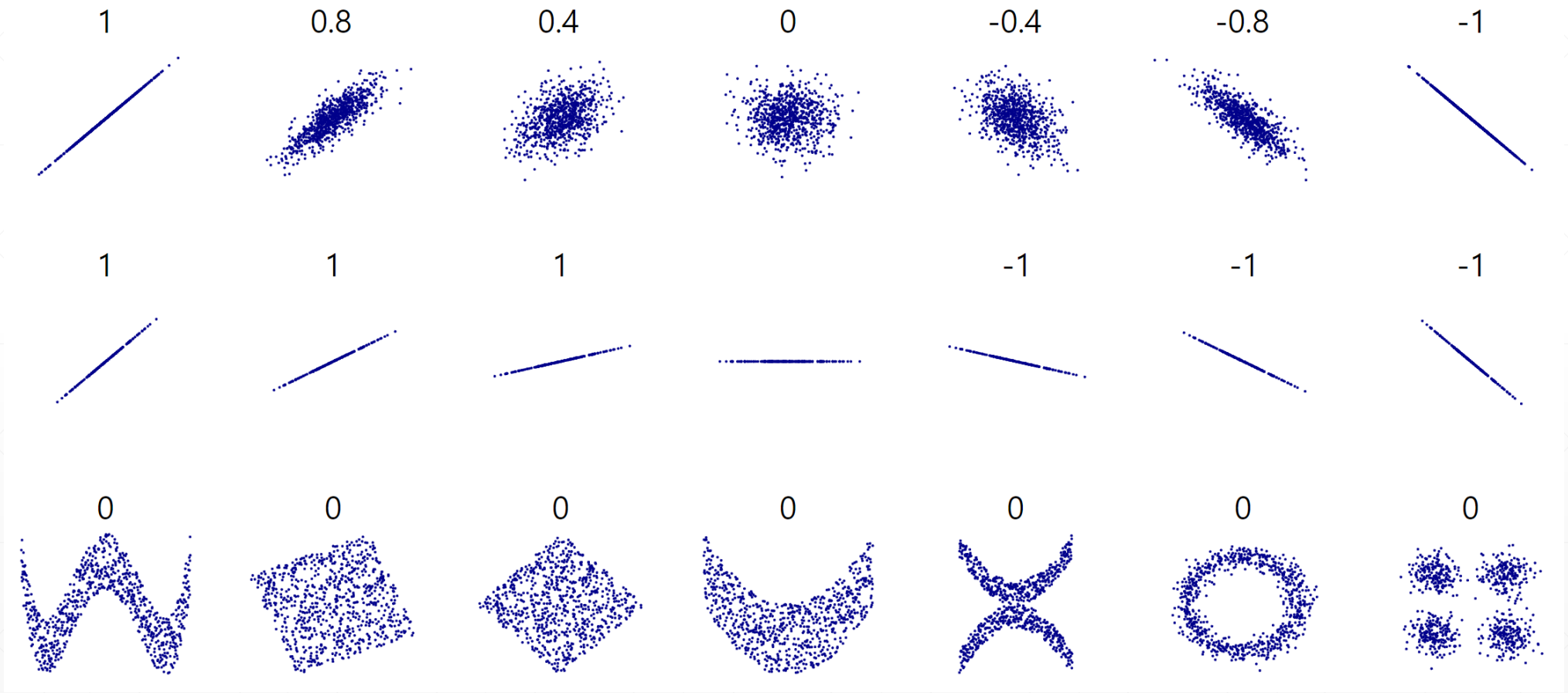
상관 분석

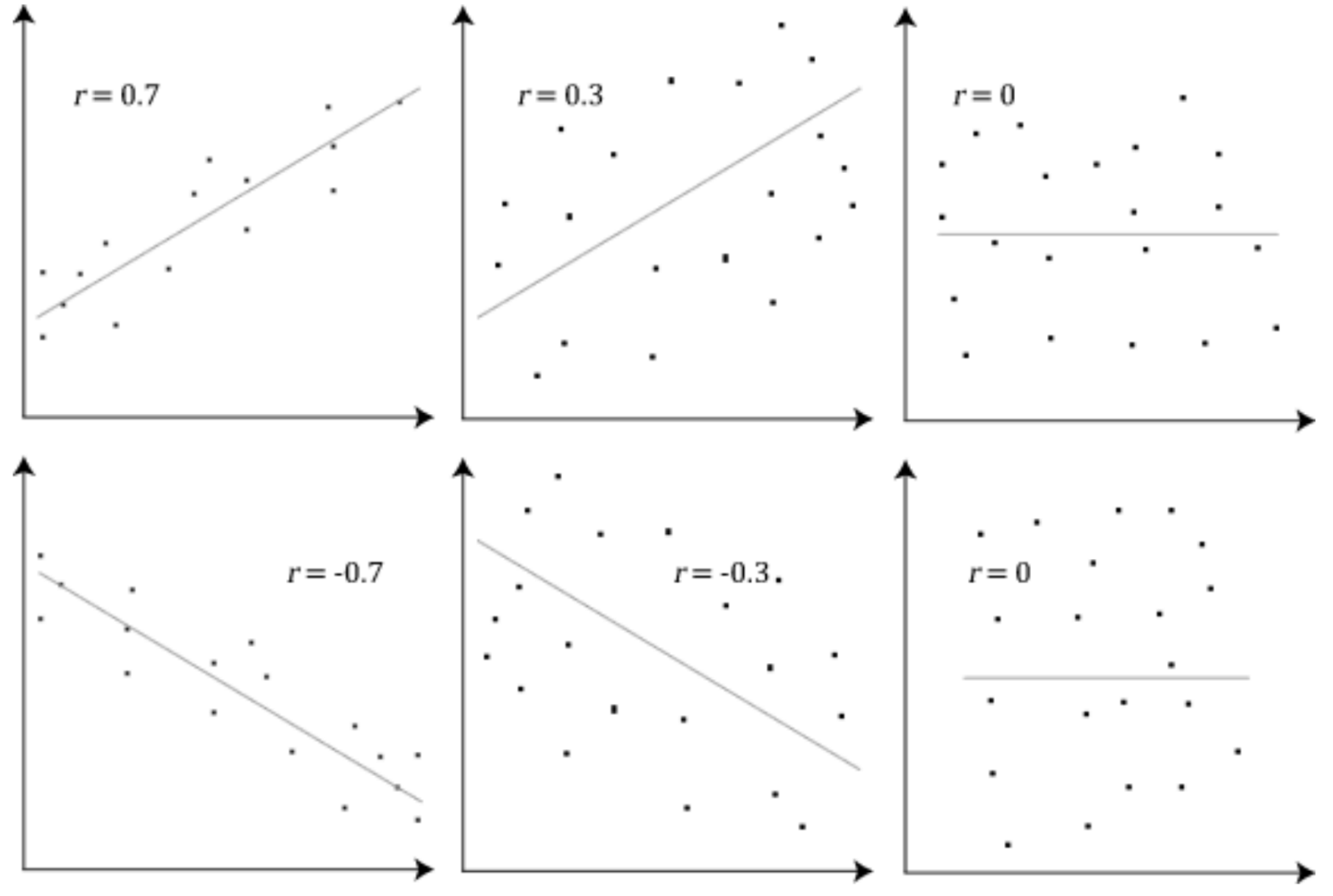
빅데이터 분석

상관 분석 (correlation analysis)



피어슨 상관 계수

$$r_{XY} = \frac{\sum_i^n \left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{\sqrt{\sum_i^n \left(X_i - \overline{X}\right)^2}\sqrt{\sum_i^n \left(Y_i - \overline{Y}\right)^2}}$$



데이터 수집

```
C:\> pip install seaborn
```

```
>>> import seaborn as sns
>>> import pandas as pd
>>> titanic=sns.load_dataset("titanic")
>>> titanic.to_csv('titanic.csv',index=False)
```

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_to	alive	alone
2	0	3	male	22	1	0	7.25	S	Third	man	TRUE		Southamp	no	FALSE
3	1	1	female	38	1	0	71.2833	C	First	woman	FALSE	C	Cherbourg	yes	FALSE
4	1	3	female	26	0	0	7.925	S	Third	woman	FALSE		Southamp	yes	TRUE
5	1	1	female	35	1	0	53.1	S	First	woman	FALSE	C	Southamp	yes	FALSE
6	0	3	male	35	0	0	8.05	S	Third	man	TRUE		Southamp	no	TRUE
7	0	3	male		0	0	8.4583	Q	Third	man	TRUE		Queensto	no	TRUE
8	0	1	male	54	0	0	51.8625	S	First	man	TRUE	E	Southamp	no	TRUE
9	0	3	male	2	3	1	21.075	S	Third	child	FALSE		Southamp	no	FALSE
10	1	3	female	27	0	2	11.1333	S	Third	woman	FALSE		Southamp	yes	FALSE
11	1	2	female	14	1	0	30.0708	C	Second	child	FALSE		Cherbourg	yes	FALSE
12	1	3	female	4	1	1	16.7	S	Third	child	FALSE	G	Southamp	yes	FALSE
13	1	1	female	58	0	0	26.55	S	First	woman	FALSE	C	Southamp	yes	TRUE
14	0	3	male	20	0	0	8.05	S	Third	man	TRUE		Southamp	no	TRUE
15	0	3	male	39	1	5	31.275	S	Third	man	TRUE		Southamp	no	FALSE

데이터 준비

데이터 정리

```
>>> titanic.isnull().sum()
>>> titanic['age']=titanic['age'].fillna(titanic['age'].median())
>>> titanic['embarked'].value_counts()
>>> titanic['embarked']=titanic['embarked'].fillna('S')
>>> titanic['embark_town'].value_counts()
>>> titanic['embark_town']=titanic['embark_town'].fillna('Southampton')
>>> titanic['deck'].value_counts()
>>> titanic['deck']=titanic['deck'].fillna('C')
>>> titanic.isnull().sum()
```

데이터 탐색

데이터의 기본 정보 탐색하기

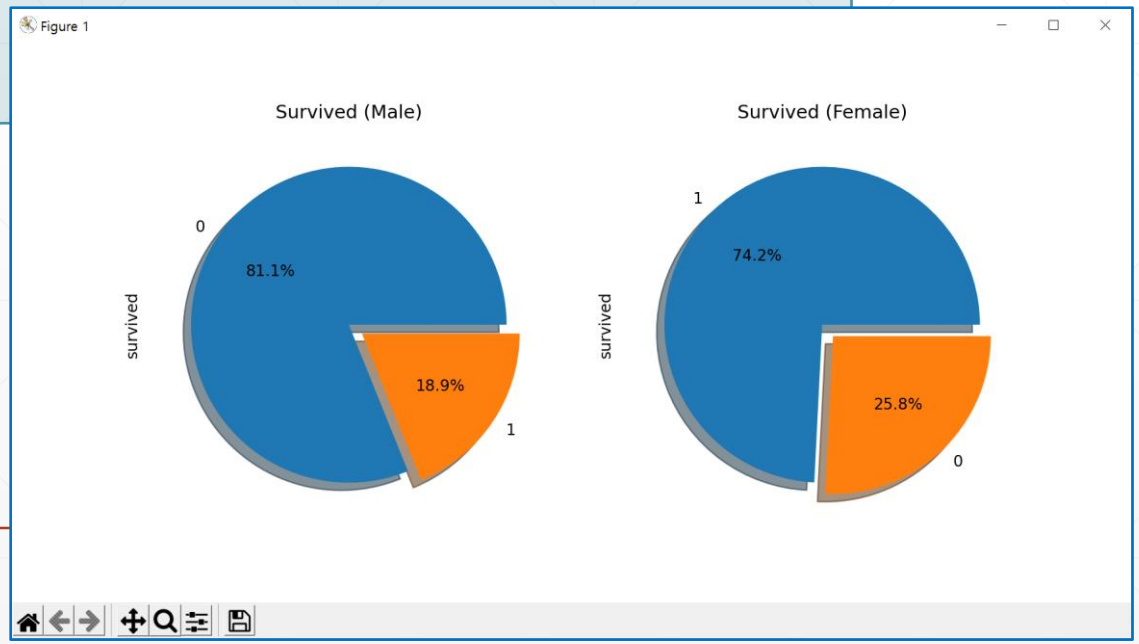
```
>>> titanic.info()
>>> titanic.survived.value_counts()
```

```
>>> titanic.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          891 non-null    float64
 4   sibsp        891 non-null    int64
 5   parch        891 non-null    int64
 6   fare         891 non-null    float64
 7   embarked     891 non-null    object
 8   class        891 non-null    category
 9   who          891 non-null    object
 10  adult_male   891 non-null    bool
 11  deck         891 non-null    category
 12  embark_town  891 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
>>> titanic.survived.value_counts()
0    549
1    342
Name: survived, dtype: int64
```

데이터 탐색 (cont'd)

차트를 그려 데이터를 시각적으로 탐색하기

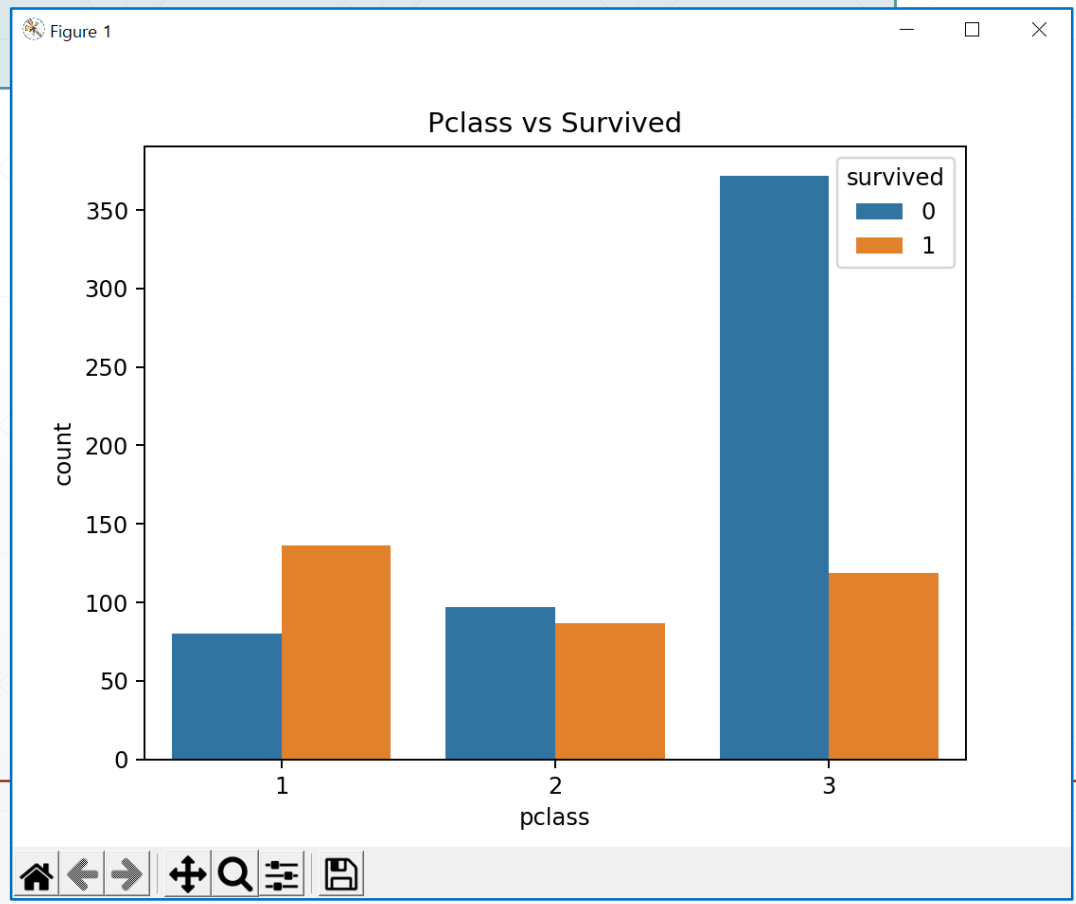
```
>>> import matplotlib.pyplot as plt
>>> f,ax=plt.subplots(1, 2, figsize=(10,5))
>>> titanic['survived'][titanic['sex']=='male'].value_counts().plot.pie(
    explode=[0,0.1],autopct='%1.1f%%',ax=ax[0],shadow=True)
>>> titanic['survived'][titanic['sex']=='female'].value_counts().plot.pie(
    explode=[0,0.1],autopct='%1.1f%%',ax=ax[1],shadow=True)
>>> ax[0].set_title('Survived (Male)')
>>> ax[1].set_title('Survived (Female)')
>>> plt.show()
```



데이터 탐색 (cont'd)

등급별 생존자 수를 차트로 나타내기

```
>>> sns.countplot(x='pclass', hue='survived', data=titanic)
>>> plt.title('Pclass vs Survived')
>>> plt.show()
```



데이터 모델링

상관 분석을 위한 상관 계수 구하고 저장하기

```
>>> titanic_corr=titanic.corr(method='pearson')
>>> titanic_corr
>>> titanic_corr.to_csv('titanic_corr.csv', index=False)
```

특정 변수 사이의 상관 계수 구하기

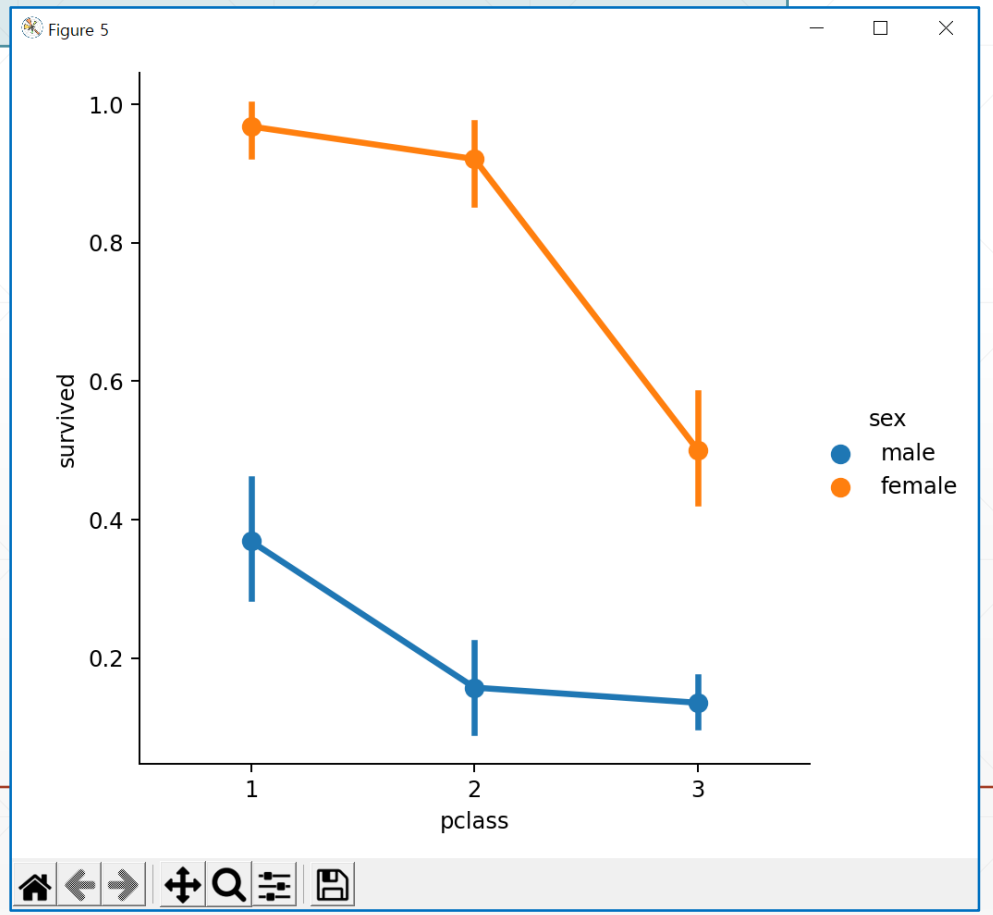
```
>>> titanic['survived'].corr(titanic['adult_male'])
>>> titanic['survived'].corr(titanic['fare'])
```

	A	B	C	D	E	F	G	H
1	survived	pclass	age	sibsp	parch	fare	adult_male	alone
2	1	-0.33848	-0.06491	-0.03532	0.081629	0.257307	-0.55708	-0.20337
3	-0.33848	1	-0.3399	0.083081	0.018443	-0.5495	0.094035	0.135207
4	-0.06491	-0.3399	1	-0.2333	-0.17248	0.096688	0.247704	0.171647
5	-0.03532	0.083081	-0.2333	1	0.414838	0.159651	-0.25359	-0.58447
6	0.081629	0.018443	-0.17248	0.414838	1	0.216225	-0.34994	-0.5834
7	0.257307	-0.5495	0.096688	0.159651	0.216225	1	-0.18202	-0.27183
8	-0.55708	0.094035	0.247704	-0.25359	-0.34994	-0.18202	1	0.404744
9	-0.20337	0.135207	0.171647	-0.58447	-0.5834	-0.27183	0.404744	1

결과 시각화

두 변수의 상관관계 시각화하기

```
>>> sns.catplot(x='pclass', y='survived', hue='sex', data=titanic, kind='point')
>>> plt.show()
```



결과 시각화 (cont'd)

```
>>> def category_age(x):
        if x < 10:
            return 0
        elif x < 20:
            return 1
        elif x <30:
            return 2
        elif x < 40:
            return 3
        elif x < 50:
            return 4
        elif x < 60:
            return 5
        elif x < 70:
            return 6
        else:
            return 7
```

변수 사이의 상관 계수를 히트맵으로 시각화하기

```
>>> titanic['age2']=titanic['age'].apply(category_age)
>>> titanic['sex']=titanic['sex'].map({'male':1, 'female':0})
>>> titanic['family']=titanic['sibsp']+titanic['parch']+1
>>> titanic.to_csv('titanic3.csv', index=False)
>>> heatmap_data=titanic[['survived', 'sex', 'age2', 'family', 'pclass', 'fare']]
>>> colormap=plt.cm.RdBu
>>> sns.heatmap(heatmap_data.astype(float).corr(), linewidths=0.1, vmax=1.0,
    square=True, cmap=colormap, linecolor='white', annot=True,
    annot_kws={"size": 10})
>>> plt.show()
```