Springboard Data Science Career Track | Nikki Seegars
Capstone Project I: Statistical Inferential Data Analysis

Null Hypothesis

The Sallie Mae loan dataset for single family loans acquired in the 4th quarter of 2008 has just under 280k closed accounts with seven different closure reason codes. The vast majority of the accounts were closed through prepaid or early payoff of the loans. Closures classified as REO, real estate owned, are those in which the borrower(s) defaulted and were foreclosed by Fannie Mae. There is a belief that the REO closures have borrowers with initial credit scores on average that are lower than the borrowers who prepaid their accounts.

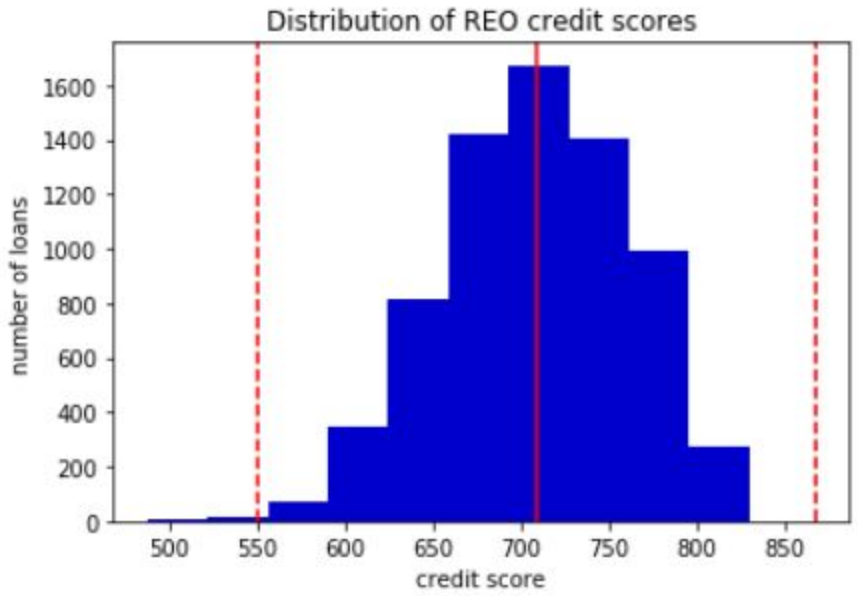
The null hypothesis is that there is no difference between borrower credit scores that were foreclosed (REO loans) and those loans that were prepaid.

During data exploration it was revealed that the vast majority of the loans for the 4th quarter were closed through prepayment. However, there were 7,022 loans that were closed through foreclosure. Summary statistics were calculated for all closed loans, prepaid loans, and REO loans and are listed below.

Category	Statistic	Value
All Closed Loans	Median	761.0
All Closed Loans	Mean	749.3
All Closed Loans	Standard Deviation	48.8
Prepaid	Median	763.0
Prepaid	Mean	751.1
Prepaid	Standard Deviation	47.9
REO	Median	710.0
REO	Mean	708.9
REO	Standard Deviation	52.9

The data points of all the borrower's credit scores for REO loans are in the plot below and has a left skew.

```
# Plot of distribution of borrower credit scores of REO loans
_ = plt.hist(df_reo['Borrower_Credit_Score'], color='mediumblue')
_ = plt.xlabel('credit score')
_ = plt.ylabel('number of loans')
_ = plt.title('Distribution of REO credit scores')
_ = plt.axvline(pop_mean, color='r')
_ = plt.axvline(pop_mean + 3 * pop_std, color='r', linestyle='--')
_ = plt.axvline(pop_mean - 3 * pop_std, color='r', linestyle='--')
```



Statistical Inference

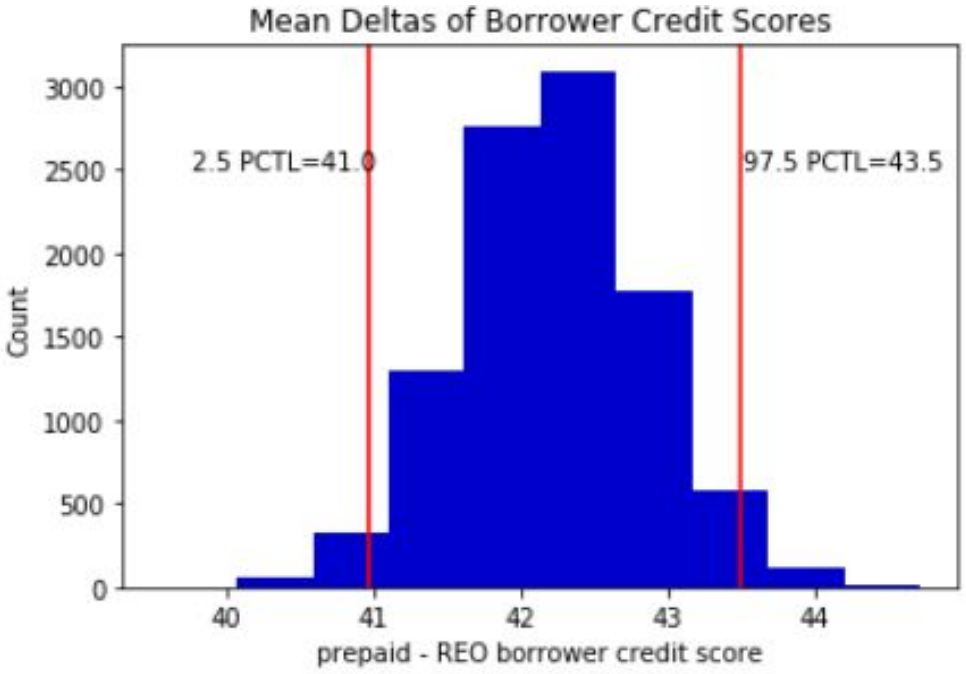
Bootstrapping methodology with permutation was used to compare the difference between REO loan credit scores and those that were prepaid in order to see if there is any significant difference. Three functions were created to perform the sampling of the data.

Function bootstrap_replicate takes a random sample with replacement for each of the datasets and returns a summary statistic, in this case the difference between means. The function is used to iterate through 10,000 samples and captured the means of the prepaid and REO credit scores and taking their difference to arrive at a value that represents and compares the two datasets.

```
# Create function that randomly samples with replacement
def bootstrap_replicate(data, func):
    """Select random sample without replacement and apply designated function on sample"""
    bs_sample = np.random.choice(data, size=len(data))
    return func(bs_sample)
```

```
size = 10000
bs_delta = np.empty(size)
for i in range(size):
    reo_loans = bootstrap_replicate(df_reo.Borrower_Credit_Score, np.mean)
    prepaid_loans = bootstrap_replicate(df_prepaid.Borrower_Credit_Score, np.mean)
    bs_delta[i] = prepaid_loans - reo_loans
lower_limit, upper_limit = np.percentile(bs_delta,[2.5, 97.5])
print('The 2.5 percentile of the difference between prepaid mean and REO mean is', lower_limit)
print('The 97.5 percentile of the difference between prepaid mean and REO mean is', upper_limit)
print('Array of sampling of the difference between prepaid mean and REO mean is', bs_delta)
```

The distribution of the differences is shown in the graph below.



Function permutation_sample is used to combine the prepaid and REO loan data into one indistinguishable dataset to be used to calculate the p-value.

```
# Combine and permute two datasets
def permutation_sample(data1, data2, func):
    """Generate a permutation sample from the two datasets."""

    # Concatenate the data sets: data
    data = np.concatenate((data1, data2))

    # Permute the concatenated array: permuted_data
    permuted_data = np.random.permutation(data)

    # Split the permuted array into two: perm_sample_1, perm_sample_2
    perm_sample_1 = permuted_data[:len(data1)]
    perm_sample_2 = permuted_data[len(data1):]

    diff = func(perm_sample_1, perm_sample_2)

    return diff
```

Function diff_of_mean subtracts the mean of REO credit scores from the mean of the prepaid credit scores.

```
# Take difference of means between two datasets
def diff_of_mean(perm_sample_1, perm_sample_2):
    """Difference of mean of two arrays."""

    # The difference of means of data_1, data_2: diff
    diff = np.mean(perm_sample_1) - np.mean(perm_sample_2)

    return diff
```

The perm_replicates array is used in a loop through 10,000 samples and store random samples of the difference between the permuted and combined datasets. From there the p-value is calculated by comparing the absolute value of the samples to the calculated difference between the two datasets captured in the bs_delta variable.

```
perm_replicates = np.empty(size)
reo_data = np.array(df_reo.Borrower_Credit_Score)

for i in range(size):
    perm_replicates[i] = permutation_sample(
        reo_data, np.random.choice(np.array(df_prepaid.Borrower_Credit_Score),
                                   len(reo_data)), diff_of_mean)
print(perm_replicates)
```

The p-value is calculated as 0 indicating that the null hypothesis that there is no difference between borrower credit scores and those that were foreclosed (REO loans) should be rejected since the p-value is zero which is <.05.

```
p = np.sum(abs(perm_replicates) >= np.mean(bs_delta)) / len(perm_replicates)
```