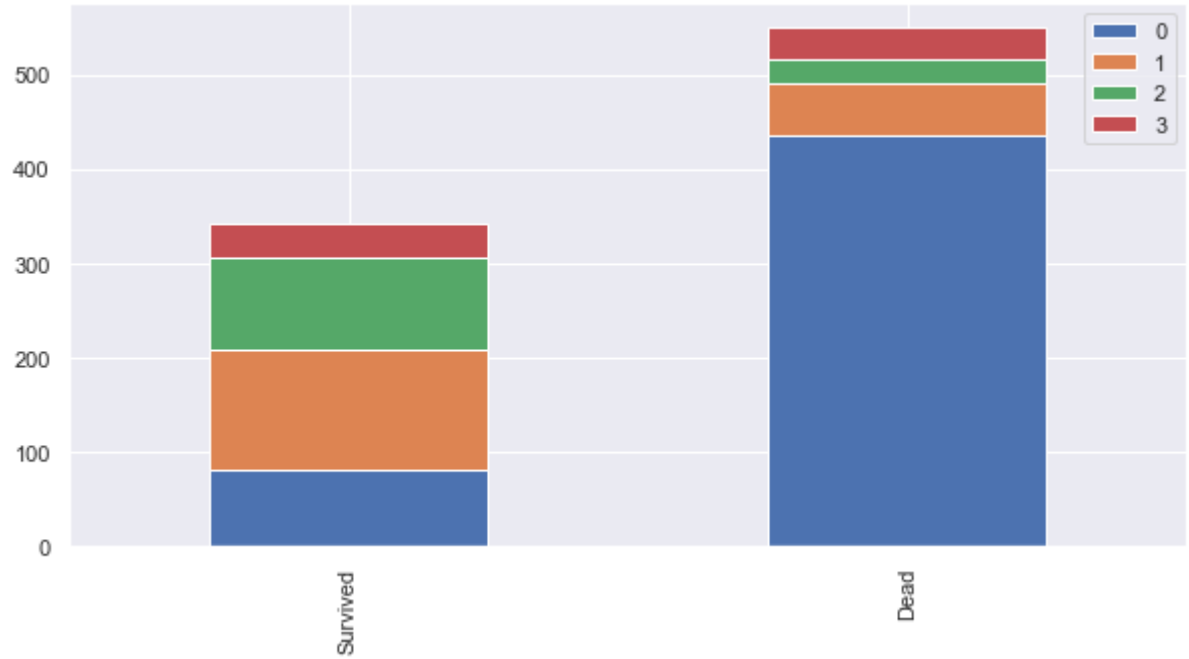
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabi
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	Na
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C8
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	Na
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C12
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	Na

In [42]:
```
test.head()
```

Out[42]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarke
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 [43]:
```
bar_chart('Title')
```



In [44]:
```
#delete unnecessary feature from dataset
train.drop('Name',axis = 1, inplace = True)
test.drop('Name',axis = 1, inplace = True)
```

In [45]:
```
train.head()
```

Out[45]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarke
0	1	0	3	male	22.0	1	0	A/5 21171	7.2500	NaN	
1	2	1	1	female	38.0	1	0	PC 17599	71.2833	C85	
2	3	1	3	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
3	4	1	1	female	35.0	1	0	113803	53.1000	C123	
4	5	0	3	male	35.0	0	0	373450	8.0500	NaN	

