In [ ]:
```
import pandas as pd
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.feature_selection import chi2
```

In [ ]:
```
df  = pd.read_csv("survey.csv",index_col=0, na_values=["NaN",])
df.dropna(axis=0,inplace=True)
df.describe()
```

Out[ ]:

	Age
count	86.000000
mean	34.662791
std	8.676690
min	-1.000000
25%	30.000000
50%	35.000000
75%	39.750000
max	56.000000

In [ ]:
```
working_data = df[["Age","Gender","family_history","remote_work","treatment"]]
working_data.is_copy = False
working_data["Gender"] = working_data["Gender"].map({
    "male":0,
    "Male":0,
    "M":0,
    "female":1,
    "Female":1,
    "F":1,
})
working_data["Gender"] = working_data["Gender"]
working_data["family_history"] = working_data["family_history"].map({
    "Yes":1,
    "No":0,
})
working_data["remote_work"] = working_data["remote_work"].map({
    "Yes":1,
    "No":0,
})
working_data["treatment"] = working_data["treatment"].map({
    "Yes":1,
    "No":0,
})
working_data.dropna(axis=0,inplace=True)
working_data.reset_index()
working_data.describe()
```

```
C:\Users\rinko\AppData\Local\Temp/ipykernel_6880/3301706824.py:3: SettingWithCopyWarn
ing:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  working_data["Gender"] = working_data["Gender"].map({
C:\Users\rinko\AppData\Local\Temp/ipykernel_6880/3301706824.py:11: SettingWithCopyWar
ning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  working_data["Gender"] = working_data["Gender"]
C:\Users\rinko\AppData\Local\Temp/ipykernel_6880/3301706824.py:12: SettingWithCopyWar
ning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  working_data["family_history"] = working_data["family_history"].map({
C:\Users\rinko\AppData\Local\Temp/ipykernel_6880/3301706824.py:16: SettingWithCopyWar
ning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  working_data["remote_work"] = working_data["remote_work"].map({
C:\Users\rinko\AppData\Local\Temp/ipykernel_6880/3301706824.py:20: SettingWithCopyWar
ning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  working_data["treatment"] = working_data["treatment"].map({
C:\Users\rinko\AppData\Local\Temp/ipykernel_6880/3301706824.py:24: SettingWithCopyWar
ning:
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
  working_data.dropna(axis=0,inplace=True)
```

Out[ ]:

	Age	Gender	family_history	remote_work	treatment
count	74.000000	74.000000	74.000000	74.000000	74.000000
mean	35.324324	0.283784	0.554054	0.391892	0.756757
std	7.443133	0.453911	0.500463	0.491505	0.431969
min	21.000000	0.000000	0.000000	0.000000	0.000000
25%	30.000000	0.000000	0.000000	0.000000	1.000000
50%	35.000000	0.000000	1.000000	0.000000	1.000000
75%	40.000000	1.000000	1.000000	1.000000	1.000000
max	56.000000	1.000000	1.000000	1.000000	1.000000

In [ ]:
```
X = np.array(working_data["Age"]).reshape(-1, 1)
y = np.array(working_data["treatment"]).reshape(-1, 1)
```

Hypothesis Testing with chi2

Ho: Age & treatment have no association

H1: Age & treatment are associated

In [ ]:
```
chi2_val , p_val =chi2(X,y)
chi2_val,p_val
```

Out[ ]:
```
(array([4.0531995]), array([0.04408771]))
```

In [ ]:
```
if p_val[0] < 0.05:
    print("Null Hypothesis Rejected")
else:
    print("Null Hypothesis Accepted")
```

```
Null Hypothesis Rejected
```

In [ ]: