Michael Frangos
FIN 6456-100: Quantitative Methods
Brian Silverstein, CFA, ASA
Project #1

Task #1 - Summary Statistics

Part A, B, C, & D - *Report summary statistics of the sample over 3 windows: 1/1974-12/2019, 1/1974-12/1987, and 1/1988 – 12/2019.*

TABLE 1 – Hedged & Unhedged Summary Statistics

Summary Statistics		Total Sample 1974:1 - 2019:12				Period 2 1988:1 - 2019:12				Period 1 1974:1 - 1987:12			
		Mean	Std. Dev	Skewness	Kurtosis	Mean	Std. Dev	Skewness	Kurtosis	Mean	Std. Dev	Skewness	Kurtosis
Canada	Unhedged	0.0061371	0.056283	-0.45594	2.311607	0.0062	0.0529	-0.5762	2.7750	0.01	0.06	-0.28	1.46
	Hedged	0.0050	0.0461	-0.5041	2.5520	0.00	0.04	-0.65	2.51	0.01	0.06	-0.35	1.57
	Diff	0.0012	0.0101	0.0481	-0.2404	0.0013	0.0127	0.0770	0.2615	0.0008	0.0058	0.0711	-0.1079
	t-stat	1.38				1.18				0.77			
France	Unhedged	0.01	0.06	-0.13	1.36	0.01	0.06	-0.30	0.89	0.01	0.08	0.01	1.14
	Hedged	0.01	0.06	-0.15	1.12	0.01	0.05	-0.21	0.69	0.01	0.07	-0.07	1.12
	Diff	0.0013	0.0070	0.0214	0.2441	0.0005	0.0052	-0.0806	0.2026	0.0033	0.0103	0.0791	0.0198
	t-stat	1.04				0.33				1.30			
Germany	Unhedged	0.01	0.06	-0.33	1.36	0.01	0.06	-0.43	1.53	0.01	0.06	-0.05	0.71
	Hedged	0.01	0.06	-0.52	2.17	0.01	0.06	-0.52	1.87	0.01	0.05	-0.48	2.95
	Diff	0.0009	0.0068	0.1902	-0.8174	0.0003	0.0055	0.0979	-0.3394	0.0024	0.0099	0.4282	-2.2476
	t-stat	0.69				0.17				0.91			
Japan	Unhedged	0.01	0.06	0.28	1.04	0.00	0.06	0.20	1.23	0.02	0.06	0.39	0.60
	Hedged	0.01	0.05	-0.16	1.26	0.00	0.05	-0.24	0.91	0.01	0.05	0.38	2.01
	Diff	0.0002	0.0066	0.4431	-0.2186	-0.001	0.0025	0.4382	0.3220	0.0036	0.0158	0.0074	-1.4164
	t-stat	0.17				-0.77				1.31			
United Kingdom	Unhedged	0.01	0.06	1.22	11.84	0.00	0.05	-0.11	0.91	0.01	0.09	1.37	8.60
	Hedged	0.01	0.05	1.31	17.29	0.00	0.04	-0.32	0.60	0.01	0.08	1.44	12.13
	Diff	0.0011	0.0074	-0.0850	-5.4529	0.0007	0.0069	0.2076	0.3089	0.0019	0.0088	-0.0697	-3.5261
	t-stat	0.87				0.51				0.76			

Part E - *Comment on the t-statistics for each. Does it seem worthwhile to hedge currency in any country? Does your conclusion change depending on what window you use?*

The data in Table 1 suggests that hedging any country's currency may depend on the time window used. However, the results should be taken with a grain of salt since almost all were not statistically significant, evident by the resulting t-stats. If period dependency is true, then the dependency may be caused by real economic factors, such as changes in each country's interest rates over time.

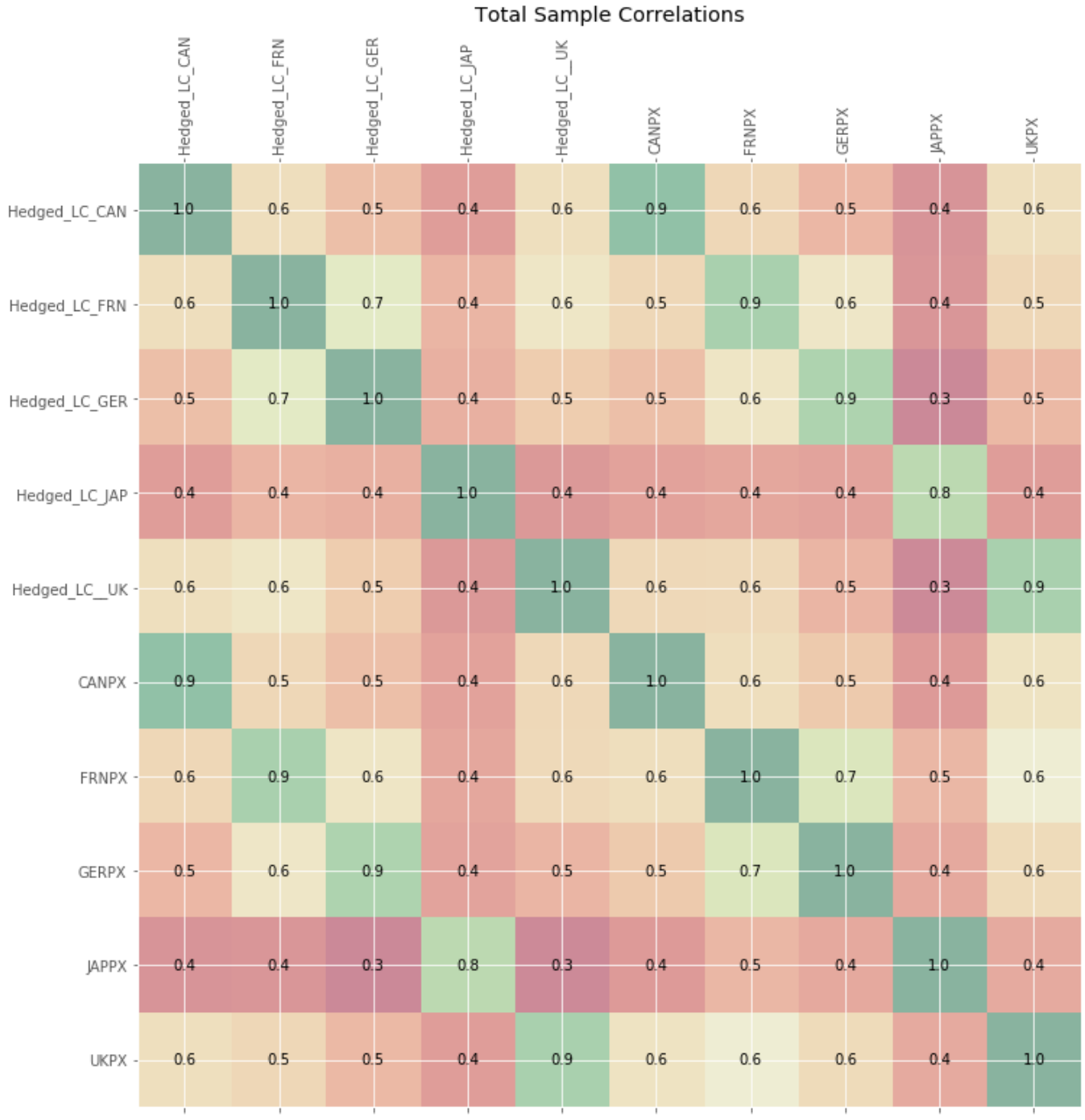
During period 2, currency hedging reduced volatility of Canadian currency returns by 0.0127, while volatility of Canadian currency returns was only reduced by 0.0058 in period 1. This suggests that period 2 was a better time for hedging. Furthermore, return variances were reduced for all currencies in all periods, but this may be a sampling error (The t-stats in Table 1 only reflect the differences in the mean returns, so additional t-tests on the differences in variances are required).

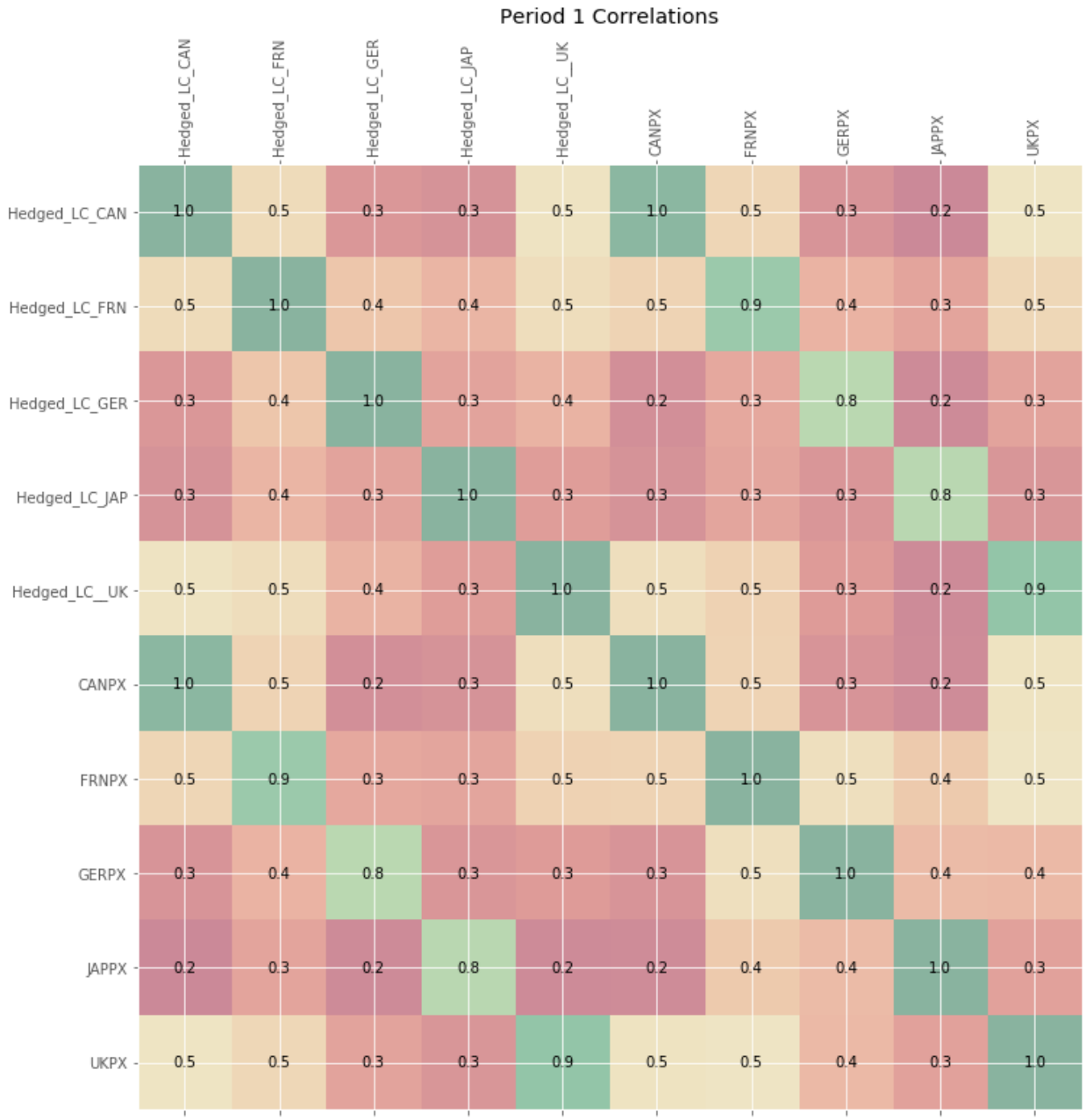
Canada's higher t-stat during period 2 of 1.18, compared with 0.77 during period 1 suggests that the mean returns may have been different. However, a t-stat of 1.18 is not statistically significant since it falls short of reaching the 80% confidence level. Consequently, we fail to reject the null hypothesis that the mean hedged returns are the same as mean unhedged returns. A higher t-stat represents a higher likelihood that the difference in the mean returns is statistically meaningful. The t-test results were similar for all other currency returns in that they all had low statistical significance.

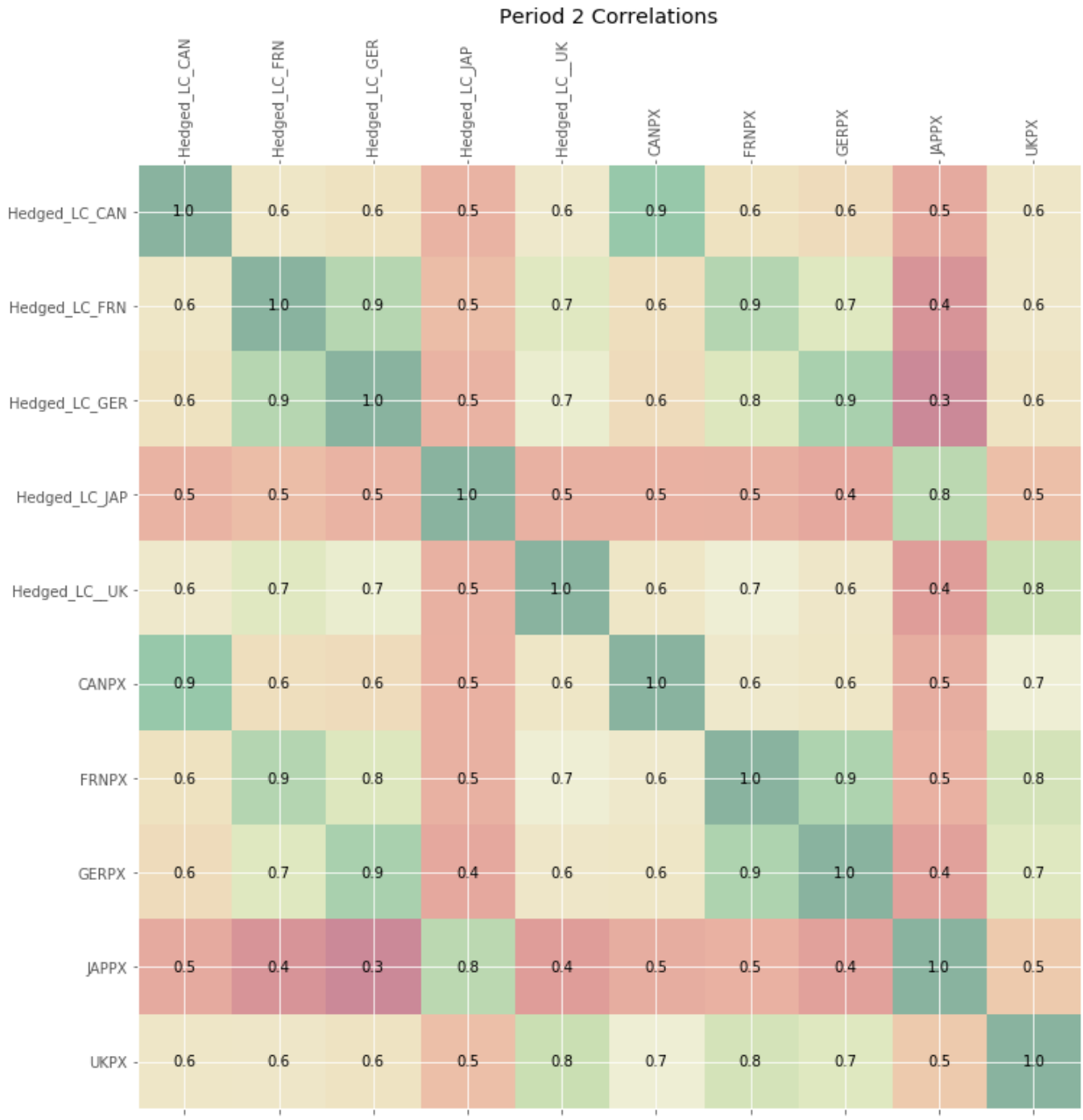
A mean difference in returns after hedging was most evident for Canadian currency returns over the entire sample period with a t-stat of 1.38. Unfortunately, this t-stat still barely exceeded the 80% confidence level in a two tailed t-stat distribution. Consequently, evidence that average unhedged returns are different from hedged returns is weak in strength. While not statistically significant, mean hedged returns were lower in 14 of 15 cases. Hedged returns may be biased upward since we are not accounting for transaction costs, which may cause a significant difference in means if accounted for.

Task #2 - Calculate correlation matrices

Part A - *Calculate a correlation matrix for the entire sample period for local currency unhedged returns and currency returns.*




Period 1 Correlations
Hedged_LC_CAN
Hedged_LC_FRN
Hedged_LC_GER
Hedged_LC_JAP
Hedged_LC__UK
CANPX
FRNPX
GERPX
JAPPX
UKPX
Hedged_LC_CAN 1.0 0.5 0.3 0.3 0.5 1.0 0.5 0.3 0.2 0.5
Hedged_LC_FRN 0.5 1.0 0.4 0.4 0.5 0.5 0.9 0.4 0.3 0.5
Hedged_LC_GER 0.3 0.4 1.0 0.3 0.4 0.2 0.3 0.8 0.2 0.3
Hedged_LC_JAP 0.3 0.4 0.3 1.0 0.3 0.3 0.3 0.3 0.8 0.3
Hedged_LC__UK 0.5 0.5 0.4 0.3 1.0 0.5 0.5 0.3 0.2 0.9
CANPX 1.0 0.5 0.2 0.3 0.5 1.0 0.5 0.3 0.2 0.5
FRNPX 0.5 0.9 0.3 0.3 0.5 0.5 1.0 0.5 0.4 0.5
GERPX 0.3 0.4 0.8 0.3 0.3 0.3 0.5 1.0 0.4 0.4
JAPPX 0.2 0.3 0.2 0.8 0.2 0.2 0.4 0.4 1.0 0.3
UKPX 0.5 0.5 0.3 0.3 0.9 0.5 0.5 0.4 0.3 1.0



Part B - *Comment on any "extreme values" (Those close to zero, -1 or 1)*

With correlations near zero, Japan's currency returns notably did not correlate well with any other currency over all periods. France and Germany had very strong correlations during period 2, possibly because they became part of the same monetary union in 1992. Although the UK also had higher correlations with its European Union peers relative to other currencies, the UK was not part of the monetary union, which may explain its relatively lower currency return correlations during all periods. High correlations between the hedged and unhedged returns for the same currency present evidence that the mean returns of hedged and unhedged currency returns are not very different. Consequently, the high correlations observed in the tables

corroborates with the t-tests, which signify that the returns are the same. There were no negative correlations during any period.

Task #3 - Determine a minimum-variance hedge ratio

Part A - *Determine the OHR for each of the 5 countries, over the 3 sample periods using linear regression with suppression of the intercept.*
Part B – *Report the unconstrained OHR, t-statistic, 95% Confidence Interval, and Constrained OHR. (The constrained OHR will just be the unconstrained limited to [0,1]).*

Panel A: 1974:1 - 2019:12

	Ex-post Optimal Hedge Ratio			
	Unconstrained			Constrained
	OHR	t-stat	95% CI	OHR
Canada	1.870	20.14	[1.689 - 2.054]	1.00
France	0.910	10.47	[.0737 - 1.077]	0.91
Germany	0.832	9.940	[0.667 - 0.996]	0.83
Japan	0.822	11.73	[0.685 - 0.960]	0.82
United Kingdom	0.999	12.46	[0.841 - 1.156]	1.00

Panel B: 1988:1 - 2019:12

	Ex-post Optimal Hedge Ratio			
	Unconstrained			Constrained
	OHR	t-stat	95% CI	OHR
Canada	1.7687	20.70	[1.601 - 1.937]	1.00
France	0.8421	9.036	[0.0 - 0.659]	0.84
Germany	0.8952	8.742	[0.694 - 1.097]	0.90
Japan	0.6197	6.975	[0.445 - 0.794]	0.62
United Kingdom	0.8990	12.02	[0.752 - 1.046]	0.90

Panel C: 1974:1 - 1987:12

	Ex-post Optimal Hedge Ratio			
	Unconstrained			Constrained
	OHR	t-stat	95% CI	OHR
Canada	2.496	8.01	[1.881 - 3.112]	1.00
France	1.054	5.59	[0.682 - 1.425]	1.00
Germany	0.697	4.830	[0.412 - 0.982]	0.70
Japan	1.192	11.10	[0.980 - 1.404]	1.00
United Kingdom	1.159	6.25	[0.793 - 1.525]	1.00

Part C - *Comment on the results. What do the t-stats tell you about the minimum variance OHR? Are they significantly different from zero? What does this mean for the philosophy that hedging is inefficient?*

The high t-stats we observe in all panels signify that the results are markedly different than zero at confidence levels greater than 95%. Since the averages of the optimal hedge ratios are all near 1 with statistical significance, it means that hedging is efficient for reducing the variance of hedged returns.

Task #4 - Rolling minimum-variance OHR - *The previous analysis is "after the fact" or "ex-post." You are concerned that someone will question the merit of your analysis or call it "academic" since you are determining an OHR on years of past data. Hence, there is no guarantee that your OHR will work for future hedging practices. As a practitioner you need to address this concern. You propose a monthly currency hedging policy based on performing the same minimum-variance OHR calculation in Task 3, but on the trailing 36 months. At the end of the 36 months you will use that minimum variance OHR, then repeat at the end of each month. At the start of the month you will recalculate the minimum-variance OHR and adjust the currency hedge. In this manner, you can implement and test an ex-ante OHR policy that evolves with changes in the market structure. You will now analyze this distribution of the generated rolling minimum-variance OHRs.*

Part A & C - *For each country report the mean, 95% confidence interval, standard deviation, minimum, Q1, median, Q3 and Max of the distribution of rolling OHRs, for each country in each period. Using the Rolling OHRs perform two tests, test the null hypothesis that the OHR=.5 and the null hypothesis that the rolling OHR = the ex-post OHR from step 3.*

Panel A: 1974:1 - 2019:12

n = 516	Ex-ante Optimal Hedge Ratio											
	36 Rolling Months								t-test Results			
	Mean	95% CI	Std Dev.	Min	Q1	Median	Q3	Max	OHR=0	OHR=0.5	OHR=1	OHR=Ex-post
Canada	1.00	[nan - nan]	0.00	1.00	1.00	1.00	1.00	1.00	-	-	-	-
France	0.70	[0.672 - 0.726]	0.31	0.00	0.47	0.76	1.00	1.00	50.57	14.43	-21.71	-(15.21)
Germany	0.65	[0.618 - 0.672]	0.31	0.00	0.44	0.67	1.00	1.00	46.90	10.56	-25.78	-(13.54)
Japan	0.67	[0.637 - 0.711]	0.43	0.00	0.03	0.96	1.00	1.00	35.46	9.17	-17.12	-(7.77)
United Kingdom	0.79	[0.765 - 0.810]	0.26	0.00	0.67	0.88	1.00	1.00	68.72	25.13	-18.45	-(18.34)

Panel B: 1988:1 - 2019:12

n = 384	Ex-ante Optimal Hedge Ratio											
	36 Rolling Months								t-test Results			
	Mean	95% CI	Std Dev.	Min	Q1	Median	Q3	Max	OHR=0	OHR=0.5	OHR=1	OHR=Ex-post
Canada	1.00	[nan - nan]	0.00	1.00	1.00	1.00	1.00	1.00	-	-	-	-
France	0.63	[0.596 - 0.660]	0.32	0.00	0.43	0.61	1.00	1.00	38.76	7.94	-22.89	-(13.16)
Germany	0.62	[0.583 - 0.652]	0.34	0.00	0.37	0.60	1.00	1.00	35.24	6.73	-21.78	-(15.80)
Japan	0.57	[0.52 - 0.610]	0.45	0.00	0.00	0.85	1.00	1.00	24.52	2.84	-18.84	-(2.35)
United Kingdom	0.73	[0.698 - 0.752]	0.27	0.00	0.54	0.78	1.00	1.00	52.37	16.27	-19.82	-(12.50)

Panel C: 1974:1 - 1987:12

n = 132	Ex-ante Optimal Hedge Ratio											
	36 Rolling Months								t-test Results			
	Mean	95% CI	Std Dev.	Min	Q1	Median	Q3	Max	OHR=0	OHR=0.5	OHR=1	OHR=Ex-post
Canada	1.00	[nan - nan]	0.00	1.00	1.00	1.00	1.00	1.00	-	-	-	-
France	0.91	[0.873 - 0.938]	0.19	0.30	0.95	1.00	1.00	1.00	55.38	24.82	-5.75	-(9.02)
Germany	0.72	[0.694 - 0.754]	0.17	0.38	0.61	0.71	0.85	1.00	47.94	14.86	-18.23	(1.82)
Japan	0.99	[0.986 - 0.996]	0.03	0.86	1.00	1.00	1.00	1.00	387.67	192.11	-3.44	-(78.61)
United Kingdom	0.97	[0.957 - 0.984]	0.08	0.59	1.00	1.00	1.00	1.00	141.88	68.85	-4.19	-(27.44)

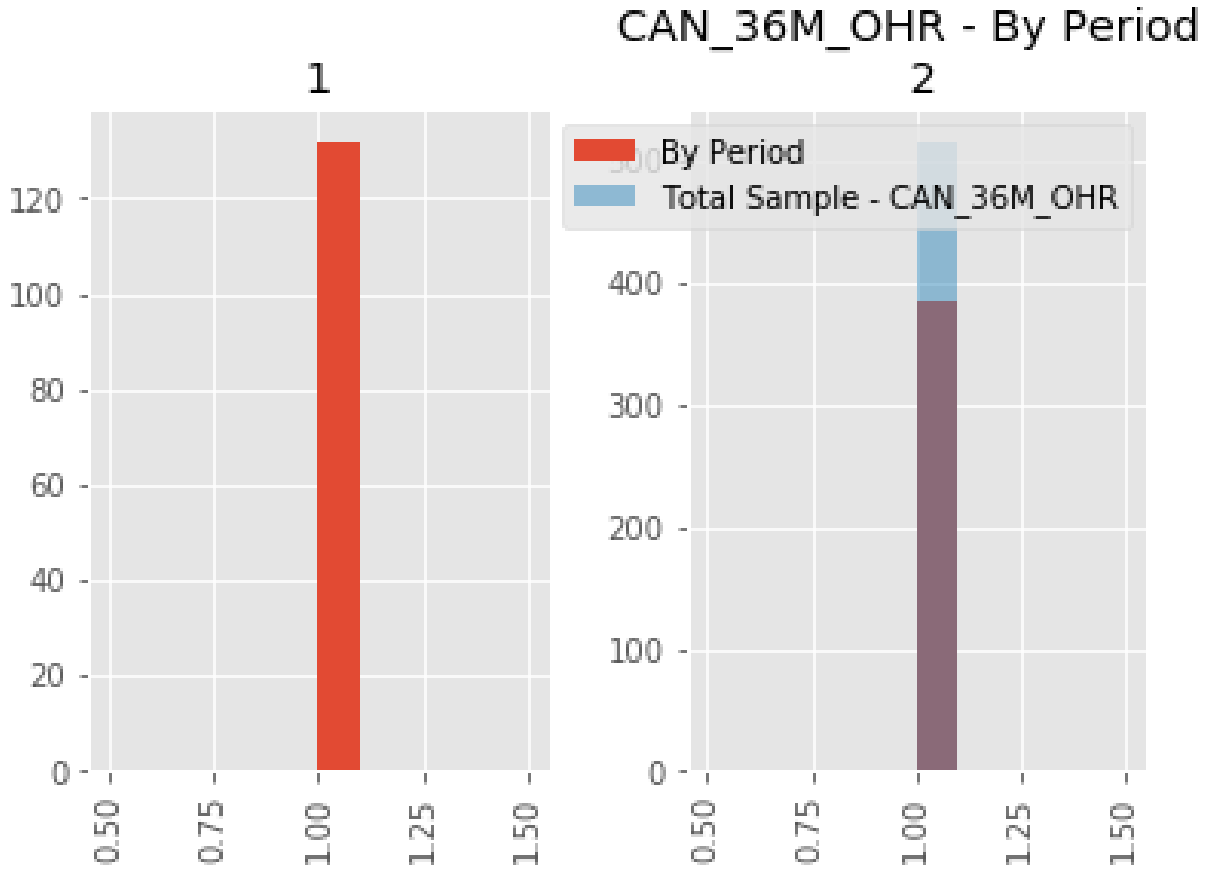
Part D - *Comment on the results. Is this rolling hedging policy any different from the ex-post methodology? How about the ad-hoc OHR=.5 standard used by many?*

The rolling hedging policy is more robust than all methodologies since it is updating OHRs over time. The ex-post methodology relies on the assumption that OHR distributions remain the same throughout time (which our findings suggest is a false assumption). Consequently, these methodologies are very different, especially evident by the statistically

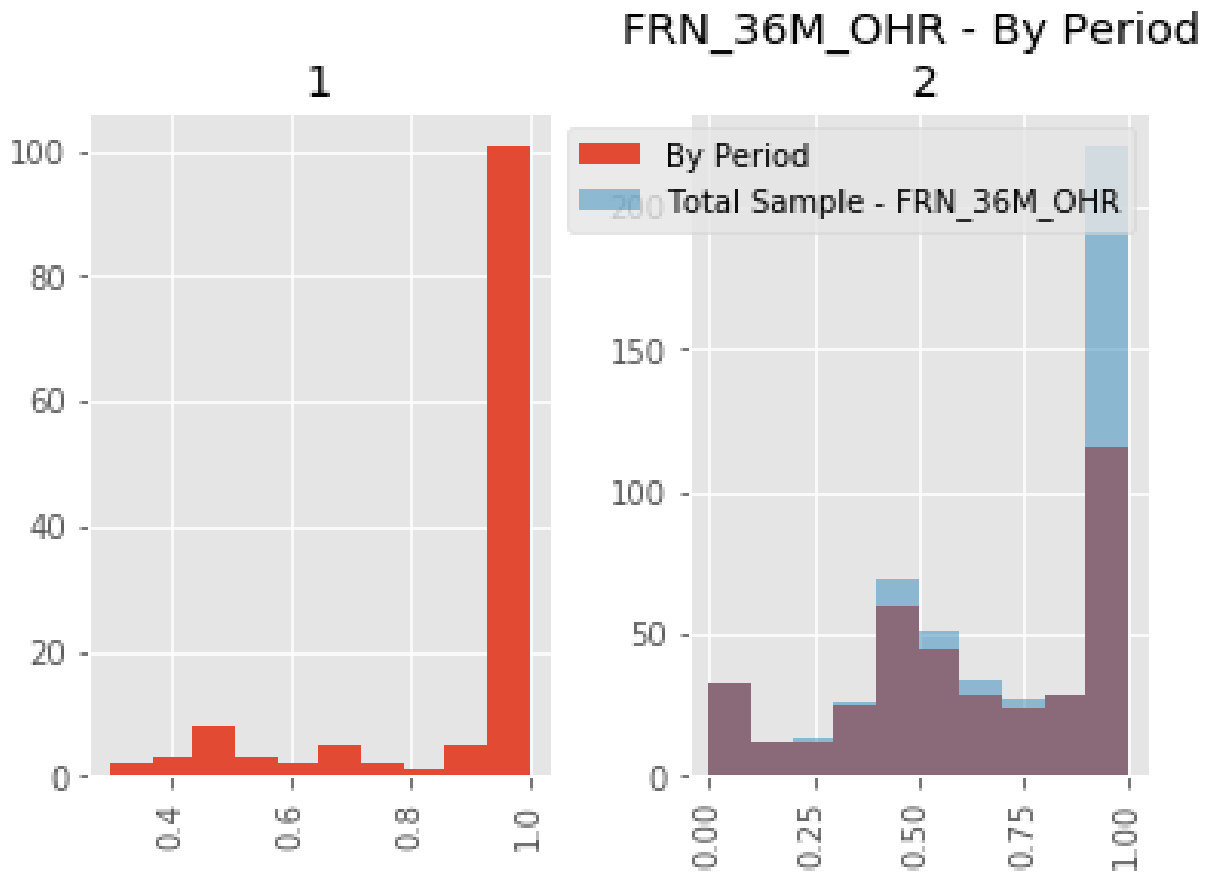
significant OHR=Ex-post t-stat. From the table, we reject the null hypothesis that the OHR = Ex-post in all cases. In other words, we can infer with a greater than a 95% confidence level that the mean return is NOT equal to the ex-post OHR. However, the ex-post methodology's t-stats were typically lower in absolute magnitude than the OHR=0 and OHR = 0.5 standards, which means that the ex-post methodology provided better OHRs than the rule of thumb standards used by many. For example, France's OHR during period one was almost definitely not zero with a t-stat of 55.38, but more likely to be the ex-post with a t-stat of -9.02.

Overall, we conclude that the rolling hedge policy is the best policy to follow. The real economic world changes rapidly, and the rolling model can better adapt to it than the ex-post methodology.

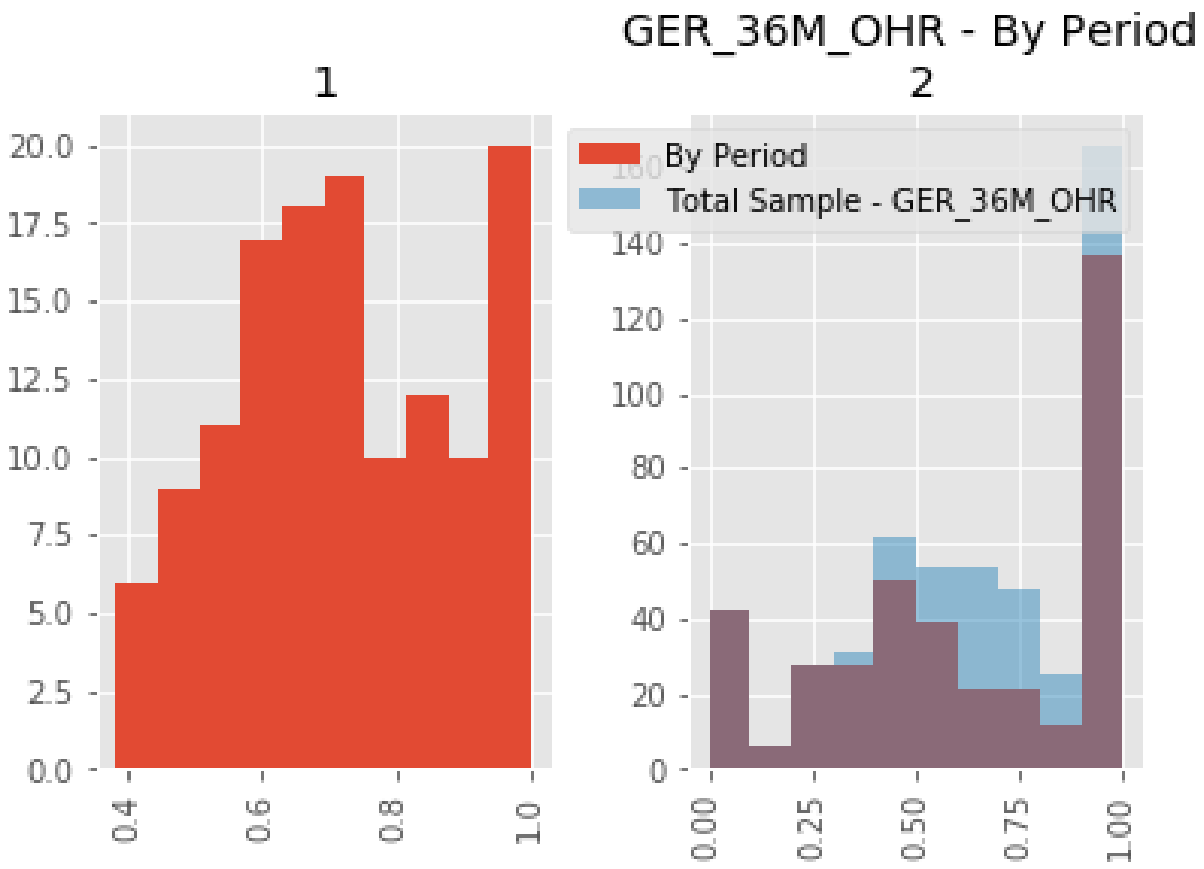
Part B – *Examine histograms of the OHRs for each country across sample periods.*



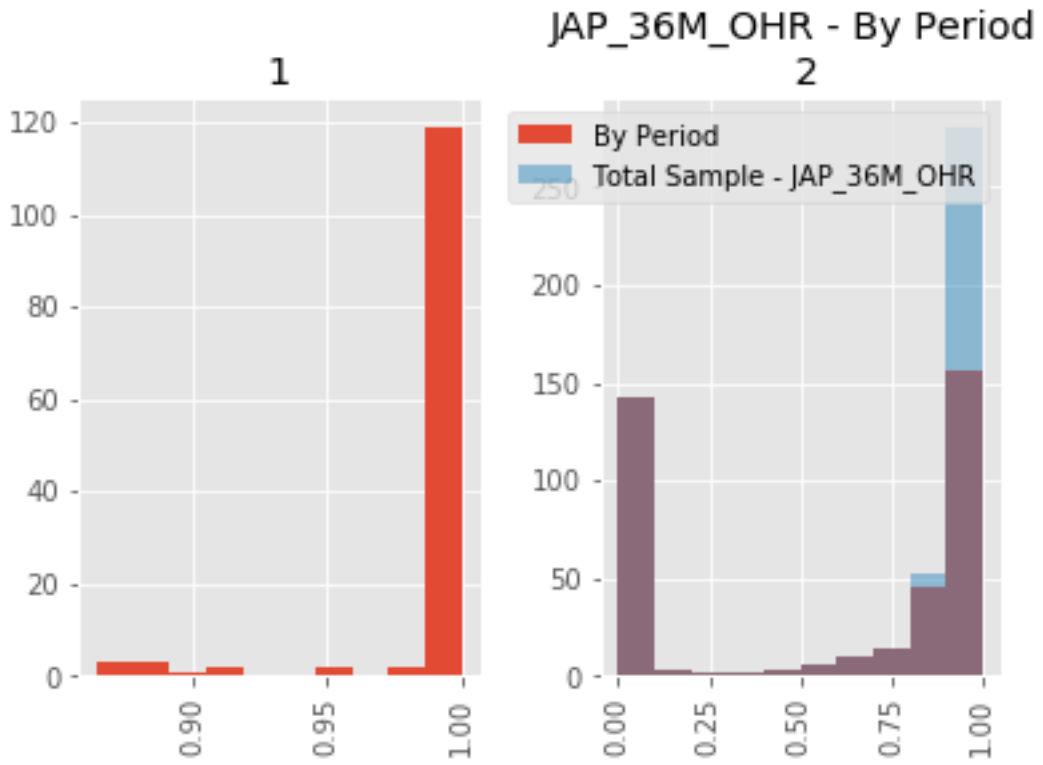
The Canadian currency OHRs were all bounded to 1 to prevent over-hedging since most of the calculated OHR values exceeded 1.



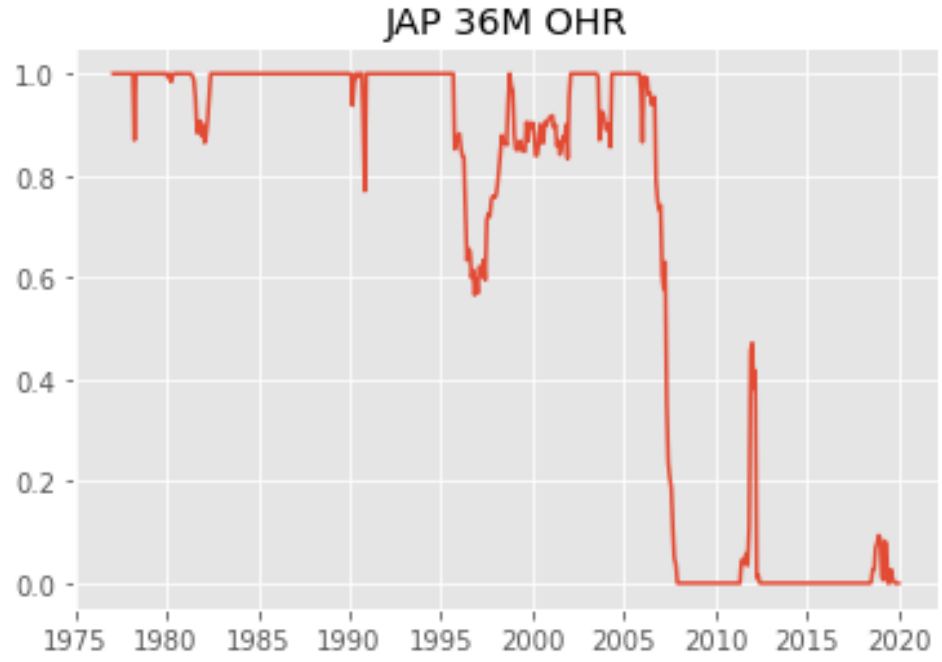
Regarding the French OHR ratios: Fully hedging exposure was highly effective most of the time during the first period. On the other hand, period 2 had much more variability in the amount of hedging required to minimize variability of returns. Covariances were likely higher during certain periods in the period 2 sample, which can reduce the need to fully hedge. The French OHR ratio distribution during period 2 closely resembles Germany's period 2 distribution, likely due to the unification of their monetary systems in 1992.

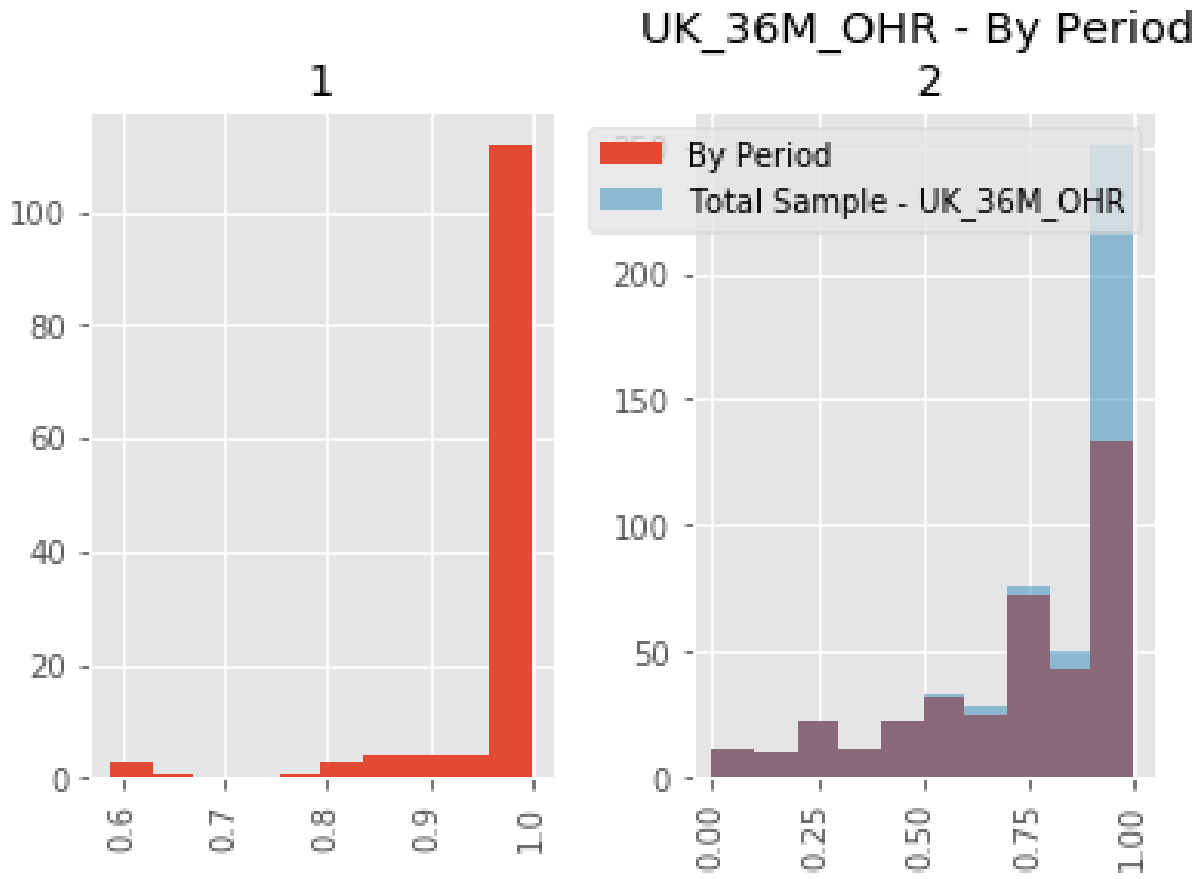


For Germany, its optimal hedge ratio distribution had high variability during period one, which reduced drastically during period 2. This drastic change in distribution presents evidence that hedge ratios are also a function of time. This is logical since the economic environments surrounding these countries and their currencies change over time. A possible catalyst for much of the change was the unification of the Euro monetary zone. Full hedges for Germany were usually more favorable during period 2.



Japan's optimal hedge ratios during period 2 were unusual in comparison to the other currencies since they formed a bimodal distribution. Consequently, there were occasions where it made sense not to hedge at all and occasions that called for full hedges, with very sparse partial hedges in between. This may signify that there are two separate populations within period 2 for Japan. Further investigation of the Japan's 36-month OHR series reveals that hedging was no longer optimal after August of 2006, which provides further evidence that our conclusions can change depending on the time window used.





Both periods of the UK's OHR distributions resembled those of France (and resembled Germany's during for period 2). One major difference between the UK and the other European countries is that the UK never joined the EU monetary union, yet its currency's usefulness as a hedge became more variable during period 2, much like the other European currencies. This suggests there are other unexplained underlying economic factors affecting the OHR distributions of European currencies.

Python Code

```
# -*- coding: utf-8 -*-
"""
Created on Mon Jul  6 18:26:12 2020

@author: Michael Frangos
"""
#Import number processing library
import numpy as np

#TASK #1

#Change working directory
import os
main_directory = 'C:/Users/Bunnita/Desktop/MSF/QMB Finance w Brian Silverstein/Project 1 (Due July 13)'
os.chdir(main_directory)

#load data
import pandas as pd
data = pd.read_excel("project1_data.xlsx")

#Create year column
data["year"] = [date.year for date in data["Date"]]

def create_period_column():
    #Initialize empty [period] column
    data["period"] = 0
    #Create date filters. This will be used to classify each date as period 1 or 2
    period_1 = (data['Date'] > '1974-1-1') & (data['Date'] <= '1988-1-1') #Set up period 1, 1974 - 1988
    period_2 = (data['Date'] > '1988-1-1') & (data['Date'] <= '2020-1-1') #Set up period 2, 1988 - 2020
    print("There is",len(data.loc[period_1]), "entries in period 1")
    print("There is",len(data.loc[period_2]), "entries in period 2")
    #By using the date filters above, replace 0 with 1 in the [period] column for each respective date
    data.loc[data[period_1].index,"period"] = data.loc[data[period_1].index, 'period'].replace(to_replace = 0,value = 1)
    data.loc[data[period_2].index,"period"] = data.loc[data[period_2].index, 'period'].replace(to_replace = 0,value = 2)

#Create the period column and append it to the dataframe
create_period_column()

#Create counter column and append it to the dataframe
data["counter"] = range(len(data))

#####Create Hedged Index#####

#Define a function to create forward premium columns for all of the forward ratios by subtracting 1. Appends the new columns to the main dataframe
def create_forward_premium_columns():
    forward_ratio_list = ['USCAN_fwd', 'USFRN_fwd', 'USGER_fwd', 'USJAP_fwd', 'USUK_fwd']
    for forward_ratio in forward_ratio_list:
        print(f"Successfully created {forward_ratio} premium column")
        data[f"{forward_ratio} premium"] = data[forward_ratio] - 1
#Execute function to create forward premium columns and append to the dataframe
create_forward_premium_columns()


```

```
#Define function to add to return index to create hedged return index. Appends the new columns to the main dataframe
def create_hedged_currency_returns_columns():
    forward_premium_list = ['USCAN_fwd premium', 'USFRN_fwd premium', 'USGER_fwd premium', 'USJAP_fwd premium', 'USUK_fwd premium']
    unhedged_local_currency_list = ['LCUH_CAN', 'LCUH_FRN', 'LCUH_GER', 'LCUH_JAP', 'LCUH_UK']
    #For each column in the unhedged local currency list and the forward ratio list, add each column together to create the hedged return index columns
    for unhedged_local_currency,forward_premium in zip(unhedged_local_currency_list,forward_premium_list):
        #print(unhedged_local_currency,forward_ratio)
        ##Create the column for US dollar based returns.
        data[f"Hedged_LC_{unhedged_local_currency[-3:]}"] = data[unhedged_local_currency] + data[forward_premium] #forward premium is the hedge.
        print(f"Successfully created and appended Hedged_LC_{unhedged_local_currency[-3:]} column")

#Execute function to create hedged return index. These are the currency returns assuming you hedged
create_hedged_currency_returns_columns()


#Import describe function to generate summary statistics
from scipy.stats import describe
#Define a function to compute summary statistics for each hedged/unhedged currency return column
def create_summary_stats_dataframe(data):
    #We will iterate through these columns
    hedged_currency_return_list = ["Hedged_LC_CAN","Hedged_LC_FRN","Hedged_LC_GER","Hedged_LC_JAP","Hedged_LC__UK"] #These are canadian market returns but in US dollar returns.
    unhedged_market_return_list = ['CANPX', 'FRNPX', 'GERPX', 'JAPPX', 'UKPX','USPX'] #These are canadian market returns but in US dollar returns.
    #Initialize empty dataframe
    dataframe = pd.DataFrame()
    #Initialize empty list to Keep track of row names
    row_name_list = []
    #We will begin by iterating through each column of hedged and unhedged returns
    for hedged_return,unhedged_return in zip(hedged_currency_return_list,unhedged_market_return_list):
        #Compute summary statistics for the hedged return column
        summary = describe(data[f"{hedged_return}"], axis=0)

        #Creates first row of the summary table if it doesn't exist yet
        if len(dataframe) == 0:
            #Create the summary table by initiating the first row.
            dataframe = pd.DataFrame([summary], columns=summary._fields)
            #Generate and append the first unhedged return summary stats row to the summary table

            unhedged_summary = describe(data[f"{unhedged_return}"], axis=0)
            dataframe = dataframe.append(pd.DataFrame([unhedged_summary], columns=unhedged_summary._fields))
        #If first row is already created, created the new rows
        else:
            #append hedged return summary row to the summary statistics table
            dataframe = dataframe.append(pd.DataFrame([summary], columns=summary._fields))
            print(dataframe)

            #append unhedged return summary row to the summary statistics table
            unhedged_summary = describe(data[f"{unhedged_return}"], axis=0)
            dataframe = dataframe.append(pd.DataFrame([unhedged_summary], columns=unhedged_summary._fields))
        #Keep track of each row name. We will use this list to name the rows later
```

```
        row_name_list.append(f"{hedged_return}")
        row_name_list.append(f"{unhedged_return}")

    #Set the names of each row to the dataframe
    dataframe.index = row_name_list

    return dataframe

#Filter the data by [period] column so we can tabulate results for each [period] column
period_1_data = data.loc[data["period"] == 1]
period_2_data = data.loc[data["period"] == 2]

#Create summary statistics tables
total_sample_summary_stats_dataframe = create_summary_stats_dataframe(data)
period_1_summary_stats_dataframe = create_summary_stats_dataframe(period_1_data)
period_2_summary_stats_dataframe = create_summary_stats_dataframe(period_2_data)

#Export summary statistics tables to excel
total_sample_summary_stats_dataframe.to_excel("total_sample_summary_stats_dataframe.xlsx")
period_1_summary_stats_dataframe.to_excel("period_1_summary_stats_dataframe.xlsx")
period_2_summary_stats_dataframe.to_excel("period_2_summary_stats_dataframe.xlsx")

###Perform t_test###
#import t-test function
from scipy.stats import ttest_rel

#Defines function to perform t-
tests on TWO RELATED samples of scores, hedged vs unhedged currency returns,
#to see if their mean returns are different.
#Iterates through hedged/unhedged currency return columns and returns t-stat tables.
def perform_t_tests_on_data(data):
    unhedged_market_return_list = ['CANPX', 'FRNPX', 'GERPX', 'JAPPX', 'UKPX']
    hedged_currency_return_list = ["Hedged_LC_CAN","Hedged_LC_FRN","Hedged_LC_GER","Hedged_LC_JAP"
,"Hedged_LC__UK"]
    t_statistic_results_list = []
    p_statistic_results_list = []
    row_name_list = []
    #For each set of columns, Calculate the t-test on TWO RELATED samples of scores, a and b.
    for unhedged_return_array, hedged_return_array in zip(unhedged_market_return_list,hedged_curre
ncy_return_list):
        #Perform t-test
        t, p = ttest_rel(data[unhedged_return_array],data[hedged_return_array])
        #Keep track of the results in lists
        t_statistic_results_list.append(t)
        p_statistic_results_list.append(p)
        row_name_list.append(unhedged_return_array[:-2])

    #Compile the results by appending our lists together into a dataframe
    results = pd.DataFrame({"t":t_statistic_results_list,
                            "p":p_statistic_results_list},
                            index = row_name_list)
    return results

#Perform t-
tests for hedged/unhedged currency returns during period 1, period 2, and the total sample.
t_test_total_sample_dataframe = perform_t_tests_on_data(data)
t_test_period_1_dataframe = perform_t_tests_on_data(period_1_data)
t_test_period_2_dataframe = perform_t_tests_on_data(period_2_data)

#Export t-test tables to excel
#Perform t-tests
```

```
t_test_total_sample_dataframe.to_excel("t_test_total_sample_dataframe.xlsx")
t_test_period_1_dataframe.to_excel("t_test_period_1_dataframe.xlsx")
t_test_period_2_dataframe.to_excel("t_test_period_2_dataframe.xlsx")


#TASK #2

#Import libraries for data visualization
import matplotlib.pyplot as plt
from matplotlib import style
style.use('ggplot')

#Create currency correlation tables
def create_correlation_tables():
    #Create a list of column names
    columns = data[["Hedged_LC_CAN","Hedged_LC_FRN","Hedged_LC_GER"
                        ,"Hedged_LC_JAP","Hedged_LC__UK",'CANPX',
                        'FRNPX', 'GERPX', 'JAPPX', 'UKPX']].columns

    #Create correlation table
    total_sample_corr_table = data[columns].corr()
    #Create correlation table
    period_1_corr_table = period_1_data[columns].corr()
    #Create correlation table
    period_2_corr_table = period_2_data[columns].corr()

    return total_sample_corr_table, period_1_corr_table, period_2_corr_table

#Create currency correlation tables
total_sample_corr_table, period_1_corr_table, period_2_corr_table = create_correlation_tables()

#Visualize and export hedged and unhedged currency correlation tables
def visualize_currency_correlation_tables():
    #Create a list of column names
    columns = data[["Hedged_LC_CAN","Hedged_LC_FRN","Hedged_LC_GER"
                        ,"Hedged_LC_JAP","Hedged_LC__UK",'CANPX',
                        'FRNPX', 'GERPX', 'JAPPX', 'UKPX']].columns
    #Place the column names into a list to make the correlation table visualization.
    labels = [x for x in columns]

    #Creates lists for that contain the numerical tables and their names
    tables = [total_sample_corr_table, period_1_corr_table, period_2_corr_table]
    table_names = ["Total Sample Correlations", "Period 1 Correlations", "Period 2 Correlations"]


    for table,table_name in zip(tables,table_names):
        #Create empty figure objects that contain visuals. We will populate them with the next few
    lines of code
        fig = plt.figure(figsize = (12,12))
        ax = fig.add_subplot(111)
        #Select the table used to create the visualization, and populate the figure object
        ax.matshow(table,cmap = plt.cm.RdYlGn,alpha=0.4)
        #Set the names of the x ticks & y ticks
        ax.set_yticklabels(labels)
        ax.set_xticklabels(labels)
        #Set title
        plt.title(f"{table_name}", y=1.14)
        #Force show all of the labels since they are being truncated by default
        ax.set_xticks(np.arange(len(labels)))
        ax.set_yticks(np.arange(len(labels)))
```

```
218.
219.        #Rotate the xticks vertically
220.        ax.set_xticklabels(labels,rotation=90)
221.
222.        #Visualize the numbers onto the correlation table
223.        numbers_to_vizualize = np.array(table)
224.        for (i, j), z in np.ndenumerate(numbers_to_vizualize):
225.            ax.text(j, i, '{:0.1f}'.format(z), ha='center', va='center')
226.
227.        #Export numerical tables
228.        table.to_excel("Total_Sample_Correlations.xlsx")
229.        #Export Table visualizations
230.        fig.savefig(f"{table_names}.png")
231.
232.#Visualize and export hedged and unhedged currency correlation tables
233.visualize_currency_correlation_tables()
234.
235.#Task #3
236.'''''
237.#Create the independent variables for the linear regression.
238.'''
239.
240.#Import libraries for regression
241.import statsmodels.api as models
242.def perform_regressions(dataset, data_set_name):
243.    #Set up lists to iterate through to generate our new variables
244.    independent_var_name_list = ["x_can","x_frn","x_ger","x_jap","x_uk"]
245.    unhedged_market_return_list = ['CANPX', 'FRNPX', 'GERPX', 'JAPPX', 'UKPX']
246.    currency_return_list = ['CUR_RET_CAN', 'CUR_RET_FRN', 'CUR_RET_GER', 'CUR_RET_JAP','CUR_RET_UK
    ']
247.    forward_premium_list = ['USCAN_fwd premium', 'USFRN_fwd premium', 'USGER_fwd premium', 'USJAP_
    fwd premium', 'USUK_fwd premium']
248.    #Initialize empty lists to populate with standard errors and OHR by currency
249.    standard_error_list = []
250.    Optimal_Hedge_ratio_list = []
251.    #Iterate through all three lists at the same time
252.    for independent_var,unhedged_return,currency_return, forward_premium in zip(independent_var_na
    me_list,unhedged_market_return_list,currency_return_list,forward_premium_list):
253.        #Print new line, name and variable
254.        print(dataset_name, f" - {independent_var}")
255.        #The dependent variables are unhedged returns
256.        #Create the independent variable columns and append to the main dataframe
257.        dataset[f"{independent_var}"] = dataset[currency_return] - dataset[forward_premium]
258.        #Set up for regression
259.        #x = data[[f"{independent_var}"]]
260.        x = dataset[[f"{independent_var}"]]
261.        y = dataset[[f"{unhedged_return}"]]
262.
263.        # Fit & print regression model
264.        regr = models.OLS(y, x).fit()
265.        print_model = regr.summary()
266.        print(print_model)
267.
268.        standard_error_list.append(regr.bse)
269.        Optimal_Hedge_ratio_list.append(regr.params)
270.
271.    return standard_error_list, Optimal_Hedge_ratio_list
272.
273.
274.#Update date filters
275.#Filter the data by [period] column so we can tabulate results for each [period] column
```

```
period_1_data = data.loc[data["period"] == 1]
period_2_data = data.loc[data["period"] == 2]

#Execute regressions for all currencies and all data sets. #Beta is the optimal hedge ratio
list_of_datasets = [data, period_1_data, period_2_data]
list_of_dataset_names = ["Entire Sample", "Period 1", "Period 2"]
for dataset, dataset_name in zip(list_of_datasets,list_of_dataset_names):
    perform_regressions(dataset,dataset_name)
```

```
####TASK #4 ####
#PART A

#Perform regression over 36 months and sliding by 1 increment
t = 0 #Time
window = 36
Trailing_36m_SE_row = []  #We will populate this empty list with each iteration's results by currency. Ex. [SE_can,SE_UK ....]
Trailing_36m_OHR_row = [] #We will populate this empty list with each iteration's results by currency.
#Iterate through the dataset using 36 month windows to calculate rolling regressions
for t in range(len(data)-36):
    print(t)
    #Slice and roll through the dataset by the window index
    sliced_dataset = data[t:t+window]
    print(sliced_dataset)
    #Execute regressions for all currencies. #Beta is the optimal hedge ratio
    standard_error, Optimal_Hedge_Ratio = perform_regressions(sliced_dataset,"Entire Sample")
    #Keep track of the results. We will append these new columns to our dataframe at the end.
    Trailing_36m_SE_row.append(standard_error)
    Trailing_36m_OHR_row.append(Optimal_Hedge_Ratio)

#Now that the rolling regressions have been calculated, let's append them to our dataframe
def append_rolling_regression_SE_columns_to_dataframe():
    #Initialize empty lists and generate NANs by the size of the window
    CAN_36M_SE, FRN_36M_SE, GER_36M_SE, JAP_36M_SE, UK_36M_SE = [np.nan for x in range(window)],[np.nan for x in range(window)],[np.nan for x in range(window)],[np.nan for x in range(window)],[np.nan for x in range(window)]

    #Populate the empty lists with the calculated data
    i=0
    for CAN, FRN, GER, JAP, UK in Trailing_36m_SE_row:
        print("\n",CAN, FRN, GER, JAP, UK,i)
        i=i+1
        CAN_36M_SE.append(float(CAN))
        FRN_36M_SE.append(float(FRN))
        GER_36M_SE.append(float(GER))
        JAP_36M_SE.append(float(JAP))
        UK_36M_SE.append( float(UK) )
    #Append columns to the main dataframe
    data["CAN_36M_SE"] = CAN_36M_SE
    data["FRN_36M_SE"] = FRN_36M_SE
    data["GER_36M_SE"] = GER_36M_SE
```

```
    data["JAP_36M_SE"] = JAP_36M_SE
    data["UK_36M_SE"] = UK_36M_SE

#Now that the rolling regressions have been calculated, let's append them to our dataframe
def append_rolling_regression_coefficient_columns_to_dataframe():
    #Initialize empty lists
    CAN_36M_OHR, FRN_36M_OHR, GER_36M_OHR, JAP_36M_OHR, UK_36M_OHR = [np.nan for x in range(window
    )],[np.nan for x in range(window)],[np.nan for x in range(window)],[np.nan for x in range(window)]
    ,[np.nan for x in range(window)]
    #Populate the empty lists
    for CAN, FRN, GER, JAP, UK in Trailing_36m_OHR_row:
        CAN_36M_OHR.append(float(CAN))
        FRN_36M_OHR.append(float(FRN))
        GER_36M_OHR.append(float(GER))
        JAP_36M_OHR.append(float(JAP))
        UK_36M_OHR.append(float(UK))
    #Append columns to the main dataframe
    data["CAN_36M_OHR"] = CAN_36M_OHR
    data["FRN_36M_OHR"] = FRN_36M_OHR
    data["GER_36M_OHR"] = GER_36M_OHR
    data["JAP_36M_OHR"] = JAP_36M_OHR
    data["UK_36M_OHR"] = UK_36M_OHR

#Execute functions: Append the new columns to the dataframe
append_rolling_regression_SE_columns_to_dataframe()
append_rolling_regression_coefficient_columns_to_dataframe()

#Replace values in the rolling regressions columns to prevent overhedging. Then print histograms
_36M_OHR_list = ["CAN_36M_OHR","FRN_36M_OHR","GER_36M_OHR","JAP_36M_OHR","UK_36M_OHR"]
for column in _36M_OHR_list :
    #Replace values less than zero with zero
    data[f"{column}"].loc[(data[f"{column}"] < 0 )] = 0
    #Replace values greater than one with one
    data[f"{column}"].loc[(data[f"{column}"] > 1 )] = 1
    #Print histograms
    plt.hist(data[f"{column}"])
    plt.title(f"{column}")
    plt.show()


#Tabstat Optimal Hedge Ratios
subdir_1 = "OHR Regression Tabstats"
Total_Period = pd.DataFrame(data[_36M_OHR_list].describe().T)
a = data[["CAN_36M_OHR","period"]].groupby("period").describe()
b = data[["FRN_36M_OHR","period"]].groupby("period").describe()
c = data[["GER_36M_OHR","period"]].groupby("period").describe()
d = data[["JAP_36M_OHR","period"]].groupby("period").describe()
e = data[["UK_36M_OHR","period"]].groupby("period").describe()
#Export OHR tabstats to excel
File_names = ['Total_Sample','CAN_36M_OHR','FRN_36M_OHR','GER_36M_OHR','JAP_36M_OHR','UK_36M_OHR']

OHR_Reg_tabstats = [Total_Period,a,b,c,d,e]
for file, filename in zip (OHR_Reg_tabstats,File_names):
    file.to_excel(f"{main_directory}/{subdir_1}/{filename}.xlsx")


#Calculate confidence intervals for the OHR rolling Regressions
def calculate_OMH_regression_confidence_intervals():
    a_group = data[["CAN_36M_OHR","period"]].groupby("period")
```

```
    b_group = data[["FRN_36M_OHR","period"]].groupby("period")
    c_group = data[["GER_36M_OHR","period"]].groupby("period")
    d_group = data[["JAP_36M_OHR","period"]].groupby("period")
    e_group = data[["UK_36M_OHR","period"]].groupby("period")

    #Print confidence intervals for periods
    groups = [a_group,b_group,c_group,d_group,e_group]
    group_names = ['CAN_36M_OHR','FRN_36M_OHR','GER_36M_OHR','JAP_36M_OHR','UK_36M_OHR']
    from scipy.stats import norm
    for group,group_name in zip(groups,group_names):
        mu = group.mean()
        standard_error = group.std()/np.sqrt(group.count())
        confidence_interval = lower, upper = norm.interval(0.95, loc=mu, scale=standard_error)
        print(group_name, "Lower Confidence Intervals:", list(lower), "Upper Confidence Intervals:
", upper)

    #Print confidence intervals for total sample for each country
    Total_Sample_group = pd.DataFrame(data[_36M_OHR_list])
    mu = Total_Sample_group.mean()
    standard_error = Total_Sample_group.std()/np.sqrt(Total_Sample_group.count())
    confidence_interval = lower, upper = norm.interval(0.95, loc=mu, scale=standard_error)
    print("Total_Sample_Group", "Lower Confidence Intervals:", list(lower), "Upper Confidence Inte
rvals:", upper)

#Execute function
calculate_OMH_regression_confidence_intervals()


#PART B
#Print histograms by currency & period group
_36M_OHR_list = ["CAN_36M_OHR","FRN_36M_OHR","GER_36M_OHR","JAP_36M_OHR","UK_36M_OHR"]
for column in _36M_OHR_list:
    ###By period - Red
    data[f"{column}"].hist(by=data['period'],alpha=1)
    plt.title(f"{column} - By Period \n 2")

    ###Total sample - Blue
    data[f"{column}"].hist(alpha=.50)
    plt.legend(["By Period", f"Total Sample - {column}"])
    plt.show()



#Japan OHR plot
plt.plot(data["Date"],data["JAP_36M_OHR"])
plt.title("JAP 36M OHR")

#Export ending dataset

Ending_dataset = data.to_excel("Ending_dataset.xlsx")
```