Case Study for Statistical Methods

In [1]:
```
# import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
import statsmodels.api as sm
from statsmodels.formula.api import ols
import statsmodels.multivariate.manova as manova
```

Chi-Squared Test

Assume the expected distribution is 50% vegetarian and 50% non-vegetarian. You want to test if your sample data matches this distribution. Goodness of fit.

In [2]:
```
import numpy as np
from scipy.stats import chisquare

# Hypothetical observed frequencies
observed_frequencies = np.array([30, 70])  # e.g., 30 vegetarian and 70 non-vegetarian meals

# Expected frequencies (assuming a 50-50 distribution)
total_meals = observed_frequencies.sum()
expected_frequencies = np.array([total_meals * 0.5, total_meals * 0.5])

# Perform the Chi-Squared test
chi2_stat, p_value = chisquare(observed_frequencies, f_exp=expected_frequencies)

# Output the results
print(f"Chi-Squared Statistic: {chi2_stat}")
print(f"P-value: {p_value}")

# print results with if else conditions
if p_value < 0.05:
```

```
    print("Reject null hypothesis, there is a significant difference between the groups")
else:
    print("Fail to reject null hypothesis, there is no significant difference between the groups ")
```

```
Chi-Squared Statistic: 16.0
P-value: 6.334248366623988e-05
Reject null hypothesis, there is a significant difference between the groups
```

In [3]:
```
import pandas as pd
import numpy as np
import seaborn as sns
from scipy.stats import chi2_contingency

# data load
df= sns.load_dataset('tips')
df.head()
```

Out[3]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

In [4]:
```
# creata a contigency table
contigency_table = pd.crosstab(df['sex'], df['day'])
contigency_table
```

Out[4]:

day	Thur	Fri	Sat	Sun
sex				
Male	30	10	59	58
Female	32	9	28	18

In [5]:
```
# perform the chi-squared test
chi2, p, dof, expected = chi2_contingency(contigency_table)

# output results
print(f"Chi-Squared Statistic: {chi2}")
print(f"P-value: {p}")
print(f"Degrees of Freedom: {dof}")
print("Expected: \n", np.array(expected))

# print the results based on if else conditions
if p < 0.05:
    print("Reject null hypothesis, there is a significant difference between the groups")
else:
    print("Fail to reject null hypothesis, there is no significant difference between the groups ")
```

```
Chi-Squared Statistic: 13.22200137240661
P-value: 0.004180302092822257
Degrees of Freedom: 3
Expected:
 [[39.89344262 12.22540984 55.9795082  48.90163934]
 [22.10655738  6.77459016 31.0204918  27.09836066]]
Reject null hypothesis, there is a significant difference between the groups
```

---

t-test

1. one sample t-test

In [6]:
```
# one sample t-test
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats

# data load
```

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

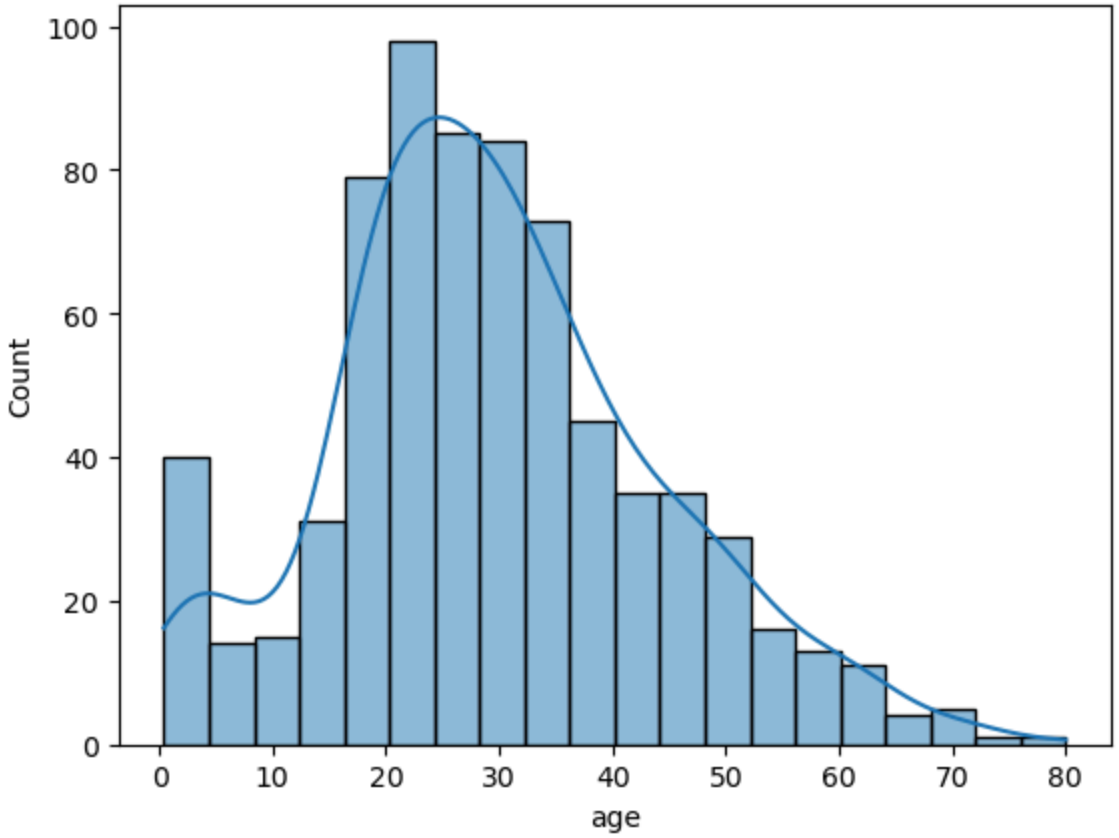
Out[6]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True

In [7]:
```
sns.histplot(df['age'], kde=True)
```

```
C:\Users\ustb\.anaconda\anwaar\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[7]:
```
<Axes: xlabel='age', ylabel='Count'>
```



In [8]:
```
# impute null values
df['age'] = df['age'].fillna(df['age'].median())
df['fare'] = df['fare'].fillna(df['fare'].median())
```

In [9]:
```
# shpiro wilk test to see the normality of the data
statistic, p = stats.shapiro(df['age'])
# print the results based on p value and if else conditions
if p < 0.05:
    print("Reject null hypothesis, data is not normally distributed")
else:
    print("Fail to reject null hypothesis, data is normally distributed")
```

```
Reject null hypothesis, data is not normally distributed
```

Reject null hypothesis, data is not normally distributed

mu = 35

is there any difference between ages of people and mu?

In [10]:
```
df['age'].mean()
```

Out[10]: 29.36158249158249

In [11]:
```
statistic, p = stats.ttest_1samp(df['age'], 35)

# print the results based on p value and if else conditions
if p < 0.05:
    print("Reject null hypothesis, there is a significant difference between the groups")
else:
    print("Fail to reject null hypothesis, there is no significant difference between the groups ")
```

```
Reject null hypothesis, there is a significant difference between the groups
```

2. Two sample independent t-test

In [13]:
```
# lets run the two sample t test on ages of male and female from titanic

age_male = df[df['sex'] == 'male']['age']
age_female = df[df['sex'] == 'female']['age']

statistic, p = stats.ttest_ind(age_male, age_female)

# print the results based on p value and if else conditions
if p < 0.05:
    print(f"Reject null hypothesis, because p vaule: {p} ,there is a significant difference between ages of male and
else:
    print(f"Fail to reject null hypothesis,because p vaule: {p}, there is not any differences between ages of female
```

```
Reject null hypothesis, because p vaule: 0.015381536989875313 ,there is a significant difference between ages of male
and female passengers of titanic.
```

ANOVA

In [14]:
```
df.head()
```

Out[14]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	Fals
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	Fals
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	Tru
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	Fals
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	Tru

In [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          891 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
```

one-way ANOVA

1. There is a difference between ages of First, second and third class passengers?

H0: There is no difference.

H1: There is a significant Difference.

In [16]:
```
df[['age', 'pclass']].head()
```

Out[16]:

	age	pclass
0	22.0	3
1	38.0	1
2	26.0	3
3	35.0	1
4	35.0	3

In [17]:
```
# stats models
import statsmodels.api as sm
from statsmodels.formula.api import ols

# ANOVa test for age and class
model = ols("age ~ pclass", data=df).fit()
anova_table = sm.stats.anova_lm(model, typ=2)
print(anova_table)

# print the results based on p value and if else conditions
if anova_table['PR(>F)'][0] < 0.05:
    print(f"Reject null hypothesis, because p vaule: {anova_table['PR(>F)'][0]} , there is a significant difference b
else:
    print(f"Fail to reject null hypothesis,because p vaule: {anova_table['PR(>F)'][0]}, there is not any differences
```

```
                sum_sq     df           F        PR(>F)
pclass     17429.695048    1.0  116.122704  1.558883e-25
Residual  133436.428421  889.0         NaN           NaN
Reject null hypothesis, because p vaule: 1.5588829926632724e-25 , there is a significant difference between ages of p
assengers in different classes.
C:\Users\ustb\AppData\Local\Temp\ipykernel_11392\2670501721.py:11: FutureWarning: Series.__getitem__ treating keys as
positions is deprecated. In a future version, integer keys will always be treated as labels (consistent with DataFram
e behavior). To access a value by position, use `ser.iloc[pos]`
  if anova_table['PR(>F)'][0] < 0.05:
C:\Users\ustb\AppData\Local\Temp\ipykernel_11392\2670501721.py:12: FutureWarning: Series.__getitem__ treating keys as
positions is deprecated. In a future version, integer keys will always be treated as labels (consistent with DataFram
e behavior). To access a value by position, use `ser.iloc[pos]`
  print(f"Reject null hypothesis, because p vaule: {anova_table['PR(>F)'][0]} , there is a significant difference bet
ween ages of passengers in different classes.")
```

PostHoc Test

In [18]:
```
# Tukey HSD pairwise comparison
from statsmodels.stats.multicomp import pairwise_tukeyhsd

tukey = pairwise_tukeyhsd(endog=df['age'], groups=df['pclass'], alpha=0.05)
tukey.summary()
```

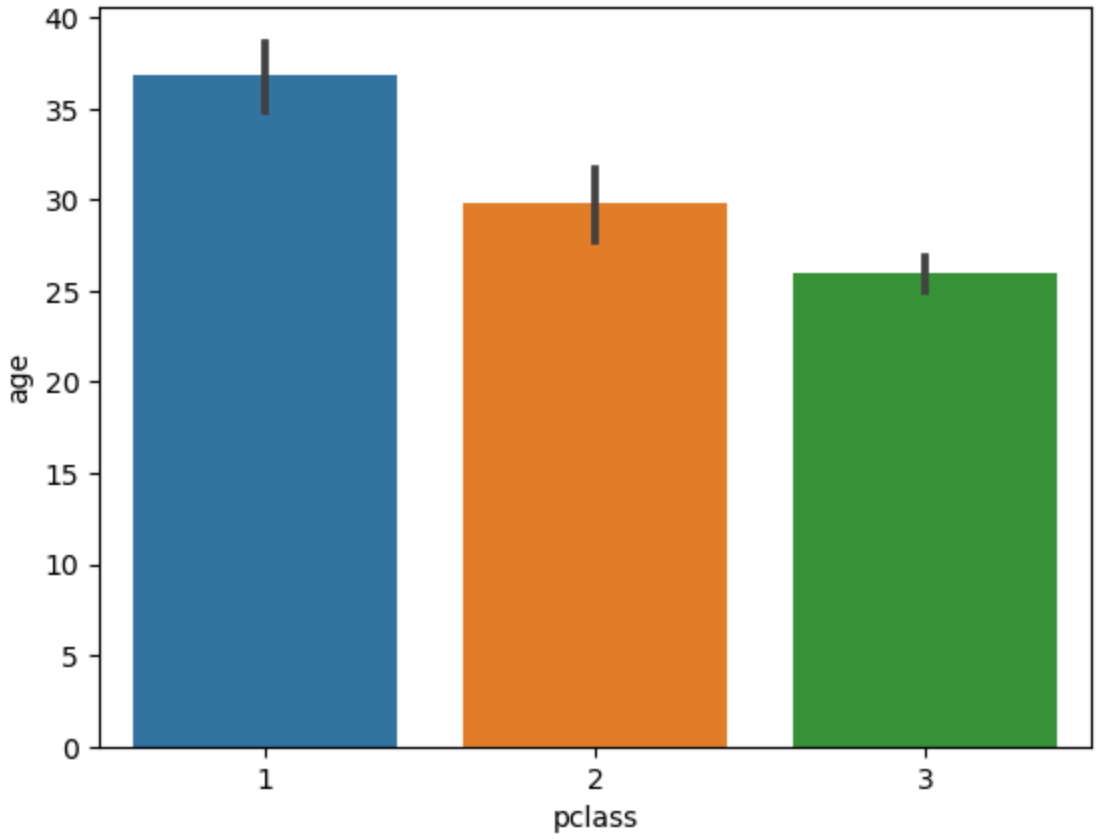
Out[18]:

Multiple Comparison of Means - Tukey HSD, FWER=0.05

group1	group2	meandiff	p-adj	lower	upper	reject
1	2	-7.0467	0.0	-9.9299	-4.1636	True
1	3	-10.8795	0.0	-13.226	-8.533	True
2	3	-3.8328	0.0009	-6.3169	-1.3486	True

In [19]:
```
sns.barplot(x='pclass', y='age', data=df)
```

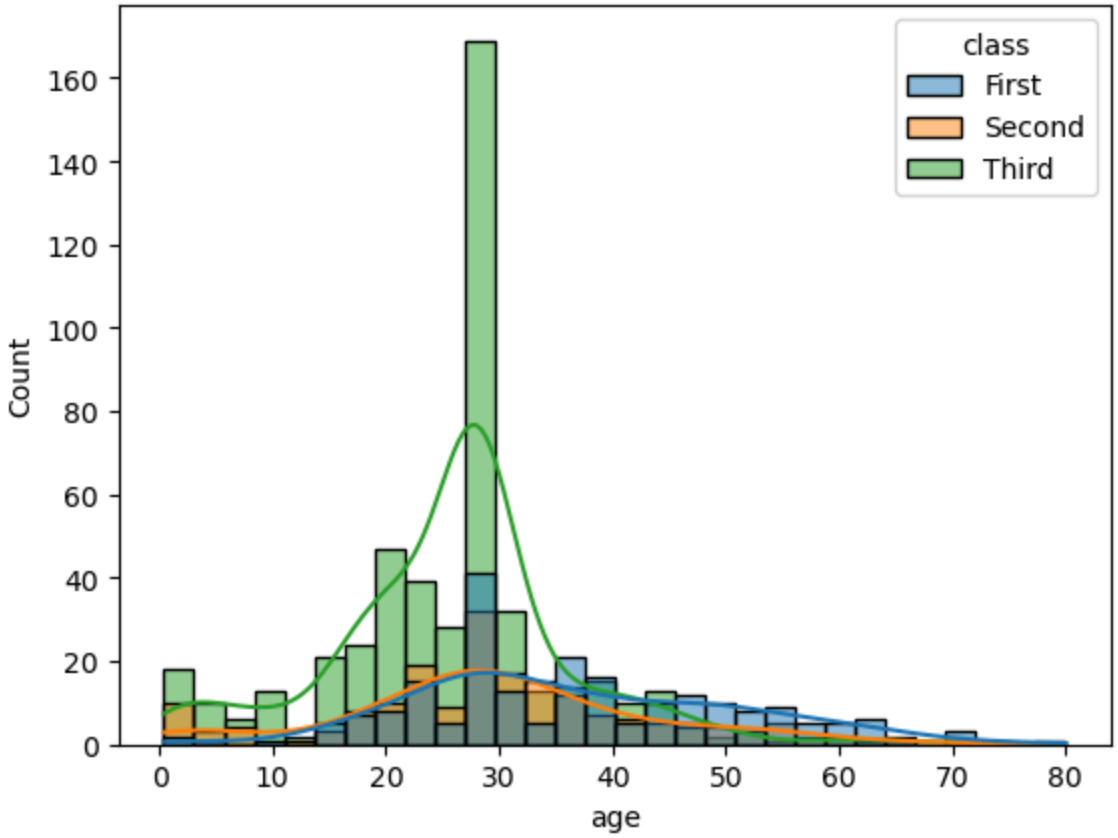
Out[19]:
```
<Axes: xlabel='pclass', ylabel='age'>
```



In [20]:
```
# hist plot with pclass grouping
sns.histplot(df, x='age', hue='class', kde=True)
```

```
C:\Users\ustb\.anaconda\anwaar\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\ustb\.anaconda\anwaar\Lib\site-packages\seaborn\_oldcore.py:1057: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.
  grouped_data = data.groupby(
C:\Users\ustb\.anaconda\anwaar\Lib\site-packages\seaborn\_oldcore.py:1075: FutureWarning: When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.
  data_subset = grouped_data.get_group(pd_key)
C:\Users\ustb\.anaconda\anwaar\Lib\site-packages\seaborn\_oldcore.py:1075: FutureWarning: When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.
  data_subset = grouped_data.get_group(pd_key)
C:\Users\ustb\.anaconda\anwaar\Lib\site-packages\seaborn\_oldcore.py:1075: FutureWarning: When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.
  data_subset = grouped_data.get_group(pd_key)
```

```
Out[20]: <Axes: xlabel='age', ylabel='Count'>
```



TWO Way ANOVA

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

Out[21]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True

In [22]:
```
# stats models
import statsmodels.api as sm
from statsmodels.formula.api import ols

# ANOVa test for age and class
# model = ols("age ~ pclass + who + pclass:who", data=df).fit()
# or this
model = ols("age ~ sex * who", data=df).fit()
anova_table = sm.stats.anova_lm(model, typ=2)
print(anova_table)
```

```
                sum_sq     df             F        PR(>F)
sex                NaN    1.0           NaN           NaN
who       3.314573e-10    2.0  1.436751e-12  9.999990e-01
sex:who   5.233208e+04    2.0  2.268412e+02  8.119221e-46
Residual  1.023151e+05  887.0           NaN           NaN
```

```
C:\Users\ustb\.anaconda\anwaar\Lib\site-packages\statsmodels\base\model.py:1888: ValueWarning: covariance of constrai
nts does not have full rank. The number of constraints is 1, but rank is 0
  warnings.warn('covariance of constraints does not have full '
C:\Users\ustb\.anaconda\anwaar\Lib\site-packages\statsmodels\base\model.py:1917: RuntimeWarning: invalid value encoun
tered in divide
  F /= J
C:\Users\ustb\.anaconda\anwaar\Lib\site-packages\statsmodels\base\model.py:1888: ValueWarning: covariance of constrai
nts does not have full rank. The number of constraints is 2, but rank is 1
  warnings.warn('covariance of constraints does not have full '
C:\Users\ustb\.anaconda\anwaar\Lib\site-packages\statsmodels\base\model.py:1888: ValueWarning: covariance of constrai
nts does not have full rank. The number of constraints is 2, but rank is 1
  warnings.warn('covariance of constraints does not have full '
```

In [23]:
```
# Tukey HSD pairwise comparison
from statsmodels.stats.multicomp import pairwise_tukeyhsd

tukey = pairwise_tukeyhsd(df['age'], df['sex'] + df['who'] , alpha=0.05)
tukey.summary()
```

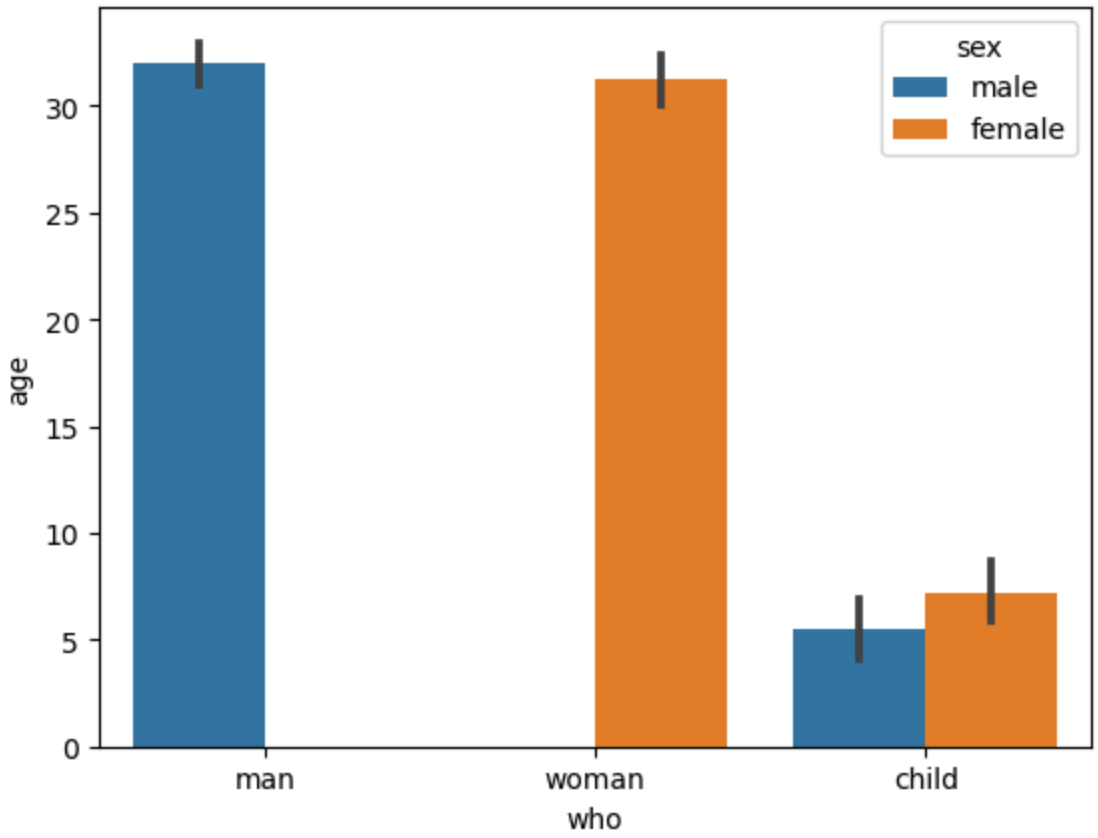
Out[23]:

Multiple Comparison of Means - Tukey HSD, FWER=0.05

group1	group2	meandiff	p-adj	lower	upper	reject
femalechild	femalewoman	24.0084	0.0	19.4706	28.5462	True
femalechild	malechild	-1.7426	0.8815	-7.8152	4.3301	False
femalechild	maleman	24.7693	0.0	20.3881	29.1505	True
femalewoman	malechild	-25.751	0.0	-30.4333	-21.0686	True
femalewoman	maleman	0.7609	0.7773	-1.299	2.8207	False
malechild	maleman	26.5118	0.0	21.9811	31.0426	True

In [24]:
```
sns.barplot(df, x= 'who', y = 'age', hue='sex')
```

Out[24]:
```
<Axes: xlabel='who', ylabel='age'>
```



Correlation

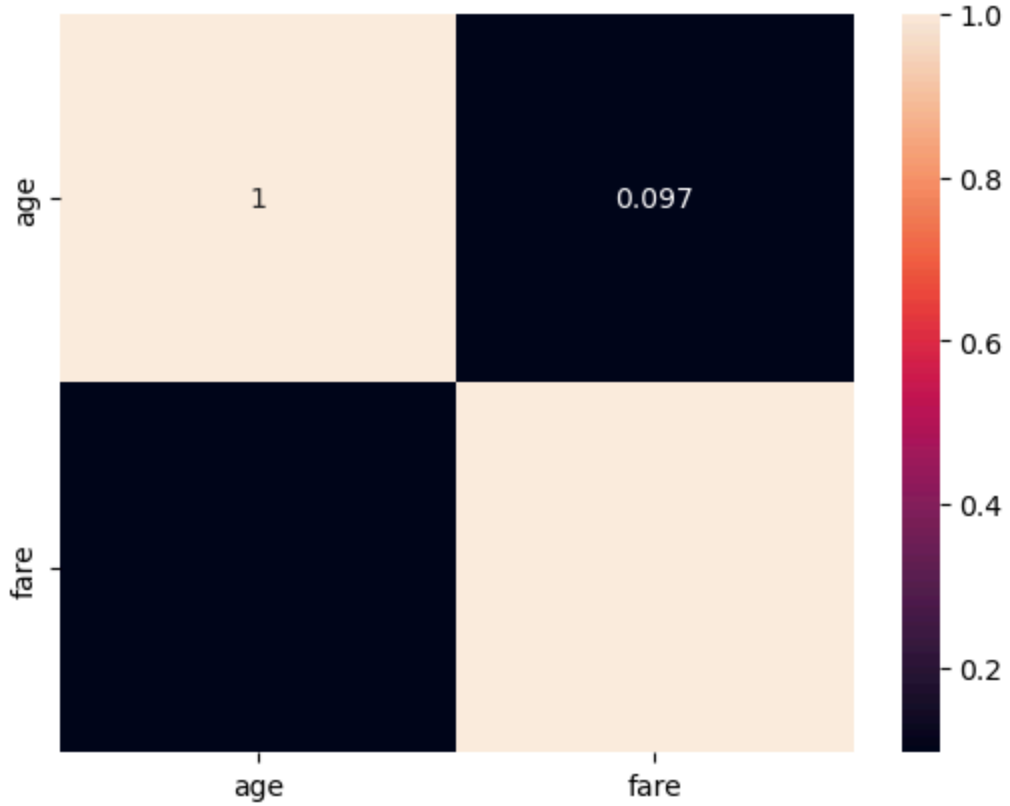
In [25]:
```
df.head()
```

Out[25]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	Fals
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	Fals
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	Tru
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	Fals
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	Tru

In [26]:
```
# creata a heatmap of correlation
sns.heatmap(df[['age', 'fare']].corr(), annot=True)
```

Out[26]: <Axes: >



In [27]:
```
# tips and correlation
tips = sns.load_dataset('tips')
tips.head()
```

Out[27]:

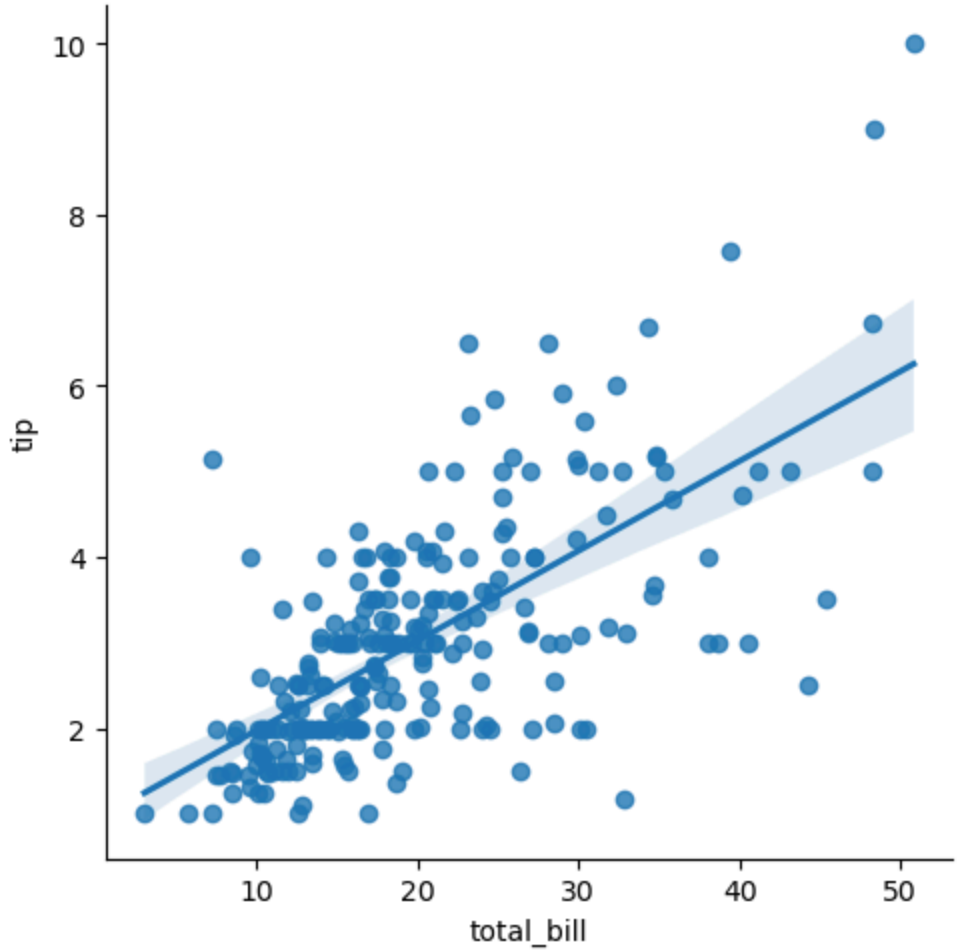
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

In [28]:
```
tips['total_bill'].corr(tips['tip'])
```

Out[28]:
```
0.6757341092113641
```

In [29]:
```
sns.lmplot(x='total_bill', y='tip', data=tips)
```

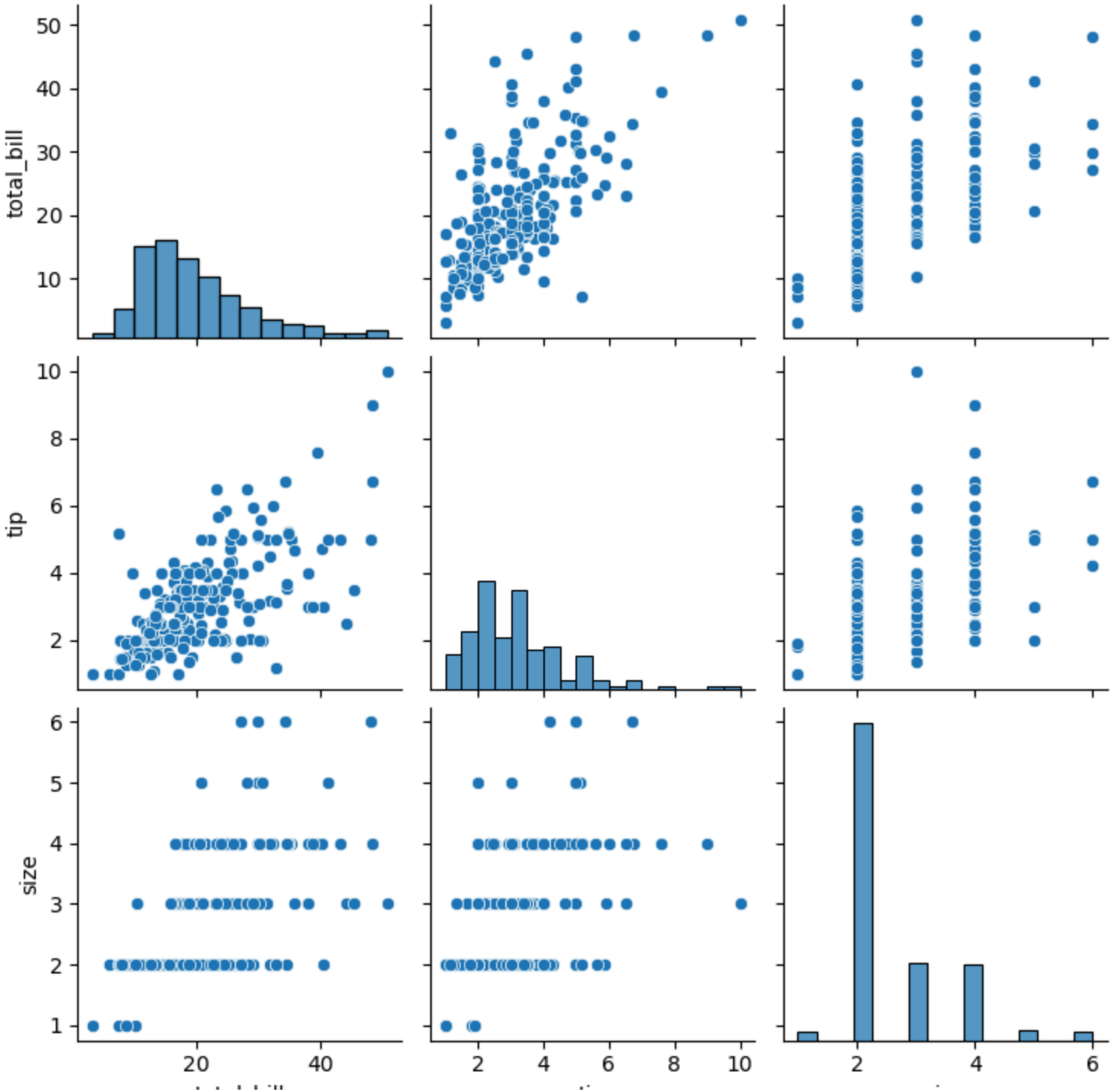
Out[29]:
```
<seaborn.axisgrid.FacetGrid at 0x188b0ba3850>
```



In [30]:
```
sns.pairplot(tips)
```

```
C:\Users\ustb\.anaconda\anwaar\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is dep
recated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\ustb\.anaconda\anwaar\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is dep
recated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\ustb\.anaconda\anwaar\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is dep
recated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

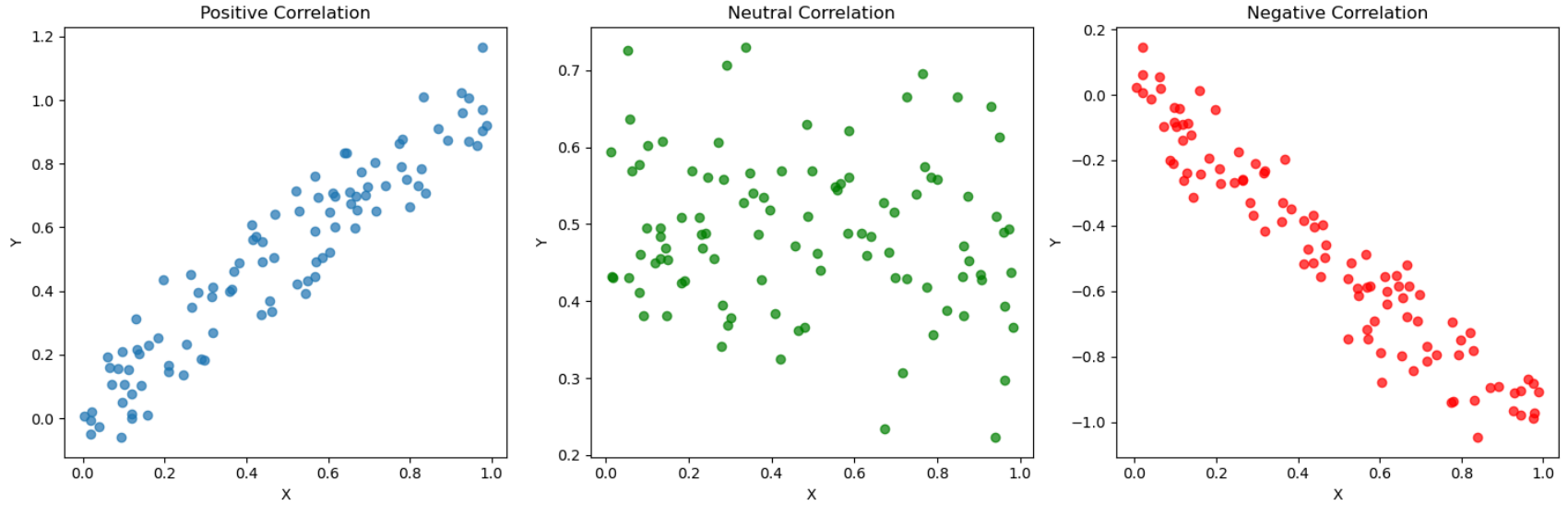
Out[30]:
```
<seaborn.axisgrid.PairGrid at 0x188b0c56210>
```


total_bill
tip
size
50
40
30
20
10
10
8
6
4
2
6
5
4
3
2
1
20
40
2
4
6
8
10
2
4
6

total_bill tip size

In [31]:
```
import numpy as np
import matplotlib.pyplot as plt

# Set random seed for reproducibility
np.random.seed(0)

# Generate synthetic data
# Positive correlation
x = np.random.rand(100)
y_pos = x + np.random.normal(0, 0.1, 100)

# Neutral correlation
x_neutral = np.random.rand(100)
y_neutral = np.random.normal(0.5, 0.1, 100)

# Negative correlation
y_neg = -x + np.random.normal(0, 0.1, 100)

# Plotting the correlations
plt.figure(figsize=(15, 5))

# Positive Correlation
plt.subplot(1, 3, 1)
plt.scatter(x, y_pos, alpha=0.7)
plt.title('Positive Correlation')
plt.xlabel('X')
plt.ylabel('Y')

# Neutral Correlation
plt.subplot(1, 3, 2)
plt.scatter(x_neutral, y_neutral, alpha=0.7, color='green')
plt.title('Neutral Correlation')
plt.xlabel('X')
plt.ylabel('Y')

# Negative Correlation
plt.subplot(1, 3, 3)
plt.scatter(x, y_neg, alpha=0.7, color='red')
plt.title('Negative Correlation')
```

```
plt.xlabel('X')
plt.ylabel('Y')

plt.tight_layout()
plt.show()
```



In [ ]: