Lending club hypothesis testing

The business problem is to analyze Lending Club loans data to test hypotheses regarding the relationships between interest rates and loan amounts, loan length and interest rates, interest rates and loan purposes, and the relationship between FICO scores and home ownership.

In [42]:
```
#Importing Necessary Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
import statsmodels
from statsmodels.formula.api import ols
```

In [2]:
```
#loading the dataset
df = pd.read_csv("C:\\Users\\sujoydutta\\Desktop\\Data analysis\\Datasets for ML\\Hypothesis testing\\LoansData
df.head()
```

Out[2]:

	Amount.Requested	Amount.Funded.By.Investors	Interest.Rate	Loan.Length	Loan.Purpose	Debt.To.Income.Ratio	State	Home.Ownership
0	20000.0	20000.0	8.90%	36 months	debt_consolidation	14.90%	SC	MORTGAGE
1	19200.0	19200.0	12.12%	36 months	debt_consolidation	28.36%	TX	MORTGAGE
2	35000.0	35000.0	21.98%	60 months	debt_consolidation	23.81%	CA	MORTGAGE
3	10000.0	9975.0	9.99%	36 months	debt_consolidation	14.30%	KS	MORTGAGE
4	12000.0	12000.0	11.71%	36 months	credit_card	18.78%	NJ	RENT

In [3]:
```
# Getting information about the data types and missing values
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2500 entries, 0 to 2499
Data columns (total 14 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Amount.Requested           2499 non-null   float64
 1   Amount.Funded.By.Investors 2499 non-null   float64
 2   Interest.Rate              2498 non-null   object 
 3   Loan.Length                2500 non-null   object 
 4   Loan.Purpose               2500 non-null   object 
 5   Debt.To.Income.Ratio       2499 non-null   object 
 6   State                      2500 non-null   object 
 7   Home.Ownership             2499 non-null   object 
 8   Monthly.Income             2499 non-null   float64
 9   FICO.Range                 2498 non-null   object 
 10  Open.CREDIT.Lines          2497 non-null   float64
 11  Revolving.CREDIT.Balance   2497 non-null   float64
 12  Inquiries.in.the.Last.6.Months  2497 non-null   float64
 13  Employment.Length          2423 non-null   object 
dtypes: float64(6), object(8)
memory usage: 273.6+ KB
None
```

In [4]:
```
#dropping null values
df=df.dropna()
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2413 entries, 0 to 2499
Data columns (total 14 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Amount.Requested           2413 non-null   float64
 1   Amount.Funded.By.Investors 2413 non-null   float64
 2   Interest.Rate              2413 non-null   object 
 3   Loan.Length                2413 non-null   object 
 4   Loan.Purpose               2413 non-null   object 
 5   Debt.To.Income.Ratio       2413 non-null   object 
 6   State                      2413 non-null   object 
 7   Home.Ownership             2413 non-null   object 
 8   Monthly.Income             2413 non-null   float64
 9   FICO.Range                 2413 non-null   object 
 10  Open.CREDIT.Lines          2413 non-null   float64
 11  Revolving.CREDIT.Balance   2413 non-null   float64
 12  Inquiries.in.the.Last.6.Months  2413 non-null   float64
 13  Employment.Length          2413 non-null   object 
dtypes: float64(6), object(8)
memory usage: 282.8+ KB
```

In [5]:
```
#data cleaning and formatting
df['Interest.Rate'] = df['Interest.Rate'].str.rstrip('%').astype(float)
df['Debt.To.Income.Ratio'] = df['Debt.To.Income.Ratio'].str.rstrip('%').astype(float)
df['Loan.Length'] = df['Loan.Length'].str.rstrip('months').astype(int)
```

Interest Rate vs. Loan Amount

Let us test if interest rates vary for different loan amounts, we will use a statistical test such as ANOVA. First,we will group the data by loan amounts and calculate the mean interest rates for each group. Then,we will perform the ANOVA test.

In [6]:
```
#performing the anova for loan amount groups
loan_amount_groups = df.groupby('Amount.Requested')
f_statistic, p_value = stats.f_oneway(*[group['Interest.Rate'] for name, group in loan_amount_groups])
```

In [7]:
```
#printing the results
print('ANOVA results for loan_amount_groups:')
print('F statistic:', f_statistic)
print('P value:', p_value)
```

```
ANOVA results for loan_amount_groups:
F statistic: 1.9852843268876887
P value: 9.5115543943547776e-21
```

Remark: Since P value is very less than 0.05 we can say there is a significant difference across loan amounts with respect to interest rates.Hence there are different interest rates for each amount.

Loan Length vs. Interest Rate

Let us see if loan length directly affects interest rates using a Pearson's correlation test.

In [10]:
```
#performing the anova for Loan length and Interest rate
correlation_coefficient, p_value = stats.pearsonr(df['Loan.Length'], df['Interest.Rate'])
```

In [12]:
```
#printing the results
print('Correlation results for Loan Length vs. Interest Rate:')
print('correlation coefficient:', correlation_coefficient)
print('P value:', p_value)
```

```
Correlation results for Loan Length vs. Interest Rate:
correlation coefficient: 0.4250573823094765
P value: 1.7938010673370282e-106
```

Remark: Since P value is very less than 0.05 we can say there is a significant correlation between Loan Length and Interest Rate and there is a moderately positive correlation between the two values.

Interest Rate vs. Loan Purpose

Let us test if interest rates vary for different loan amounts, we will use a statistical test such as ANOVA. First,we will group the data by loan amounts and calculate the mean interest rates for each group. Then,we will perform the ANOVA test.

In [13]:
```
#performing the anova for Loan Purposes
f_statistic, p_value = stats.f_oneway(*[group['Interest.Rate'] for name, group in df.groupby('Loan.Purpose')])
```

In [14]:
```
#printing the results
print('ANOVA results for Loan Purposes:')
print('F statistic:', f_statistic)
print('P value:', p_value)
```

```
ANOVA results for Loan Purposes:
F statistic: 7.330838185919651
P value: 2.7646672581411367e-14
```

Remark: Since P value is very less than 0.05 we can say there is a significant difference across loan purposes with respect to interest rates.Hence there are different interest rates for each purpose.

In [19]:
```
#Feature engineering for fico score
df[['FICO.Min', 'FICO.Max']] = df['FICO.Range'].str.split('-', expand=True).astype(int)
```

In [20]:
```
#getting the average FICO score
df['FICO.score'] = df[['FICO.Min', 'FICO.Max']].mean(axis=1)
df['FICO.score']
```

Out[20]:
```
0       737.0
1       717.0
2       692.0
3       697.0
4       697.0
        ...
2495    707.0
2496    742.0
2497    682.0
2498    677.0
2499    672.0
Name: FICO.score, Length: 2413, dtype: float64
```

Relationship Between FICO Scores and Home Ownership

In order to analyze the relationship between FICO scores and home ownership, we can compare the FICO scores for different home ownership categories using T-test of independent samples.

In [44]:
```
# Separating FICO scores
fico_home_owners = df[df['Home.Ownership'] == 'OWN']['FICO.score']
fico_non_home_owners = df[df['Home.Ownership'] != 'OWN']['FICO.score']
```

In [45]:
```
# Performing an independent samples t-test
t_statistic, p_value = stats.ttest_ind(fico_home_owners, fico_non_home_owners)
```

In [46]:
```
# Printing the results
print("T-Statistic:", t_statistic)
print("P-Value:", p_value)
```

```
T-Statistic: 0.4755690628354589
P-Value: 0.63442450330406779
```

In [47]:
```
# Determining the significance
alpha = 0.05
if p_value < alpha:
    print("There is a significant difference between FICO scores of home owners and non-home owners.")
else:
    print("There is no significant difference between FICO scores of home owners and non-home owners.")
```

```
There is no significant difference between FICO scores of home owners and non-home owners.
```

Remark: Since P value is higher than Alpha we can say Home ownership is not affected by FICO scores.