Chi-Squared Test (Test of Independence)

A chi-squared test (χ^2 test) is used to decide whether there is a relationship exists between two categorical variables of a sample.

In this exercise, we will perform χ^2 test based on 'Immunotherapy.csv' and 'titanic.csv' dataset.

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
```

Immunotherapy Dataset

In [2]:
```
df_immuno = pd.read_csv("https://raw.githubusercontent.com/ThammakornS/ProgStat/mai
df_immuno.head()
```

Out[2]:

	gender	age	time	number_of_warts	type	area	induration_diameter	response
0	Female	15	1.75	1	Plantar	49	7	No
1	Female	38	2.50	1	Both	43	50	Yes
2	Female	24	4.25	1	common	174	30	Yes
3	Female	34	8.50	1	Plantar	163	7	No
4	Female	53	10.00	1	Plantar	30	25	Yes

Adjust Data Type

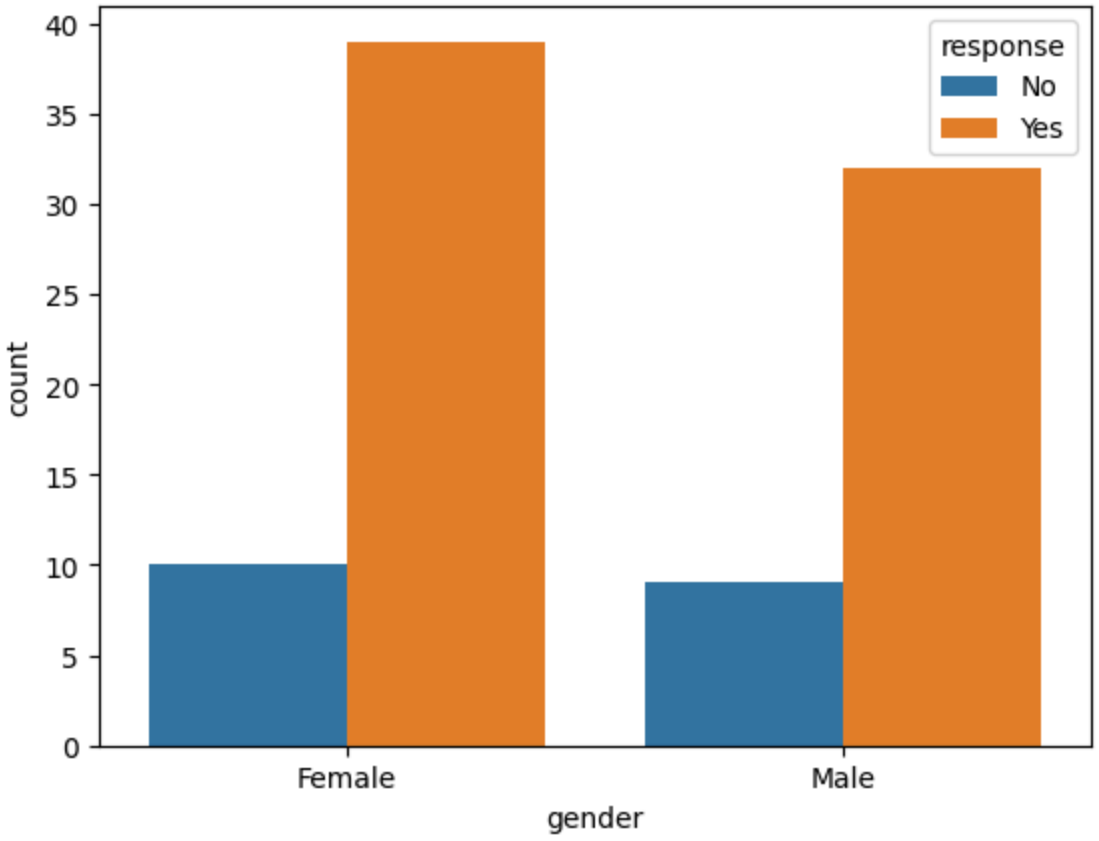
In [3]:
```
df_immuno = df_immuno.astype({
    'gender': 'category',
    'type': 'category',
    'response': 'category'
})
df_immuno.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 90 entries, 0 to 89
Data columns (total 8 columns):
 #   Column               Non-Null Count  Dtype   
---  ------               --------------  -----   
 0   gender               90 non-null     category
 1   age                  90 non-null     int64   
 2   time                 90 non-null     float64 
 3   number_of_warts      90 non-null     int64   
 4   type                 90 non-null     category
 5   area                 90 non-null     int64   
 6   induration_diameter  90 non-null     int64   
 7   response             90 non-null     category
dtypes: category(3), float64(1), int64(4)
memory usage: 4.3 KB
```

Visualization of Categorical Data

In [4]:
```
sns.countplot(data=df_immuno,
              x='gender',
              hue='response')
```

Out[4]: <Axes: xlabel='gender', ylabel='count'>



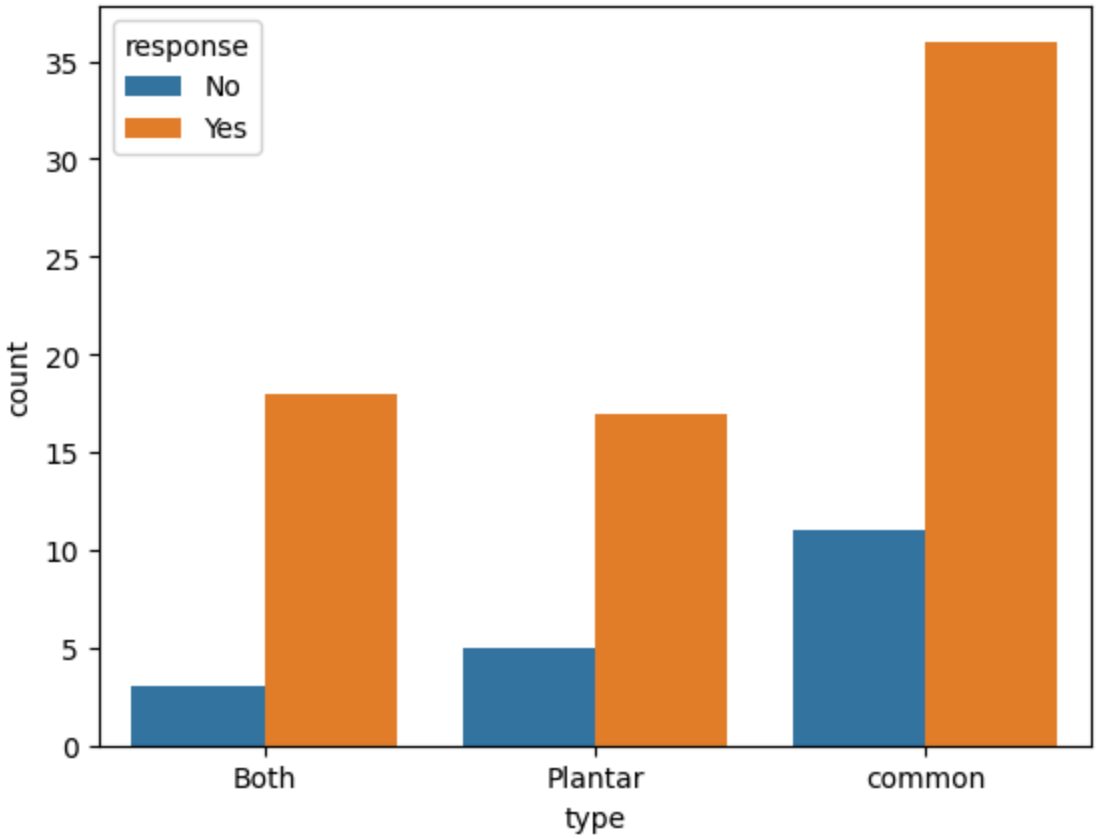
In [5]:
```
pd.crosstab(df_immuno.gender,
            df_immuno.response,
            margins=True)
```

Out[5]:

response	No	Yes	All
gender			
Female	10	39	49
Male	9	32	41
All	19	71	90

In [6]:
```
sns.countplot(data=df_immuno,
              x='type',
              hue='response')
```

Out[6]:
```
<Axes: xlabel='type', ylabel='count'>
```



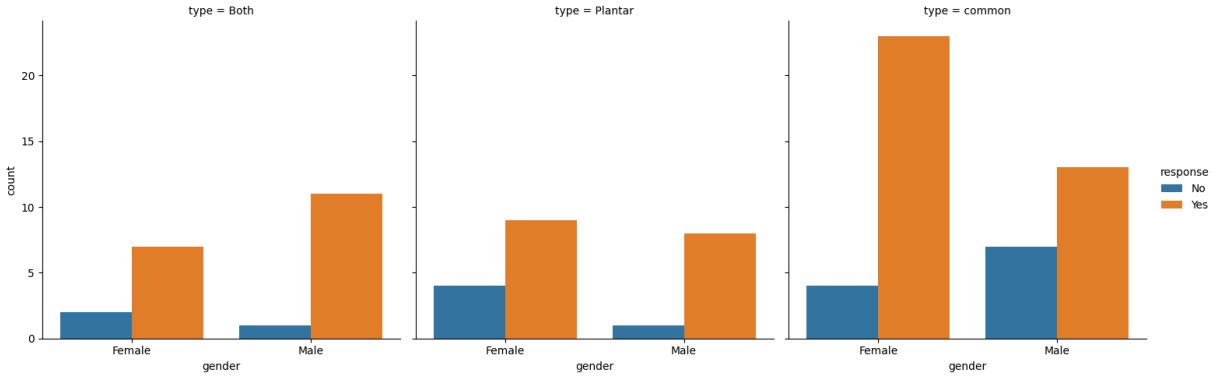
In [7]:
```
pd.crosstab(df_immuno.type,
            df_immuno.response,
            margins=True)
```

Out[7]:

response	No	Yes	All
type			
Both	3	18	21
Plantar	5	17	22
common	11	36	47
All	19	71	90

In [8]:
```
sns.catplot(data=df_immuno,
            kind='count',
            col='type',
            x='gender',
            hue='response')
```

Out[8]: <seaborn.axisgrid.FacetGrid at 0x20bfc4c5400>



Gender & Response

Create Contingency Table

To run the Chi-Square Test, the easiest way is to convert the data into a contingency table with frequencies.

In [9]:
```
contab = pd.crosstab(df_immuno.gender,
                     df_immuno.response)
contab
```

Out[9]:

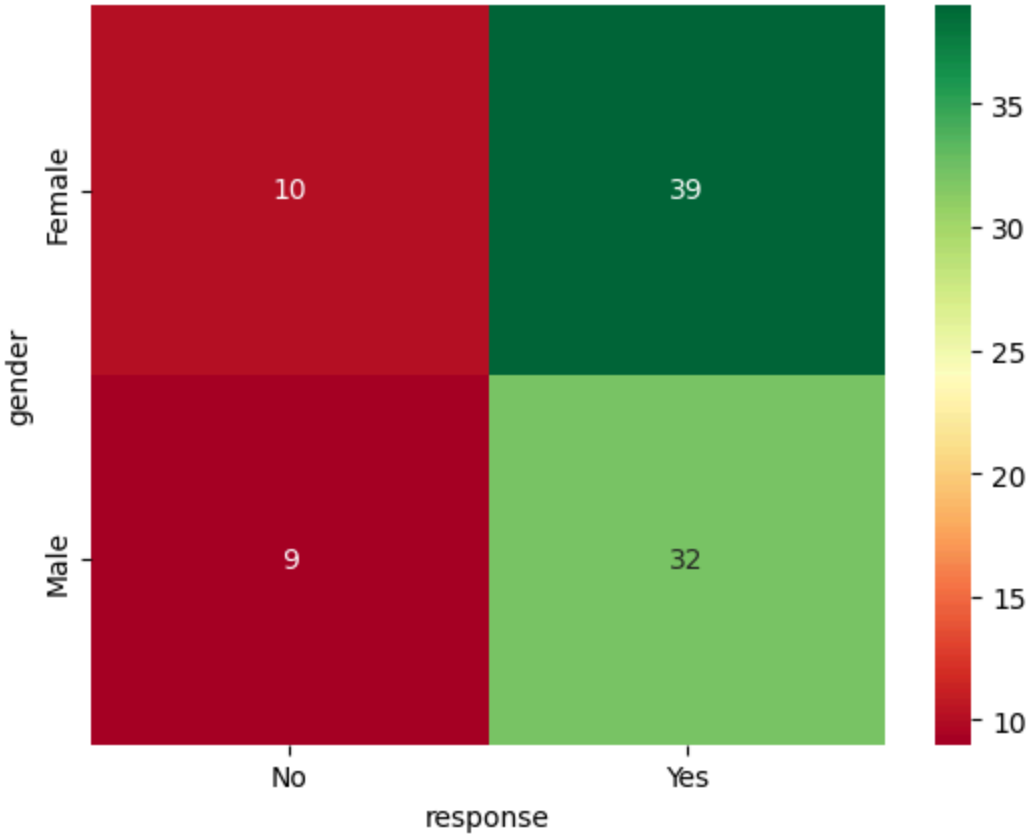
response	No	Yes
gender		
Female	10	39
Male	9	32

Visualize Contingency Table

An easy way to visualize a contingency table is to draw a heatmap.

```
In [10]: sns.heatmap(contab, annot=True, cmap='RdYlGn')
```

```
Out[10]: <Axes: xlabel='response', ylabel='gender'>
```



Chi-Square Test

Now that we have built the contingency table, we can pass it to chi2_contingency() from the scipy library which returns:

- The test statistic (c)
- The p-value of the test (p)
- Degrees of freedom (dof)
- The expected frequencies, based on the marginal sums of the table (expected)

__Hypothesis__

H_0: the gender & response have *no relationship*

H_A: there is a relationship between gender & response

In [11]:
```
c, p, dof, expected = stats.chi2_contingency(contab)
```

In [12]:
```
print("p-value:", p)
```

```
p-value: 1.0
```

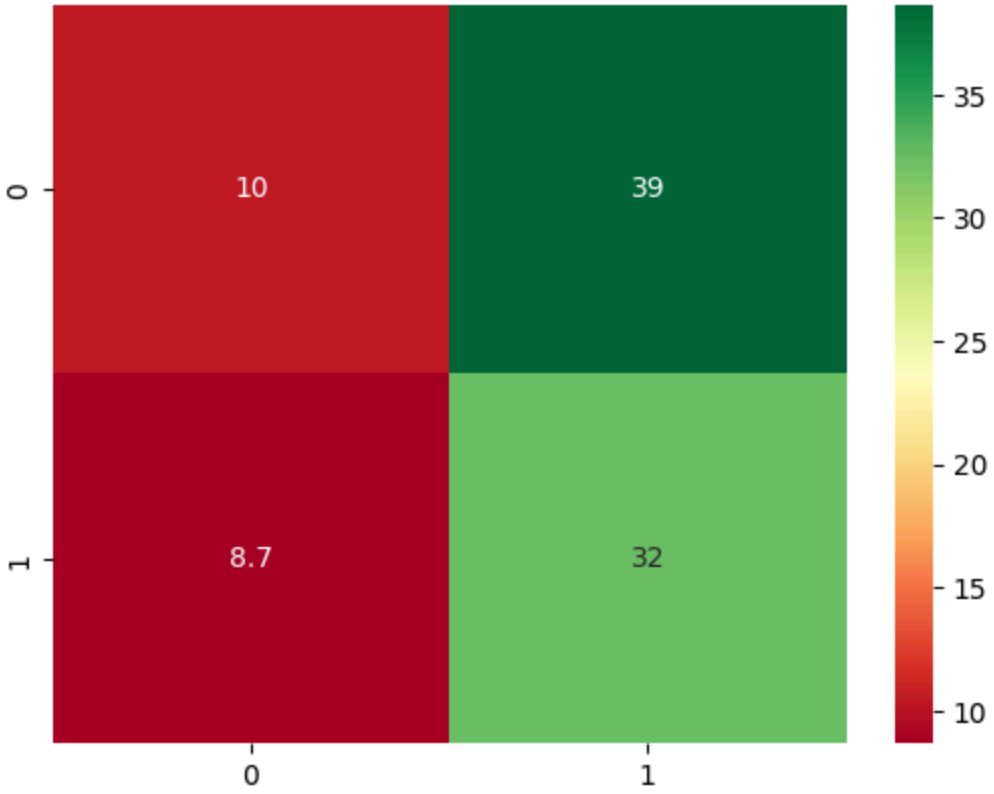
p-value > 0.05, fail to reject H_0.

The p-value is 1 which means that we do not reject the H_0 at 95% level of confidence.

Expected frequencies (no relationship):

In [13]:
```
sns.heatmap(expected, annot=True, cmap='RdYlGn')
```

Out[13]: <Axes: >



The expected value at row i and column j can be calculated using below equation.

$$\frac{\mathbf{R_i\ Total \times C_j\ Total}}{\mathbf{Grand\ Total}}$$

Let's see the contingency table again before calculating expected values.

In [14]:
```
pd.crosstab(df_immuno.gender,
            df_immuno.response,
            margins=True)
```

Out[14]:

response	No	Yes	All
gender			
Female	10	39	49
Male	9	32	41
All	19	71	90

Now calculate expected value at gender=Female, response=No

In [15]:
```
(49*19)/90
```

Out[15]:
```
10.344444444444445
```

Calculate expected value at gender=male, response=No

In [16]:
```
(41*19)/90
```

Out[16]:
```
8.655555555555555
```

Type & Response

Create Contingency Table

In [17]:
```
contab = pd.crosstab(df_immuno.type,
                     df_immuno.response)
contab
```

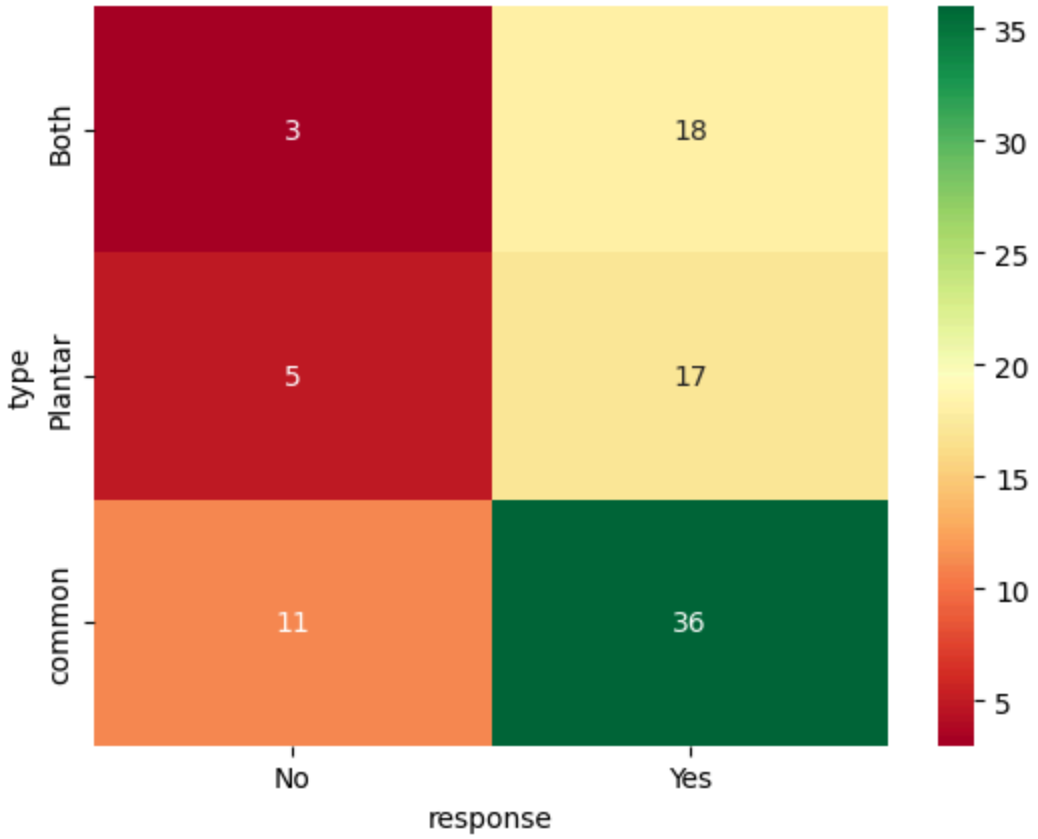
Out[17]:

response	No	Yes
type		
Both	3	18
Plantar	5	17
common	11	36

Visualize Contingency Table

In [18]:
```
sns.heatmap(contab, annot=True, cmap='RdYlGn')
```

Out[18]: <Axes: xlabel='response', ylabel='type'>



Chi-Square Test

__Hypothesis__

H_0: the type & response variables have *no relationship*

H_A: there is a relationship between type & response variables

In [19]:
```
c, p, dof, expected = stats.chi2_contingency(contab)
```

In [20]:
```
print("p-value:", p)
```

```
p-value: 0.6803408056744953
```

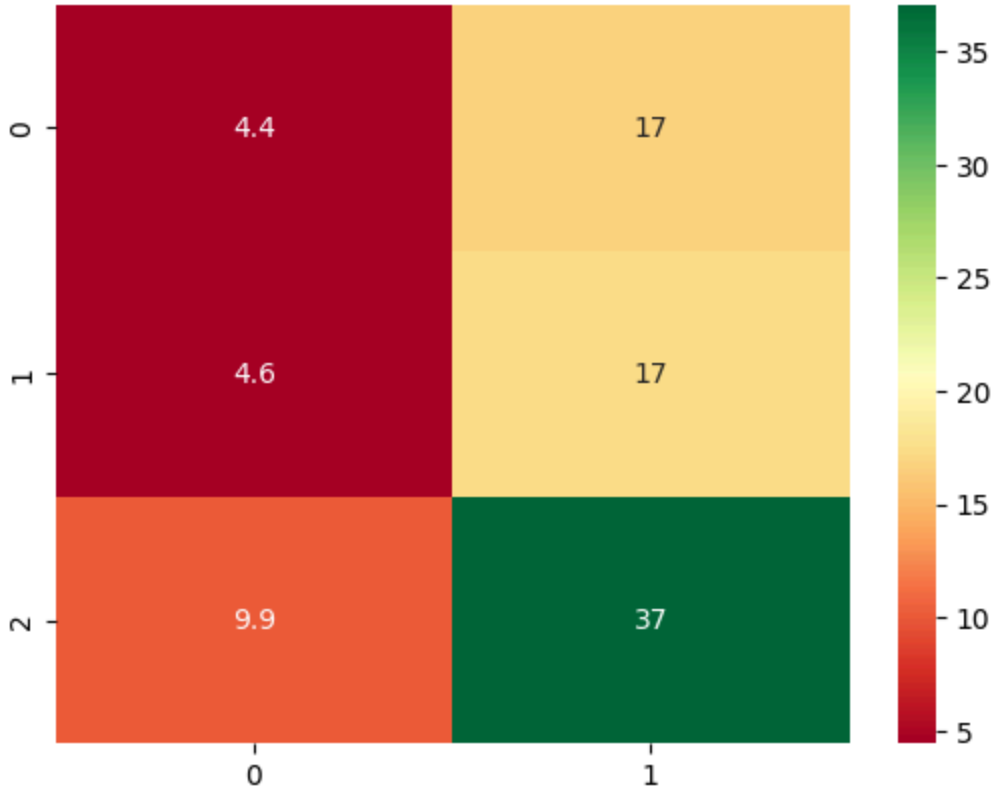
p-value > 0.05, fail to reject H_0.

The p-value is 1 which means that we do not reject the H_0 at 95% level of confidence.

Expected frequencies:

In [21]:
```
sns.heatmap(expected, annot=True, cmap='RdYlGn')
```

Out[21]: <Axes: >



Titanic Dataset

In [22]:
```
df_titan = pd.read_csv("https://raw.githubusercontent.com/ThammakornS/ProgStat/main
df_titan.head()
```

Out[22]:

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

Adjust Data Type

In [23]:

```
df_titan = df_titan.astype({
    'Survived': 'category',
    'Pclass': 'category',
    'Sex': 'category',
    'Cabin': 'category',
    'Embarked': 'category'
})
df_titan.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    category
 2   Pclass       891 non-null    category
 3   Name         891 non-null    object
 4   Sex          891 non-null    category
 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    category
 11  Embarked     889 non-null    category
dtypes: category(5), float64(2), int64(3), object(2)
memory usage: 59.8+ KB
```

Gender & Survived

Create Contingency Table

In [24]:
```
contab = pd.crosstab(df_titan.Sex,
                     df_titan.Survived,)
contab
```

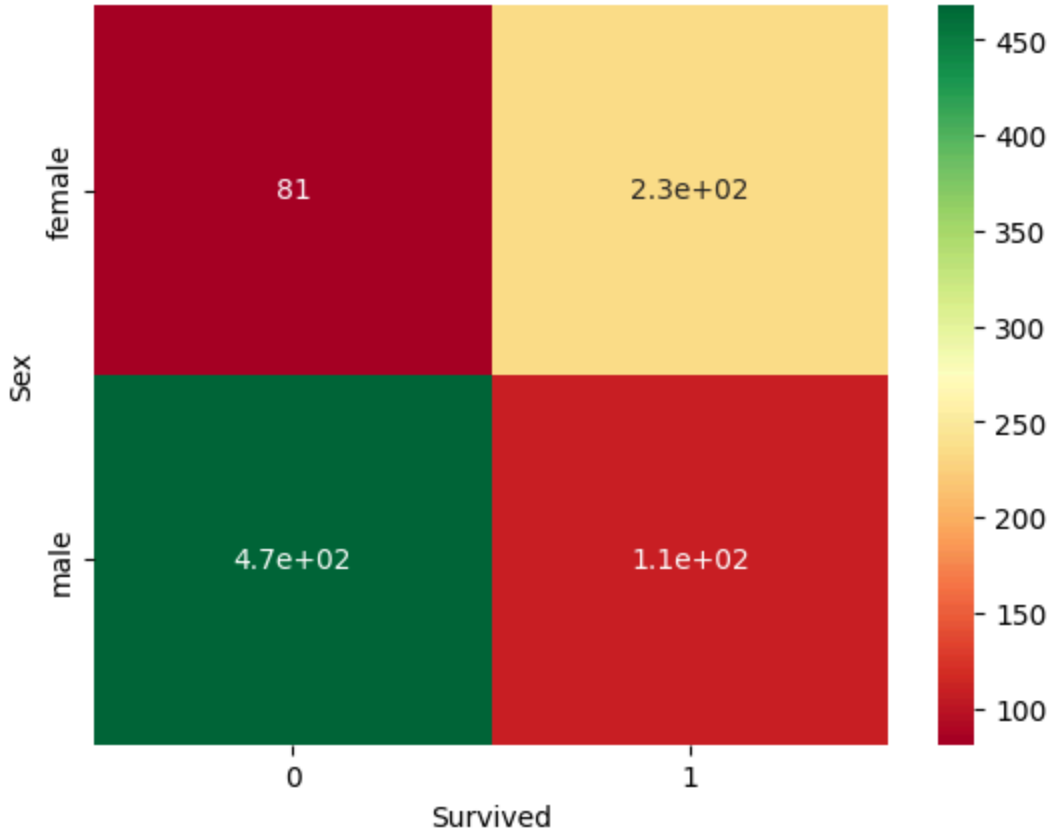
Out[24]:

Survived	0	1
Sex		
female	81	233
male	468	109

Visualize Contingency Table

In [25]:
```
sns.heatmap(contab, annot=True, cmap='RdYlGn')
```

Out[25]:
```
<Axes: xlabel='Survived', ylabel='Sex'>
```



Chi-Square Test

__Hypothesis__

H_0: the gender & survived variables have *no relationship*

H_A: there is a relationship between gender & survived variables

In [26]:
```
c, p, dof, expected = stats.chi2_contingency(contab)
```

In [27]:
```
print("p-value:", p)
```

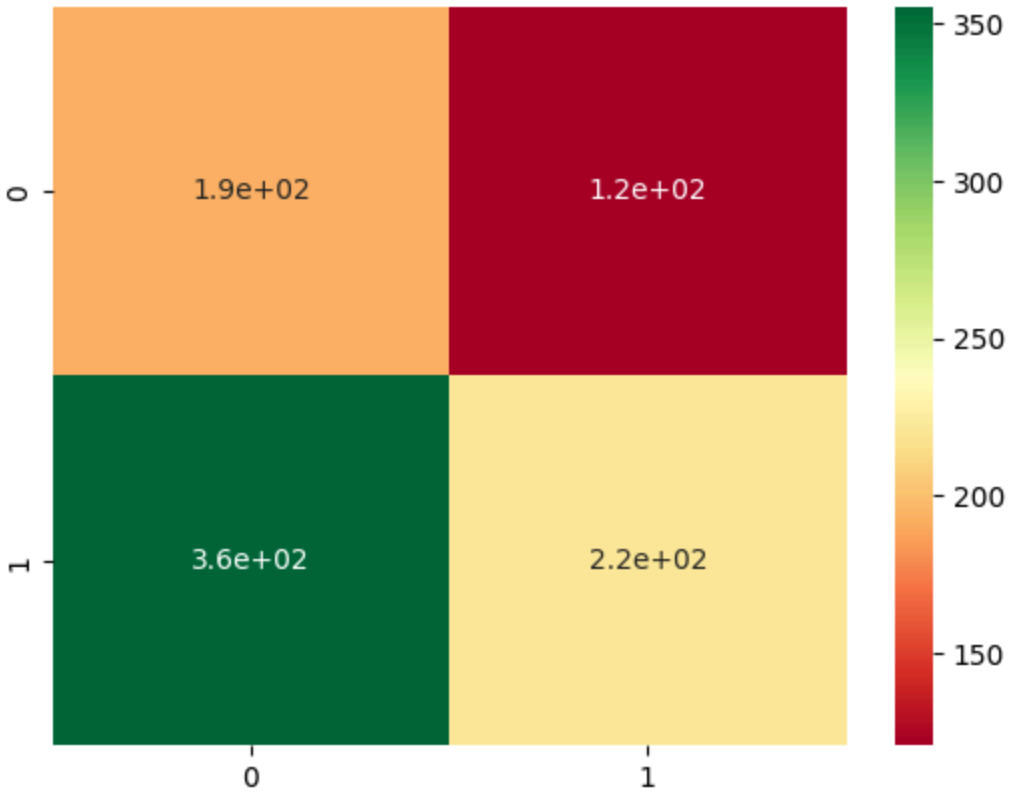
```
p-value: 1.1973570627755645e-58
```

p-value <= 0.05, reject H_0, accept H_A.

Expected frequencies:

In [28]:
```
sns.heatmap(expected, annot=True, cmap='RdYlGn')
```

Out[28]:
```
<Axes: >
```



Pclass & Survived

Create Contingency Table

In [29]:
```
contab = pd.crosstab(df_titan.Pclass,
                     df_titan.Survived)
contab
```

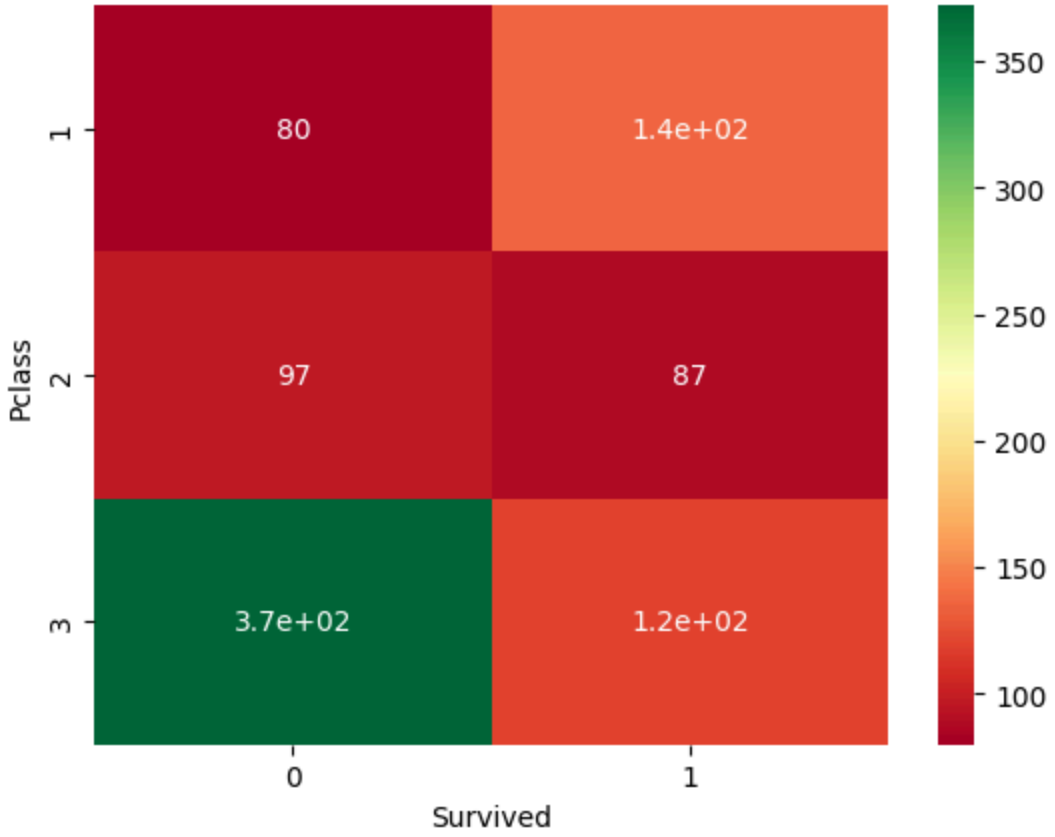
Out[29]:

Survived	0	1
Pclass		
1	80	136
2	97	87
3	372	119

Visualize Contingency Table

In [30]:
```
sns.heatmap(contab, annot=True, cmap='RdYlGn')
```

Out[30]:
```
<Axes: xlabel='Survived', ylabel='Pclass'>
```



Chi-Square Test

__Hypothesis__

H_0: the pclass & survived variables have *no relationship*

H_A: there is a relationship between pclass & survived variables

In [31]:
```
c, p, dof, expected = stats.chi2_contingency(contab)
```

In [32]:
```
print("p-value:", p)
```

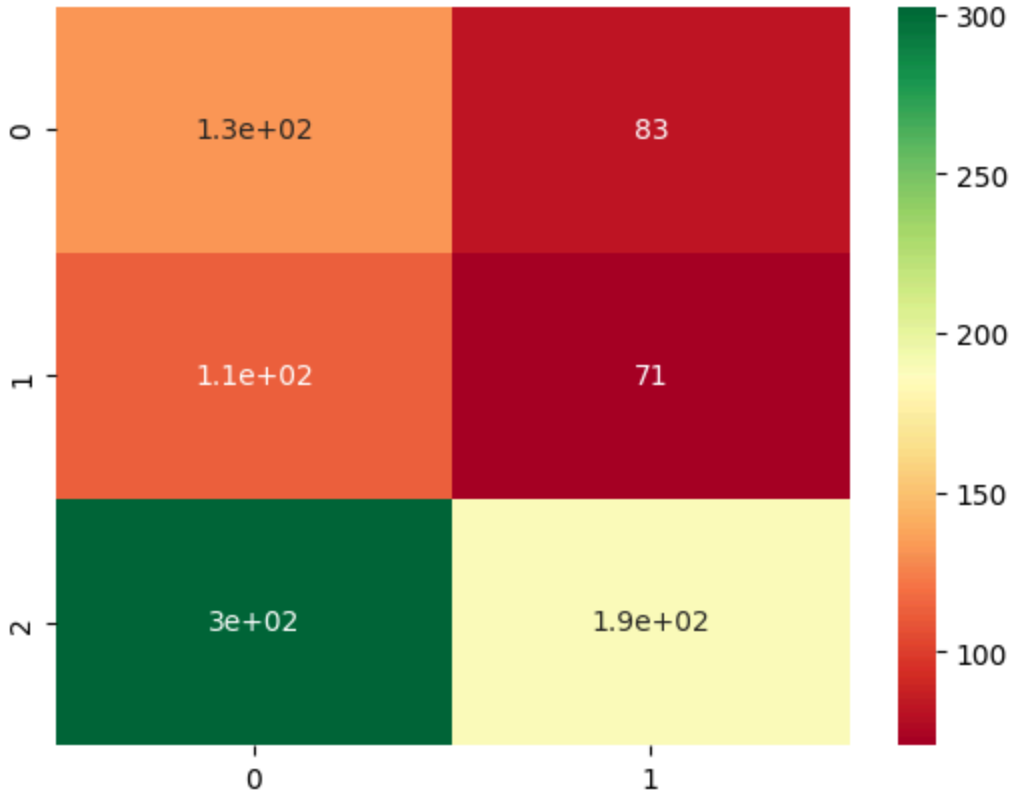
```
p-value: 4.549251711298793e-23
```

p-value <= 0.05, reject H_0, accept H_A.

Expected frequencies:

In [33]:
```
sns.heatmap(expected, annot=True, cmap='RdYlGn')
```

Out[33]:
```
<Axes: >
```



In [ ]: