Report of Data Analysis

Xizhe Cheng

July 7, 2021

Abstract

A series of analysis have been conducted on the sample data provided by Millennium. In general, two stages of investigations have been performed. The anomalies in the sample dataset are firstly detected and replaced with reasonable values, followed by a detailed regression analysis on the relationship between the provided 'Signal' and 'Adjusted Close Price' series. To this end, the effectiveness of the 'Signal' series to forecast the 'Adjusted Close Price' series is. Econometric techniques including Granger Causality test, vector error correction model (VECM) as well as Newey-West Standard Errors are employed.

1 Data Cleaning

The provided sample data consists of 1 time column and 6 associated time series: 'Signal', 'Open', 'High', 'Low', 'Close' and 'Adjusted Close'. Other than the 'Signal' column, each of the others refers to certain price information of the underlying ETF. After a bit search, the ETF is found to be **iShares Russell 2000 ETF** with the ticker symbol **IWM**. As such, the correct data can be downloaded from Yahoo Finance (*https://finance.yahoo.com/quote/IWM*). The actual data is thereafter used as a reference for checking the data cleaning result.

The data cleaning operations are achieved using Python and the intermediate steps are recorded in the attached jupyter notebook file.

1.1 Anomalies in Date Column

It is firstly spotted that the date column bears some anomalies. Some invalid trading days are present as well as several valid trading days being missing. Compared to the historical data achieved from open-source python package, the invalid trading dates together with the missing ones are present in Table 1

Table 1: Anomalies in Date Column

Invalid Dates	Missing Dates
2017-07-04	2018-11-12
2018-05-19	2018-11-13
2018-05-20	2018-11-14
2018-06-23	2018-11-15
2018-06-24	2018-11-16

To fix the anomaly data rows spotted in the 'Date' column, the invalid data rows specified with the invalid dates are simply deleted. On the other hand, if we turn our blinds eyes to the missing data, there can exist a break of continuity of the stock price since the missing rows are clustered together and fit right into a week. We choose to insert the missing data rows. The fixes are listed below:

Invalid Date Simply discard the data rows

Missing Date Insert the data rows, then fill the corresponding 'Signal' values using a rolling mean (windows size = 11). Fill the other values using a log-linear interpolation, which is linear interpolation in the log values of the columns.

In the pandas DataFrame object, we have append an 'is_new' column to indicate if a certain row is newly inserted. Besides, Columns 'Signal Filled', 'Open Filled', 'High Filled', 'Low Filled', 'Close Filled' and 'Adj Close Filled' are all added to indicate if the referred data has experienced any modification.

1.2 Outliers

'Outliers' refer to the datapoints that are extraordinarily different from the left ones within the same dataset. Outliers are found in the provided sample dataset. Generally speaking, for a stationary time series, one could apply the 3σ *principle* to decide which datapoints are outliers. Specifically, 3σ *principle* states that a datapoint is an outlier if it is more that 3 times the standard deviations of the series away from the mean value. In this case, we cannot directly adopt this idea as the included time series are clearly not stationary but entail stochastic trends to some extend. Alternatively, we use the rolling mean and rolling standard deviation to replace the global mean and global standard deviation values to determine the outlier. The rolling window is chosen to be two weeks (10 trading days in total, excluding the inpected data itself). The outliers are detected and listed in Table 2:

Table 2: Outliers in the Dataset

Date	Signal	Open	High	Low	Close	Adj
2016-07-13	O	–	–	–	–	–
2016-11-30	O	–	–	–	–	–
2016-12-05	–	–	–	–	–	O
2017-03-22	O	–	–	–	–	–
2017-03-27	–	–	–	–	–	O
2017-05-17	–	–	–	O	–	–
2017-09-25	–	–	–	–	O	O
2017-11-13	O	–	–	–	–	–
2018-02-05	–	–	–	O	O	O
2018-03-19	–	–	–	–	O	–
2018-10-09	O	–	–	–	–	–
2018-10-10	–	–	–	–	–	O
2019-11-25	–	–	O	–	O	O
2019-12-27	O	–	–	–	–	–
2019-12-30	O	–	–	–	–	–
2019-12-31	O	–	–	–	–	–
2020-01-02	O	–	–	–	–	–
2020-01-03	O	–	–	–	–	–
2020-01-06	O	–	–	–	–	–

To fix the outlier issues, we simply follow what is described above in the last subsection:

Signal Outlier Use a rolling mean with the windows size of 11.

Other Outliers Use a log-linear interpolation, which is linear interpolation in the log values of the columns.

In the pandas DataFrame object, the columns 'Signal Filled', 'Open Filled', 'High Filled', 'Low Filled', 'Close Filled' and 'Adj Close Filled' are all modified according to this step of anomaly detection and modification.

1.3 Semantic/Logical Error

By semantic error we refer to those data rows where 'High' is not the highest ones or 'Low' is not the lowest ones indeed. We find this kind of error is also very common in the provided sample dataset. the errors are listed in Table 3

Table 3: Semantic Errors

Dates	Problematic High	Problematic Low
2017-05-17	N	Y
2017-08-07	Y	N
2017-09-11	N	Y
2017-09-12	N	Y
2017-09-13	N	Y
2017-09-14	N	Y
2017-09-15	N	Y
2017-09-18	N	Y
2017-09-19	N	Y
2017-09-20	N	Y
2017-09-21	N	Y
2017-09-22	N	Y
2018-02-05	N	Y
2018-03-07	Y	Y
2018-07-16	Y	Y
2018-10-19	N	Y
2018-12-06	Y	N
2019-06-10	N	Y
2019-09-24	N	Y
2019-10-17	Y	N
2019-11-25	Y	N

For this type of semantic errors within the data rows, there is not enough information for us to decide which fields within those abnormal rows actually cause the problems. As a result, we applied the following fix:

Rows with Semantic Errors Ignore all the fields other than the Dates with in the rows. Then use rolling mean to fill the 'Signal' fields and log-linear interpolation to fill the other ones, respectively.

In the Pandas DataFrame object, two columns, i.e., True High and True Low are added to indicate if the rows consist the true 'High' and true 'Low' values.

1.4 Quality Check against True Values

As we have downloaded the true data from Yahoo Finance, we can check the modified sample data against the downloaded version. Note that the downloaded true data has not been employed for the fixation of the data anomalies in the provided sample data.

After entry-wise data check, we summarize the left anomalies in Table 4, where the data anomalies are defined as those deviating more than 1% from the true values. Note that there is systematic differences between the adjusted close prices in the provided sample data and those in the true data. so do not include that field in this quality check.

Table 4: Quality Check against True Values

Date	Open	High	Low	Close
2017-05-17	–	–	–	P
2018-02-05	–	–	–	P
2018-11-12	P	–	–	P
2018-11-13	P	P	–	P
2018-11-14	P	P	P	P
2018-11-15	P	P	P	–
2018-11-16	P	–	–	–
2018-12-06	P	P	P	–
2019-09-24	–	–	–	P

In the Pandas DataFramem, we have appended the columns 'Open Check', 'High Check', 'Low Check', 'Close Check' to indicate the problematic values in each field. One can easily filled out these data rows using these appended columns.

We take a sneak peek into the problematic rows:

2018-11-12 – 2018-11-16 These rows are purely inserted to the original sample and filled using rolling means and interpolations.

2017-05-17, 2019-09-24 Informed from the true data downloaded from Yahoo Finance, the volatility in this day belongs to the extreme kind. The close price that day is 3% lower compared to that one day before, which is relatively large for an ETF.

2018-12-06 Informed from the true data downloaded from Yahoo Finance, the volatility in the next day belongs to the extreme kind compared to that day.

It turns out that all of the problematic days are actually special, which to some extent explain the result.

2 Regression Analysis

A series of regression analysis has been run to determine the causation and forecast-type relationship between the provided 'Signal' and 'Adjusted Close' time series. For convenience, all the analysis is done using **R** language, of which the source code is also included.

For time series of prices, it is more appropriate to deal with the percentage changes in these kind of variables. As a

result, in our regression analysis, the logarithm of the 'Adjusted Close' values is taken. On the other hand, the properties of the 'Signal' value is unknown, so both of the log–log model and linear–log model have been analyzed.

2.1 Log–Log Model

In the log–log model, we regress the log value of the adjusted close prices on the log value of signal. The stationarity of the two series are firstly investigated using Augmented Dickey Fuller test. The results are listed in Table 5. It turns out that both time series are not stationary. We then take the first difference of the two time series and conduct the Augmented Dickey Fuller test on the both of them. The test results clearly confirm the stationarity of the both of the first difference series. In another word, both series are of the integration order 1 $I(1)$.

Table 5: Augmented Dickey Fuller Test

	Ln(Sig)	*Ln(Adj)*	*d(Ln(Sig))*	*d(Ln(Adj))*
D–F	-2.36	-2.33	-9.72	-12.31
p	0.47	0.44	≤0.01	≤0.01

2.1.1 Causality Test

Since nonstationarity, the direct Granger causality test cannot be run on *Log(Adj) versus Log(Sig)*. Instead, we resort to the procedure proposed by **Toda-Yamamoto**. Four information criteria are are calculated with different lag orders to select the optimal lag order. The Results are summarized in Table 6.

Table 6: Optimal Lag Order

Info Criterion	**AIC**	**HQ**	**SC**	**FPE**
Optimal Lag Order	17	3	3	17

The optimal lag order can be either 3 or 17 according to the different information criteria. We now proceed to determine which of the two to choose for further analysis. In order to do this, two Vector Auto Regressions (VARs) have been run with the 2 lags are run with the autocorrelations of the 2 residual series being tested. The results are summarized in Table 7

Table 7: Autocorrelation (Edgerton-Shukur) Test Results

Lag Order	3	17
p-value	8.27×10^{-6}	0.44

From Table 7, the *p-value* of the Edgerton-Shukur test in the case of the lag order of 17 is larger, implying that the regression using the lag order of 17 is less likely to be serially correlated.

After that, we continue on following the **Toda-Yamamoto** procedure to perform the causality test between the two log series. Instead of the lag order of 17, we use 18(=17+1) as the lag order to set up a Vector Autoregressive (VAR) model and use the Granger test to test the Granger causalities within the bivariate system. The *p-values* of the causality tests are listed in Table 8

Table 8: Causality Test (**Toda-Yamamoto** Procedure)

Null Hypothesis	*p-value*
Adj_Close do not Granger-cause *Sig*	0.23
Sig do not Granger-cause *Adj_Close*	2.2×10^{-6}

According to the test result, it can be concluded that *Signal* Granger-causes *Adjusted Close*, but *Adjusted Close* does not Granger-cause *Signal*. Thus it can be concluded that Signal can be used to forecast the price of the ETF.

2.1.2 Vector Error Correction Model

More quantitative assessment of the relationship between *Signal* and *Adjusted Close* are done using Vector Error Correction Model. The first step is to study the cointegration relationship between the two log series. Johansen's procedure has been adopted for the test of cointegration. The Max-Eigen statistics are summarized in Table 9.

Table 9: Johansen Cointegration Test

	statistics	10%	5%	1%
$r \leq 1$	3.46	7.52	9.24	12.97
$r = 0$	15.78	13.75	15.67	20.20

Thus the two log series are cointegrated with the confidence level of 5%. Besides, the cointegration equation reads:

$$log(Adj\_Close) = 0.87 \times log(Signal) + 2.45$$

The residual of the Vector Error Correction Model (VECM) are tested in terms of autocorrelation, heteroskedasticity as well as normality with the results listed in Table 10

Table 10: Test of the VECM Residual

Field	Test	*p-value*	Result
Autocorrelation	ES Test	0.42	No
Heteroskedasticity	ARCH	1.99×10^{-11}	Yes
Normality	JB Test	$< 2.2 \times 10^{-16}$	No

According to the test results, we can see that there is no serial correlations within the residuals, but heteroskedasticity problem exists with the residuals deviating from normal distribution. Since no autocorrelation or serial correlation exists within the residual series, the linear estimators are actually **unbiased** and **consistent**. However, heteroskedasticity can result in the **inefficiency** of the linear estimators. The standard errors and the test statistics is not corrected in this kind of estimators. To correct for the heteroskedasticity, we adopt the Newey West-type HAC robust standard errors with the final results listed in Table 11

Table 11: VECM Result for *Adj_Close* with HAC S.D.

Coef	Value	SD	HAC SD	t (HAC)	Sig
ect	-0.061	0.016	0.016	-3.782	***
adj[-1]	-0.102	0.034	0.041	-2.489	*
adj[-2]	-0.086	0.034	0.038	-2.294	*
adj[-3]	-0.089	0.034	0.030	-2.958	**
adj[-4]	-0.092	0.033	0.036	-2.570	*
adj[-5]	-0.072	0.033	0.038	-1.925	*
adj[-6]	-0.063	0.033	0.040	-1.590	–
adj[-7]	-0.007	0.032	0.034	0.832	–
adj[-8]	-0.077	0.032	0.030	-2.557	*
adj[-9]	-0.037	0.031	0.032	-1.165	–
adj[-10]	-0.024	0.031	0.030	-0.800	–
adj[-11]	-0.003	0.029	0.037	-0.070	–
adj[-12]	-0.000	0.029	0.030	-0.015	–
adj[-13]	-0.019	0.029	0.028	-0.697	–
adj[-14]	-0.055	0.029	0.035	-1.560	–
adj[-15]	-0.060	0.029	0.028	-2.103	*
adj[-16]	-0.063	0.029	0.032	1.960	*
sig[-1]	-0.015	0.016	0.018	-0.878	–
sig[-2]	0.041	0.017	0.019	2.159	*
sig[-3]	0.081	0.017	0.021	3.873	***
sig[-4]	0.092	0.017	0.020	4.609	***
sig[-5]	0.103	0.018	0.020	5.177	***
sig[-6]	0.132	0.018	0.022	6.082	***
sig[-7]	0.110	0.018	0.021	5.253	***
sig[-8]	0.102	0.018	0.020	5.105	***
sig[-9]	0.096	0.018	0.019	5.146	***
sig[-10]	0.092	0.017	0.019	4.718	***
sig[-11]	0.074	0.017	0.020	3.646	***
sig[-12]	0.064	0.016	0.020	3.215	**
sig[-13]	0.054	0.015	0.018	3.059	**
sig[-14]	0.031	0.014	0.015	2.052	**
sig[-15]	0.021	0.012	0.012	1.704	–
sig[-16]	0.016	0.009	0.008	1.961	*

From the VECM result, it can be seen that most of the lags of *Log(Sig)* is significantly involved in the equation for *Log(Adj_Close)*. While much fewer lags of *Log(Adj_Close)* is significantly involved. For a clear and intuitive interpretataion of the regression result, we plot the impulse response function (IRF, Figure 1) as well as the variance decomposition of forecast errors (FEVD, Figure 2) of the equation of *Log(Adj_Close)*.

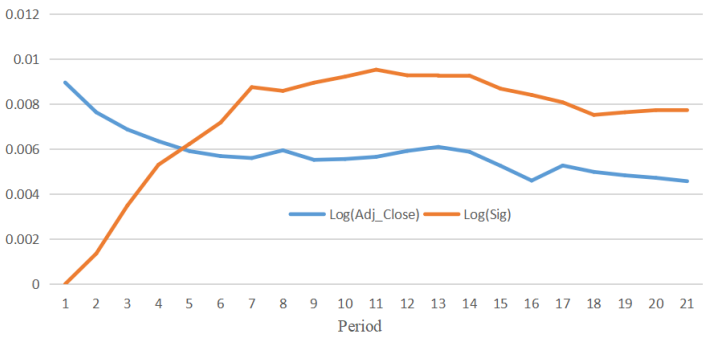

Figure 1: Impulse Response of *log(Adj_Close)*

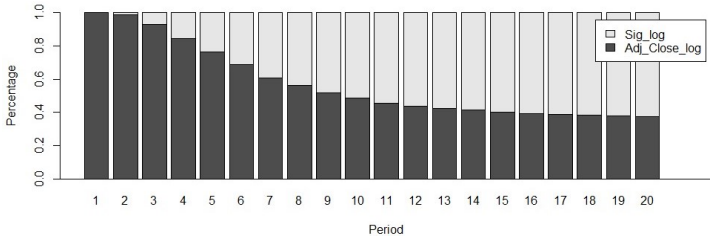

Figure 2: Variance Decomposition of *log(Adj_Close)*

From the IRF plot, it can be told that an one-standard-deviation shock in *Log(Sig)* can result in a stable 0.8% change in *Log(Adj_Close)*, which is more significant than that (0.6%) resulted from an one-standard-deviation shock in *Log(Adj_Close)* itself. (The 95% confidence sidebands of the two actually overlap, which means that compared to the lags of *Log(Adj_Close)*, *Log(Sig)* is only marginally better.) From the FEVD result, it can be seen in the long run, *Log(Sig)* contributes 60% to *Log(Adj_Close)*, which dominates the contribution from the lags of *Log(Adj_Close)* itself.

To conclude, from the results in the VECM analysis. *Log(Sig)* can be effective (more than the lags of *Log(Adj_Close)* itself) in forecasting *Log(Adj_Close)*. This result is consistent with that yield from the Granger causality test.

2.1.3 Forecasting Test

An actual forecasting test according to the VECM workflow presented above is conducted. To do this, the last 17 data points (test set) are separated from the left data (train set). A new VECM is estimated based on the train set and used to predict the data in the test set.

The predicted values against the real value is pasted in Table 12

Table 12: Prediction Result of Adj_Close (Not Logged)

Date	Ture Value	Predicted Value	Deviation
2019-12-11	160.132	160.453	0.2%
2019-12-12	161.580	161.257	-0.2%
2019-12-13	160.774	161.580	0.5%
2019-12-16	162.065	161.903	-0.1%
2019-12-17	162.878	161.742	-0.7%
2019-12-18	163.204	161.742	-0.7%
2019-12-19	163.858	161.903	-1.2%
2019-12-20	164.186	162.390	-1.1%
2019-12-23	164.515	162.390	-1.3%
2019-12-24	164.844	162.228	-1.6%
2019-12-26	164.844	161.903	-1.8%
2019-12-27	164.022	161.742	-1.4%
2019-12-30	163.694	161.580	-1.3%
2019-12-31	163.858	161.580	-1.4%
2020-01-02	164.022	161.257	-1.7%
2020-01-03	163.367	161.257	-1.3%
2020-01-06	163.531	161.257	-1.4%

The fanchart is shown in Figure 3

According to the result, the deviation of the predicted values from the true values is always within 2%. Importantly, the model successfully predict the peak position at 2019-12-24–2019-12-26, validating the effectiveness of *Signal* in forecasting *Adj_Close*

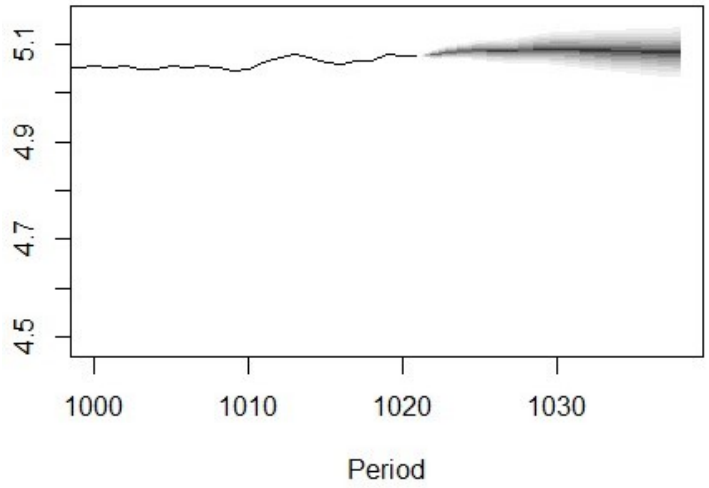

Figure 3: Fanchart of the Forecast Result

2.2 Linear-Log Model

The liear-log model is also studied, where the *Log(Adj_Close)* is regressed on the original value of *Signal*.

The situation is slightly different in this case. First, ADF test once again confirm the staionarities of both the first difference series of *Signal* and *Log(Adj_Close)* but not the original series. Information in this case gives possible lag order to be any one of 3, 8, or 17. Following **Yato-Yamamoto**'s procedure, the optimal lag is still determined to be 17. The causality test also support that *Signal* causes *Log(Adj_Close)* but not the other way around. So far, so good. However, Johensen's test shows that with a lag order of 17, the two series are not cointegrated. But with a lag order of 3 or 8, the residuals are severely autocorrelated. We therefore conclude that this is not an appropriate model in this situation.

3 Conclusion

From the above results, it can be concluded that:

- The quality of the provided sample data is low;
- Compared to the model built on *Signal* and *Log(Adj_Close)*, that built on *Log(Signal)* and *Log(Adj_Close)* is more reasonable;
- *Log(Signal)* can be used to forecast *Log(Adj_Close)*, given the results from the causality tests, the VECM, as well as the in-sample test. But *Log(Signal)* is only marginally better compared to the lags of *Log(Adj_Close)*.

I hold cautious optimism about this forecasting product and would require more data for further investigations and study. I would advise to negotiate with the source company on the following points:

- Give more similar pieces of data regarding other financial instruments for more investigations;
- Include more periods of the data, especially those when the volatility is high, e.g., March 2020;