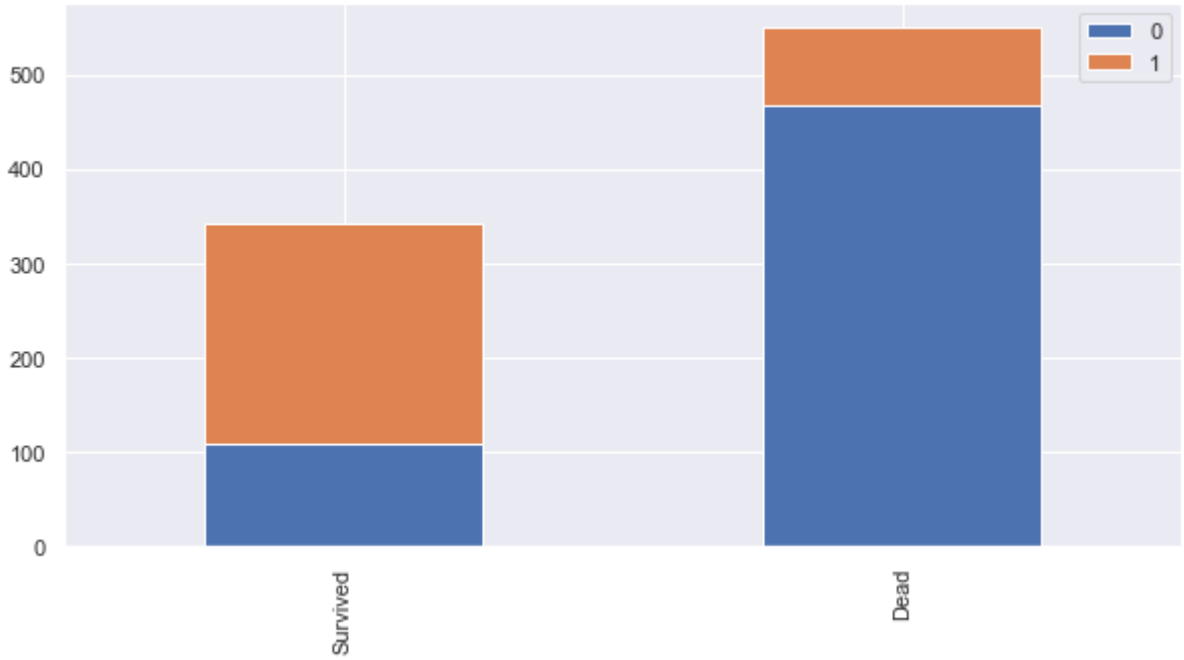
In [46]:
```
test.head()
```

Out[46]:

	PassengerId	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Title
0	892	3	male	34.5	0	0	330911	7.8292	NaN	Q	0
1	893	3	female	47.0	1	0	363272	7.0000	NaN	S	2
2	894	2	male	62.0	0	0	240276	9.6875	NaN	Q	0
3	895	3	male	27.0	0	0	315154	8.6625	NaN	S	0
4	896	3	female	22.0	1	1	3101298	12.2875	NaN	S	2

In [47]:
```
sex_mapping = {"male": 0, "female": 1}
for dataset in train_test_data:
    dataset['Sex'] = dataset['Sex'].map(sex_mapping)
```

In [48]:
```
bar_chart('Sex')
```



In [49]:
```
train.head()
```

Out[49]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	0	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	1	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	1	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	1	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	0	35.0	0	0	373450	8.0500	NaN	S

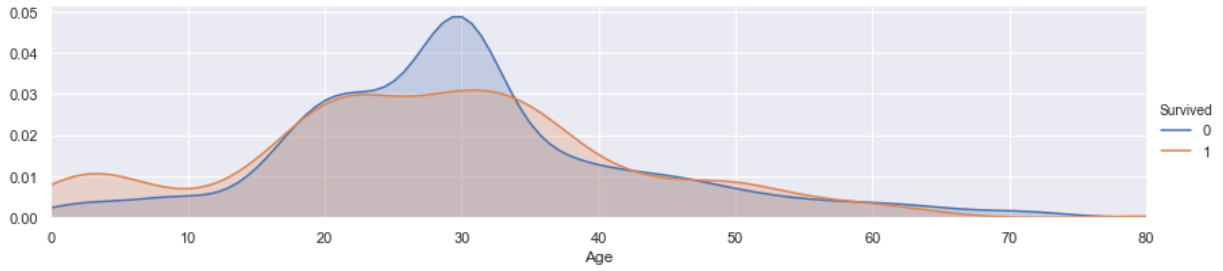
In [54]:
```
#fill missing age with median age for each title(Mr,Mrs,Miss,Others)
train["Age"].fillna(train.groupby("Title")["Age"].transform("median"), inplace =
test["Age"].fillna(test.groupby("Title")["Age"].transform("median"), inplace =Tru
```

In [55]:
```
train.groupby("Title")["Age"].transform("median")
train.head()
```

Out[55]:

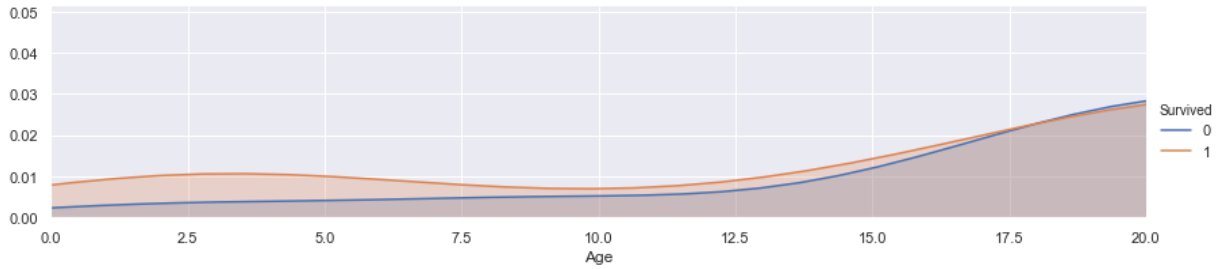
	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	0	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	1	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	1	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	1	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	0	35.0	0	0	373450	8.0500	NaN	S

In [57]:
```
facet = sns.FacetGrid(train, hue = "Survived", aspect = 4)
facet.map(sns.kdeplot,'Age',shade = True)
facet.set(xlim = (0, train['Age'].max()))
facet.add_legend()
plt.show()
```



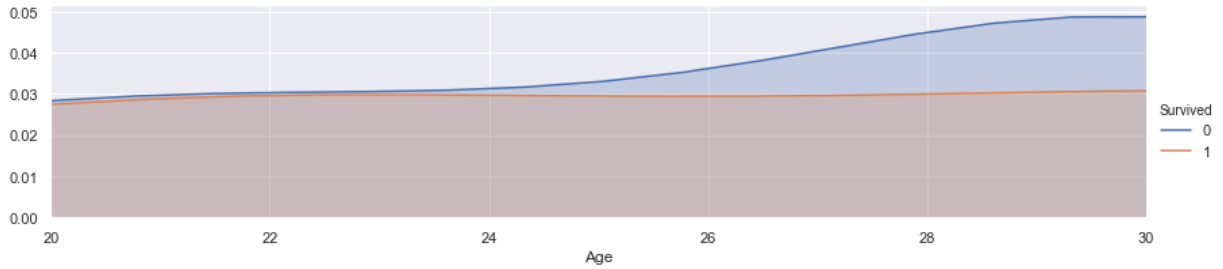
In [60]:
```
facet = sns.FacetGrid(train, hue = "Survived", aspect = 4)
facet.map(sns.kdeplot,'Age',shade = True)
facet.set(xlim = (0, train['Age'].max()))
facet.add_legend()
plt.xlim(0,20)
```

Out[60]: (0, 20)



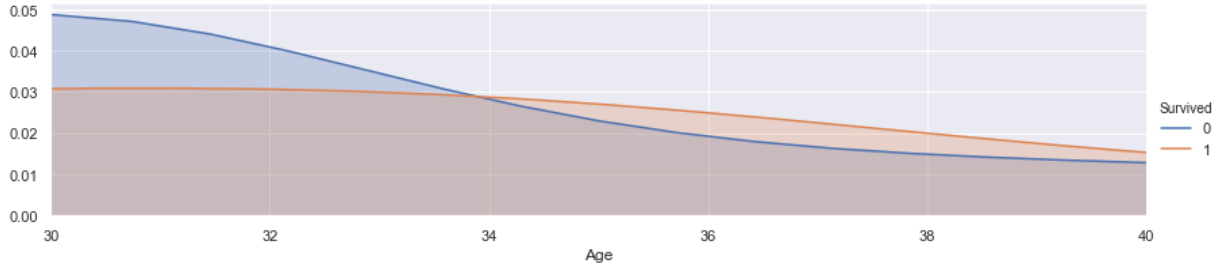
In [61]:
```
facet = sns.FacetGrid(train, hue = "Survived", aspect = 4)
facet.map(sns.kdeplot,'Age',shade = True)
facet.set(xlim = (0, train['Age'].max()))
facet.add_legend()
plt.xlim(20,30)
```

Out[61]: (20, 30)



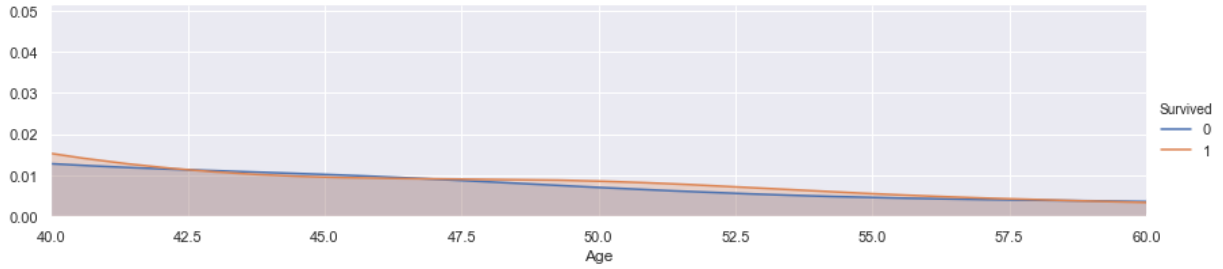
In [62]:
```
facet = sns.FacetGrid(train, hue = "Survived", aspect = 4)
facet.map(sns.kdeplot,'Age',shade = True)
facet.set(xlim = (0, train['Age'].max()))
facet.add_legend()
plt.xlim(30,40)
```

Out[62]: (30, 40)



In [63]:
```
facet = sns.FacetGrid(train, hue = "Survived", aspect = 4)
facet.map(sns.kdeplot,'Age',shade = True)
facet.set(xlim = (0, train['Age'].max()))
facet.add_legend()
plt.xlim(40,60)
```

Out[63]: (40, 60)



In [64]:
```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
PassengerId    891 non-null int64
Survived       891 non-null int64
Pclass         891 non-null int64
Sex            891 non-null int64
Age            891 non-null float64
SibSp          891 non-null int64
Parch          891 non-null int64
Ticket         891 non-null object
Fare           891 non-null float64
Cabin          204 non-null object
Embarked       889 non-null object
Title          891 non-null int64
dtypes: float64(2), int64(7), object(3)
memory usage: 83.6+ KB
```

In [65]:
```
test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 11 columns):
PassengerId    418 non-null int64
Pclass         418 non-null int64
Sex            418 non-null int64
Age            418 non-null float64
SibSp          418 non-null int64
Parch          418 non-null int64
Ticket         418 non-null object
Fare           417 non-null float64
Cabin          91 non-null object
Embarked       418 non-null object
Title          418 non-null int64
dtypes: float64(2), int64(6), object(3)
memory usage: 36.0+ KB
```

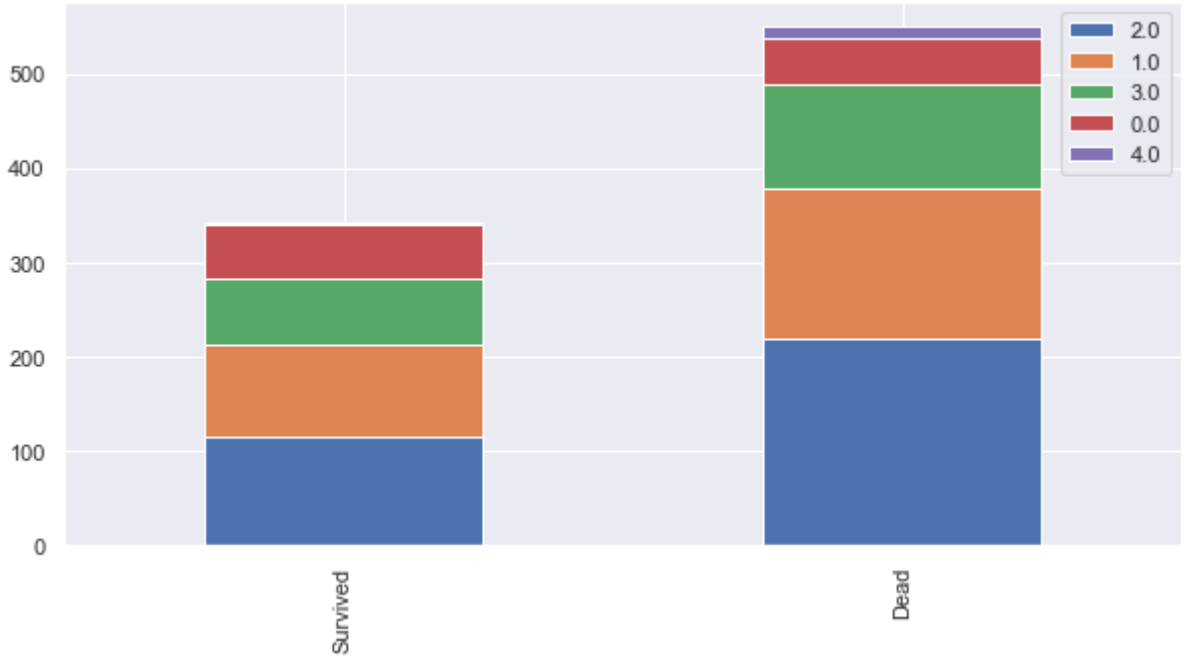
In [97]:
```
for dataset in train_test_data:
    dataset.loc[dataset['Age'] <= 16,'Age'] = 0,
    dataset.loc[(dataset['Age'] > 16) & (dataset['Age'] <= 26), 'Age'] = 1,
    dataset.loc[(dataset['Age'] > 26) & (dataset['Age'] <= 36), 'Age'] = 2,
    dataset.loc[(dataset['Age'] > 36) & (dataset['Age'] <= 62), 'Age'] = 3,
    dataset.loc[dataset['Age'] > 62,'Age'] = 4
```

In [99]:
```
train.head()
```

Out[99]:

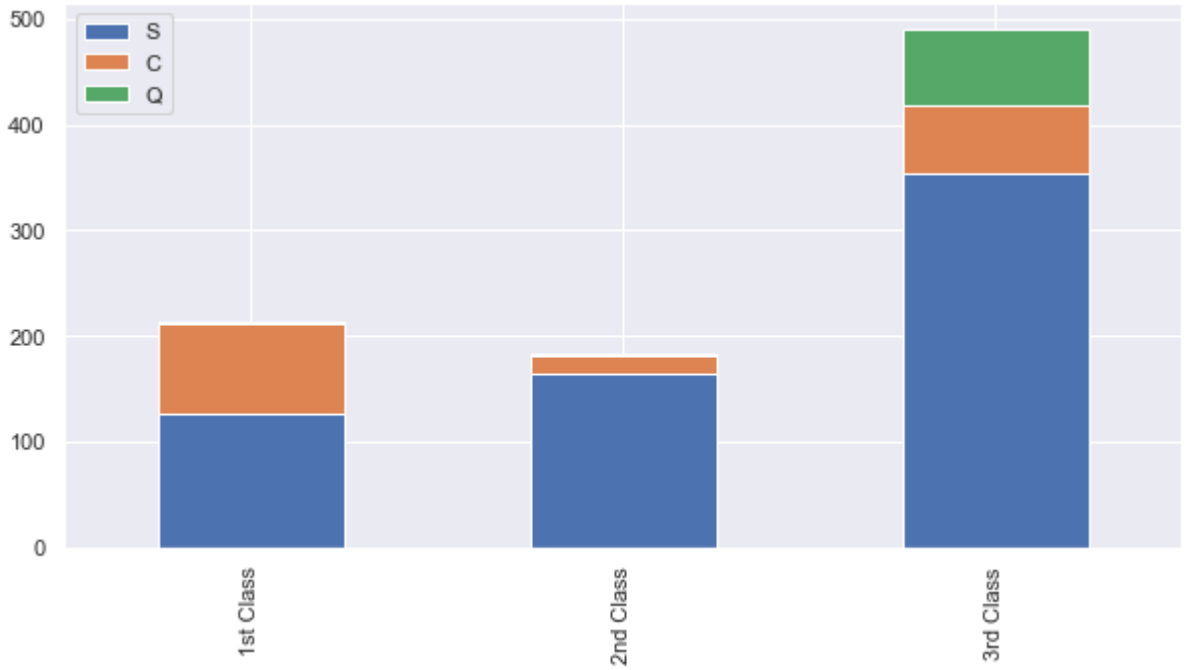
	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	0	1.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	1	3.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	1	1.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	1	2.0	1	0	113803	53.1000	C123	S
4	5	0	3	0	2.0	0	0	373450	8.0500	NaN	S

In [100]:
```
bar_chart('Age')
```



In [102]:
```
Pclass1 = train[ train[ 'Pclass'] == 1]['Embarked'].value_counts()
Pclass2 = train[ train[ 'Pclass'] == 2]['Embarked'].value_counts()
Pclass3 = train[ train[ 'Pclass'] == 3]['Embarked'].value_counts()
df = pd.DataFrame([Pclass1,Pclass2,Pclass3])
df.index = [ '1st Class','2nd Class','3rd Class']
df.plot(kind = 'bar',stacked = True, figsize = (10,5))
```

Out[102]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x14c89d48cc0>
```



In [103]:
```
for dataset in train_test_data:
    dataset['Embarked'] = dataset['Embarked'].fillna('S')
```

In [104]:
```
train.head()
```

Out[104]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	0	1.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	1	3.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	1	1.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	1	2.0	1	0	113803	53.1000	C123	S
4	5	0	3	0	2.0	0	0	373450	8.0500	NaN	S

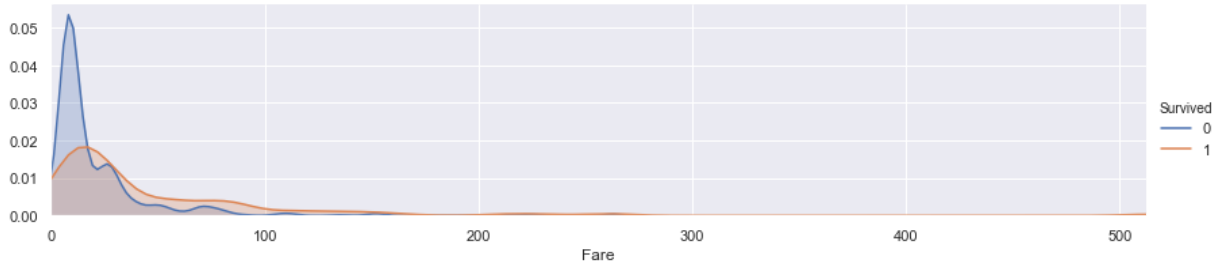
In [105]:
```
embarked_mapping = {"S":0, "C": 1,"Q": 2}
for dataset in train_test_data:
    dataset['Embarked'] = dataset['Embarked'].map(embarked_mapping)
```

In [106]:
```
#fill missing fare with median fare for each Pclass
train["Fare"].fillna(train.groupby("Pclass")["Fare"].transform("median"), inplace
test["Fare"].fillna(test.groupby("Pclass")["Fare"].transform("median"), inplace =
train.head(5)
```

Out[106]:

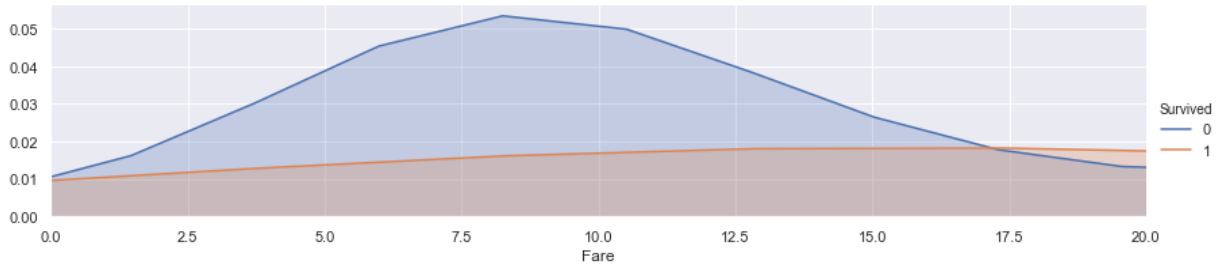
	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	0	1.0	1	0	A/5 21171	7.2500	NaN	0
1	2	1	1	1	3.0	1	0	PC 17599	71.2833	C85	1
2	3	1	3	1	1.0	0	0	STON/O2. 3101282	7.9250	NaN	0
3	4	1	1	1	2.0	1	0	113803	53.1000	C123	0
4	5	0	3	0	2.0	0	0	373450	8.0500	NaN	0

In [107]:
```
facet = sns.FacetGrid(train, hue = "Survived", aspect = 4)
facet.map(sns.kdeplot,'Fare',shade = True)
facet.set(xlim = (0, train['Fare'].max()))
facet.add_legend()
plt.show()
```



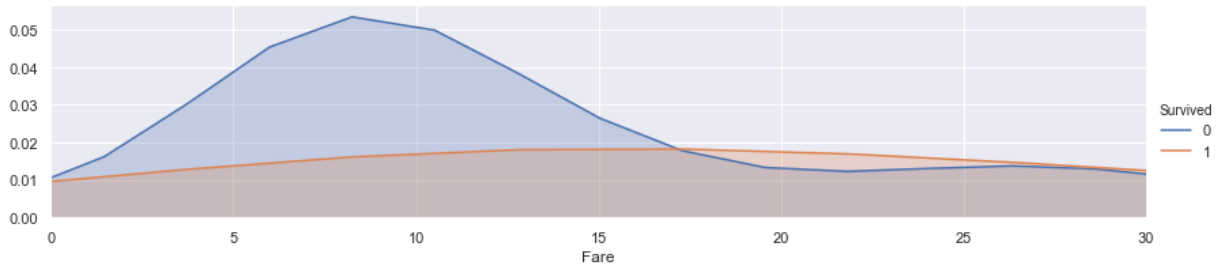
In [108]:
```
facet = sns.FacetGrid(train, hue = "Survived", aspect = 4)
facet.map(sns.kdeplot,'Fare',shade = True)
facet.set(xlim = (0, train['Fare'].max()))
facet.add_legend()
plt.xlim(0,20)
```

Out[108]: (0, 20)



In [109]:
```
facet = sns.FacetGrid(train, hue = "Survived", aspect = 4)
facet.map(sns.kdeplot,'Fare',shade = True)
facet.set(xlim = (0, train['Fare'].max()))
facet.add_legend()
plt.xlim(0,30)
```

Out[109]: (0, 30)



In [110]:
```
for dataset in train_test_data:
    dataset.loc[dataset['Fare'] <= 17,'Fare'] = 0,
    dataset.loc[(dataset['Fare'] > 17) & (dataset['Fare'] <= 30), 'Fare'] = 1,
    dataset.loc[(dataset['Fare'] > 30) & (dataset['Fare'] <= 100), 'Fare'] = 2,
    dataset.loc[dataset['Fare'] > 100,'Fare'] = 3
```

In [111]:
```
train.head()
```

Out[111]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Ti
0	1	0	3	0	1.0	1	0	A/5 21171	0.0	NaN	0	
1	2	1	1	1	3.0	1	0	PC 17599	2.0	C85	1	
2	3	1	3	1	1.0	0	0	STON/O2. 3101282	0.0	NaN	0	
3	4	1	1	1	2.0	1	0	113803	2.0	C123	0	
4	5	0	3	0	2.0	0	0	373450	0.0	NaN	0	

In [112]:
```
train.Cabin.value_counts()
```

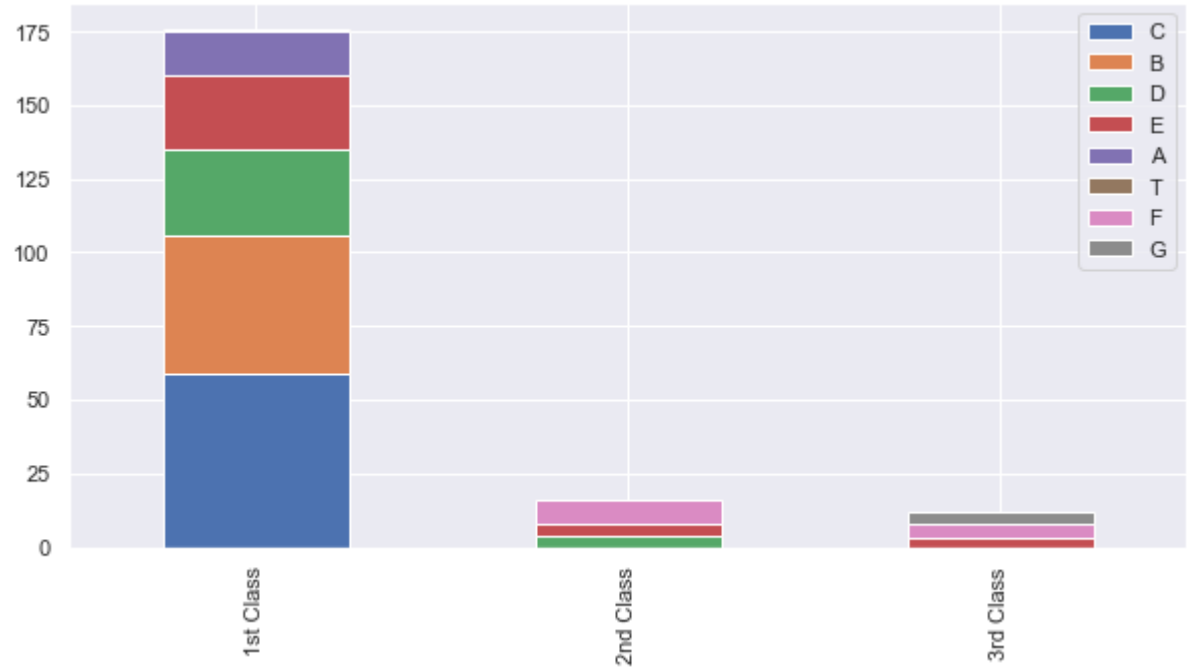
Out[112]:
```
B96 B98        4
G6             4
C23 C25 C27    4
C22 C26        3
F33            3
D              3
E101           3
F2             3
B22            2
E8             2
B77            2
B51 B53 B55    2
C68            2
B28            2
D33            2
C125           2
B20            2
C123           2
D17            2
C124           2
C126           2
E44            2
E24            2
E121           2
C52            2
B18            2
B58 B60        2
E25            2
F G73          2
E33            2
              ..
C50            1
C82            1
A10            1
B37            1
B73            1
C101           1
E31            1
B102           1
C87            1
B19            1
A36            1
E68            1
F G63          1
E38            1
E10            1
A34            1
A20            1
D56            1
C70            1
D45            1
E34            1
B42            1
B38            1
```

```
E17            1
C47            1
B86            1
C32            1
A31            1
C106           1
E63            1
Name: Cabin, Length: 147, dtype: int64
```

In [113]:
```
for dataset in train_test_data:
    dataset['Cabin'] = dataset['Cabin'].str[:1]
```

In [114]:
```
Pclass1 = train[ train[ 'Pclass'] == 1]['Cabin'].value_counts()
Pclass2 = train[ train[ 'Pclass'] == 2]['Cabin'].value_counts()
Pclass3 = train[ train[ 'Pclass'] == 3]['Cabin'].value_counts()
df = pd.DataFrame([Pclass1,Pclass2,Pclass3])
df.index = [ '1st Class','2nd Class','3rd Class']
df.plot(kind = 'bar',stacked = True, figsize = (10,5))
```

Out[114]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x14c898c8ef0>
```



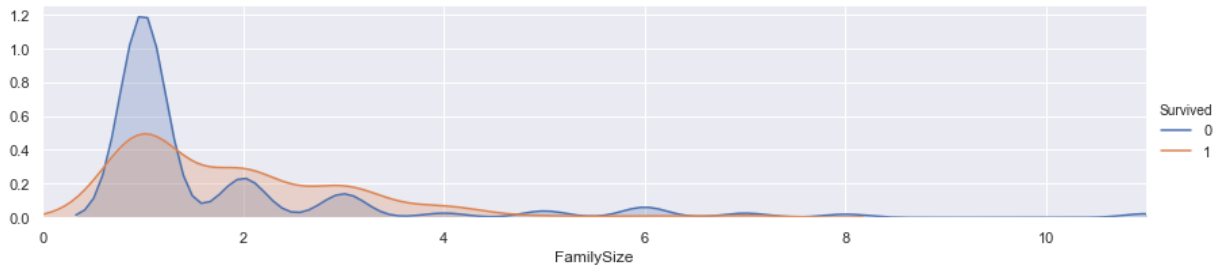
In [115]:
```
cabin_mapping = {"A":0, "B": 0.4,"C": 0.8,"D": 1.2, "E": 1.6,"F": 2,"G": 2.4, "T
for dataset in train_test_data:
    dataset['Cabin'] = dataset['Cabin'].map(cabin_mapping)
```

In [116]:
```
train["Cabin"].fillna(train.groupby("Pclass")["Cabin"].transform("median"), inpl
test["Cabin"].fillna(test.groupby("Pclass")["Cabin"].transform("median"), inplac
```

In [117]:
```
train["FamilySize"] = train["SibSp"] + train["Parch"] + 1
test["FamilySize"] = test["SibSp"] + test["Parch"] + 1
```

In [118]:
```
facet = sns.FacetGrid(train, hue = "Survived", aspect = 4)
facet.map(sns.kdeplot,'FamilySize',shade = True)
facet.set(xlim = (0, train['FamilySize'].max()))
facet.add_legend()
plt.xlim(0)
```

Out[118]: (0, 11.0)



In [119]:
```
family_mapping = {1:0, 2: 0.4,3: 0.8,4: 1.2, 5: 1.6,6: 2,7: 2.4, 8: 2.8,9: 3.2,1
for dataset in train_test_data:
    dataset['FamilySize'] = dataset['FamilySize'].map(family_mapping)
```

In [120]:
```
train.head()
```

Out[120]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Ti
0	1	0	3	0	1.0	1	0	A/5 21171	0.0	2.0	0	
1	2	1	1	1	3.0	1	0	PC 17599	2.0	0.8	1	
2	3	1	3	1	1.0	0	0	STON/O2. 3101282	0.0	2.0	0	
3	4	1	1	1	2.0	1	0	113803	2.0	0.8	0	
4	5	0	3	0	2.0	0	0	373450	0.0	2.0	0	

In [121]:
```
train.head()
```

Out[121]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Ti
0	1	0	3	0	1.0	1	0	A/5 21171	0.0	2.0	0	
1	2	1	1	1	3.0	1	0	PC 17599	2.0	0.8	1	
2	3	1	3	1	1.0	0	0	STON/O2. 3101282	0.0	2.0	0	
3	4	1	1	1	2.0	1	0	113803	2.0	0.8	0	
4	5	0	3	0	2.0	0	0	373450	0.0	2.0	0	

In [122]:
```
feature_drop = ['Ticket','SibSp','Parch']
train = train.drop(feature_drop,axis = 1)
test = test.drop(feature_drop,axis = 1)
train = train.drop(['PassengerId'],axis = 1)
```

In [123]:
```
train_data = train.drop('Survived', axis = 1)
target = train['Survived']
train_data.shape, target.shape
```

Out[123]: ((891, 8), (891,))

In [125]:
```
train_data.head()
```

Out[125]:

	Pclass	Sex	Age	Fare	Cabin	Embarked	Title	FamilySize
0	3	0	1.0	0.0	2.0	0	0	0.4
1	1	1	3.0	2.0	0.8	1	2	0.4
2	3	1	1.0	0.0	2.0	0	1	0.0
3	1	1	2.0	2.0	0.8	0	2	0.4
4	3	0	2.0	0.0	2.0	0	0	0.0

In [126]:
```
# Importing classifier Modules
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
import numpy as np
```

In [127]:
```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 9 columns):
Survived      891 non-null int64
Pclass        891 non-null int64
Sex           891 non-null int64
Age           891 non-null float64
Fare          891 non-null float64
Cabin         891 non-null float64
Embarked      891 non-null int64
Title         891 non-null int64
FamilySize    891 non-null float64
dtypes: float64(4), int64(5)
memory usage: 62.7 KB
```

In [128]:
```
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
k_fold = KFold(n_splits = 10,shuffle = True,random_state = 0)
```

In [130]:
```
clf = DecisionTreeClassifier()
scoring = 'accuracy'
score = cross_val_score(clf, train_data, target, cv=k_fold, n_jobs=1, scoring=sc
print(score)
```

```
[0.76666667 0.83146067 0.76404494 0.7752809  0.8988764  0.76404494
 0.83146067 0.82022472 0.74157303 0.78651685]
```

In [131]:
```
#decision tree score
round(np.mean(score)*100,2)
```

Out[131]: 79.8

In [132]:
```
clf = RandomForestClassifier(n_estimators = 13)
scoring = 'accuracy'
score = cross_val_score(clf, train_data, target, cv=k_fold, n_jobs=1, scoring=sc
print(score)
```

```
[0.82222222 0.83146067 0.78651685 0.76404494 0.91011236 0.79775281
 0.78651685 0.80898876 0.73033708 0.83146067]
```

In [133]:
```
#Random forest score
round(np.mean(score)*100,2)
```

Out[133]: 80.69

In [134]:
```
clf = RandomForestClassifier(n_estimators = 13)
clf.fit(train_data, target)
test_data = test.drop("PassengerId", axis = 1).copy()
prediction = clf.predict(test_data)
```

In [136]:
```
submission = pd.DataFrame({
    "PassengerId" : test["PassengerId"],"Survived" : prediction
})
submission.to_csv('submission.csv',index = False)
```

In [139]:
```
submission = pd.read_csv('submission.csv')
submission.head()
```

Out[139]:

	PassengerId	Survived
0	892	0
1	893	0
2	894	0
3	895	0
4	896	0

In [ ]:
```

```