In [29]:

```
import pandas as pd
import numpy as np
import scipy.stats as stats
data = pd.read_csv('FowlerModule04-2.csv', usecols=['Homeowner', 'Credit Sco
homeowner = data[data['Homeowner'] == 'Y']
non_homeowner = data[data['Homeowner'] == 'N']
print(data.head())
print(homeowner.head())
print(non_homeowner.head())
```

```
  Homeowner  Credit Score
0         Y           725
1         Y           573
2         Y           677
3         N           625
4         N           527
  Homeowner  Credit Score
0         Y           725
1         Y           573
2         Y           677
5         Y           795
8         Y           591
   Homeowner  Credit Score
3          N           625
4          N           527
6          N           733
7          N           620
13         N           620
```

In [30]:

```
# Calculate the mean and standard error of the mean (SEM) for homeowners
mean_homeowner = homeowner['Credit Score'].mean()
sem_homeowner = stats.sem(homeowner['Credit Score'])

# Calculate the confidence interval
confidence_interval_homeowner = stats.t.interval(0.95, len(homeowner['Credit
print("Confidence interval for homeowners' credit score:", confidence_interv
```

```
Confidence interval for homeowners' credit score: (np.float64(660.7745695482
795), np.float64(724.5587637850538))
```

In [31]:

```
# Calculate the mean and standard error of the mean (SEM) for non-homeowners
mean_non_homeowner = non_homeowner['Credit Score'].mean()
sem_non_homeowner = stats.sem(non_homeowner['Credit Score'])

# Calculate the confidence interval
confidence_interval_non_homeowner = stats.t.interval(0.95, len(non_homeowner
print("Confidence interval for non-homeowners' credit score:", confidence_ir
```

```
Confidence interval for non-homeowners' credit score: (np.float64(571.323716
5728597), np.float64(638.4154138619228))
```

In [32]:

```
# Calculate the prediction interval for homeowners
n_homeowner = len(homeowner['Credit Score'])
t_value_homeowner = stats.t.ppf(0.975, n_homeowner - 1)
margin_of_error_homeowner = t_value_homeowner * sem_homeowner * np.sqrt(1 +
prediction_interval_homeowner = (mean_homeowner - margin_of_error_homeowner,
print("Prediction interval for homeowners' credit score:", prediction_interv
```

```
Prediction interval for homeowners' credit score: (np.float64(660.1893446409
127), np.float64(725.1439886924205))
```

In [33]:

```
# Calculate the prediction interval for non-homeowners
n_non_homeowner = len(non_homeowner['Credit Score'])
t_value_non_homeowner = stats.t.ppf(0.975, n_non_homeowner - 1)
margin_of_error_non_homeowner = t_value_non_homeowner * sem_non_homeowner *

prediction_interval_non_homeowner = (mean_non_homeowner - margin_of_error_nc
print("prediction interval for non-homeowners: ", prediction_interval_non_hc
```

```
prediction interval for non-homeowners:  (np.float64(570.6022179311888), np.
float64(639.1369125035937))
```