```
import seaborn as sns
```

```
sns.get_dataset_names()
```

```
['anagrams',
 'anscombe',
 'attention',
 'brain_networks',
 'car_crashes',
 'diamonds',
 'dots',
 'dowjones',
 'exercise',
 'flights',
 'fmri',
 'geyser',
 'glue',
 'healthexp',
 'iris',
 'mpg',
 'penguins',
 'planets',
 'seaice',
 'taxis',
 'tips',
 'titanic']
```

```
df=sns.load_dataset('titanic')
```

```
df
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_t
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southamp
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbo
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southamp
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southamp
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southamp
...	...	...	...	...	...	...	...	...	...	...	...	...	
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True	NaN	Southamp
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False	B	Southamp
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False	NaN	Southamp
889	1	1	male	26.0	0	0	30.0000	C	First	man	True	C	Cherbo
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True	NaN	Queenst

891 rows × 15 columns

```
df
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_t
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southamp
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbo
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southamp
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southamp
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southamp
...	...	...	...	...	...	...	...	...	...	...	...	...	
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True	NaN	Southamp
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False	B	Southamp
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False	NaN	Southamp
889	1	1	male	26.0	0	0	30.0000	C	First	man	True	C	Cherbo
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True	NaN	Queenst

891 rows × 15 columns

```
import  pandas as pd
```

```
df.head()
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton

```
df.tail()
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_tow
886	0	2	male	27.0	0	0	13.00	S	Second	man	True	NaN	Southampto
887	1	1	female	19.0	0	0	30.00	S	First	woman	False	B	Southampto
888	0	3	female	NaN	1	2	23.45	S	Third	woman	False	NaN	Southampto
889	1	1	male	26.0	0	0	30.00	C	First	man	True	C	Cherbour
890	0	3	male	32.0	0	0	7.75	Q	Third	man	True	NaN	Queenstow

```
df.sample()
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	a
640	0	3	male	20.0	0	0	7.8542	S	Third	man	True	NaN	Southampton	

```
df.shape
```

```
(891, 15)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   survived     891 non-null    int64
 1   pclass       891 non-null    int64
 2   sex          891 non-null    object
 3   age          714 non-null    float64
 4   sibsp        891 non-null    int64
 5   parch        891 non-null    int64
 6   fare         891 non-null    float64
 7   embarked     889 non-null    object
 8   class        891 non-null    category
 9   who          891 non-null    object
 10  adult_male   891 non-null    bool
 11  deck         203 non-null    category
 12  embark_town  889 non-null    object
 13  alive        891 non-null    object
 14  alone        891 non-null    bool
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.7+ KB
```

```
df.describe()
```

	survived	pclass	age	sibsp	parch	fare
count	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	0.486592	0.836071	13.002015	1.102743	0.806057	49.693429
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	0.000000	2.000000	22.000000	0.000000	0.000000	7.910400
50%	0.000000	3.000000	29.699118	0.000000	0.000000	14.454200
75%	1.000000	3.000000	35.000000	1.000000	0.000000	31.000000
max	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

```
df.dtypes
```

	0
survived	int64
pclass	int64
sex	object
age	float64
sibsp	int64
parch	int64
fare	float64
embarked	object
class	category
who	object
adult_male	bool
deck	category
embark_town	object
alive	object
alone	bool

dtype: object

```
df.columns
```

```
Index(['survived', 'pclass', 'sex', 'age', 'sibsp', 'parch', 'fare',
       'embarked', 'class', 'who', 'adult_male', 'deck', 'embark_town',
       'alive', 'alone'],
      dtype='object')
```

```
df.index
```

```
RangeIndex(start=0, stop=891, step=1)
```

```
df.isnull().sum()
```

	0
survived	0
pclass	0
sex	0
sibsp	0
parch	0
embarked	2
who	0
deck	688
alive	0

dtype: int64

```
df['age'].mean()
```

```
np.float64(29.69911764705882)
```

```
df['age']=df['age'].fillna(df['age'].mean())
```

```
df
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	emb
0	0	3	male	22.000000	1	0	7.2500	S	Third	man	True	NaN	Sou
1	1	1	female	38.000000	1	0	71.2833	C	First	woman	False	C	(
2	1	3	female	26.000000	0	0	7.9250	S	Third	woman	False	NaN	Sou
3	1	1	female	35.000000	1	0	53.1000	S	First	woman	False	C	Sou
4	0	3	male	35.000000	0	0	8.0500	S	Third	man	True	NaN	Sou
...	...	...	...	...	...	...	...	...	...	...	...	...	
886	0	2	male	27.000000	0	0	13.0000	S	Second	man	True	NaN	Sou
887	1	1	female	19.000000	0	0	30.0000	S	First	woman	False	B	Sou
888	0	3	female	29.699118	1	2	23.4500	S	Third	woman	False	NaN	Sou
889	1	1	male	26.000000	0	0	30.0000	C	First	man	True	C	(
890	0	3	male	32.000000	0	0	7.7500	Q	Third	man	True	NaN	Qu

891 rows × 15 columns

```
df=df.drop("deck",axis=1)
```

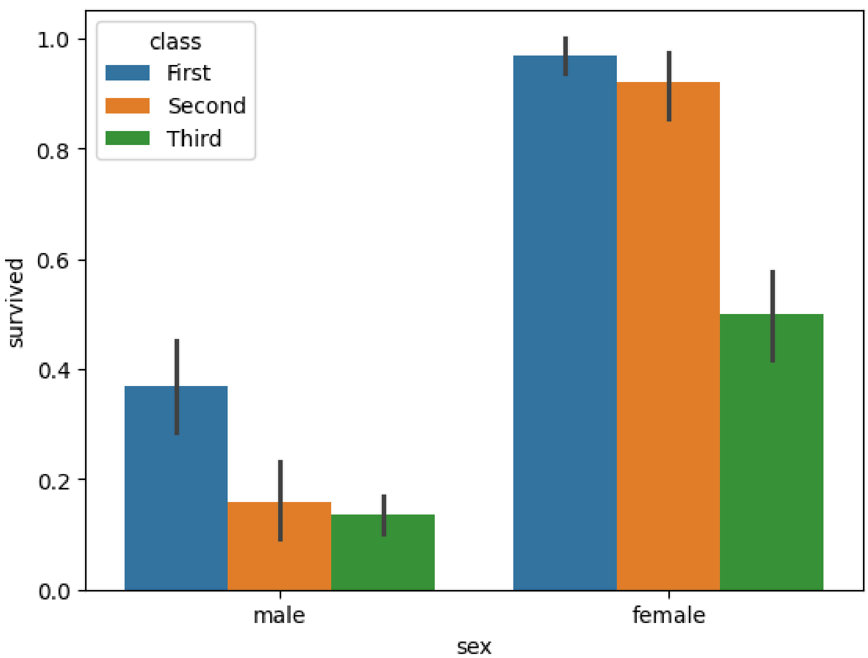
```
df
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	embark_to
0	0	3	male	22.000000	1	0	7.2500	S	Third	man	True	Southampt
1	1	1	female	38.000000	1	0	71.2833	C	First	woman	False	Cherbou
2	1	3	female	26.000000	0	0	7.9250	S	Third	woman	False	Southampt
3	1	1	female	35.000000	1	0	53.1000	S	First	woman	False	Southampt
4	0	3	male	35.000000	0	0	8.0500	S	Third	man	True	Southampt
...	...	...	...	...	...	...	...	...	...	...	...	
886	0	2	male	27.000000	0	0	13.0000	S	Second	man	True	Southampt
887	1	1	female	19.000000	0	0	30.0000	S	First	woman	False	Southampt
888	0	3	female	29.699118	1	2	23.4500	S	Third	woman	False	Southampt
889	1	1	male	26.000000	0	0	30.0000	C	First	man	True	Cherbou
890	0	3	male	32.000000	0	0	7.7500	Q	Third	man	True	Queensto

891 rows × 14 columns

```
sns.barplot(x='sex',y='survived', hue='class',data=df)
```

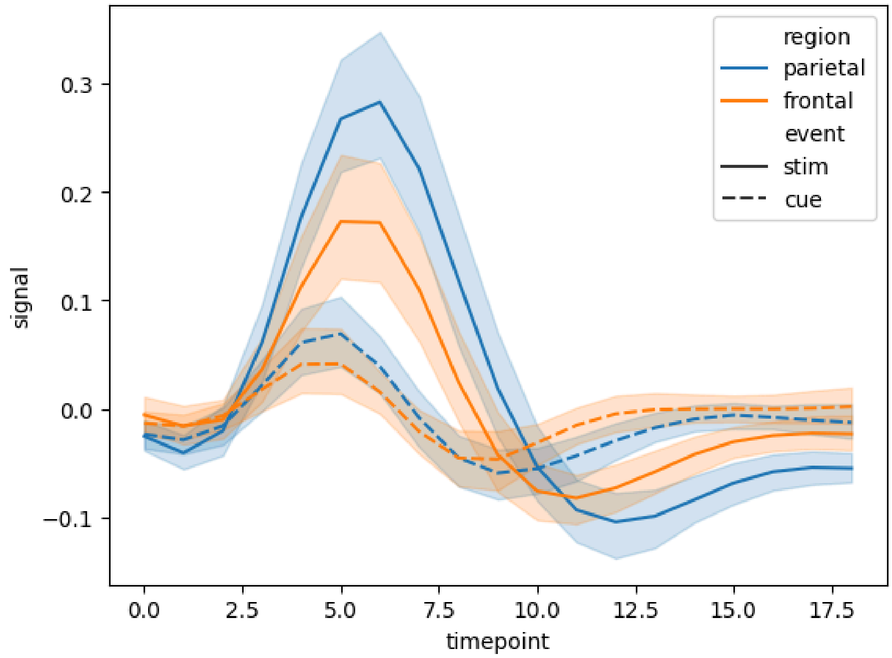
```
<Axes: xlabel='sex', ylabel='survived'>
```




```
plt.figure(figsize=(3,2))
ax = sns.lineplot(x="survived" , y = "sex", hue = "class" , data=df)
ax.set_title("survived vs sex")
plt.show()
```

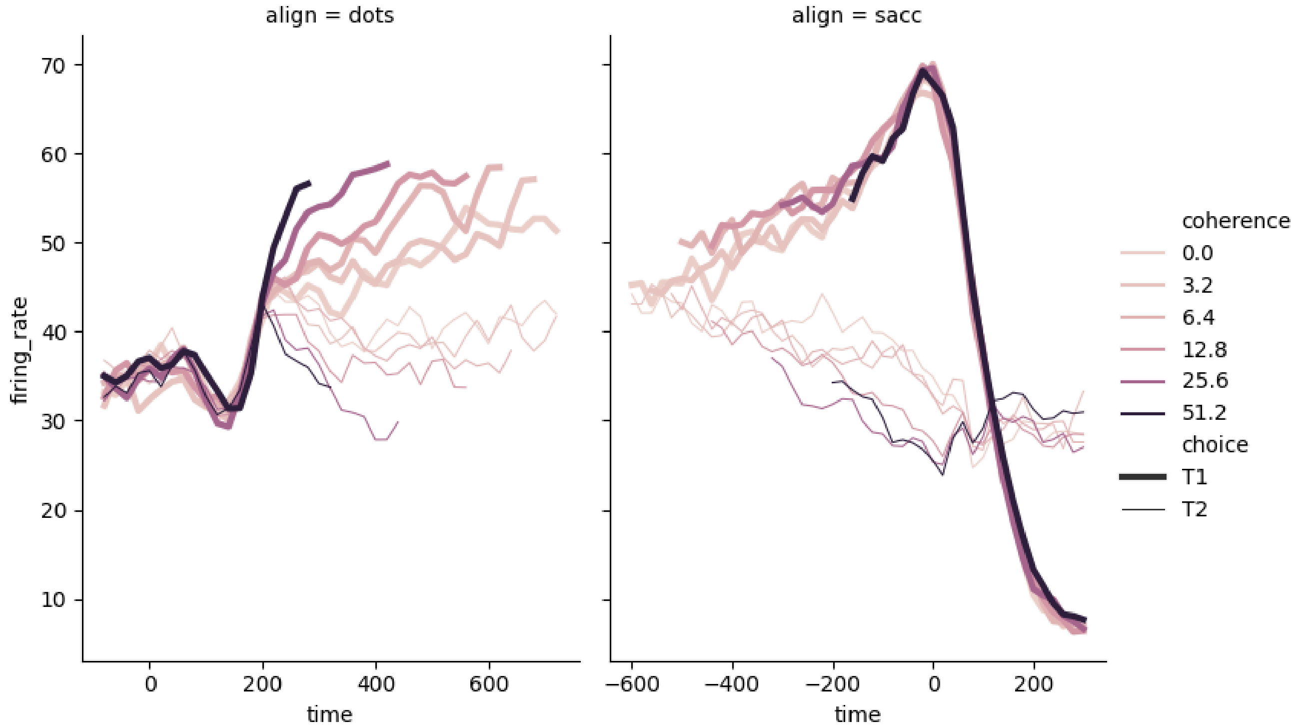

```
fmri = sns.load_dataset('fmri')
sns.lineplot(x="timepoint", y="signal",
             hue="region", style="event",
             data=fmri)
```

```
<Axes: xlabel='timepoint', ylabel='signal'>
```



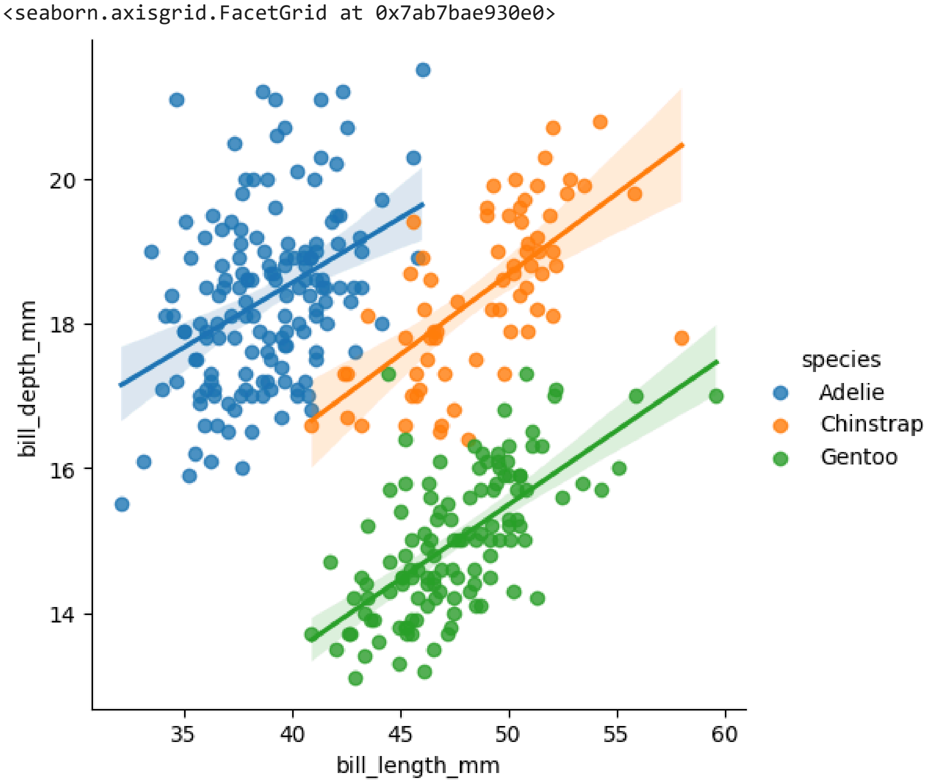

```
dots = sns.load_dataset('dots')
sns.relplot(
    data=dots,
    x="time", y="firing_rate",
    hue="coherence", size="choice", col="align",
    kind="line", size_order=["T1", "T2"],
    height=5, aspect=.75, facet_kws=dict(sharex=False),
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7ab7bea89cd0>
```



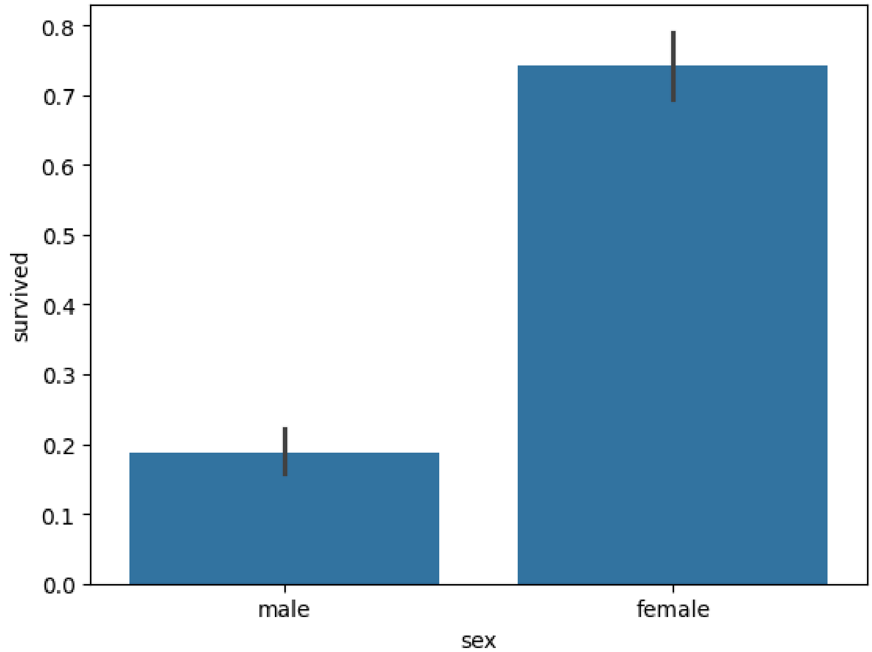

```
penguins = sns.load_dataset('penguins')
sns.lmplot(
    data=penguins,
    x="bill_length_mm", y="bill_depth_mm", hue="species",
    height=5
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7ab7bae930e0>
```



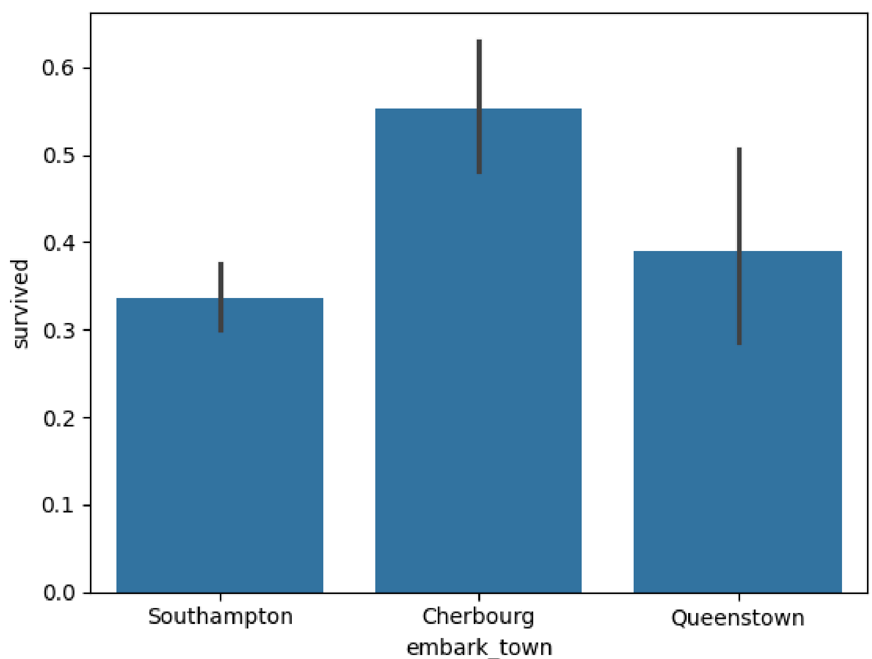

```
sns.barplot(x='sex', y='survived', data=df)
```

```
<Axes: xlabel='sex', ylabel='survived'>
```



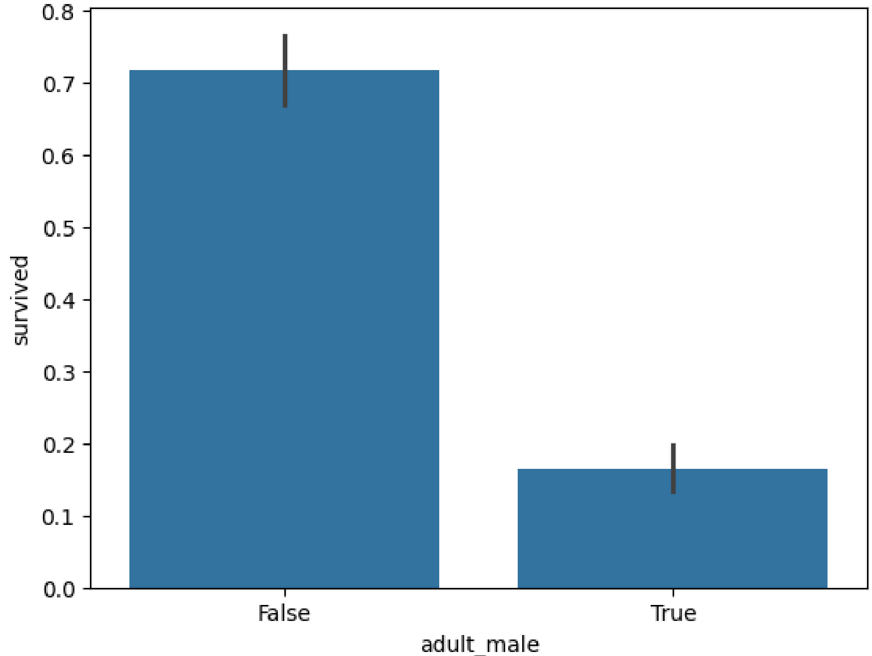

```
sns.barplot(x='embark_town', y='survived', data=df)
```

```
<Axes: xlabel='embark_town', ylabel='survived'>
```




```
sns.barplot(x='adult_male', y='survived', data=df)
```

```
<Axes: xlabel='adult_male', ylabel='survived'>
```




```
display(df.corr(numeric_only=True))
```

	survived	pclass	age	sibsp	parch	fare	adult_male	alone
survived	1.000000	-0.338481	-0.069809	-0.035322	0.081629	0.257307	-0.557080	-0.203367
pclass	-0.338481	1.000000	-0.331339	0.083081	0.018443	-0.549500	0.094035	0.135207
age	-0.069809	-0.331339	1.000000	-0.232625	-0.179191	0.091566	0.253236	0.179775
sibsp	-0.035322	0.083081	-0.232625	1.000000	0.414838	0.159651	-0.253586	-0.584471
parch	0.081629	0.018443	-0.179191	0.414838	1.000000	0.216225	-0.349943	-0.583398
fare	0.257307	-0.549500	0.091566	0.159651	0.216225	1.000000	-0.182024	-0.271832
adult_male	-0.557080	0.094035	0.253236	-0.253586	-0.349943	-0.182024	1.000000	0.404744
alone	-0.203367	0.135207	0.179775	-0.584471	-0.583398	-0.271832	0.404744	1.000000

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
a=['sex', 'embarked', 'class', 'who', 'adult_male', 'embark_town', 'alive', 'alone']
for i in a:
    df[i]=le.fit_transform(df[i])
```

```
df
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	embark_town
0	0	3	1	22.000000	1	0	7.2500	2	2	1	1	2
1	1	1	0	38.000000	1	0	71.2833	0	0	2	0	0
2	1	3	0	26.000000	0	0	7.9250	2	2	2	0	2
3	1	1	0	35.000000	1	0	53.1000	2	0	2	0	2
4	0	3	1	35.000000	0	0	8.0500	2	2	1	1	2
887	1	1	0	19.000000	0	0	30.0000	2	0	2	0	2
889	1	1	1	26.000000	0	0	30.0000	0	0	1	1	0

```
df.corr()
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	
survived	1.000000	-0.338481	-0.543351	-0.069809	-0.035322	0.081629	0.257307	-0.163517	-0.338481	0.325
pclass	-0.338481	1.000000	0.131900	-0.331339	0.083081	0.018443	-0.549500	0.157112	1.000000	-0.196