Kanwarpartap Singh Brar

Analysis of Treasury Yields as Leading Indicators for Mortgage Rates

Part 1 (Question Formulation)/Introduction:

The question I will be diving into is:

"Are changes in treasury yields leading indicators of changes in mortgage rates, or do they follow a lagged relationship?"

Mortgage rates are the interest rate that is charged on a mortgage when purchasing a property and there are different types of mortgage rates. I will be looking at 30-year fixed rate mortgage rates as well as 15-year fixed rate mortgage rates over the last 4 years (01/01/2020 - 01/01/2024). A multitude of factors such as inflation, monetary policy, securities, etc impacts mortgage rates. In this project, I will focus on treasury yields as a key factor influencing mortgage rates.

Treasury yields are interest rates of securities that the government pays investors for lending money. When it comes to mortgages, the 10-year treasury yield is significant because it helps indicate long-term interest rates as mortgages are usually long-term loans (15 or 30 years). In this project, I aim to analyze how long-term Treasury yields influence mortgage rates and whether changes in Treasury yields can predict mortgage rates based on their relationship over the past 24 years.

I will collect data from 3 datasets using FRED (Federal Reserve Economic Data) and join them on their dates to perform a time-series analysis. The datasets I will use are 'Market Yield on U.S. *Treasury Securities at 10-Year Constant Maturity', '15-Year Fixed Rate Mortgage Average in the United States', and '30-Year Fixed Rate Mortgage Average in the United States'.*

Part 2 (Data Collection & Pre-Processing):

I collected my data from FRED (Federal Reserve Economic Data), an online database system that holds various economic data. As mentioned above the datasets I will use are Market Yield on U.S. *Treasury Securities at 10-Year Constant Maturity', '15-Year Fixed Rate Mortgage Average in the United States', and '30-Year Fixed Rate Mortgage Average in the United States'*. Below is my code for loading the datasets as well as cleaning and organizing the data so it is ready for analysis:

```
#PART 2 (Data Collection)

#read treasury yield data, 15-year mortgage rate data, and 30-year mortgage rate data

treasury_yield_data = pd.read_csv('/Users/shawnbrar/Desktop/Hofstra_Classes/DS_221/Final_Project/Treasury_Yield_Data.csv')
#print(treasury_yield_data.head())

#rename DGS10 column to yield
treasury_yield_data.rename(columns = {'DGS10':'Yield'},inplace = True)

Mortgage_Data_15 = pd.read_csv('/Users/shawnbrar/Desktop/Hofstra_Classes/DS_221/Final_Project/15_Year_Mortgage_Data.csv')

Mortgage_Data_30 = pd.read_csv('/Users/shawnbrar/Desktop/Hofstra_Classes/DS_221/Final_Project/30_Year_Mortgage_Data.csv')


#check to see if any null values in the data
#print('\n')
#print('Tres Data: ')
#print(treasury_yield_data.isnull().any())
#print(treasury_yield_data.info())
#checking for non-numeric values b/c Yield is an object not float64
#print(treasury_yield_data['Yield'].unique())


#convert date to datetime instead of object

treasury_yield_data['DATE'] = pd.to_datetime(treasury_yield_data['DATE'])
Mortgage_Data_15['DATE'] = pd.to_datetime(Mortgage_Data_15['DATE'])
Mortgage_Data_30['DATE'] = pd.to_datetime(Mortgage_Data_30['DATE'])


#acciental '.' instead of a value so I will replace convert column to numeric and then use linear interpolation to fill missing value
treasury_yield_data['Yield'] = treasury_yield_data['Yield'].replace('.', np.nan)
treasury_yield_data['Yield'] = pd.to_numeric(treasury_yield_data['Yield'], errors='coerce')
treasury_yield_data['Yield'] = treasury_yield_data['Yield'].interpolate(method='linear')
```

```
print('\n')
print('15-Year Data: ')
print(Mortgage_Data_15.isnull().any())
print(Mortgage_Data_15.info())

print('\n')
print('30-Year Data: ')
print(Mortgage_Data_30.isnull().any())
print(Mortgage_Data_30.info())

print('\n')
print('Tres Info: ')
print(treasury_yield_data.isnull().any())
print(treasury_yield_data.info()) #Yield has been converted to a numeric datatype


print(treasury_yield_data.head())
print(Mortgage_Data_15.head())
print(Mortgage_Data_30.head())



#merge 3 datasets using inner join so they all have the same dates (weekly dates)

fp_merged_data = pd.merge(treasury_yield_data, Mortgage_Data_15, on = 'DATE', how = 'inner')
fp_merged_data = pd.merge(fp_merged_data, Mortgage_Data_30, on = 'DATE', how = 'inner')

#checking to see if data is clean
print('\n')
print('Merged Data Info: ')
print(fp_merged_data.isnull().any())
print(fp_merged_data.shape)
print(fp_merged_data.info())

#saving my merged dataset as a csv file
#fp_merged_data.to_csv('/Users/shawnbrar/Desktop/Hofstra_Classes/DS_221/Final_Project/fp_merged_data.csv')
```

