In [65]:
```
import pandas as pd
from pandas import Series, DataFrame
# Set up of the titanic data as a Data Frame
titanic_df = pd.read_csv('/Users/rodri/Downloads/train.csv')
# Data preview
titanic_df.head()
```

Out[65]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	E
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	

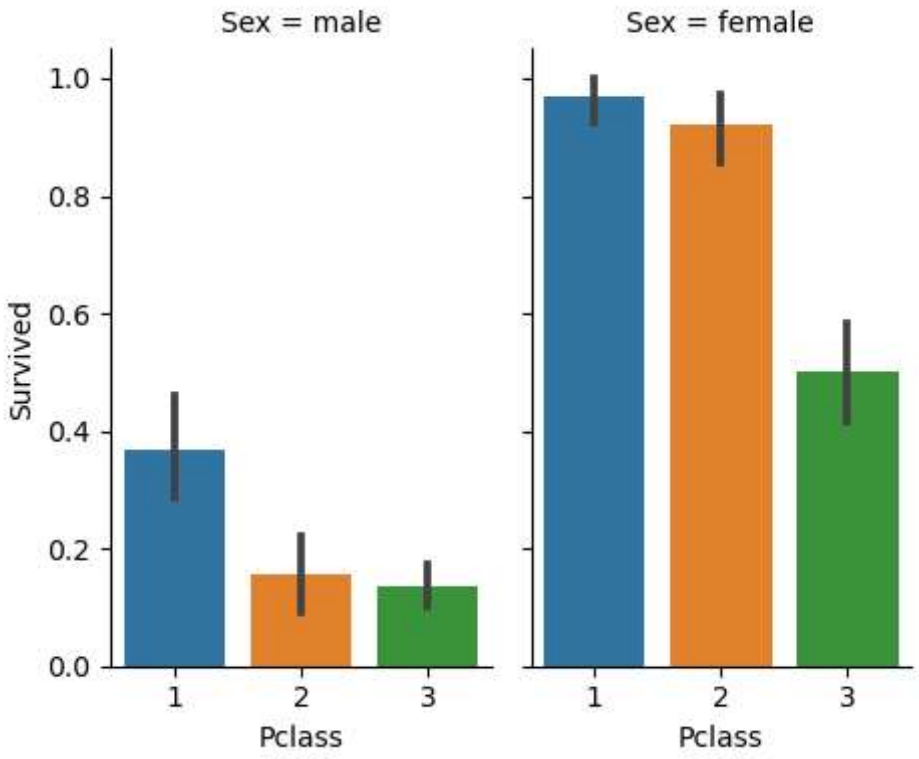
In [66]:
```
# Data structures overview
titanic_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [67]:
```
# Tools for Data Visualization and Analysis
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [68]:
```
# First Analysis using Factor Plot, which only needs the column argument, understandir
sns.catplot(
    data=titanic_df, x="Pclass", y="Survived", col="Sex",
    kind="bar", height=4, aspect=.6,
)
```

Out[68]:
```
<seaborn.axisgrid.FacetGrid at 0x1e2687f5f10>
```



In [69]:
```
# Utilizing apply and creating a function, time to see how many children survived

def kids(passanger):
    age,sex = passanger
    if age < 16:
        return 'child'
    else:
        return sex

titanic_df['Person'] = titanic_df[['Age','Sex']].apply(kids,axis = 1)
```

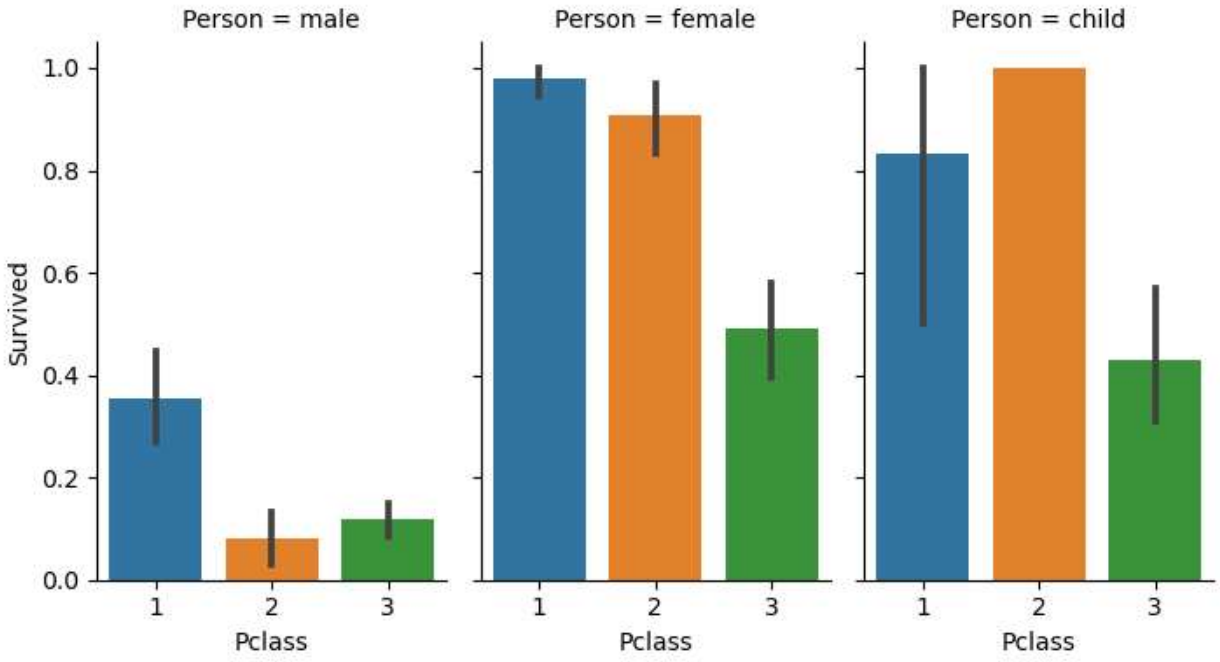
In [70]:
```
titanic_df[0:10]
```

Out[70]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	E
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	

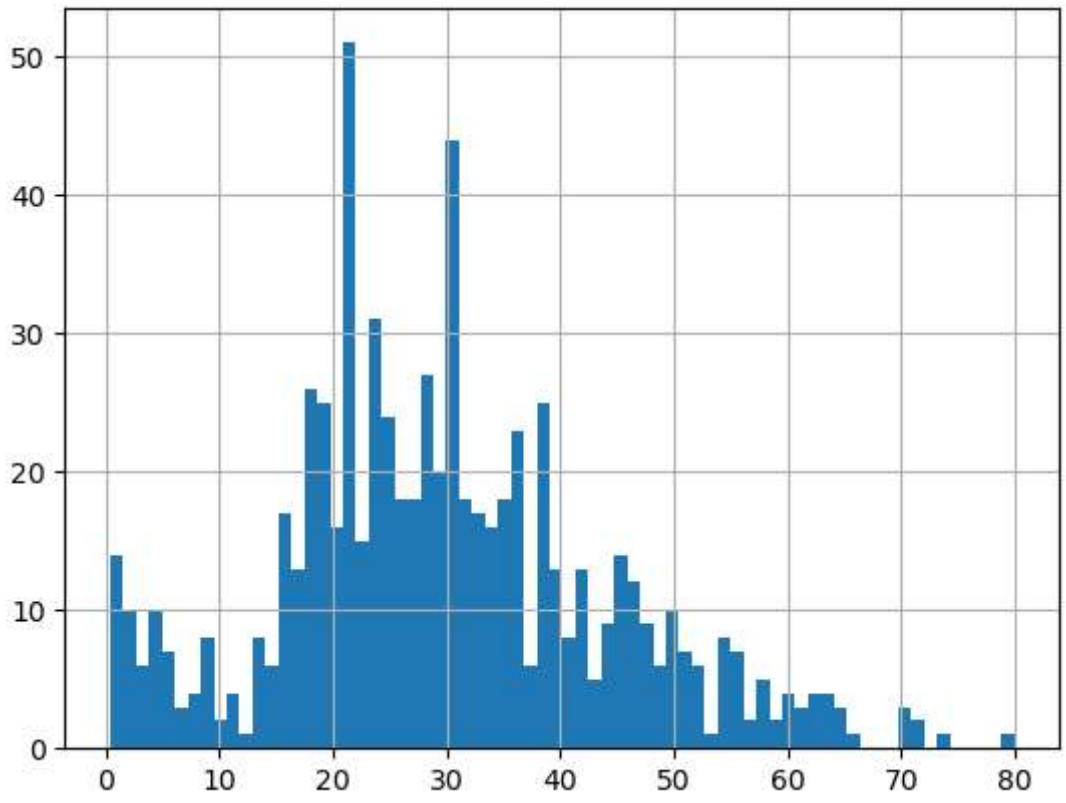
In [71]:
```
# Repeating the Analysis adding the new column created
sns.catplot(
    data=titanic_df, x="Pclass", y="Survived", col="Person",
    kind="bar", height=4, aspect=.6,
)
```

Out[71]:
```
<seaborn.axisgrid.FacetGrid at 0x1e2698dd6d0>
```



In [72]:
```
# Histogram to check the Ages
titanic_df['Age'].hist(bins=70)
```

Out[72]:
```
<AxesSubplot:>
```



In [73]:
```
titanic_df['Age'].mean()
```

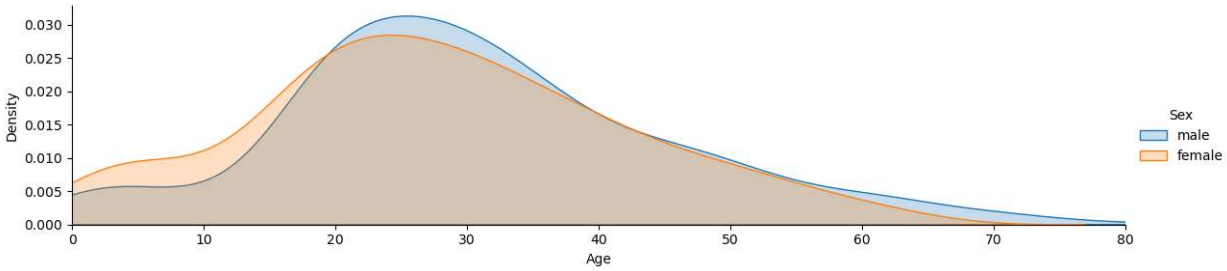
Out[73]:
```
29.69911764705882
```

In [74]:
```
titanic_df['Person'].value_counts()
```

Out[74]:
```
male      537
female    271
child      83
Name: Person, dtype: int64
```

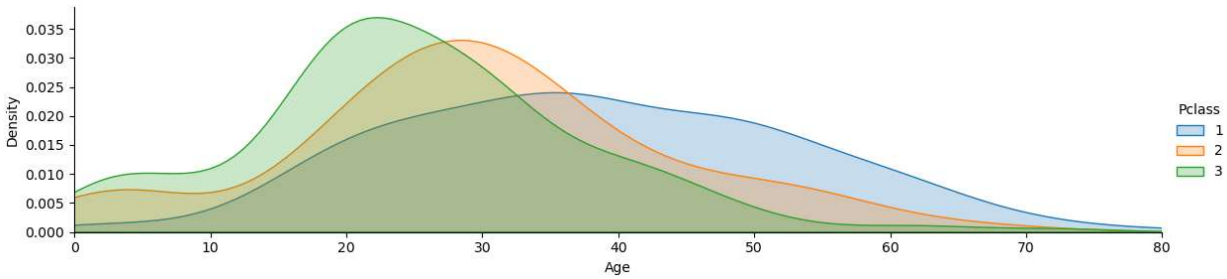
In [75]:
```
# Plot of the distribution of Sex by age
fig = sns.FacetGrid(titanic_df,hue='Sex',aspect=4)
fig.map(sns.kdeplot,'Age',shade=True)
oldest = titanic_df['Age'].max()
fig.set(xlim=(0,oldest))
fig.add_legend()
```

Out[75]:
```
<seaborn.axisgrid.FacetGrid at 0x1e269a1fcd0>
```



In [76]:
```
# Plot of the distribution of Classes by age
fig = sns.FacetGrid(titanic_df,hue='Pclass',aspect=4)
fig.map(sns.kdeplot,'Age',shade=True)
oldest = titanic_df['Age'].max()
fig.set(xlim=(0,oldest))
fig.add_legend()
```

Out[76]:
```
<seaborn.axisgrid.FacetGrid at 0x1e269ba2040>
```



In [77]:
```
deck = titanic_df['Cabin'].dropna()
```

In [78]:
```
deck.head()
```

Out[78]:
```
1      C85
3     C123
6      E46
10      G6
11    C103
Name: Cabin, dtype: object
```

In [79]:
```
levels = []

for level in deck:
    levels.append(level[0])
```

In [80]:
```
cabin_df = DataFrame(levels)
cabin_df.columns = ['Cabin']
```

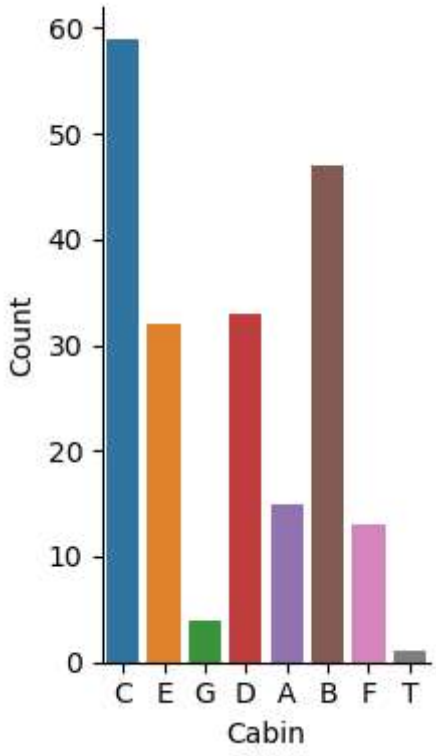
```
cabin_df['Count'] = cabin_df.groupby('Cabin')['Cabin'].transform('count')
cabin_df.head()
```

Out[80]:

	Cabin	Count
0	C	59
1	C	59
2	E	32
3	G	4
4	C	59

In [81]:
```
# Graph to visualize the Cabin Count
sns.catplot(
    data=cabin_df, x="Cabin", y="Count",
    kind="bar", height=4, aspect=.6,
)
```

Out[81]:
```
<seaborn.axisgrid.FacetGrid at 0x1e269ab6e20>
```



In [82]:
```
# Returning to the original DataFrame to start analyzing the Embarked Data
titanic_df.head()
```

Out[82]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	E
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	

In [83]:
```
# creating a new embarked column

embarked_df = titanic_df[['PassengerId','Embarked','Pclass']]
embarked_df.head()
#remover a coluna passengerId e criar o count para countar o numero de embarcados
```

Out[83]:

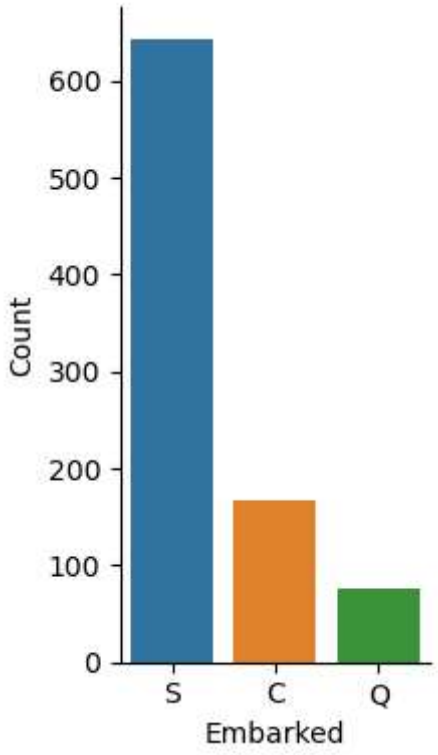
	PassengerId	Embarked	Pclass
0	1	S	3
1	2	C	1
2	3	S	3
3	4	S	1
4	5	S	3

In [84]:
```
embarked_df = embarked_df.rename(columns={'PassengerId':'Count'})
```

In [85]:
```
embarked_df['Count'] = embarked_df.groupby('Embarked')['Embarked'].transform('count')
```

In [86]:
```
# Graph to visualize the Embarked Count
sns.catplot(
    data=embarked_df, x="Embarked", y="Count",
    kind="bar", height=4, aspect=.6,
)
```

Out[86]:
```
<seaborn.axisgrid.FacetGrid at 0x1e269d0afd0>
```



In [87]:
```
# Who was alone and who was with family
titanic_df['Alone'] = titanic_df.SibSp + titanic_df.Parch
titanic_df['Alone'].head()
```

Out[87]:
```
0    1
1    1
2    0
3    1
4    0
Name: Alone, dtype: int64
```

In [88]:
```
titanic_df['Alone'].loc[titanic_df['Alone']>0] = 'With Family'
titanic_df['Alone'].loc[titanic_df['Alone'] == 0] = 'Alone'
```

```
C:\Users\rodri\AppData\Local\Temp\ipykernel_15136\230142470.py:1: SettingWithCopyWarn
ing:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  titanic_df['Alone'].loc[titanic_df['Alone']>0] = 'With Family'
```

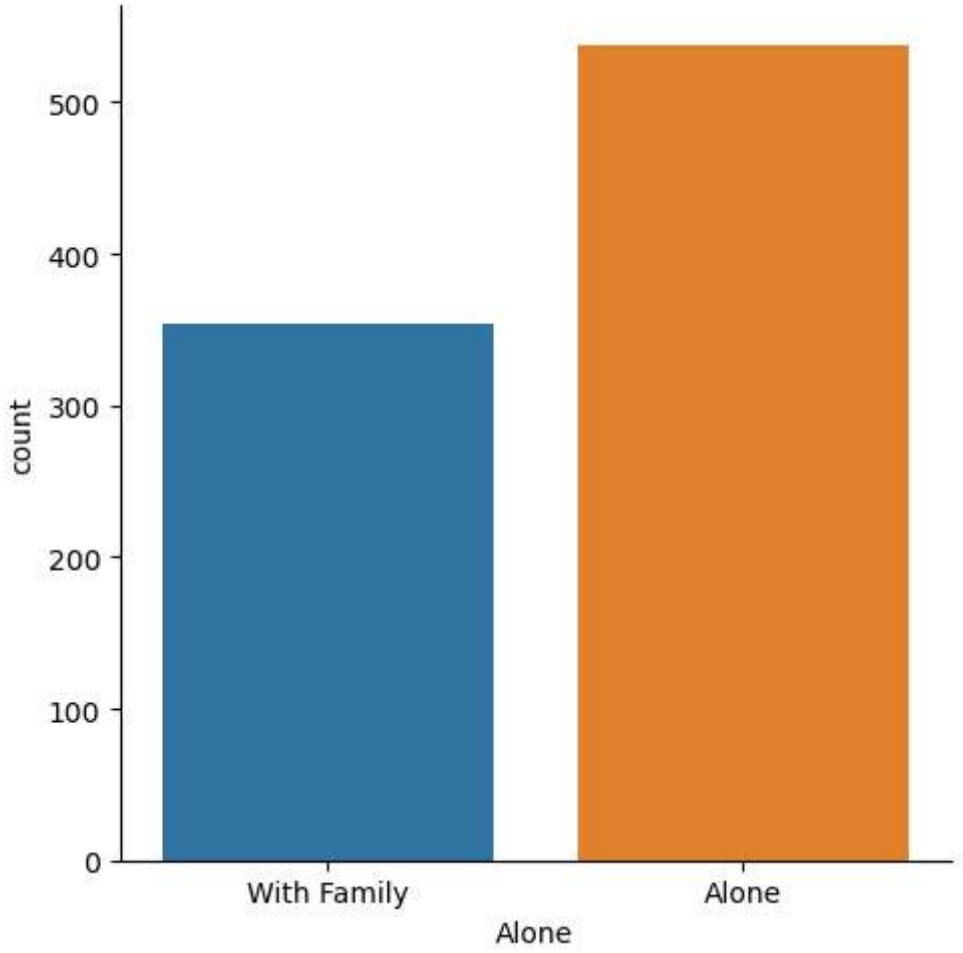
In [89]:
```
titanic_df.head()
```

Out[89]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	E
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	

In [90]:
```
sns.catplot(x="Alone",
                kind="count",
                data=titanic_df)
```

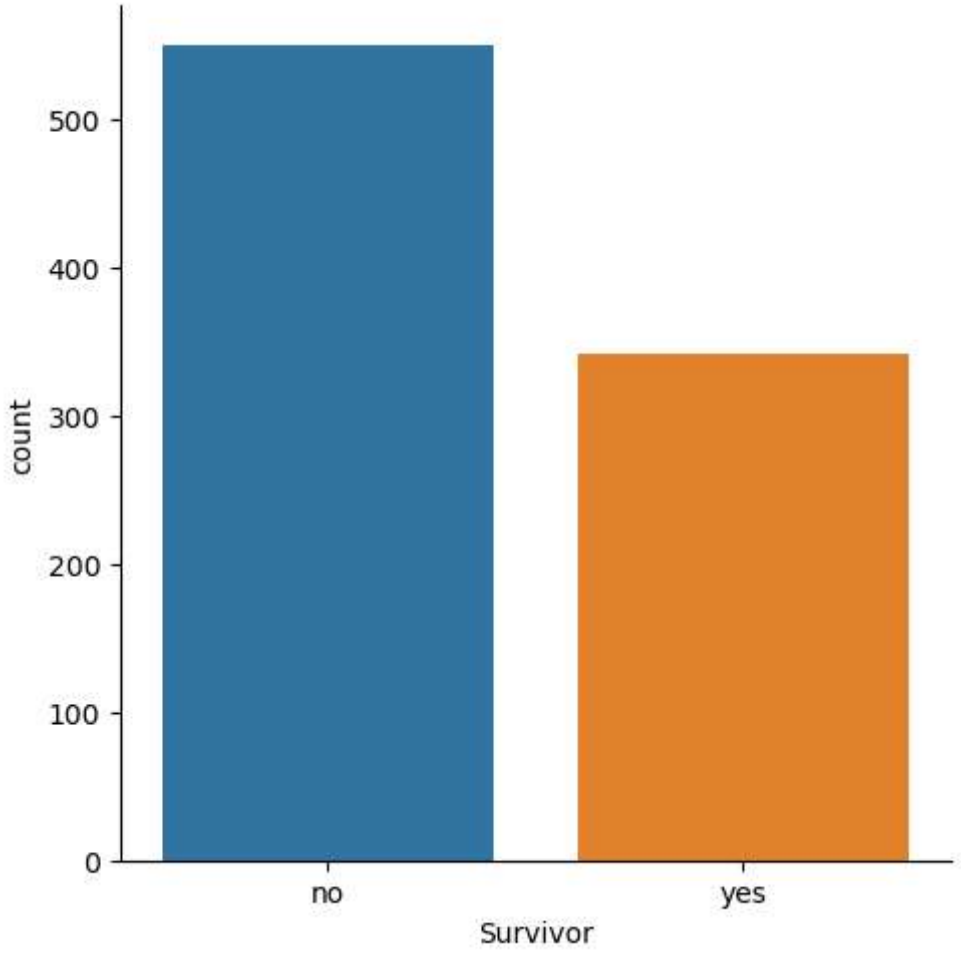
Out[90]:
```
<seaborn.axisgrid.FacetGrid at 0x1e269d4cfa0>
```



In [91]:
```
# search how to replace values in a Pandas column based in a condition
titanic_df['Survivor'] = titanic_df.Survived.map({0:'no',1:'yes'})
```

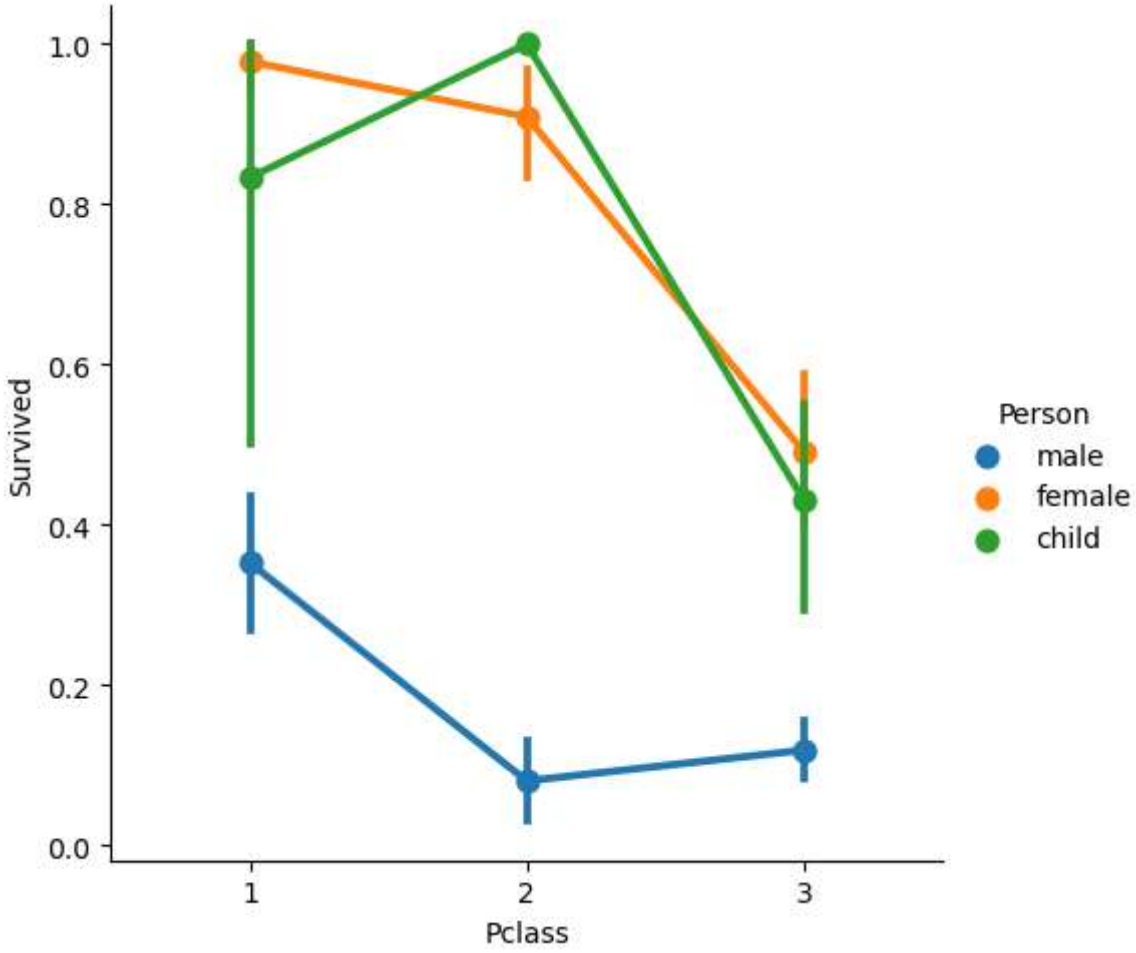
In [92]:
```
sns.catplot(x="Survivor",
                kind="count",
                data=titanic_df)
```

Out[92]:
```
<seaborn.axisgrid.FacetGrid at 0x1e269dd0f70>
```



In [98]:
```
sns.factorplot('Pclass','Survived',hue='Person',data=titanic_df)
```

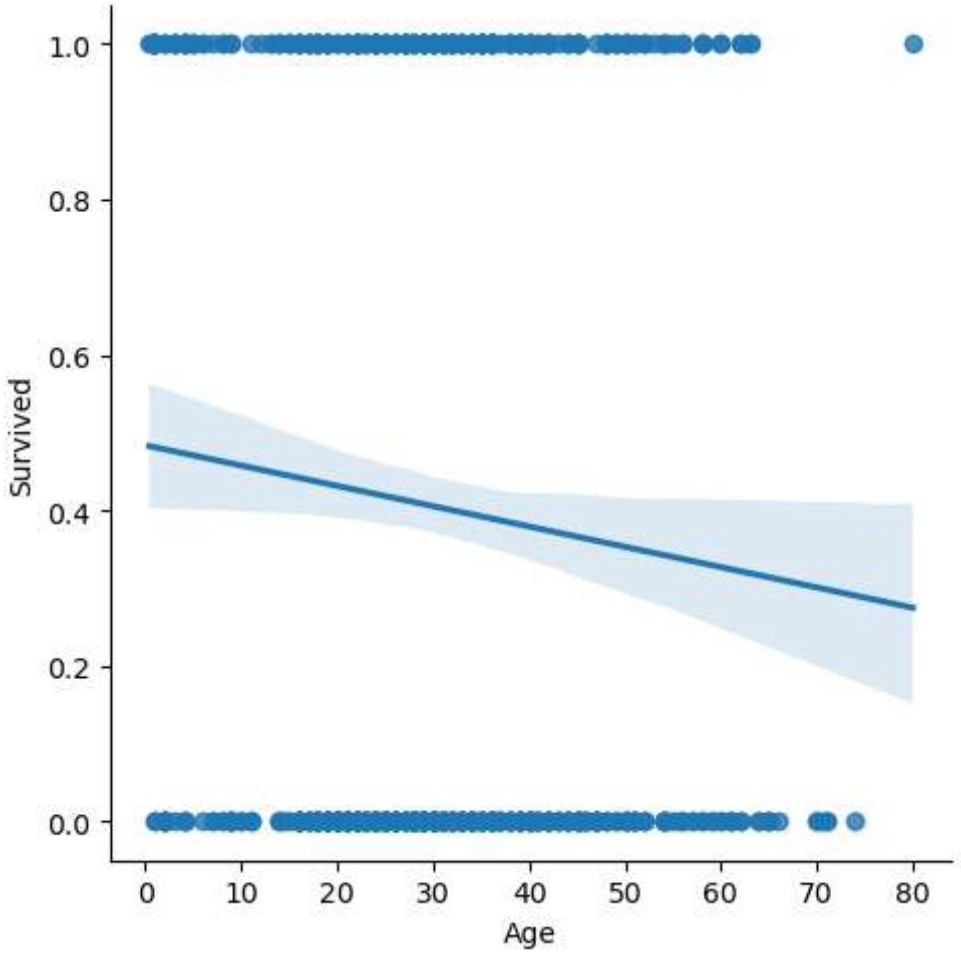
Out[98]:
```
<seaborn.axisgrid.FacetGrid at 0x1e26b3f01c0>
```



In [99]:
```
# Using a linear regression plot, let's see who survived by age
sns.lmplot('Age','Survived',data=titanic_df)
```

```
C:\Users\rodri\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variables as keyword args: x, y. From version 0.12, the only valid
positional argument will be `data`, and passing other arguments without an explicit k
eyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[99]:
```
<seaborn.axisgrid.FacetGrid at 0x1e26b485dc0>
```

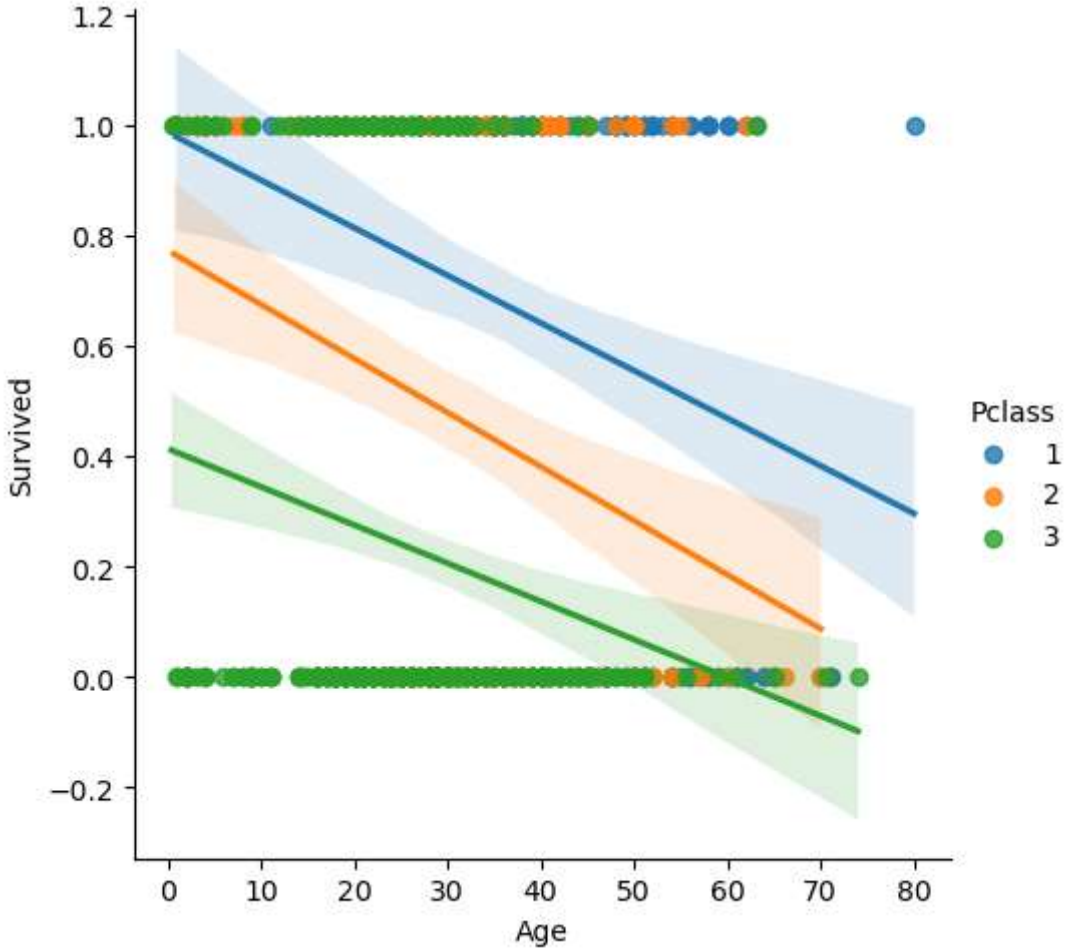


The general trend shows that, how older the passanger was, less likely he was to survive

In [100…
```
sns.lmplot('Age','Survived',hue='Pclass',data=titanic_df)
```

```
C:\Users\rodri\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variables as keyword args: x, y. From version 0.12, the only valid
positional argument will be `data`, and passing other arguments without an explicit k
eyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[100]:
```
<seaborn.axisgrid.FacetGrid at 0x1e26b4f92b0>
```

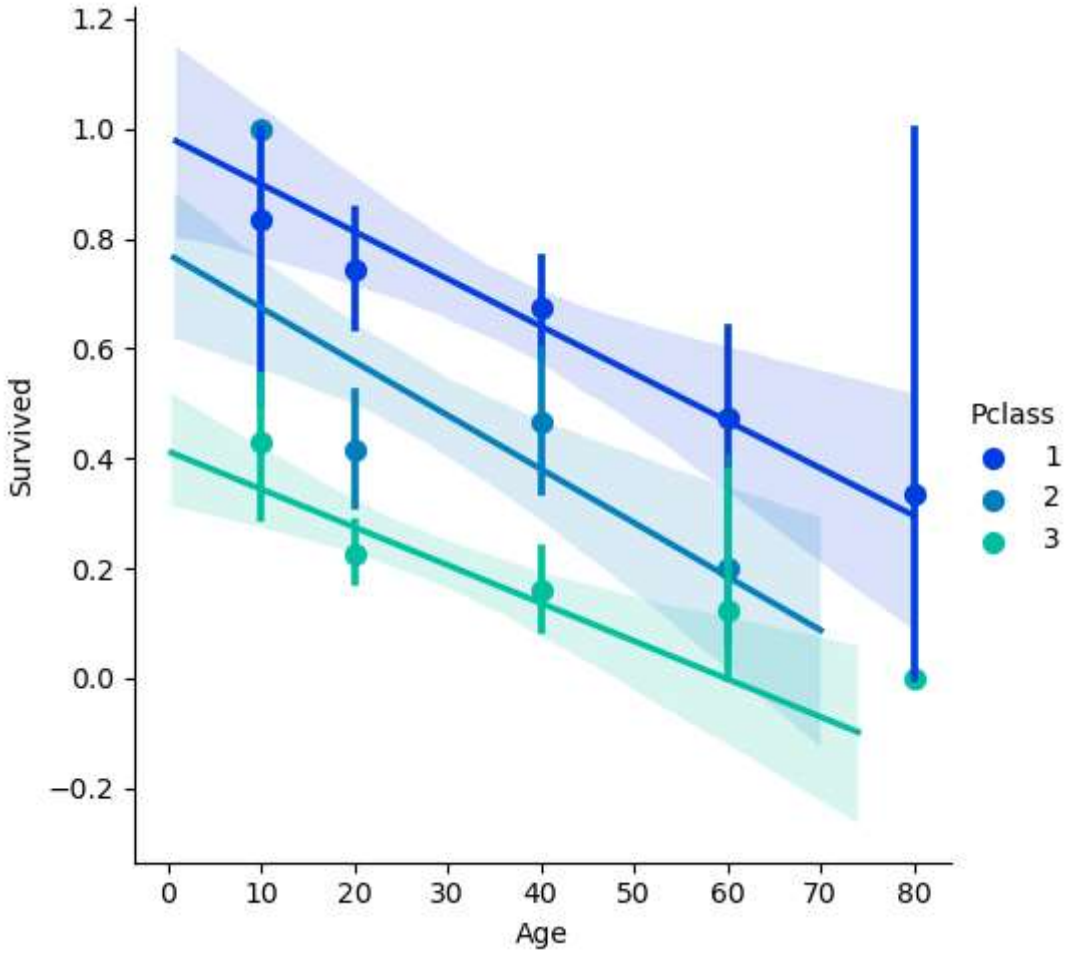


In [101…
```
generations = [10,20,40,60,80]

sns.lmplot('Age','Survived',hue='Pclass',data=titanic_df,palette='winter',x_bins=gener
```

```
C:\Users\rodri\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variables as keyword args: x, y. From version 0.12, the only valid
positional argument will be `data`, and passing other arguments without an explicit k
eyword will result in an error or misinterpretation.
  warnings.warn(
```

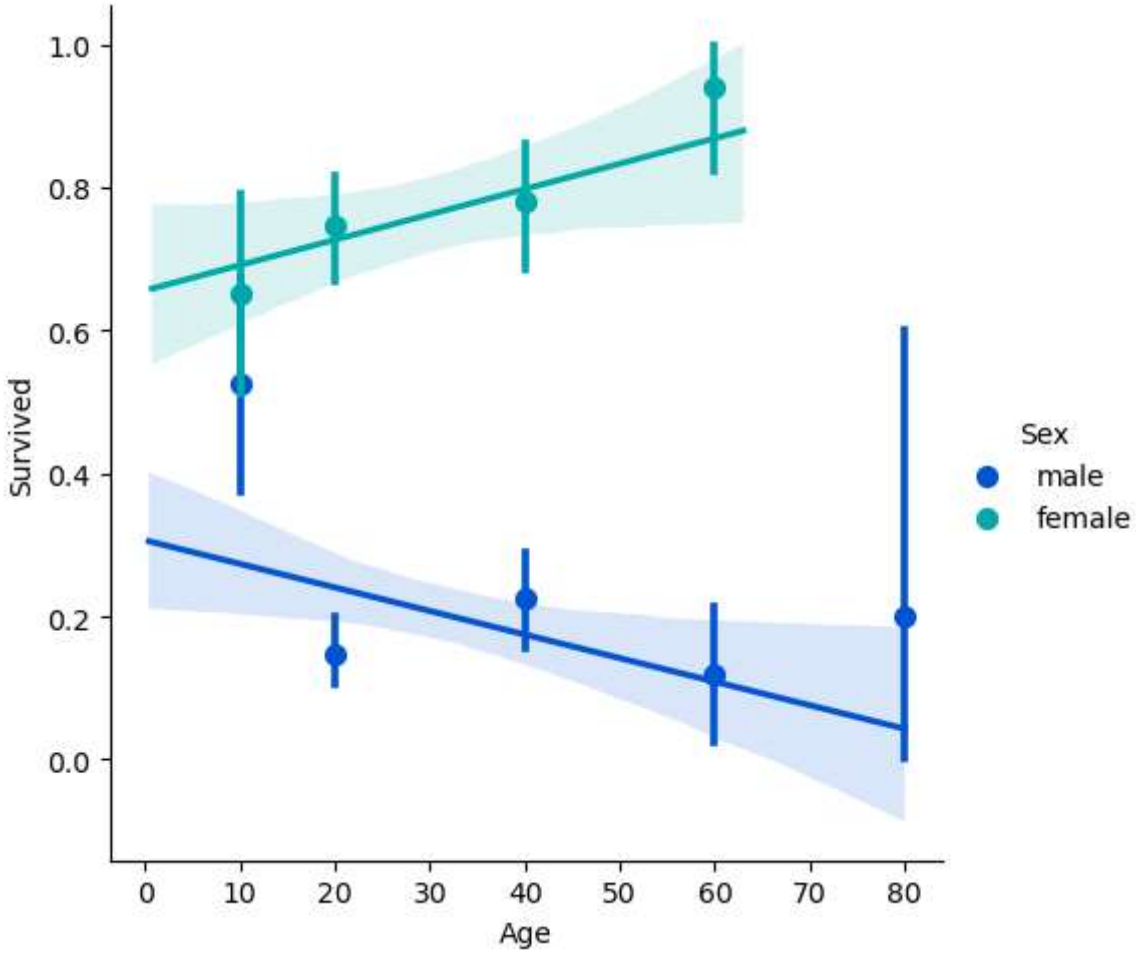
Out[101]:
```
<seaborn.axisgrid.FacetGrid at 0x1e26b541760>
```



In [102…
```
sns.lmplot('Age','Survived',hue='Sex',data=titanic_df,palette='winter',x_bins=generati
```

```
C:\Users\rodri\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variables as keyword args: x, y. From version 0.12, the only valid
positional argument will be `data`, and passing other arguments without an explicit k
eyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[102]:
```
<seaborn.axisgrid.FacetGrid at 0x1e26b485fa0>
```



In [103…

```
titanic_df.head()
```

Out[103]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	E
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	

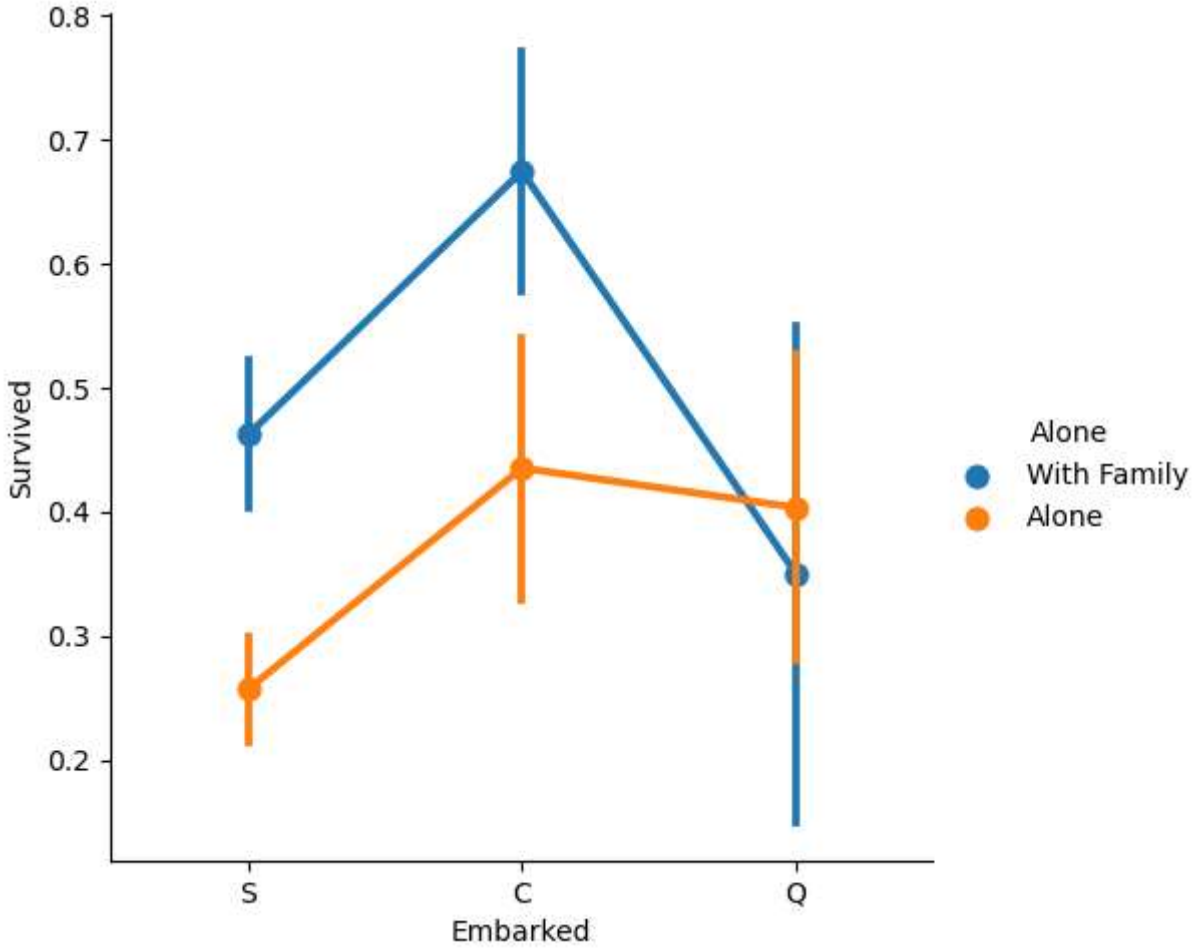
In [104…

```
sns.factorplot('Embarked','Survived',hue='Alone',data=titanic_df)
```

```
C:\Users\rodri\anaconda3\lib\site-packages\seaborn\categorical.py:3717: UserWarning:
The `factorplot` function has been renamed to `catplot`. The original name will be re
moved in a future release. Please update your code. Note that the default `kind` in `
factorplot` (`'point'`) has changed `'strip'` in `catplot`.
  warnings.warn(msg)
C:\Users\rodri\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variables as keyword args: x, y. From version 0.12, the only valid
positional argument will be `data`, and passing other arguments without an explicit k
eyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[104]:

```
<seaborn.axisgrid.FacetGrid at 0x1e26b85ee80>
```


0.8
0.7
0.6
0.5
0.4
0.3
0.2
Survived
S
C
Q
Embarked
Alone
With Family
Alone

In [ ]: