

A Programmatic Analysis of Brexit Events

PROGRAMMING FOR FINANCE FINAL

qt890669

Table of Contents

Introduction	2
Question 1.....	2
Table 1 – Key Brexit Events.....	2
Figure 1 – Annotated Timeline	3
Question 2.....	3
Table 2 – Key Dates Returns	3
Table 3 – Regression Results.....	4
Table 4 – Correlations Between Key Event Indicator and Asset Class	4
Question 3.....	4
Table 5 – Returns on Key Dates	5
Table 6 – Regression Results.....	5
Table 7 – Correlations Between Key Event Indicator and Asset Class	6
Question 4.....	6
Table 7 – Returns +/- 5 Days Around June 24 th	6
Figure 7 – Returns	7
Question 5.....	7
Table 8 – Key Event Classifications.....	8
Table 9 – Surrounding Day Classification.....	8
Table 10 – Surrounding Day Classification All Assets	9
Table 11 – Regression Results.....	10
Conclusion.....	10
Appendix 1	11
Appendix 2	13
Appendix 3	15
Appendix 4	17
Appendix 5	19

Introduction

In this report, I present my findings on the financial market's reaction to several key events leading up to and following the United Kingdom's referendum to leave the European Union using data across multiple asset classes, all processed using the Python programming language and associated libraries. The appendices contain all code used in the production of this report.

Question 1

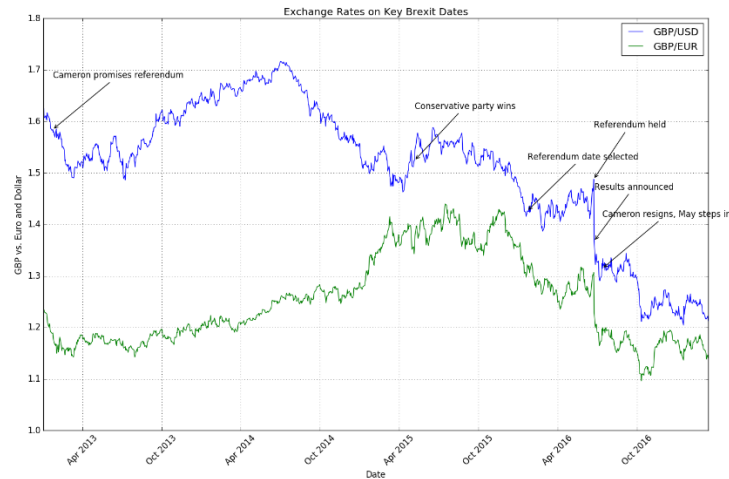
The first step in determining market reactions to events is to select those meaningful events the UK Government took in leaving the EU. I selected for dates that had meaning in some way in direct relationship to the Brexit vote, beginning with early 2013 when David Cameron promised a referendum. My sample ends shortly after Cameron resigns, though significant developments have occurred recently with the formal process for leaving the EU¹ having been triggered. Table 1 provides a list of the events with a description of each, and Figure 1 shows an annotated timeline of each event on both the GBP/USD and GBP/EUR spot exchange rates. Appendix 2 contains the code used to produce Figure 1.

Table 1 – Key Brexit Events

Dates	Description
1/23/2013	Cameron Promises Referendum
5/7/2015	Conservative party wins re-election
1/22/2016	Referendum date selected
6/23/2016	Referendum held
6/24/2016	Results announced
7/13/2016	Cameron resigns, Theresa May takes over

¹ Article 50, the formal process for leaving the EU, was triggered on March 29th, 2017. This fell beyond the scope of my data and thus doesn't factor into my analysis, though casual analysis suggests that the actual triggering of Article 50 had little effect compared to other events.

Figure 1 – Annotated Timeline



Question 2

As Figure 1 already shows, the foreign exchange markets have not responded favorably to the prospect of Britain leaving the EU. I have further investigated the returns on other markets, such as the UK 30-year gilts, as well as the FTSE. Table 2 provides a list of returns by asset class on key dates, along with summary statistics. Table 2 suggests that only the last two dates were negative as measured by the most indicators, with the exception of the UK Gilts, which is expected, as investors shift from now-riskier equities to safer government debt. The events on the 24th of June and the 13th of July are both substantially negative, enough so to drag the average return across all the events down. We can see this outlier effect in the median for each asset class, which are all positive. Appendix 3 contains the relevant code.

Table 2 – Key Dates Returns

	GBPUSD	GBPEUR	FTSE	UKGILT
1/23/2013	0.000442	0.000673	0.002989	0.004385
5/7/2015	0	0.007371	-0.00675	0.008292
1/22/2016	0.003236	0.011167	0.021861	-0.00446
6/23/2016	0.011902	0.003995	0.012284	-0.00799
6/24/2016	-0.08066	-0.06	-0.03146	0.048839
7/13/2016	-0.00733	-0.01011	-0.00154	0.015226
Average	-0.01207	-0.00782	-0.00044	0.010716
Median	0.000221	0.002334	0.000724	0.006338
Std. Dev.	0.034169	0.026574	0.018299	0.020492

I also perform four regressions of the following form:

$$y = \alpha + \beta_1 x$$

Where y is the set of all variables, α is a constant, and x is simply a dummy variable for key events taking 1 on key event dates and zero otherwise. I extract each regression's coefficient, p-value, and r-squared. Table 3 presents a summary of the findings. The regressions suggest significant negative returns on key event dates for foreign exchange, insignificant (but negative) returns for the FTSE, and positive and significant returns for sovereign debt.

Table 3 – Regression Results

Variable	Coefficient	P-Value	R-Squared
GBPUSD	-0.011927	1.8883E-07	0.020782
GBPEUR	-0.007829	3.0045E-04	0.010059
FTSE	-0.000656	8.5740E-01	0.000025
UKGILT	0.01088	4.1627E-02	0.003206

In evaluating correlation between the key events variable and the above variables, I present relationships between asset classes and the key variable indicator in Table 4. This table reaffirms the findings from the regressions, primarily that foreign exchange declines on key events, sovereign debt prices rise, and the FTSE tends to decline slightly.

Table 4 – Correlations Between Key Event Indicator and Asset Class

	GBPUSD	GBPEUR	UKGILT	FTSE
KY	-0.14416	-0.1003	0.056621	-0.005

Markets obviously have an unfavorable view of Brexit. The fact that returns for each of the variables in Table 3 decline on key events dates suggests that the market expects the quality of Britain's economy to decline.

Question 3

UK assets performed poorly, as we saw in the previous section. But what about other markets across the globe? Tables 5-7 present information similar to the tables in Question 2, with additional asset classes added. The key takeaway from these tables is that key events are statistically significant for asset classes linked to UK macroeconomic factors (exchange rates, sovereign debt) and indicate a downturn in the economy. It is less clear what the effect is on global marketplaces, as no regression is significant at the 5% level for any exchange index

except for the Italian FTMIB. The coefficients suggest a negative relationship between key event dates and returns on European indices, and slight positive relationships in American and Russian markets. Appendix 3 contains code relevant to this question.

Table 5 – Returns on Key Dates

	GBPUSD	GBPEUR	FTSE	SNP	UKGILT	HSI	DAX	CAC	FTMIB	MICEX
1/23/2013	0.000442	0.000673	0.002989	0.001507	0.004385	-0.00101	0.001472	-0.00397	-0.00773	0.008754
5/7/2015	0	0.007371	-0.00675	0.003774	0.008292	-0.0127	0.005094	-0.00288	0.008035	-0.0148
1/22/2016	0.003236	0.011167	0.021861	0.020284	-0.00446	0.029034	0.01992	0.030974	0.016301	0.02939
6/23/2016	0.011902	0.003995	0.012284	0.013364	-0.00799	0.003521	0.018466	0.019605	0.037112	0.01648
6/24/2016	-0.08066	-0.06	-0.03146	-0.03592	0.048839	-0.02919	-0.06823	-0.08042	-0.12481	-0.02048
7/13/2016	-0.00733	-0.01011	-0.00154	0.000135	0.015226	0.0046	-0.00335	0.000896	-0.01155	-0.00238
Average	-0.01207	-0.00782	-0.00044	0.000524	0.010716	-0.00096	-0.00444	-0.00597	-0.01377	0.002826
Median	0.000221	0.002334	0.000724	0.002641	0.006338	0.001256	0.003283	-0.00099	0.000154	0.003184
Std. Dev.	0.034169	0.026574	0.018299	0.019469	0.020492	0.019419	0.032603	0.039047	0.057166	0.019017

Table 6 – Regression Results

Variable	Coefficient	P-Value	R-Squared
GBPUSD	-0.01193	0.0000	0.0208
GBPEUR	-0.00783	0.0003	0.0101
FTSE	-0.00066	0.8574	0.0000
UKGILT	0.01088	0.0416	0.0032
SNP	0.000092	0.9772	0.0000
HSI	-0.00111	0.7948	0.0001
DAX	-0.00494	0.3051	0.0008
CAC	-0.00635	0.1951	0.0013
FTMIB	-0.0141	0.0301	0.0036
MICEX	0.002615	0.5990	0.0002

Table 7 – Correlations Between Key Event Indicator and Asset Class

	GBPUSD	GBPEUR	FTSE	SNP	UKGILT	HSI	DAX	CAC	FTMIB	MICEX
KY	0.144161	0.100297	0.004998	0.000794	0.056621	0.007233	0.028521	0.036028	0.060269	0.014624

Question 4

To investigate a single event more thoroughly, Table 7 charts the returns of all asset classes for 5 days before and after the June 24th, 2016 event. The UK Gilt returns show a steady stream of negative values leading up to the 24th, and immediately switch the day of to become positive returns. Returns across asset classes for the 24th and the following trading day, the 27th, are all negative returns, though they quickly reverse and resume increasing.

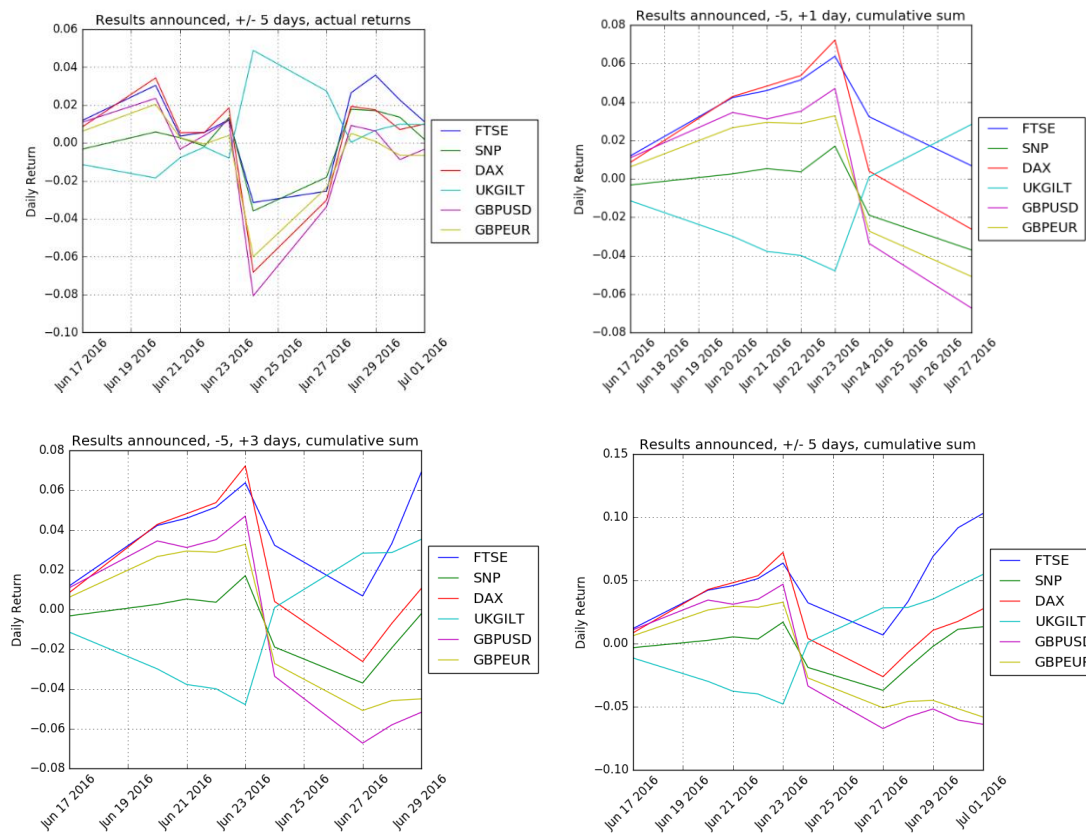
Appendix 4 contains all relevant code for this question.

Table 7 – Returns +/- 5 Days Around June 24th

	GBPUSD	GBPEUR	FTSE	SNP	UKGILT	HSI	DAX	CAC	FTMIB	MICEX	KY
6/17/2016	0.0109	0.0062	0.0119	-0.0033	-0.0114	0.0066	0.0085	0.0098	0.0349	0.0094	0
6/20/2016	0.0235	0.0204	0.0304	0.0058	-0.0185	0.0169	0.0343	0.0350	0.0254	0.0109	0
6/21/2016	-0.0033	0.0028	0.0036	0.0027	-0.0078	0.0077	0.0054	0.0061	0.0045	-0.0007	0
6/22/2016	0.0040	-0.0006	0.0056	-0.0017	-0.0021	0.0061	0.0055	0.0029	-0.0062	-0.0013	0
6/23/2016	0.0119	0.0040	0.0123	0.0134	-0.0080	0.0035	0.0185	0.0196	0.0371	0.0165	1
6/24/2016	-0.0807	-0.0600	-0.0315	-0.0359	0.0488	-0.0292	-0.0682	-0.0804	-0.1248	-0.0205	1
6/27/2016	-0.0336	-0.0236	-0.0255	-0.0181	0.0274	-0.0016	-0.0302	-0.0297	-0.0394	-0.0236	0
6/28/2016	0.0092	0.0050	0.0264	0.0178	0.0003	-0.0027	0.0193	0.0261	0.0330	0.0043	0
6/29/2016	0.0063	0.0008	0.0358	0.0170	0.0066	0.0131	0.0175	0.0260	0.0221	0.0175	0
6/30/2016	-0.0088	-0.0065	0.0227	0.0136	0.0099	0.0175	0.0071	0.0100	0.0157	0.0019	0
7/1/2016	-0.0034	-0.0066	0.0113	0.0019	0.0096	0.0000	0.0099	0.0086	0.0061	0.0031	0
Average	-0.0058	-0.0053	0.0094	0.0012	0.0050	0.0034	0.0025	0.0031	0.0008	0.0016	-
Median	0.0040	0.0008	0.0119	0.0027	0.0003	0.0061	0.0085	0.0098	0.0157	0.0031	-
Std. Dev	0.0288	0.0211	0.0214	0.0162	0.0192	0.0128	0.0282	0.0324	0.0472	0.0133	-

Figure 7 plots four graphs. The first shows actual returns plus and minus five days, the second shows cumulative returns to one day after the event, the third shows cumulative returns three days after the event, and the fourth shows cumulative returns five days after the event.

Figure 7 – Returns



The graphs in Figure 7 indicate several things. First, there's little to show that the market expected the outcome of the referendum, with the precipitous drop in both actual and cumulative returns reflecting an information shock. Second, some markets weathered the results of the vote better than others. The FTSE, for example, seems to have resumed its growth, while the pound's exchange rates have lowered and remained there, and even shown a slight decline. Third, the results of the referendum seem to have introduced uncertainty across the market: the actual returns for the period following the event are less cointegrated than before, and demonstrate more spread.

Question 5

The day the results were announced was not a good one for financial markets. Every asset class (except sovereign debt) declined, and some indicators such as foreign exchange have stayed low and even continued to decline. It's easy to say that this day was a negative one because there is overwhelming evidence. But what is the best way to formalize whether an event is positive or negative?

I present a method to determine whether a day's returns are out of the ordinary for the past thirty days. For every day in the sample, a rolling mean and standard deviation is computed using the previous 30 day's returns on the GBP/EUR exchange rate. If a day's actual return is more than the sum of twice the standard deviation and the mean, then it is classified as a positive event. If it is less than that sum, it's negative. If the actual return falls within two

standard deviations, it is classified as neutral. The code in the Appendix 5 is designed to be easily modifiable; the standard deviation metric, rolling window size, and measured variable. Table 8 presents the classifications of key events.

Table 8 – Key Event Classifications

Date	Mean	Stdev	GBPEUR	Classification	Description
1/23/2013	-0.00137	0.00354	0.00067	Neutral	Cameron promises referendum
5/7/2015	-0.00030	0.00597	0.00737	Positive	Conservative party wins
1/22/2016	-0.00191	0.00539	0.01117	Positive	Referendum date selected
6/23/2016	0.00100	0.00799	0.00400	Neutral	Referendum held
6/24/2016	0.00098	0.00798	-0.06000	Negative	Results announced
7/13/2016	-0.00266	0.01400	-0.01011	Neutral	Cameron resigns, May steps in

This method does a fair job at classifying events, and can be used on any element in the sample except for the first 30 days. Table 9 presents five days before and after the June 24th event, with event classifications included. Table 10 shows the same set of days with event classification included across all asset classes.

Table 9 – Surrounding Day Classification

Date	Mean	Stdev	GBPEUR	KY	Classification
6/17/2016	-0.00010	0.00711	0.00617	0	Neutral
6/20/2016	0.00025	0.00715	0.02043	0	Positive
6/21/2016	0.00090	0.00804	0.00277	0	Neutral
6/22/2016	0.00088	0.00804	-0.00061	0	Neutral
6/23/2016	0.00100	0.00799	0.00400	1	Neutral
6/24/2016	0.00098	0.00798	-0.06000	1	Negative
6/27/2016	-0.00102	0.013701	-0.02361	0	Negative
6/28/2016	-0.00185	0.014297	0.005003	0	Neutral
6/29/2016	-0.00185	0.014299	0.00083	0	Neutral
6/30/2016	-0.00243	0.013797	-0.00655	0	Neutral
7/1/2016	-0.00272	0.01379	-0.00659	0	Neutral

Table 10 – Surrounding Day Classification All Assets

Date	GBPUSD	GBPEUR	FTSE	SNP	UKGILT	HSI	DAX	CAC	FTMIB	MICEX	Classification
6/17/2016	0.0109	0.0062	0.0119	-0.0033	-0.0114	0.0066	0.0085	0.0098	0.0349	0.0094	Neutral
6/20/2016	0.0235	0.0204	0.0304	0.0058	-0.0185	0.0169	0.0343	0.0350	0.0254	0.0109	Positive
6/21/2016	-0.0033	0.0028	0.0036	0.0027	-0.0078	0.0077	0.0054	0.0061	0.0045	-0.0007	Neutral
6/22/2016	0.0040	-0.0006	0.0056	-0.0017	-0.0021	0.0061	0.0055	0.0029	-0.0062	-0.0013	Neutral
6/23/2016	0.0119	0.0040	0.0123	0.0134	-0.0080	0.0035	0.0185	0.0196	0.0371	0.0165	Neutral
6/24/2016	-0.0807	-0.0600	-0.0315	-0.0359	0.0488	-0.0292	-0.0682	-0.0804	-0.1248	-0.0205	Negative
6/27/2016	-0.0336	-0.0236	-0.0255	-0.0181	0.0274	-0.0016	-0.0302	-0.0297	-0.0394	-0.0236	Negative
6/28/2016	0.0092	0.0050	0.0264	0.0178	0.0003	-0.0027	0.0193	0.0261	0.0330	0.0043	Neutral
6/29/2016	0.0063	0.0008	0.0358	0.0170	0.0066	0.0131	0.0175	0.0260	0.0221	0.0175	Neutral
6/30/2016	-0.0088	-0.0065	0.0227	0.0136	0.0099	0.0175	0.0071	0.0100	0.0157	0.0019	Neutral
7/1/2016	-0.0034	-0.0066	0.0113	0.0019	0.0096	0.0000	0.0099	0.0086	0.0061	0.0031	Neutral

The results shown in Table 10 suggest that the previously outlined method of classifying a day is descriptive of returns across multiple asset classes. For instance, the returns on June 20th, 2016, a positive day, are all positive except for the sovereign bond prices. The two negative days on the 24th and the 27th have negative returns across all non-debt assets. For the neutral days, there does not seem to be a clear pattern.

It's worth noting that because this method relies exclusively on the returns from the GBP/EUR rate, all classifications should be studied in the context of any given day. Using this variable evaluates the relationship between Europe and the UK, but it can be effected by other correlated assets. An extension of this methodology would require the selection of another variable specific to a series of events, or development in including multiple variables.

To test whether the method is significant, I replicate the regression found in Question 3 (Table 6), but instead of regressing on the key events variable, I regress each variable on a classification variable that takes 1 on negative days and 0 on positive days. Neutral days are excluded from the sample. Table 11 presents the results. In contrast to the previous regression, this value presents far higher R-squared, p-values, and includes all negative non-debt coefficients. This suggests that this classification method is an adequate estimator of a negative return on any asset class.

Table 11 – Regression Results

Variable	Coefficient	P-Value	R-Squared
GBPUSD	-0.0066	0.0000	0.1844
GBPEUR	-0.0149	0.0000	0.7484
FTSE	-0.0006	0.5628	0.0009
UKGILT	0.0030	0.0005	0.0307
SNP	-0.0032	0.0003	0.0336
HSI	-0.0019	0.1143	0.0065
DAX	-0.0068	0.0000	0.0642
CAC	-0.0056	0.0001	0.0399
FTMIB	-0.0062	0.0007	0.0296
MICEX	-0.0006	0.6122	0.0007

Conclusion

The data presented in this paper suggest that Brexit was both unexpected by the financial markets and viewed negatively. The average returns on key events dates, the regression results, correlations, and other information presented here prove this. Further, my method for evaluating whether an event is negative, positive, or neutral has some evidence to suggest that it is robust in determining event status on days where a Brexit-related event has happened.

Appendices

Appendix 1

This contains the code for a file named “munge.py” which is primarily a backend script to import the data from an excel file which contains all asset data across different sheets.

```
1.  """
2.  This script reads from a pre-prepared excel sheet with all data needed,
3.  separated on different sheets. Because dates may vary from asset to asset
4.  and by country to country (FTSE vs. CAC), I separate perform and outer join
5.  on each individual data series and create a composite dataframe. This script
6.  "pickles" the data frame so that other scripts can simply load it into memory
7.  when necessary.
8.  """
9.
10. import pandas as pd
11.
12. def read_excel():
13.     """
14.     This reads the excel sheet containing all relevant closing prices, and
15.     turns them into a single pandas dataframe for later use. Because trading
16.     dates may vary between assets and across countries, I separate them by
17.     asset into a unique data frame and then combine a join on each component
18.     part.
19.
20.     The worksheet index is as follows:
21.     0. GBPUSD
22.     1. GBPUSD
23.     2. FTSE
24.     3. S&P
25.     4. UKGILT
26.     5. HSI
27.     6. DAX
28.     7. CAC
29.     8. FTMIB
30.     9. MICEX
31.     10.KY (the key indicator boolean)
32.     """
33.     df = pd.DataFrame()
34.     for i in range(0,11):
35.         book = pd.read_excel("Progdata.xlsx", sheetname=i,
36.                               index_col=0)
37.         df = df.join(book, how='outer')
38.
39.     df["KY"] = df["KY"].fillna(0)
40.     df = df.fillna(method = 'ffill')
41.     df[["KY"]] = df[["KY"]].astype(int)
42.     #Cut the bottom of the dataframe off.
43.     df = df.loc["2012-03-26":]
44.     return(df)
45.
46. if __name__ == '__main__':
47.
```

```
48. df = read_excel()
49. print(df.head())
50. print(df.ix["2016-06-24"])
51. print(df.dtypes)
52. df.to_pickle("data.csv")
```

Appendix 2

This is “Timeline.py”, which generates an annotated timeline for use in question 1.

```
1. import matplotlib.pyplot as mp
2. import matplotlib.dates as md
3. import pandas as pd
4. import datetime as dt
5.
6. #Grab pound/USD information
7. fx = pd.read_pickle("data.csv")
8. fx = fx[["GBPUSD", "GBPEUR", "KY"]]
9. fx2 = fx.loc["2013-01-01":] # Remove un-needed data
10. fx3 = fx2[["GBPUSD", "GBPEUR"]]
11.
12. #Make significant dates
13. dates = [dt.date(2013,1,23), dt.date(2016,1,22), dt.date(2016,6,23),
14.          dt.date(2016,6,24), dt.date(2016,7,13)]
15.
16. #Labels for annotation
17. labels = ["Cameron promises referendum", "Conservative party wins", "Referendum date se
    lected",
18.           "Referendum held", "Results announced", "Cameron resigns, May steps in"]
19.
20. #fx2.plot()
21.
22. fig, ax = mp.subplots()
23.
24. ax.plot(fx3['GBPUSD'], label = "GBP/USD")
25. ax.plot(fx3['GBPEUR'], label = "GBP/EUR")
26.
27.
28. labi = 0
29. for i in range(0,len(fx2)):
30.     if fx2["KY"][i] == 1:
31.         ax.annotate(labels[labi], xy = (fx2.index[i], fx2['GBPUSD'][i]),
32.                     textcoords='data', xytext= (fx2.index[i], fx2['GBPUSD'][i]+0.1), arrowprops=dict(
33.                         arrowstyle='->'))
34.         labi = labi + 1
35.
36. #Place annotations here for all i in dates. The following code is suggested
37. from http://stackoverflow.com/questions/11067368/annotate-time-series-plot-in-
    matplotlib
38.
39. for i in range(0,5):
40.     ax.annotate(labels[i], (md.date2num(dates[i]), fx2.loc[str(dates[i])]),
41.                         textcoords='data', xytext= (dates[i],fx2.loc[str(dates[i])] + 0.05), ar
42.                         rowprops=dict(arrowstyle='->'))
43.
44. mp.legend()
45. mp.ylabel("GBP vs. Euro and Dollar")
46. mp.xlabel("Date")
47. mp.title("Exchange Rates on Key Brexit Dates")
48. mp.xticks(rotation=45)
49. mp.grid(True)
50. #Show the plot we've made.
51. mp.savefig("timeline.png")
```

```
52. mp.show()
```

Appendix 3

This appendix contains the relevant code for both questions 2 and 3, due to their similarity.

```
1. import pandas as pd
2. import statsmodels.formula.api as sm
3.
4. def load_data():
5.     #Import our data from munge.py
6.     data = pd.read_pickle("data.csv")
7.     print("Load data...")
8.     return data
9. def get_returns(data):
10.    #Converts the dataframe of prices into returns.
11.    returns = data.ix[:,0:10].pct_change(1)
12.    returns["KY"] = data["KY"]
13.    returns = returns[2:]
14.    returns.to_pickle("returns.csv") #In case I need this elsewhere.
15.    print("Get returns...")
16.    return returns
17. def key_returns(returns):
18.    #What were the returns on key dates?
19.    print(returns.head())
20.    k_event = returns[returns.KY!=0]
21.    print("Key returns...")
22.    k_event.to_csv("k_event.csv")
23.    return k_event
24. def regression(returns):
25.    print("Regression 1...")
26.    #Change the "predicted" variable to what you want returned.
27.    predicted = ["GBPUSD", "GBPEUR", "FTSE", "UKGILT", "SNP", "HSI", "DAX", "CAC", "FTM
IB", "MICEX"]
28.    #Iterate through each predictor variable and regress it on a constant and
29.    #the key events variable.
30.
31.    composite = pd.DataFrame(columns = ["Variable", "Coefficient", "P-Value", "R-
Squared" ])
32.    for i in range(0, len(predicted)):
33.        result = sm.ols(formula = '{ } ~ KY'.format(predicted[i]), data = returns).fit()
34.        composite.loc[i] = [predicted[i], result.params[1], result.pvalues[1], result.r
squared]
35.    composite
36.    return composite
37.
38.    return result
39. def regression3(returns):
40.    print("Regression 2...")
41.    result = sm.ols(formula = "KY ~ GBPUSD + GBPEUR + UKGILT + FTSE + SNP + HSI + DAX +
CAC",
42.    data = returns).fit()
43.    t = open("Q3 Regression.txt", "w")
44.    t.write(str(result.summary()))
45.    t.close()
46.    return result
47. def get_correlation(returns):
48.    x = returns[["KY", "GBPUSD", "GBPEUR", "UKGILT", "FTSE"]]
49.    corr = x.corr()
50.    corr.to_csv("corr.csv")
```



```

51.     return corr
52. def correlation_q3(returns):
53.     corr3 = returns.corr()
54.     corr3.to_csv("Q3 Correlation.csv")
55.     return corr3
56. def neg_returns(returns):
57.     '''
58.     This function shows pretty clearly that other markets have negative returns
59.     when the UK markets have negative returns, within a very high degree of
60.     certainty. The only market that doesn't follow this structure is the Italian
61.     FTMIB, which is just barely not significant at the 5 percent level and is
62.     also negative.
63.     '''
64.     #Dataframe of returns where the FTSE is negative.
65.     df = returns[returns.FTSE < 0]
66.     result = sm.ols(formula = ("FTSE ~ CAC + DAX + HSI + SNP + MICEX + FTMIB"), data =
df).fit()
67.     t = open("Q3 Neg Regression.txt", "w")
68.     t.write(str(result.summary()))
69.     t.close()
70.     return result
71.
72. if __name__ == '__main__':
73.     data = load_data()
74.     returns = get_returns(data)
75.
76.     #Question 2 variables.
77.     k_event = key_returns(returns)
78.     mean_kreturns = k_event.mean()
79.     median_kreturns = k_event.median()
80.     kstddev = k_event.std()
81.     corr2 = get_correlation(returns)
82.     result = regression(returns)
83.
84.     #Here are the variables for question 3.
85.     corr3 = correlation_q3(returns)
86.     corr3
87.     result3 = regression3(returns)
88.     neg = neg_returns(returns)
89.     print(neg.summary())

```

Appendix 4

This section contains the code from “Question4.py”, which is used in processing the results for question 4.

```
1. import pandas as pd
2. import statsmodels.formula.api as sm
3. import matplotlib.pyplot as mp
4. from pandas.tseries.offsets import *
5.
6. def load_returns():
7.     df = pd.read_pickle("returns.csv")
8.     return df
9. def load_prices():
10.    df = pd.read_pickle("data.csv")
11.    return df
12. def t10(date, returns, boffset = 5, eoffset = 5):
13.     '''
14.     This function returns a dataframe with the returns +/- 5 days from a specific
15.     date.
16.     '''
17.     boff = boffset * BDay()
18.     eoff = eoffset * BDay()
19.     b = date - boff
20.     e = date + eoff
21.     df = returns.loc[b:e]
22.     return df
23. def cumu_returns(df):
24.     cumu = df.cumsum(axis = 0)
25.     return cumu
26. def plot_returns(tf, num, title = "No title"):
27.     mp.clf()
28.     mp.plot(tf.index, tf["FTSE"])
29.     mp.plot(tf.index, tf["SNP"])
30.     mp.plot(tf.index, tf["DAX"])
31.     mp.plot(tf.index, tf["UKGILT"])
32.     mp.plot(tf.index, tf["GBPUSD"])
33.     mp.plot(tf.index, tf["GBPEUR"])
34.     mp.xlabel("Date")
35.     mp.ylabel("Daily Return")
36.     mp.xticks(rotation=45)
37.     mp.title(title)
38.     mp.grid(True)
39.     mp.legend(loc='center left', bbox_to_anchor=(1, 0.5))
40.     mp.gcf().subplots_adjust(bottom=0.15, right = .75)
41.     mp.savefig("{} .png".format(num))
42.     #mp.show()
43.     return 0
44. def create_matrix(date, returns):
45.     t = t10(date, returns)
46.
47.
48. if __name__ == '__main__':
49.     labels = ["Cameron promises referendum", "Conservative party wins", "Referendum dat
50. e selected",
51.              "Referendum held", "Results announced", "Cameron resigns, May steps in"
52. ]
53.     returns = load_returns()
54.     prices = load_prices()
55.     num_dates = len(returns[returns.KY == 1].index)
56.     #This iterates throughout each requirement of the question and produces
57.     #all the necessary graphs.
```

```

56.     for i in range(0,1):
57.         date = returns[returns.KY == 1].index[4]
58.         tf = t10(date, returns)
59.
60.         plot_returns(tf, 1, title = "{} , +/- 5 days, actual returns".format(labels[4]))
61.         '....'
62.         Repeat the steps 2) and 3) using cumulative returns from five days before the e
vent
63.         to one day after the event.
64.         '''
65.         tf2 = t10(date, returns, eoffset = 1)
66.         cumu = cumu_returns(tf2)
67.         plot_returns(cumu, 2, title = "{} , -
5, +1 day, cumulative sum".format(labels[4]))
68.
69.         '....'
70.         What happens if you consider the
71.         cumulative returns from five days before the event to three days after the even
t?
72.         '''
73.         tf3 = t10(date, returns, eoffset = 3)
74.         cumu2 = cumu_returns(tf3)
75.         plot_returns(cumu2, 3, title = "{} , -
5, +3 days, cumulative sum".format(labels[4]))
76.
77.         tf4 = t10(date, returns, eoffset = 5)
78.         cumu3 = cumu_returns(tf4)
79.         cumu3
80.         plot_returns(cumu3, 4, title = "{} , +/- 5 days, cumulative sum".format(labels[4
]))
81.
82.     print(date)

```

Appendix 5

This appendix contains the code for question 5.

```
1. import pandas as pd
2. from Question4 import load_returns, t10
3. import statsmodels.formula.api as sm
4.
5. def classrow(row):
6.     # Mean, Standard deviation, Actual
7.     nodevs = 1
8.     confidence = nodevs * row[1]
9.     upper = row[0] + confidence
10.    lower = row[0] - confidence
11.
12.    upper
13.    lower
14.
15.    row[2] > upper
16.    row[2] < lower
17.    if row[2] > upper:
18.        return 0
19.    if row[2] < lower:
20.        return 1
21.    else:
22.        return -1
23.
24. def classify_key1(returns):
25.     '''
26.     This method classifies events as positive or negative based simple on the
27.     mean return of all assets on the day of the event. It's a fairly crude
28.     instrument and I suspect it's a poor indicator. See classify_key2 for
29.     hopefully an improved tactic.
30.     '''
31.     ky = returns[returns.KY == 1]
32.     ky = ky[["UKGILT", "GBPUSD", "GBPEUR", "FTSE"]]
33.     finky = ky
34.     finky["MEAN"] = ky.mean(axis = 1)
35.     finky["CLS"] = 0
36.     finky.ix[1, "CLS"] = 1
37.     for i in range(0, len(finky)):
38.         if finky["MEAN"].ix[i] > 0:
39.             finky.ix[i, "CLS"] = "Positive"
40.         if finky["MEAN"].ix[i] < 0:
41.             finky.ix[i, "CLS"] = "Negative"
42.     return finky
43. def classify_key2(returns):
44.     ky = returns[returns.KY == 1]
45.     num_dates = len(ky)
46.     dates = []
47.     #get dates of key events so we can pass it to the t10 function, imported
48.     #from Question4.py
49.     for i in range(0, num_dates):
50.         dates.append(ky.index[i])
51.
52.     #this loop iterates through each date, receives a dataframe of +/- 5 days
53.     #from the t10 function, runs a regression of each variable in turn for the
54.     #11 day period, extracts the coefficient and p-value of the KY variable,,
55.     #and classifies each key event as positive/negative/neutral.
56.     ky = ky.assign(PARAM = 0, PVAL = 0, CLS = "Neutral")
57.     ky.loc[dates[1], "PARAM"]
```

```

58.     for i in range(0,num_dates):
59.         tf = t10(dates[i], returns, boffset = 20, eoffset = 0)
60.         result = sm.ols(formula = "GBPEUR ~ KY", data = tf).fit()
61.         ky.loc[dates[i], "PARAM"] = result.params[1]
62.         ky.loc[dates[i], "PVAL"] = result.pvalues[1]
63.         if ky.loc[dates[i], "PVAL"] <= 0.05 and ky.loc[dates[i], "PARAM"] < 0:
64.             ky.loc[dates[i], "CLS"] = "Negative"
65.         if ky.loc[dates[i], "PVAL"] <= 0.05 and ky.loc[dates[i], "PARAM"] > 0:
66.             ky.loc[dates[i], "CLS"] = "Positive"
67.     ky = ky[["GBPEUR", "PARAM", "PVAL", "CLS"]]
68.     ky = ky.assign(DES = ["Cameron promises referendum", "Conservative party wins",
69.         "Referendum date selected", "Referendum held", "Results announced", "Ca
meron resigns, May steps in"])
70.     return ky
71. def classify_key3(returns):
72.     '''
73.     This method classifies positive, negative, or neutral events by both average
74.     and standard deviation. For each date, we find the standard deviation and
75.     average return for the previous twenty days.
76.     '''
77.     # This variable changes how many days to use in the rolling window.
78.     con = 30
79.     var = "GBPEUR"
80.
81.     roll = returns[var].rolling(window = con)
82.     stdev = roll.std()
83.     mean = roll.mean()
84.     stdev.name = "Stdev"
85.     mean.name = "Mean"
86.     event = pd.concat([mean, stdev], axis =1)
87.     event = event.shift(1)
88.     event = pd.concat([event, returns[var], returns["KY"]], axis =1)
89.     event
90.     event["Classification"] = event.apply(classrow, axis =1)
91.     return event
92. def regression(regret):
93.     print("Regression 1...")
94.     #Change the "predicted" variable to what you want returned.
95.     predicted = ["GBPUSD", "GBPEUR", "FTSE", "UKGILT", "SNP", "HSI", "DAX", "CAC", "FTM
IB", "MICEX"]
96.     #Iterate through each predictor variable and regress it on a constant and
97.     #the key events variable.
98.
99.     composite = pd.DataFrame(columns = ["Variable","Coefficient", "P-Value", "R-
Squared" ])
100.     for i in range(0, len(predicted)):
101.         result = sm.ols(formula = '{ } ~ Classification'.format(predicted[i]), da
ta = regret).fit()
102.         composite.loc[i] = [predicted[i], result.params[1], result.pvalues[1], r
esult.rsquared]
103.     composite
104.     return composite
105.
106.     if __name__ == '__main__':
107.         returns = load_returns()
108.         classified = classify_key1(returns)
109.         classified2 = classify_key2(returns)
110.         classified3 = classify_key3(returns)
111.         key_dates = returns.index[returns.KY == 1]
112.         clsreturns = returns.join(classified3.Classification)
113.         key = classified3.loc[key_dates]

```

```
114.         key["Description"] = ["Cameron promises referendum", "Conservative party win
s", "Referendum date selected", "Referendum held", "Results announced", "Cameron resigns
, May steps in"]
115.         dateclass = t10(key_dates[4], clsreturns)
116.
117.         neg = clsreturns[clsreturns.Classification == 1]
118.         pos = clsreturns[clsreturns.Classification == 0]
119.         neu = clsreturns[clsreturns.Classification == -1]
120.
121.         regret = clsreturns[clsreturns.Classification == 1]
122.         regret = regret.append(clsreturns[clsreturns.Classification == 0])
123.
124.         regret
125.
126.         result = regression(regret)
127.         result
```