As we can see below this code allows us to check and see if our data is clean. We can see that there are no null values in any of the datasets and that the datatypes are datetime64 and float64 for all the different datasets. I've merged the data using inner join onto DATE and will use the merged dataset for analysis (fp_merged_data). In this dataset, there are 209 non-null values which are obtained weekly because most economic data is best gathered/understood either weekly or monthly.

```
15-Year Data:
DATE            False
MORTGAGE15US    False
dtype: bool
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1252 entries, 0 to 1251
Data columns (total 2 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   DATE          1252 non-null   datetime64[ns]
 1   MORTGAGE15US  1252 non-null   float64
dtypes: datetime64[ns](1), float64(1)
memory usage: 19.7 KB
None


30-Year Data:
DATE            False
MORTGAGE30US    False
dtype: bool
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1252 entries, 0 to 1251
Data columns (total 2 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   DATE          1252 non-null   datetime64[ns]
 1   MORTGAGE30US  1252 non-null   float64
dtypes: datetime64[ns](1), float64(1)
memory usage: 19.7 KB
None


Tres Info:
DATE     False
Yield    False
dtype: bool
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1043 entries, 0 to 1042
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   DATE    1043 non-null   datetime64[ns]
 1   Yield   1043 non-null   float64
dtypes: datetime64[ns](1), float64(1)
memory usage: 16.4 KB
None
```

```
None
        DATE  Yield
0 2020-01-02   1.88
1 2020-01-03   1.80
2 2020-01-06   1.81
3 2020-01-07   1.83
4 2020-01-08   1.87
        DATE  MORTGAGE15US
0 2000-01-07          7.73
1 2000-01-14          7.78
2 2000-01-21          7.86
3 2000-01-28          7.84
4 2000-02-04          7.85
        DATE  MORTGAGE30US
0 2000-01-07          8.15
1 2000-01-14          8.18
2 2000-01-21          8.26
3 2000-01-28          8.25
4 2000-02-04          8.25


Merged Data Info:
DATE            False
Yield           False
MORTGAGE15US    False
MORTGAGE30US    False
dtype: bool
(209, 4)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 209 entries, 0 to 208
Data columns (total 4 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   DATE          209 non-null    datetime64[ns]
 1   Yield         209 non-null    float64
 2   MORTGAGE15US  209 non-null    float64
 3   MORTGAGE30US  209 non-null    float64
dtypes: datetime64[ns](1), float64(3)
memory usage: 6.7 KB
None
```

Part 3 (Exploratory Data Analysis):

Summary Statistics:

```
#PART 3 (EDA)

##1. summary statistics

summary_statistics = fp_merged_data.describe()
#gets mean, median, sd, min, max, quartiles
#print(summary_statistics)

numeric_columns = fp_merged_data.select_dtypes(include = 'float64') #selecting all columns except 'Date'
var = numeric_columns.var()  #getting variance of columns
summary_statistics.loc['var'] = var
summary_statistics = summary_statistics.drop(columns = 'DATE') #drop 'Date' as it is unnecessary
print("\n")
print("Summmary Statistics: ")
print(summary_statistics)
```

```
Summmary Statistics:
            Yield  MORTGAGE15US  MORTGAGE30US
count  209.000000    209.000000    209.000000
mean     2.308660      3.886938      4.548134
min      0.550000      2.100000      2.650000
25%      1.270000      2.350000      2.990000
50%      1.810000      2.990000      3.560000
75%      3.550000      5.690000      6.390000
max      4.980000      7.030000      7.790000
std      1.298315      1.677901      1.721530
var      1.685621      2.815350      2.963667
```

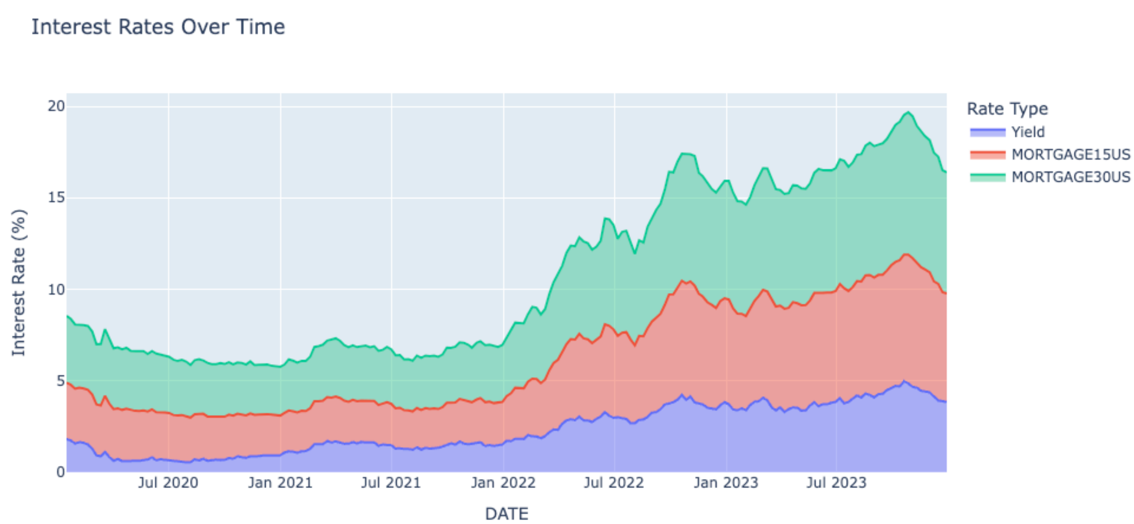
By looking at the summary statistics we can see quite a few things. There are 209 total observations and averages for the treasury yield, 15-year mortgage, and 30-year mortgage over the 4-year observed period are approximately 2.30, 3.89, and 4.55 respectively. The standard deviation was near 1.5% for all the rates suggesting that there was a somewhat considerable amount of fluctuation around the mean of the different rates over the period which makes sense. Overall we see that the variance is relatively high for all the financial data which signifies a lot of variability over the time period.

Data Visualization:

```
##2. Data Visualization

#print(fp_merged_data.head())

#area chart using plotly

fp_long = fp_merged_data.melt(id_vars = 'DATE', value_vars = ['Yield', 'MORTGAGE15US', 'MORTGAGE30US'], var_name='Rate Type', value_name='Rate (%)')


fig = px.area(fp_long, x = 'DATE', y = 'Rate (%)', color = 'Rate Type', line_group = 'Rate Type',
labels = { 'Rate (%)': 'Interest Rate (%)', 'variable':'Rate Type'},
title = 'Interest Rates Over Time')

fig.update_layout(width = 1000, height = 500)

fig.show()

#box plot for each variable

#Yield box plot

fig_yield = px.box(fp_merged_data, y = 'Yield', points = 'all',
                   labels = {'Yield': 'Treasury Yield(%)' },
                   title = 'Distribution of Treasury Yield')

fig.update_layout(width = 1000, height = 500)
fig_yield.show()

#MORTGAGE15US box plot

fig = px.box(fp_long, x = 'Rate Type', y = 'Rate (%)', color = 'Rate Type', points = 'all',
             title = 'Rate Distribution')
fig.update_layout(width = 1000, height = 500)
fig.show()
```

In this section, I will walk through two different charts. The first chart will be an area line chart which is a line graph of the three different rate types over the 4-year period and the second chart is a box plot of all 3 variables that show the rate distribution of each.

1. *Area Line Graph Of Interest Rates Over Time:*



With this graph, we can analyze the trend of treasury yields and 15/30-year mortgage rates in the U.S. from 2020 to 2024. One of the first things we see is that the rates have risen consistently and peaked around October 2023 before starting to come down. If you look closely we can see that the treasury yields are first to change and there is a buffer between the mortgage rates and the treasury yields. Buffer means that the mortgage rates are lagging and their changes follow behind the treasury yields. Even though the buffer is slight it is something to take into account. The 30-year mortgage rates are always slightly higher than the 15-year mortgage rates which makes sense because the longer the loan the higher the risk premium attached to it. We can also note that around the start of 2024, there is a steep drop off in the mortgage rates for both the 15 and 30-year rates whereas the treasury yield rates have not declined as quickly. In this analysis, I did not examine the data for after Jan 2024 but because of the steep decline in the mortgage rates and not so steep decline in the treasury yield rates, this can be an issue for the housing crisis as well as inflation. As we have gone through 2024 we have noticed how this steep drop has played a role as part of the FED expansionary policy of trying to lower interest rates with rate cuts throughout the year.

Overall in this graph, we observe that there has been generally an upward trend of rates, and more importantly that both mortgage rates lag slightly behind the treasury yield rates, indicating that they are reactive and partly influenced by changes in the treasury yields.

2. *Box Plot Rate Distribution:*

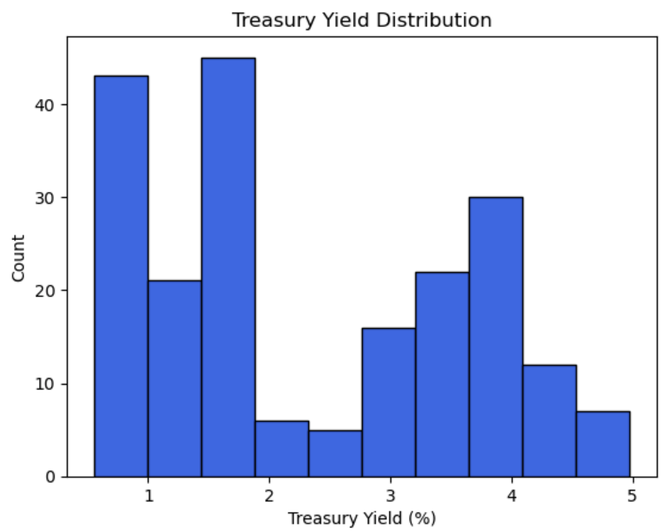
In this box plot, we can see the distribution of the different rates. We observe that the treasury yield rates seem to be more stable and less volatile when compared to mortgage rates, which can be seen by the wider ranges. This is also because long-term treasury yields like the ones observed are a low-risk investment which leads to less volatility in changes to the interest rate. This also makes sense as to why the 30-year mortgage rate is the most volatile with the widest range because it also has the highest risk premium of the three observed rates. Overall, with this box plot, we can understand the central tendencies and variabilities within each rate type.

Data Distribution:

```
##3. DATA DISTRIBUTION

#histogram for yield (right-skewed)

plt.hist(fp_merged_data['Yield'], edgecolor = 'black', color = 'royalblue')
plt.xlabel('Treasury Yield (%)')
plt.ylabel('Count')
plt.title('Treasury Yield Distribution')
plt.show()

#histogram for MORTGAGE15US (left-skewed)

plt.hist(fp_merged_data['MORTGAGE15US'], edgecolor = 'black', color = 'royalblue')
plt.xlabel('Interest Rate (%)')
plt.ylabel('Count')
plt.title('15 Year Mortgage Rate Distribution')
plt.show()

#histogram for MORTGAGE30US

plt.hist(fp_merged_data['MORTGAGE30US'], edgecolor = 'black', color = 'royalblue')
plt.xlabel('Interest Rate (%)')
plt.ylabel('Count')
plt.title('30 Year Mortgage Rate Distribution')
plt.show()

#measure value of skewness (positive = right, negative = left skew)

skewness_Yield = fp_merged_data['Yield'].skew()
print('Skewness of Yield:', skewness_Yield)

skewness_M15 = fp_merged_data['MORTGAGE15US'].skew()
print('\n')
print('Skewness of MORTGAGE15US:', skewness_M15)

skewness_M30 = fp_merged_data['MORTGAGE30US'].skew()
print('\n')
print('Skewness of MORTGAGE30US:', skewness_M30)
```

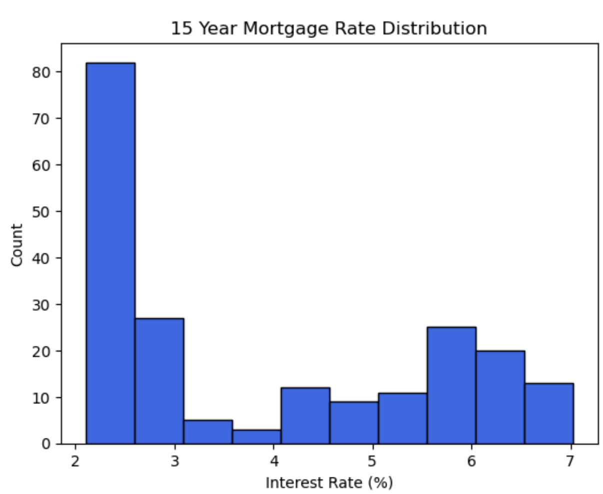
```
Skewness of Yield: 0.2958056492863812

Skewness of MORTGAGE15US: 0.45632534366218513

Skewness of MORTGAGE30US: 0.41652044127610766
```

a. *Histogram for Yield:*



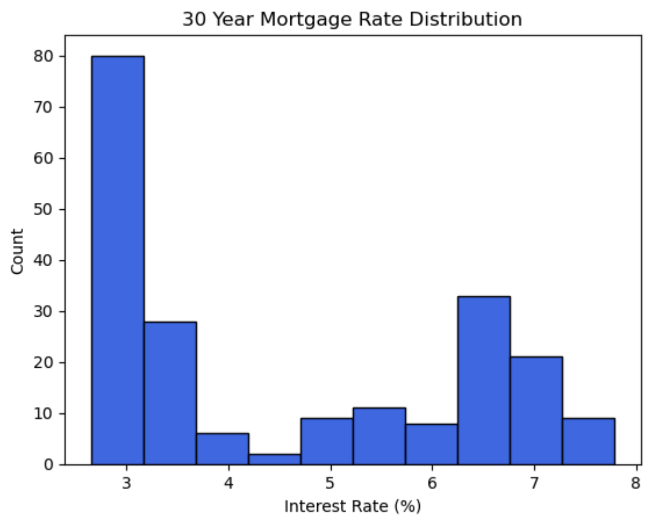
Here we can see the distribution of the treasury yield rates. We can observe that the histogram of ‘Yield’ is slightly skewed to the right picture of the distribution as well as the skewness value of approximately 0.296.

b. *Histogram for MORTGAGE15US:*



Here we can see the distribution of the 15-year mortgage rates. We can observe that the histogram of 'MORTGAGE15US' is skewed to the right because of the highly concentrated values at the lower end and the extending tail towards the right. We also see that it has a skewness value of approximately 0.46.

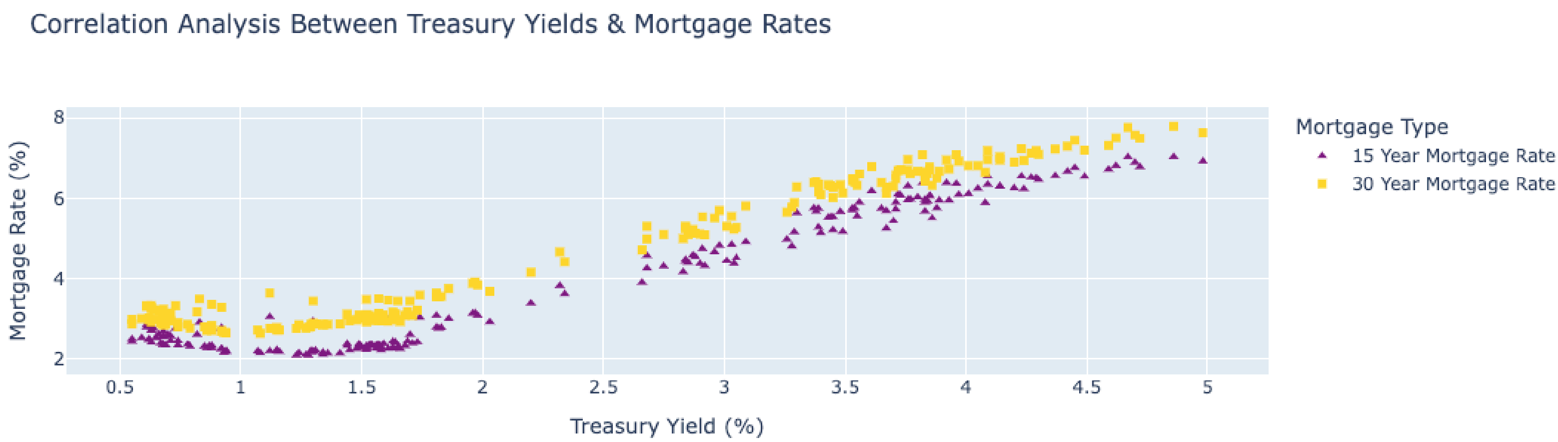
c. *Histogram for MORTGAGE30US:*



Here we can see the distribution of the 30-year mortgage rates. We can observe that the histogram of 'MORTGAGE30US' is skewed to the right because of the similar highly concentrated values at the lower rates and the extending tail towards the right. We also see that it has a skewness value of approximately 0.42.

Correlation Analysis:

```
##4. CORRELATION ANALYSIS


fp_merged_data['Yield'] = pd.to_numeric(fp_merged_data['Yield'], errors='coerce')

fig = go.Figure()

#15 year mortgage rate scatter plot

fig.add_trace(go.Scatter(x = fp_merged_data['Yield'], y = fp_merged_data['MORTGAGE15US'], mode = 'markers',
                         name = '15 Year Mortgage Rate', marker = dict(color = 'purple', symbol = 'triangle-up')))

#30 year mortgage rate scatter plot

fig.add_trace(go.Scatter(x = fp_merged_data['Yield'], y = fp_merged_data['MORTGAGE30US'], mode = 'markers',
                         name = '30 Year Mortgage Rate', marker = dict(color = 'gold', symbol = 'square')))

fig.update_layout(title = 'Correlation Analysis Between Treasury Yields & Mortgage Rates', xaxis_title = 'Treasury Yield (%)',
                  yaxis_title = 'Mortgage Rate (%)', legend_title = 'Mortgage Type')

fig.show()
```



In this scatterplot, we can observe the relationship between treasury yields and mortgage rates. We see that as treasury yield (%) increases, both the 15-year and 30-year mortgage rates increase which tells us there is a positive correlation between treasury yields and mortgage rates. We also notice that the 15-year mortgage rates for the most part are always consistently lower than the 30-year mortgage rates, this is primarily due to the risk premium that comes with a longer loan. Overall, the consistent increase in mortgage rates with treasury yields implies that treasury yields could be used as a leading indicator in financial models and mortgage rate predictions due to their close link and relationship.

Hypothesis Testing:

Hypothesis: Do changes in treasury yields have different effects on 15-year and 30-year mortgage rates?

H0 (Null): Changes in treasury yields have the same impact on both mortgage rates.

H1 (Alternative): Changes in treasury yields do not have the same impact on both mortgage rates.

```
##5. HYPOTHESIS TESTING

#Is there a time lag between changes in treasury yields mortgage rates?

#find differneces in changes in interest rates

fp_merged_data['Yield_changes'] = fp_merged_data['Yield'].diff()  #changes in Yield

fp_merged_data['M15_changes'] = fp_merged_data['MORTGAGE15US'].diff() #changes in 15 year mortgage data

fp_merged_data['M30_changes'] = fp_merged_data['MORTGAGE30US'].diff() #changes in 30 year mortgage data

fp_merged_data = fp_merged_data.dropna() #drops first row bc you cant difference the first row so it will not have anumber ar

#run ordinary least square regression

X = sm.add_constant(fp_merged_data['Yield_changes'])  #constant term

#dependednt variables whos relationship to yield changes is being observed
mort_15_change = fp_merged_data['M15_changes']
mort_30_change = fp_merged_data['M30_changes']

#OLS for 15year rates
OLS_M15 = sm.OLS(mort_15_change, X).fit()

#OLS for 30 year rates
OLS_M30 = sm.OLS(mort_30_change, X).fit()

change_on_15yr = OLS_M15.params['Yield_changes']
change_on_30yr = OLS_M30.params['Yield_changes']

#conduct ttest

statistic, p_value = ttest_ind(fp_merged_data['M15_changes'], fp_merged_data['M30_changes'], equal_var=False)

print("\n")
print('HYPOTHESIS TESTING')
print("t_stat: ", statistic, "p_value: ", p_value)
```

```
alpha = .05
if p_value < .05:
    print("Reject the null hypothesis, the result is statistically significant at the 5% significance level")
else:
    print("Failed to reject the null hypothesis")
```

```
HYPOTHESIS TESTING
t_stat:  -0.04605449637553338 p_value:  0.9632890225298327
Failed to reject the null hypothesis
```

For this hypothesis test, I chose an independent t-test. First I found the differences in changes in interest rates and I ran an ordinary least square regression (OLS) to figure out how much the mortgage rates were expected to change as the treasury yields changed by one unit. Then I ran the t-test and the two independent groups were the changes in the 15-year mortgage rates and the changes in the 30-year mortgage rates.

As a result of the test, we get a t_stat of approximately -.046 and a p-value of approximately 0.963. This p-value is very high so we failed to reject the null hypothesis, meaning that both mortgage durations (15-year & 30-year) respond similarly to adjustments in treasury yields, at least in the aspect of how much their rates change independently.

Part 4 (Machine Learning/Results & Discussion):

```
#4. MACHINE LEARNING


#predict 30 year mortgage rates by predicting yields
#predict the treasury yield rate by predicting 30 year mortgage rates


#predict 15 year mortgage rates
#Using a forest regressor model

X = fp_merged_data[['Yield', 'MORTGAGE30US']] #feature variables (dependent)
y = fp_merged_data['MORTGAGE15US'] #target varialbe (independent)

#split data so 20% is for testing and 80% is for training

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42)

random_forest_M15 = RandomForestRegressor(n_estimators= 100, random_state = 42) #100 estimators for a better prediction

#train model

random_forest_M15.fit(X_train, y_train)

#predict
y_pred_M15 = random_forest_M15.predict(X_test)

print('\n')
print('Predicted 15-year Mortgage Rates: ')
print(y_pred_M15)


#model evaluation

mse_random_forest_M15 = mean_squared_error(y_test, y_pred_M15) #get mse
rmse_random_forest_M15 = np.sqrt(mse_random_forest_M15) #get rmse
r2_random_forest_M15 = r2_score(y_test, y_pred_M15) #get r^2
```

```
print('\n')
print("Mean Squared Error Of Random Forest Regressor Model (M15): ", mse_random_forest_M15)
print('RMSE Of Random Forest Regressor Model (M15): ', rmse_random_forest_M15)
print("R2 Score of Random Forest Regressor Model (M15): ", r2_random_forest_M15 )

#visualize the model and its predictions

dates = pd.date_range(start='2020-01-01', periods=len(y_test), freq='ME') #monthly range of dates

actual_M15 = y_test #actual points
predicted_M15 = y_pred_M15 #model predicted rates

plt.figure(figsize=(10, 5))
plt.plot(dates, actual_M15, label = 'Actual 15-Year Mortgage Rates', color = 'gold', marker = 'o')
plt.plot(dates, predicted_M15, label = 'Predicted 15-Year Mortgage Rates', color = 'purple', marker = 'x')
plt.title('Predicted VS Actual 15-Year Mortgage Rates')
plt.xlabel('Date')
plt.ylabel('15-Year Mortgage Rate (%)')
plt.legend()
plt.grid(True)
plt.show()


#predict 30 year mortgage rates
#Using a forest regressor model

X_M30 = fp_merged_data[['Yield', 'MORTGAGE15US']] #feature variables (dependent)
y_M30 = fp_merged_data['MORTGAGE30US'] #target varialbe (independent)

#split data so 20% is for testing and 80% is for training
X_M30_train, X_M30_test, y_M30_train, y_M30_test = train_test_split(X_M30, y_M30, test_size = 0.2, random_state = 42)

random_forest_M30 = RandomForestRegressor(n_estimators= 100, random_state = 42) #100 estimators for a better prediction


#train model
random_forest_M30.fit(X_M30_train, y_M30_train)

#predict model
y_pred_M30 = random_forest_M30.predict(X_M30_test)
```

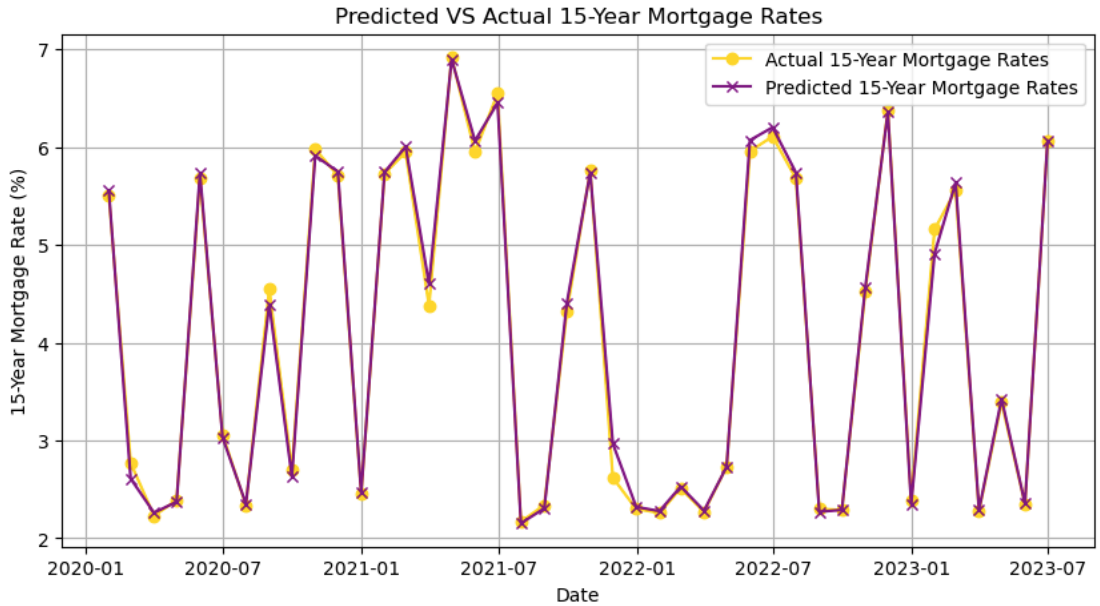
```python
print('\n')
print('Predicted 30-year Mortgage Rates: ')
print(y_pred_M30)


#model evaluation

mse_random_forest_M30 = mean_squared_error(y_M30_test, y_pred_M30) #get mse
rmse_random_forest_M30 = np.sqrt(mse_random_forest_M30) #get rmse
r2_random_forest_M30 = r2_score(y_M30_test, y_pred_M30) #get r^2

print('\n')
print("Mean Squared Error Of Random Forest Regressor Model (M30): ", mse_random_forest_M30)
print('RMSE Of Random Forest Regressor Model (M30): ', rmse_random_forest_M30)
print("R2 Score of Random Forest Regressor Model (M30): ", r2_random_forest_M30 )

#visualize the model and its predictions

dates_M30 = pd.date_range(start='2020-01-01', periods=len(y_M30_test), freq='ME') #monthly range of dates

actual_M30 = y_M30_test #actual points
predicted_M30 = y_pred_M30 #model predicted rates

plt.figure(figsize=(10, 5))
plt.plot(dates_M30, actual_M30, label = 'Actual 30-Year Mortgage Rates', color = 'gold', marker = 'o')
plt.plot(dates_M30, predicted_M30, label = 'Predicted 30-Year Mortgage Rates', color = 'purple', marker = 'x')
plt.title('Predicted VS Actual 30-Year Mortgage Rates')
plt.xlabel('Date')
plt.ylabel('30-Year Mortgage Rate (%)')
plt.legend()
plt.grid(True)
plt.show()

#predict treasury yields using mortgage rates
#using a forest regressor model

X_Yield = fp_merged_data[['MORTGAGE30US', 'MORTGAGE15US']] #feature variables (dependent)
y_Yield = fp_merged_data['Yield'] #target varialbe (independent)
```

```python
X_Yield_train, X_Yield_test, y_Yield_train, y_Yield_test = train_test_split(X_Yield, y_Yield, test_size = 0.2, random_state = 42)

random_forest_Yield = RandomForestRegressor(n_estimators= 100, random_state = 42) #100 estimators for a better prediction

#train model
random_forest_Yield.fit(X_Yield_train, y_Yield_train)

#predict model
y_pred_Yield = random_forest_Yield.predict(X_Yield_test)

print('\n')
print('Predicted Treasury Yield Rates: ')
print(y_pred_Yield)

#model evaluation

mse_random_forest_Yield = mean_squared_error(y_Yield_test, y_pred_Yield) #get mse
rmse_random_forest_Yield = np.sqrt(mse_random_forest_Yield) #get rmse
r2_random_forest_Yield = r2_score(y_Yield_test, y_pred_Yield) #get r^2

print('\n')
print("Mean Squared Error Of Random Forest Regressor Model (Treasury Yield Rate): ", mse_random_forest_Yield)
print('RMSE Of Random Forest Regressor Model (Treasury Yield Rate): ', rmse_random_forest_Yield)
print("R2 Score of Random Forest Regressor Model (Treasury Yield Rate): ", r2_random_forest_Yield)

#visualize the model and its predictions

dates_Yield = pd.date_range(start='2020-01-01', periods=len(y_Yield_test), freq='ME') #monthly range of dates

actual_Yield = y_Yield_test #actual points
predicted_Yield = y_pred_Yield #model predicted rates

plt.figure(figsize=(10, 5))
plt.plot(dates_Yield, actual_Yield, label = 'Actual Treasury Yield Rates', color = 'gold', marker = 'o')
plt.plot(dates_Yield, predicted_Yield, label = 'Predicted Treasury Yield Rates', color = 'purple', marker = 'x')
plt.title('Predicted VS Actual Treasury Yield Rates')
plt.xlabel('Date')
plt.ylabel('Treasury Yield Rate (%)')
plt.legend()
plt.grid(True)
plt.show()
```

```
Predicted 15-year Mortgage Rates:
[5.1903 2.8004 2.2611 2.2872 5.9347 2.7926 2.3473 4.4528 2.7843 6.3588
 5.738  2.3717 5.7375 5.9598 4.4032 6.9179 5.9745 6.4705 2.1452 2.2638
 4.3757 5.58   2.5083 2.3619 2.261  2.5548 2.3358 2.665  5.5813 6.199
 5.5815 2.2715 2.346  4.3959 6.3076 2.368  4.8685 5.7308 2.2946 3.1184
 2.3769 6.0356]


Mean Squared Error Of Random Forest Regressor Model (M15):  0.003733381666666286
RMSE Of Random Forest Regressor Model (M15):  0.06110140478472347
R2 Score of Random Forest Regressor Model (M15):  0.9986476942629414


Predicted 30-year Mortgage Rates:
[6.0998     3.3239     2.9608     2.9734     6.6064     3.3554
 3.0733     5.2133     3.2638     6.93831    6.4595     2.9551
 6.3819     6.68775833 5.0811     7.6121     6.755725   7.1599
 2.7608     2.9825     5.0465     6.3171     3.1724     3.01400833
 3.03056429 3.0272     3.02995595 3.3197     6.3459     6.84645
 6.3697     3.04381429 3.01661429 5.1677     6.9957     3.1096
 5.7924     6.3351     2.7558     3.7687     3.1096     6.70752   ]


Mean Squared Error Of Random Forest Regressor Model (M30):  0.003694918266601 7725
RMSE Of Random Forest Regressor Model (M30):  0.06078583935919428
R2 Score of Random Forest Regressor Model (M30):  0.9987026994458696


Predicted Treasury Yield Rates:
[3.4254     0.6533     1.55956667 1.5323     3.7917     0.7456
 1.568      2.8269     0.6837     4.0226     3.463      0.7279
 3.5545     3.8261     2.8577     4.6882     3.8182     4.3295
 1.3492     1.4607     2.8559     3.4208     1.5307     1.52994167
 1.53506667 0.6334     1.59464167 0.7276     3.4694     3.8992
 3.3892     1.53490667 1.44415    2.9057     4.0832     1.4715
 3.016      3.5545     0.8916     1.9175     1.4476     3.8272    ]


Mean Squared Error Of Random Forest Regressor Model (Treasury Yield Rate):  0.02282256702949749
RMSE Of Random Forest Regressor Model (Treasury Yield Rate):  0.15107139712565543
R2 Score of Random Forest Regressor Model (Treasury Yield Rate):  0.9856800282337208
```

a. *Model (1):*

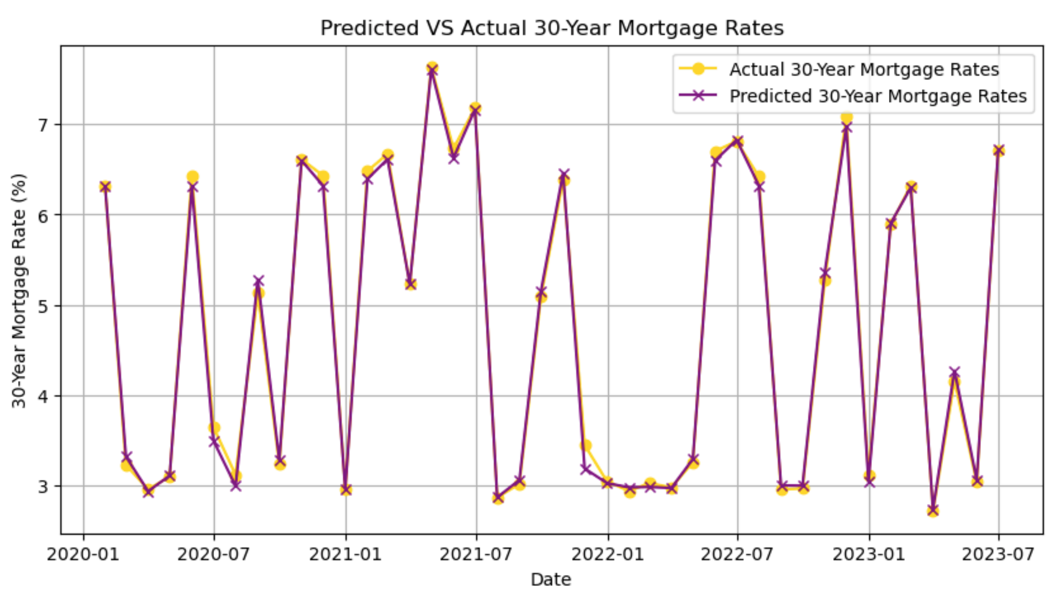
For all of the models, I used RandomForestRegressor as my model. I picked this model because of its ability to handle financial datasets due to its good overfitting and predictive performance. The first model was to predict the 15-year mortgage rate using ‘Yield’ and ‘MORTGAGE30US’ as my feature variables and ‘MORTGAGE15US’ as my target variable. When looking at the predicted 15-year mortgage rates we get a mean squared error (MSE) of approximately 0.004 which indicates a good model fit because a lower MSE indicates a better fit. The R^2 is very high at approximately 0.99, indicating that about 99% of the variance in the target variable (MORTGAGE15US) is explained by the model.



In this visualization, we can see the accuracy of the model and the consistency in its predictions. We can also see how well the graph is able to display the variability in the interest rates, and how the model is able to predict and be consistent with being close to the actual rates. This shows that this model could be effective for forecasting 15-year mortgage rates.

b. *Model (2):*

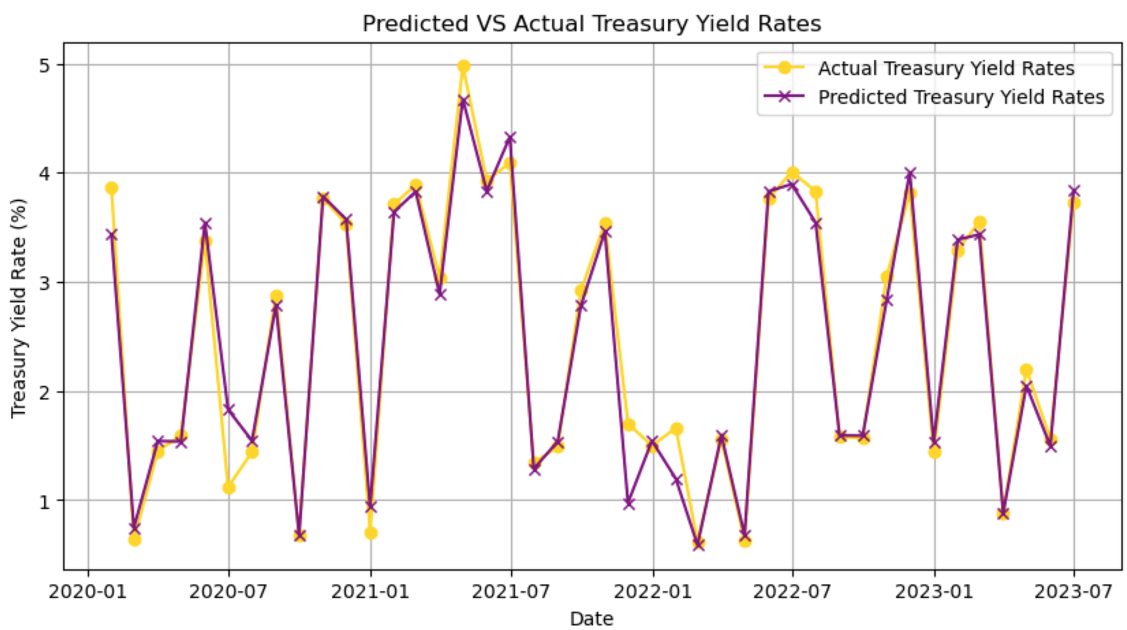
The second model was to predict the 30-year mortgage rate using 'Yield' and 'MORTGAGE15US' as my features and 'MORTGAGE30US' as my target variable. When looking at the predicted 30-year mortgage rates we get a MSE of approximately 0.004 again which is very low, indicating a good fit. We get an R^2 that is very high at approximately 0.99, indicating that 99% of the variance in the target variable (MORTGAGE30US) is explained by the model.



In this visualization, we can see similarly to the previous model, the accuracy and the consistency in its predictions. This shows us that this model can be a good indicator of effectively forecasting 30-year mortgage rates.

c. *Model (3):*

The third model now predicts treasury yields with both mortgage rates as the features and 'Yield' as the target variable. This model gets a slightly higher MSE of approximately 0.22 which is higher than the previous models but still low enough to indicate that this model is a good fit. The R^2 of this model is also very high at approximately 0.99, indicating that 99% of the variance in the target variable (Yield) is explained by the model.



In this visualization we can see that this model is fairly accurate, however by looking at the points it's safe to say it is not as accurate as the previous two models. This would also be an effective model to help forecast future treasury yield rates, however forecasting the mortgage rates would be more accurate because we now know by analyzing the data that the mortgage rates lag slightly behind the treasury yield rates.

Conclusion:

Based on the results, we can conclude that treasury yield rates (specifically long-term treasuries) do indicate changes in mortgage rates with a small lag. After running and analyzing the predictive machine learning models, I believe that due to their consistency, accuracy, and model fit, treasury yields can be used as a reliable tool with these models for forecasting mortgage rate trends. For future works, I plan to use forecasting tools to further enhance these models and see how well they do in predicting fluctuations and changes in mortgage rates.

In conclusion, the random forest regressor model was able to show the accuracy and ability of using treasury yields to help predict both 15-year and 30-year mortgage rates.

Data Attribution & References:

Treasury yield data was obtained from FRED (Federal Reserve Economic Data):
Name: Market Yield on U.S. Treasury Securities at 30-Year Constant Maturity
https://fred.stlouisfed.org/series/DGS30

15-year mortgage rate data was obtained from FRED (Federal Reserve Economic Data)
Name: 15-Year Fixed Rate Mortgage Average in the United States
https://fred.stlouisfed.org/series/MORTGAGE15US

30-year mortgage rate data was obtained from FRED (Federal Reserve Economic Data)
Name: 30-Year Fixed Rate Mortgage Average in the United States
https://fred.stlouisfed.org/series/MORTGAGE30US

Articles I used for research purposes:

"The Balance." *The Balance Money*, www.thebalancemoney.com/treasury-note-and-mortgage-rate-relationship-3305734.