In [1]:
```
print("Allah")
```

```
Allah
```

In [2]:
```
#load dataset using pandas
import pandas as p
train = p.read_csv("C:/Users/user/Desktop/kaggle/train.csv")
test = p.read_csv("C:/Users/user/Desktop/kaggle/test.csv")
```

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

Out[3]:

	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	Nal
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	Nal
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	Nal

In [4]:
```
test.head()
```

Out[4]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

In [5]:
```
train.shape
```

Out[5]: (891, 12)

In [6]:
```
test.shape
```

Out[6]: (418, 11)

In [7]:
```
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 [8]:
```
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 [9]:
```
train.isnull().sum()
```

Out[9]:
```
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
```

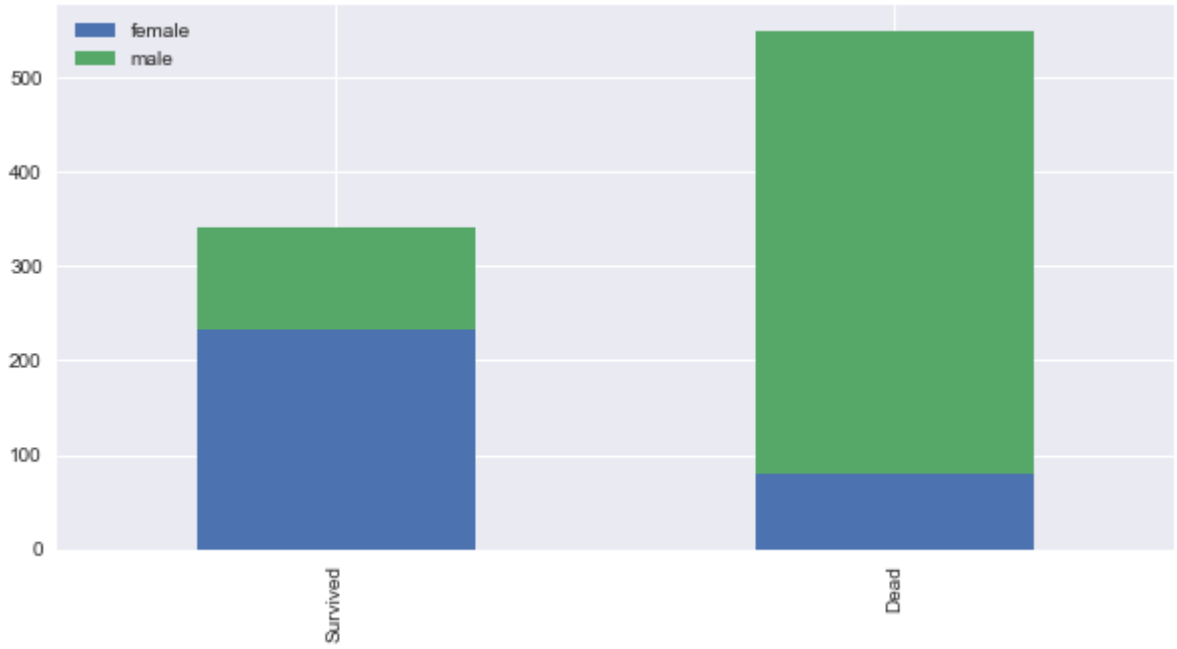
In [10]:
```
test.isnull().sum()
```

Out[10]:
```
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 [11]:
```
#import  python lib for visualization
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as  sns
sns.set() #setting default seaborn for plot
```

In [12]:
```
def bar_chart(feature):
    survived = train[train['Survived']==1][feature].value_counts()
    dead=train[train['Survived']==0][feature].value_counts()
    df=p.DataFrame([survived,dead])
    df.index=['Survived','Dead']
    df.plot(kind='bar',stacked=True,figsize=(10,5))
```

In [13]:
```
bar_chart('Sex')
```



In [14]:
```
train.head(0)
```

Out[14]:

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

In [15]:
```
train.head()
```

Out[15]:

	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	Nal
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	Nal
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	Nal

In [16]:
```
train_test_data=[train,test]
for dataset in train_test_data:
    dataset['Title'] =dataset['Name'].str.extract('([A-Za-z]+)\.',expand=False)
```

In [17]:
```
train['Title'].value_counts()
```

Out[17]:
```
Mr          517
Miss        182
Mrs         125
Master       40
Dr            7
Rev           6
Col           2
Mlle          2
Major         2
Mme           1
Don           1
Capt          1
Jonkheer      1
Ms            1
Sir           1
Countess      1
Lady          1
Name: Title, dtype: int64
```

In [18]:
```
test['Title'].value_counts()
train['Title'].value_counts()
```

Out[18]:
```
Mr          517
Miss        182
Mrs         125
Master       40
Dr            7
Rev           6
Col           2
Mlle          2
Major         2
Mme           1
Don           1
Capt          1
Jonkheer      1
Ms            1
Sir           1
Countess      1
Lady          1
Name: Title, dtype: int64
```

In [19]:
```
title_mapping={"Mr": 0,"Miss": 1,"Mrs": 2,"Master": 3,"Dr": 3, "Rev": 3,"Major":
               "Dona":3,"Mme":3,"Capt":3,"Sir": 3}
for dataset in train_test_data:
    dataset['Title'] =dataset['Title'].map(title_mapping)
```

In [20]:
```
train.head()
train['Title'].value_counts()
```

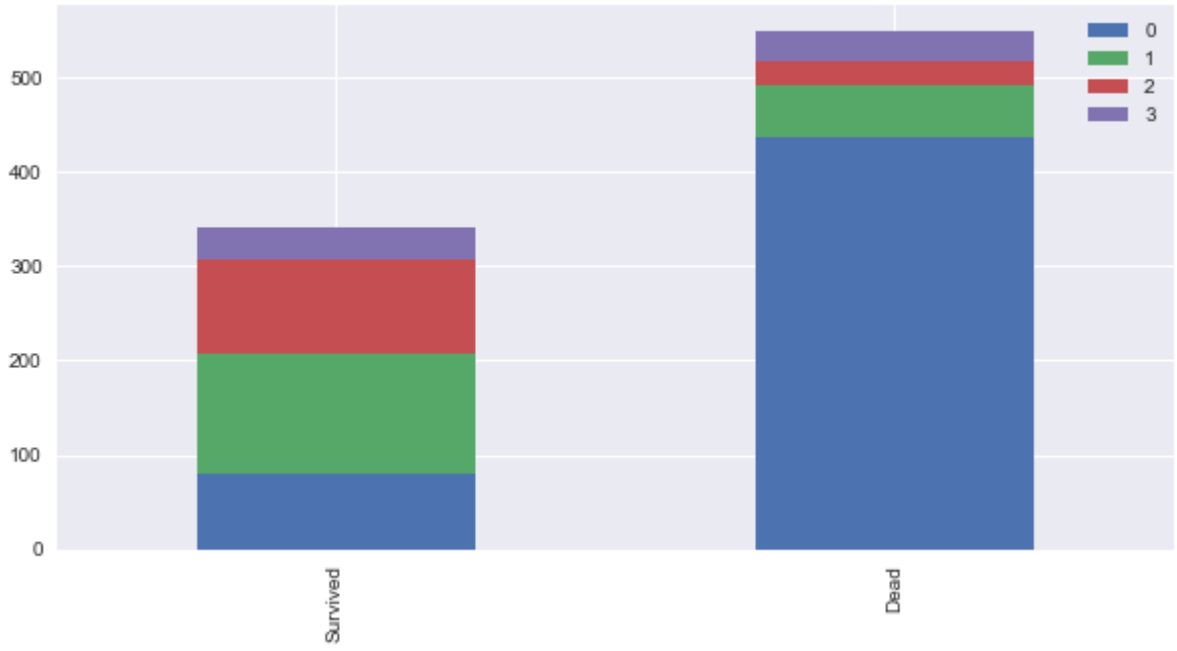
Out[20]:
```
0    517
1    182
2    125
3     67
Name: Title, dtype: int64
```

In [21]:
```
test.head()
```

Out[21]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

In [22]:
```
bar_chart('Title')
```



In [23]:
```
train.drop('Name',axis=1,inplace=True)
test.drop('Name',axis=1,inplace=True)
```

In [24]:
```
train.head()
```

Out[24]:

	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	

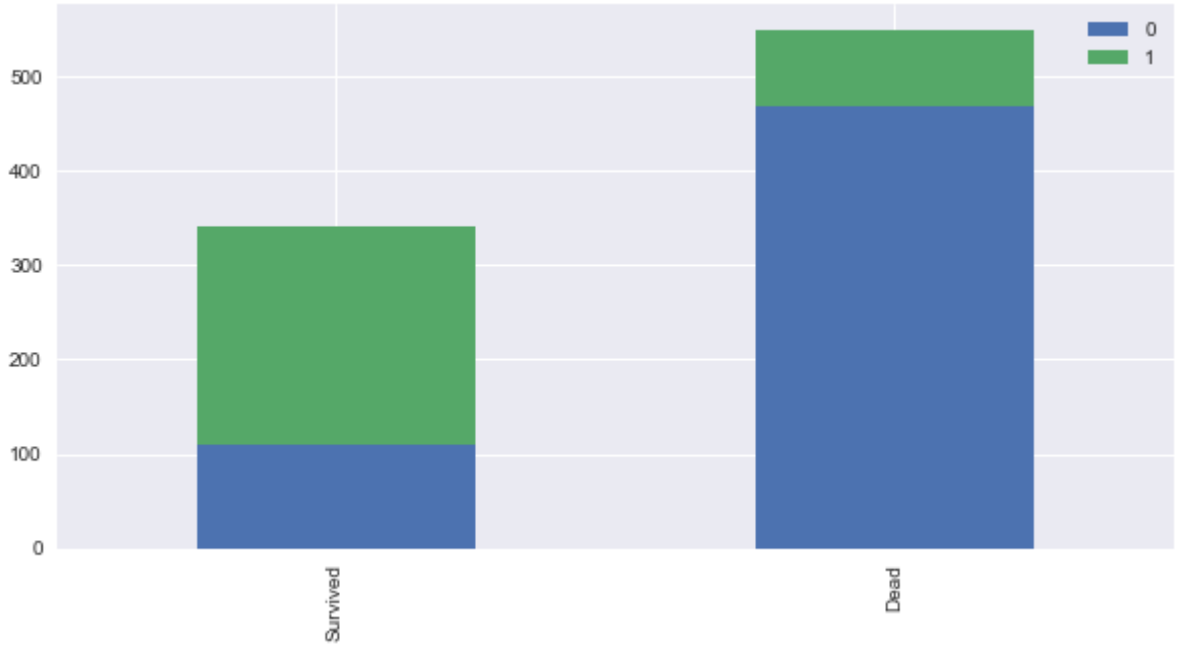
In [25]:
```
test.head()
```

Out[25]:

	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 [26]:
```
sex_mapping={"male": 0,"female": 1}
for dataset in train_test_data:
    dataset['Sex'] =dataset['Sex'].map(sex_mapping)
```

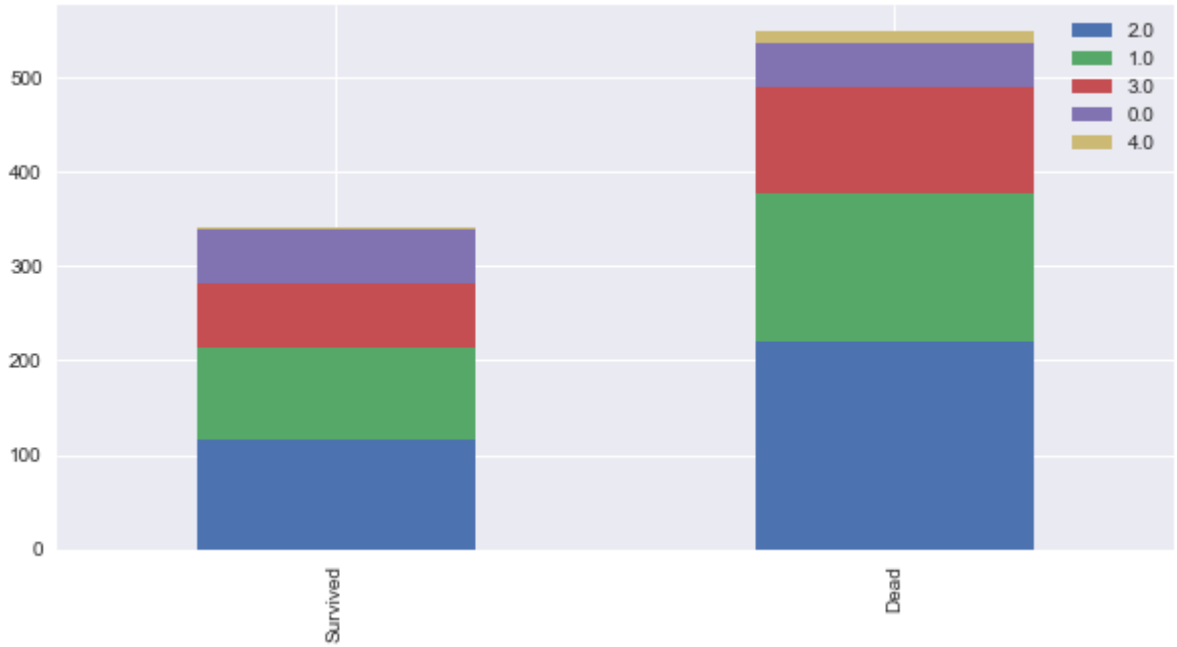
In [27]:
```
bar_chart('Sex')
```



In [28]:
```
train["Age"].fillna(train.groupby("Title")["Age"].transform("median"),inplace=Tru
test["Age"].fillna(test.groupby("Title")["Age"].transform("median"),inplace=True)
```

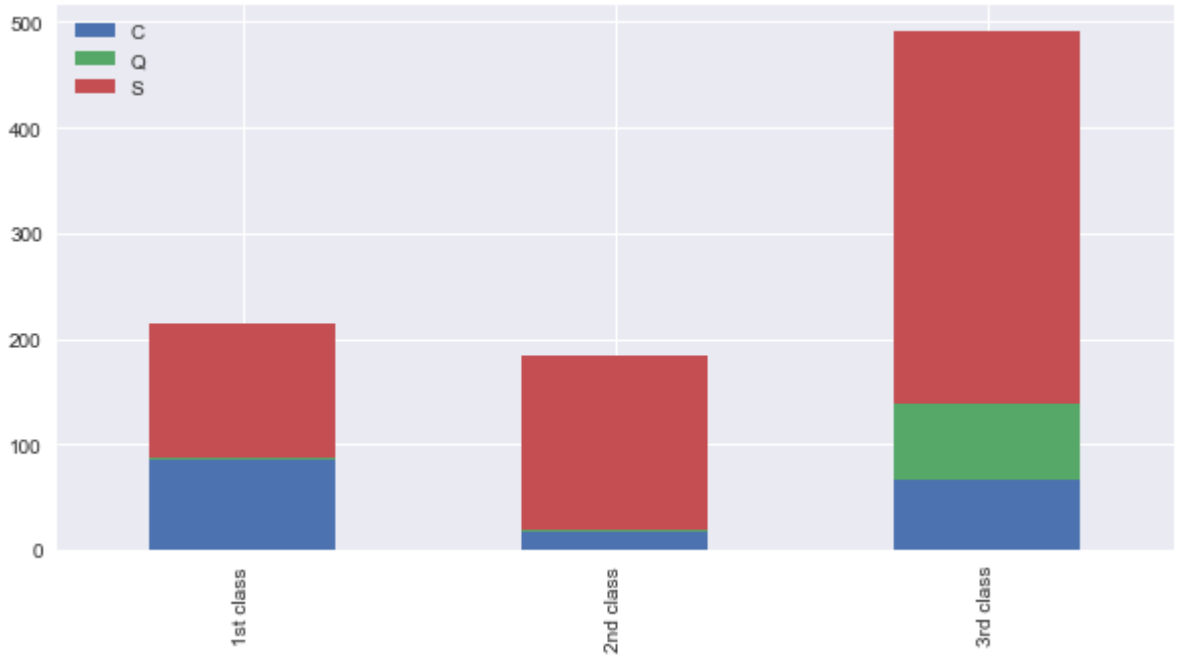
In [29]:
```
for dataset in train_test_data:
    dataset.loc[ dataset['Age']<=16,'Age']=0,
    dataset.loc[ (dataset['Age']>16)&(dataset['Age']<=26),'Age']=1,
    dataset.loc[ (dataset['Age']>26)&(dataset['Age']<=36),'Age']=2,
    dataset.loc[ (dataset['Age']>36)&(dataset['Age']<=62),'Age']=3,
    dataset.loc[ dataset['Age']>62,'Age']=4,
```

In [30]:
```
bar_chart('Age')
```



In [31]:
```
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=p.DataFrame([Pclass1,Pclass2,Pclass3])
df.index=['1st class','2nd class','3rd class']
df.plot(kind='bar',stacked=True,figsize=(10,5))
```

Out[31]: <matplotlib.axes._subplots.AxesSubplot at 0x1d87390acf8>



In [32]:
```
for dataset in train_test_data:
    dataset['Embarked'] =dataset['Embarked'].fillna('S')
```

In [33]:
```
train.head()
```

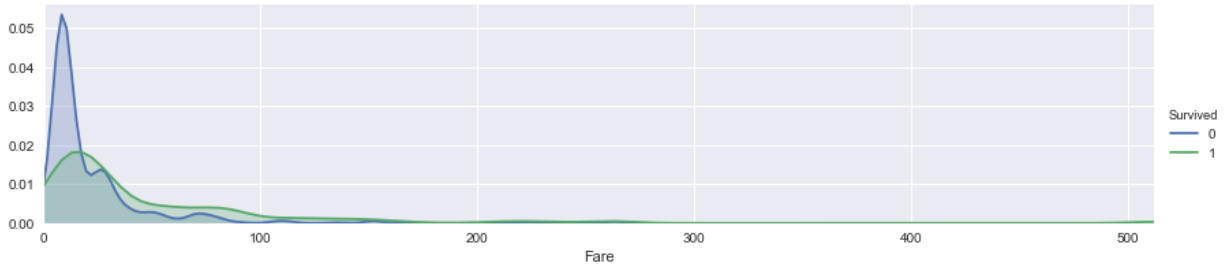
Out[33]:

	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 [34]:
```
embarked_mapping = {"S":0,"C":1,"Q":2}
for dataset in train_test_data:
    dataset['Embarked']=dataset['Embarked'].map(embarked_mapping)
```

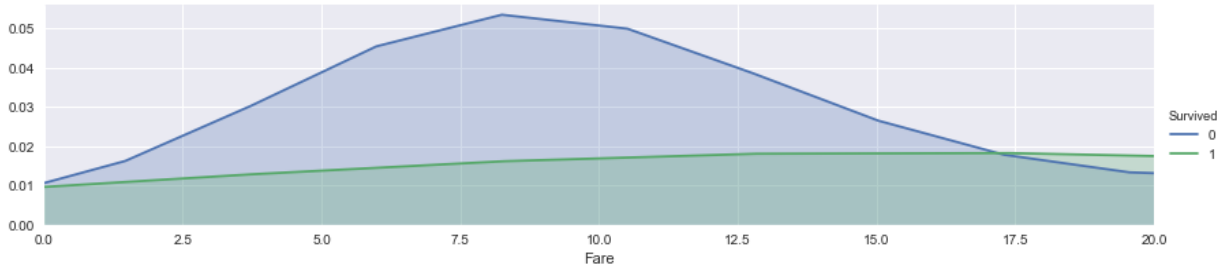
In [35]:
```
train["Fare"].fillna(train.groupby("Pclass")["Fare"].transform("median"),inplace=
test["Fare"].fillna(test.groupby("Pclass")["Fare"].transform("median"),inplace=Tr
```

In [36]:
```
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()
```



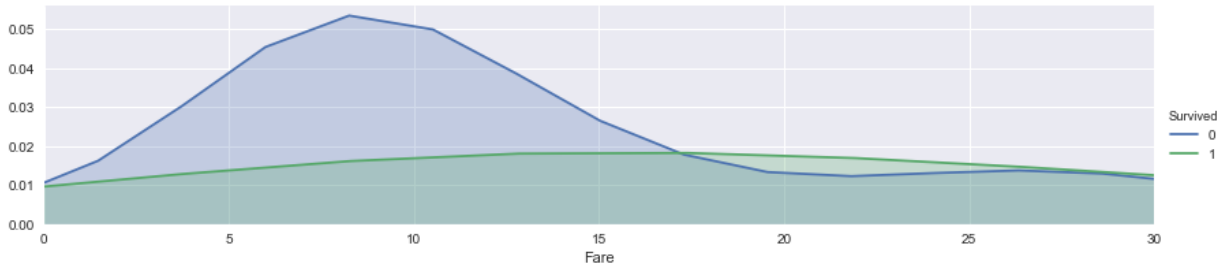
In [37]:
```
facet = sns.FacetGrid(train,hue="Survived",aspect=4)
facet.map(sns.kdeplot,'Fare',shade=True)
facet.set(xlim=(0,train['Fare'].max()))
facet.add_legend()
plt.xlim(0,20)
```

Out[37]: (0, 20)



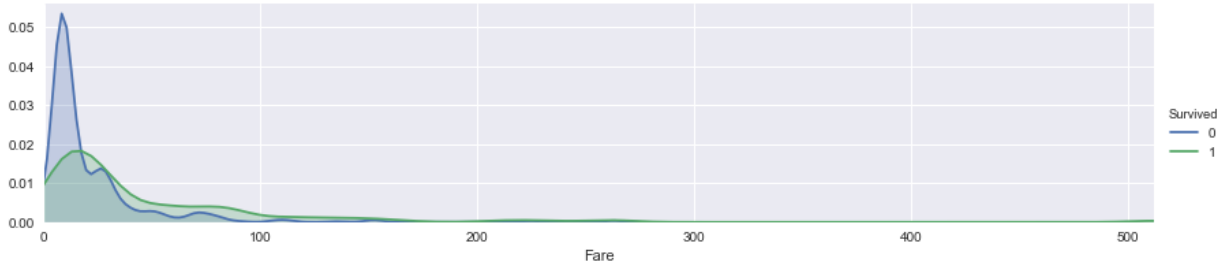
In [38]:
```
facet = sns.FacetGrid(train,hue="Survived",aspect=4)
facet.map(sns.kdeplot,'Fare',shade=True)
facet.set(xlim=(0,train['Fare'].max()))
facet.add_legend()
plt.xlim(0,30)
```

Out[38]: (0, 30)



In [39]:
```
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)
```

Out[39]: (0, 512.32920000000001)



In [40]:
```
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 [41]:
```
train.head()
```

Out[41]:

	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 [42]:
```
train.Cabin.value_counts()
```

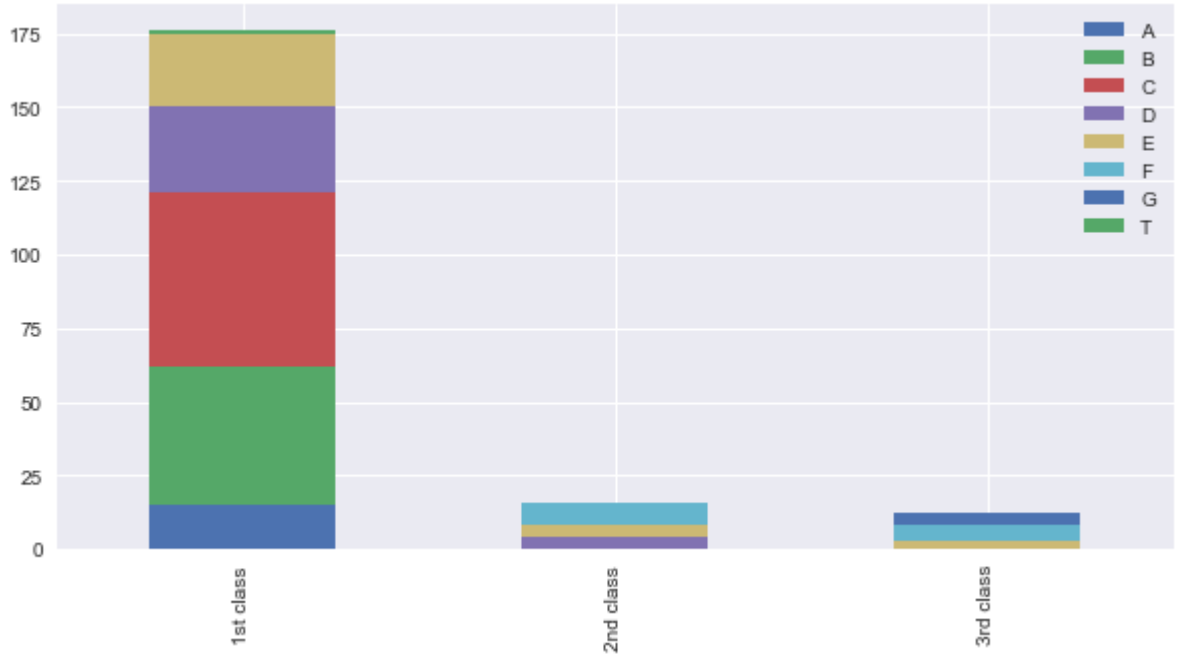
Out[42]:
```
C23 C25 C27        4
G6                 4
B96 B98            4
F2                 3
C22 C26            3
E101               3
F33                3
D                  3
D26                2
E24                2
E33                2
F4                 2
E121               2
B49                2
B58 B60            2
C52                2
B77                2
B51 B53 B55        2
B18                2
C68                2
B5                 2
C83                2
C2                 2
B57 B59 B63 B66    2
E67                2
C65                2
D33                2
F G73              2
E44                2
C124               2
                  ..
C128               1
E34                1
A31                1
A26                1
A36                1
D30                1
E38                1
B101               1
B73                1
C82                1
B38                1
C47                1
E17                1
B19                1
D37                1
A23                1
E31                1
T                  1
B78                1
C87                1
D50                1
C30                1
C86                1
E40                1
```

```
C106              1
E46               1
D56               1
E50               1
C101              1
B30               1
Name: Cabin, Length: 147, dtype: int64
```

In [43]:
```
for dataset in train_test_data:
    dataset['Cabin'] =dataset['Cabin'].str[:1]
```

In [44]:
```
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=p.DataFrame([Pclass1,Pclass2,Pclass3])
df.index=['1st class','2nd class','3rd class']
df.plot(kind='bar',stacked=True,figsize=(10,5))
```

Out[44]: <matplotlib.axes._subplots.AxesSubplot at 0x1d873bbd160>



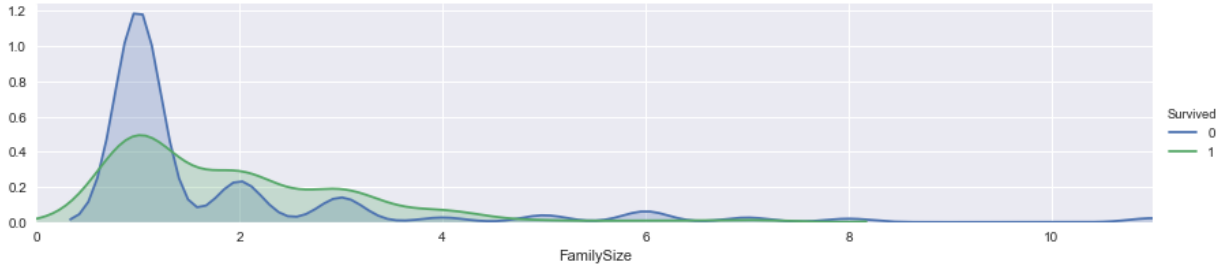
In [45]:
```
cabin_mapping = {"A":0,"B":0.4,"C":0.8,"D":1.2,"E":1.6,
                 "F":2,"G":2.4,"T":2.8}
for dataset in train_test_data:
    dataset['Cabin']=dataset['Cabin'].map(cabin_mapping)
```

In [46]:
```
train["Cabin"].fillna(train.groupby("Pclass")["Cabin"].transform("median"),inplac
test["Cabin"].fillna(test.groupby("Pclass")["Cabin"].transform("median"),inplace=
```

In [47]:
```
train["FamilySize"]= train["SibSp"]+train["Parch"]+1
test["FamilySize"]= test["SibSp"]+test["Parch"]+1
```

In [48]:
```
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[48]: (0, 11.0)



In [49]:
```
family_mapping = {1:0,2:0.4,3:0.8,4:1.2,5:1.6,6:2,7:2.4,8:2.8,
                  9:3.2,10:3.6,11:4}
for dataset in train_test_data:
    dataset['FamilySize']=dataset['FamilySize'].map(family_mapping)
```

In [50]:
```
features_drop=['Ticket','SibSp','Parch']
train=train.drop(features_drop,axis=1)
test=test.drop(features_drop,axis=1)
train=train.drop(['PassengerId'],axis=1)
```

In [51]:
```
train_data=train.drop('Survived',axis=1)
target= train['Survived']
```

In [52]:
```
train_data.shape,target.shape
```

Out[52]: ((891, 8), (891,))

In [53]:
```
train_data.head()
```

Out[53]:

	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 [54]:
```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
```

In [55]:
```
import numpy as np
```

In [56]:
```
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 [57]:
```
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 [58]:
```
clf=KNeighborsClassifier(n_neighbors=13)
```

In [59]:
```
scoring='accuracy'
```

In [60]:
```
score = cross_val_score(clf,train_data,target,cv=k_fold,n_jobs=1,scoring=scoring)
```

In [61]:
```
print(score)
```

```
[ 0.82222222  0.76404494  0.80898876  0.83146067  0.87640449  0.82022472
  0.85393258  0.79775281  0.84269663  0.84269663]
```

In [62]:
```
round(np.mean(score)*100,2)
```

Out[62]:
```
82.599999999999994
```

In [63]:
```
clf=KNeighborsClassifier()
```

In [64]:
```
clf.fit(train_data,target)
```

Out[64]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=5, p=2,
           weights='uniform')
```

In [65]:
```
test_data= test.drop("PassengerId",axis=1).copy()
```

In [66]:
```
prediction=clf.predict(test_data)
```

In [67]:
```
submission=p.DataFrame({"PassengerId":test["PassengerId"],"Survived":prediction})
```

In [68]:
```
submission.to_csv('C:/Users/user/Desktop/submission.csv',index=False)
```

In [69]:
```
submission=p.read_csv('C:/Users/user/Desktop/submission.csv')
submission.head()
```

Out[69]:

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

In [ ]:

In [ ]: