ST959 Assignment Report - Analysis of statistical techniques on the Chicago Mercantile Exchange

Denis Maruev (Student ID: 1904957)

Warwick Business School, University of Warwick, Coventry CV4 7AL, United Kingdom

I. Introduction

In this project we investigate the *sp5may.dat* data file provided to us in Ruey S. Tsay's book 'Analysis of Financial Time Series'. This data file contains three columns: log(futures price), log(spot price), and cost of carry (×100), obtained from the Chicago Mercantile Exchange (CME) for the S&P 500 stock index in May 1993 and its June futures contract with a time interval of one minute (intraday).

Our objective is to compare the prediction accuracy between two regression models: one with simple regression and the other with time series errors. To accomplish this, we will first familiarize ourselves with the methodology outlined in Section 2.9 of Ruey S. Tsay's book, 'Analysis of Financial Time Series'. This section specifically covers the process of building regression models with time series errors.

Our hypothesis is that the regression model with time series errors will provide more accurate predictions than the simple regression model. To test this hypothesis, we will perform statistical analysis on the data and compare the results of both models.

II. Inspecting *sp5may.dat*

Let us first consider the data in *sp5may.dat*; we check the the number of rows against the opening hours of the CME to get an idea of the time span covered by the data:

```
>  nrow(rawData)
[1] 7061
```

From 7061 rows of data in in *sp5may.dat* excluding the axes labels we can conclude that we have 7061 time intervals of data where each time interval is a minute. In this link: `https://www.cmegroup.com/trading-hours.html`, we see that the CME trading hours are 23.75 hours per day from Monday to Thursday, 17.75 hours on Friday, 0 hours on Saturday, and 7 hours on Sunday. It is important to note that there is a 15-minute gap in reporting from 5:45 p.m. to 6:00 p.m. CT on Monday to Thursday. The total trading time in a week is 119.75 hours, equivalent to 7185 minutes. The absence of trading hours on Saturday and the 7 hours of trading on Sunday (420 minutes) would result in a total of 6765 minutes of trading time if a weekend is not included. However, the 7061 rows of data in the ”sp5may.dat” file suggest that the data period includes a weekend, albeit some missing minutes of course. Hence, we can be almost certain that the time period spanned includes a weekend.

To further justify this, we check whether there are any missing values in the data set:

```
> sum(is.na(rawData))
[1] 0
```

Hence we can be sure in our deduction that this time period spanned includes a weekend and that no further actions are needed to manipulate the data set to perform statistical analysis.

III. Analysis of *sp5may.dat*

Consider two vectors which we denote f and s which are the log(futures prices) and log(spot prices) respectively. We plot these two vectors as time series on the same graph with spot prices being the red line and futures prices being the blue line:

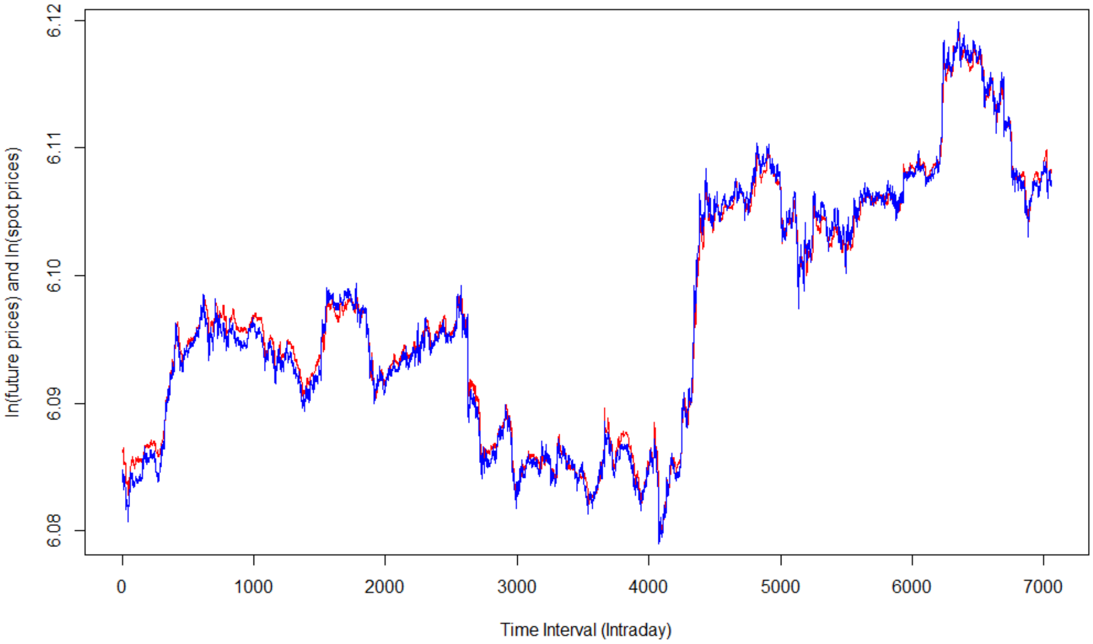

Figure 1: Plot of f (in blue) and s (in red) from the CME for the S&P 500 stock index in May 1993 and its June futures contract with a time interval of one minute (intraday)

As one might suspect, futures and spot prices are highly correlated; we know this since futures and spot prices are influenced by the same market forces, such as supply and demand, geopolitical events, and economic indicators. As a result, the two prices tend to move in the same direction and be highly correlated. This can be seen in the following figure:

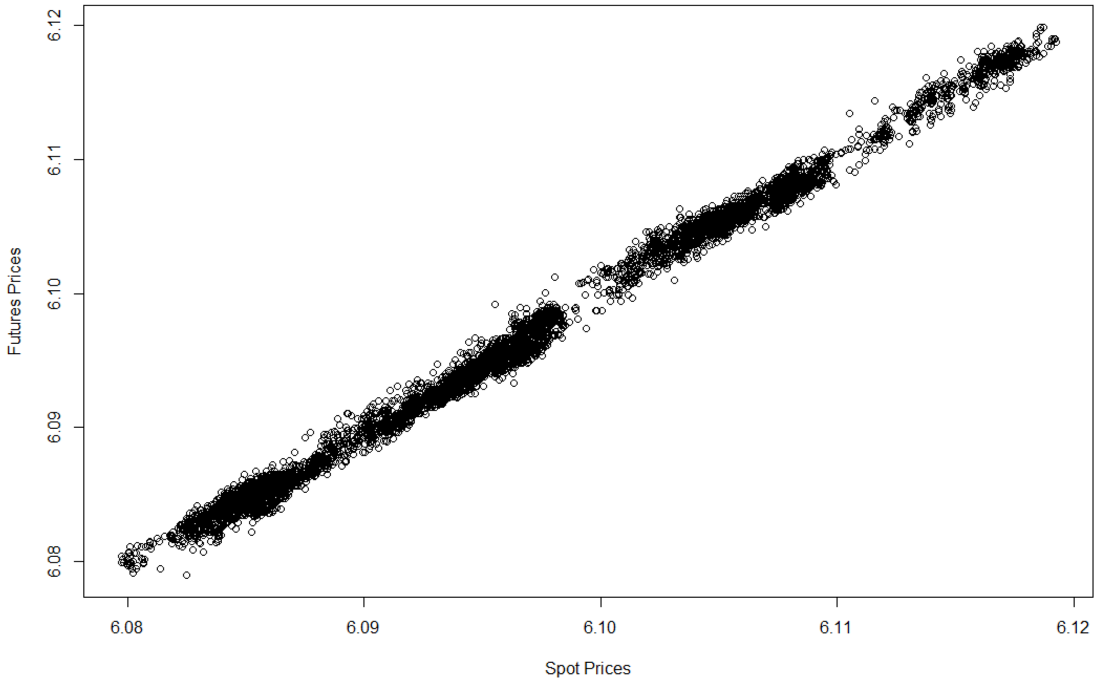

Figure 2: Plot of futures prices against spot prices, clearly showing a very strong positive correlation

It is important to note the sharp spikes which occur in the plot in Figure 1. To explain this, we shall assume that this is due to the times the CME is not open such as the 15 minutes of no reporting in Monday to Thursday. A heuristic argument for this is that when gaps in reporting occur during periods of high market volatility, the sharp price movements that occur during that time may not be reflected in the data, resulting in a discontinuity or a spike in the time series plot when the data is interpolated. Figure 2 also reflects this as we can see gaps in our hypothetical straight line of futures against spot prices.

Of course, we want to describe the relationship between the vectors f and s. We do this by fitting a simple linear regression model in R:

$$f = \alpha + \beta r_{1t} \tag{1}$$

```
> df2 <- data.frame(spotData, futureData)
> model2 <- lm(futureData ~ spotData, df2)
> summary(model2)
```

```
Call:
lm(formula = futureData ~ spotData, data = df2)

Residuals:
       Min         1Q     Median         3Q        Max
-0.0028147 -0.0004874 -0.0000513  0.0004645  0.0039005

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.1795864  0.0055263   -32.5   <2e-16 ***
spotData     1.0294208  0.0009064  1135.8   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0007243 on 7059 degrees of freedom
Multiple R-squared:  0.9946,Adjusted R-squared:  0.9946
F-statistic: 1.29e+06 on 1 and 7059 DF,  p-value: < 2.2e-16
```

This gives us a fitted model:

$$\begin{aligned} f_t &= -0.1796 + 1.0294s + \epsilon_t\,, && (2) \\ \hat{\sigma_\epsilon} &= 0.0007243\,. && (3) \end{aligned}$$

This has an R-squared value of 99.46% with a standard error of the two coefficients being 0.0055263. Hence this model supports the statement that there is a high correlation between the futures and spot prices. To make sure that this is a valid model to make such claims, we construct an ACF of the residuals of linear regression for f and s.

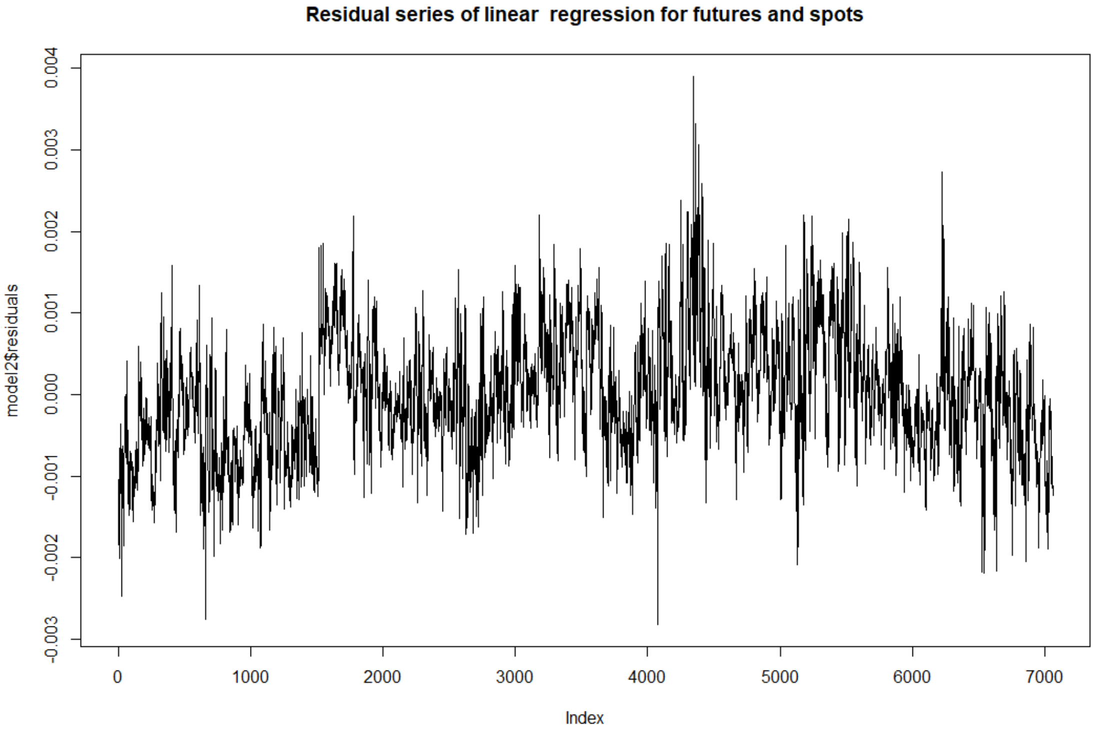

Figure 3: Residual series of the model for a linear regression f and s with a time interval of one minute (intraday)

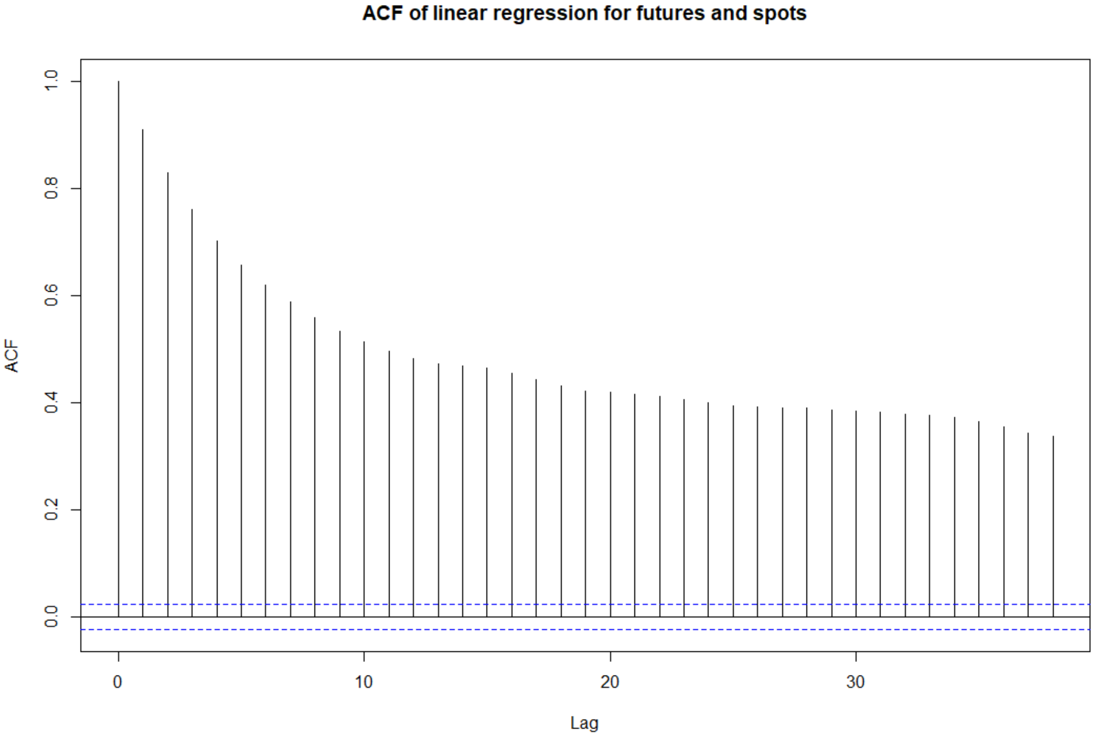

Figure 4: Autocorrelation function of linear regression for futures and spots over a lag of up to 38

Unfortunately, the ACF of the residuals is highly significant and decays very slowly as lag increases which implies that we have a unit-root nonstationary time series. Hence we must deduce that our current model is inadequate for the analysis of the *sp5may.dat* data file. This is because an ACF of a residual series measures the correlation between the residuals and their lags. In linear regression, residuals should be uncorrelated and have constant variance to indicate that the model is a good fit. If the ACF of the residual series shows significant autocorrelation, it means that the linear regression model is not capturing all the underlying patterns in the data, and is therefore an inadequate model. This suggests that other factors or a different model might be needed to better describe the relationship between the future and spot prices.

Hence, we shall consider two new vectors to counteract this new-found issue:

$$\begin{aligned} y_n &= f_n - f_{n-1}\,, & (4) \\ x_n &= s_n - s_{n-1}\,. & (5) \end{aligned}$$

One will point out that these are simply difference vectors of the futures and spot prices vectors previously defined. It is a good idea to use difference vectors because differencing involves subtracting each observation from its previous value, which can help remove linear trends and seasonality in the data. By considering the difference vectors of futures and spot prices instead of just the vectors of futures or spot prices, one can capture the change in these prices over time, rather than just their level. Hypothetically, this leads to better predictions.

To get a better understanding of the new vectors we are currently working with, we plot them where we again have x in red and y in blue:

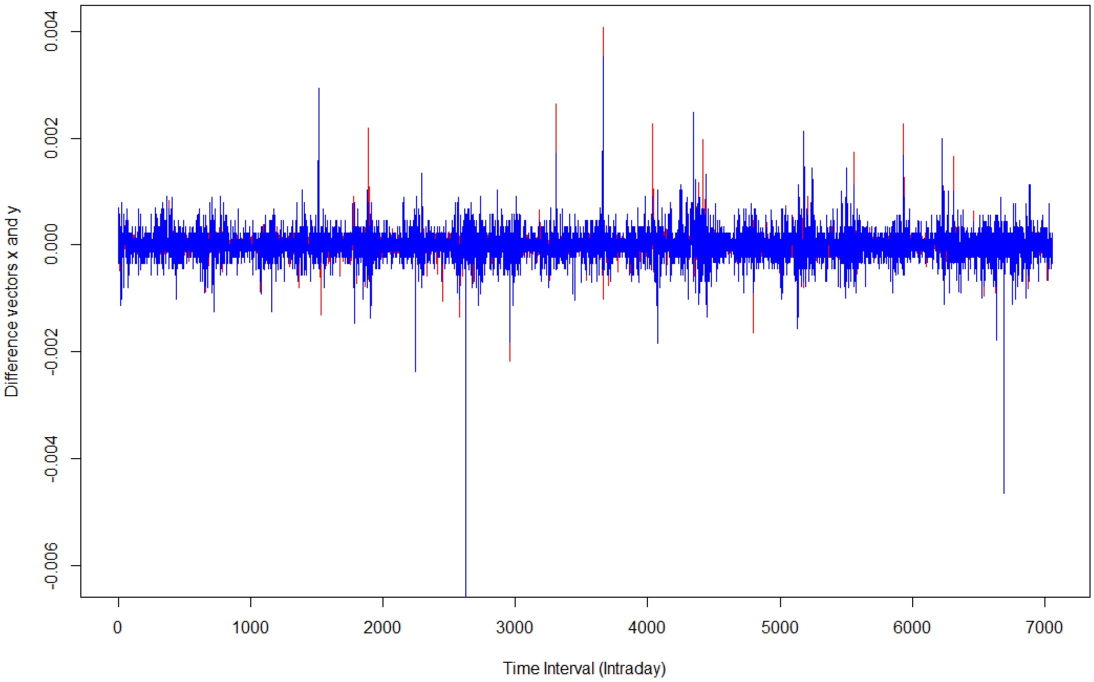

Figure 5: Difference vectors x (red) and y (blue) against time

Now let us review the model these difference vectors give us:

```
> df <- data.frame(x_n, y_n)
> model1 <- lm(y_n ~ x_n, df)
> summary(model1)

Call:
lm(formula = y_n ~ x_n, data = df)

Residuals:
       Min         1Q     Median         3Q        Max
-0.0038484 -0.0001568 -0.0000014  0.0001612  0.0026256

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.354e-06  3.509e-06   0.386      0.7
x_n         6.212e-01  1.754e-02  35.420   <2e-16 ***
---
```

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0002948 on 7058 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.1509,Adjusted R-squared:  0.1508
F-statistic:  1255 on 1 and 7058 DF,  p-value: < 2.2e-16
```

Hence, the fitted model is given by:

$$\begin{aligned} y_t &= 0.000001354 + 0.6212x + \epsilon_t \,, & (6) \\ \hat{\sigma_\epsilon} &= 0.0002948 \,. & (7) \end{aligned}$$

This has an R-squared value of 15.09% with a standard error of the two coefficients being 0.000003509. This implies that, after using difference vectors, the correlation has substantially decreased compared to what we had previously. This is the case because the correlation between futures and spot prices can be driven by factors such as linear trends, seasonality, and heteroscedasticity, and these factors can obscure the true relationship between them. By taking the difference vectors of futures and spot prices, these factors are removed, and the residuals are likely to be more homoscedastic and less correlated with each other. The following figures further support this hypothesis:

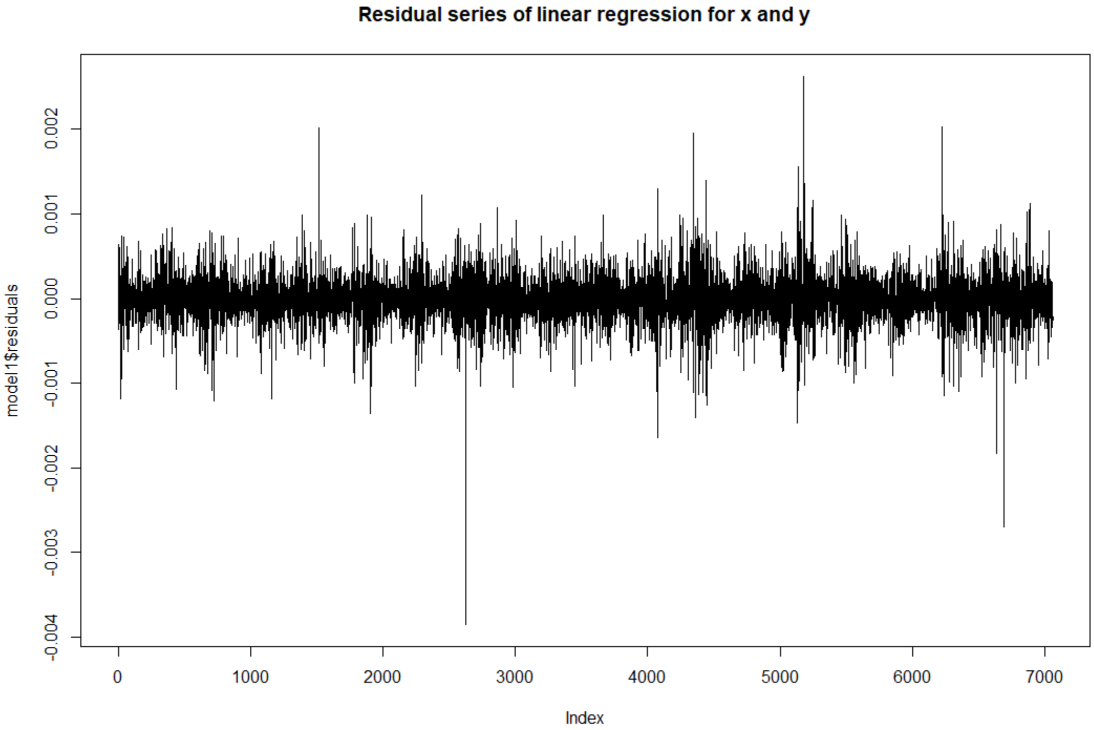

Figure 6: Residual series of the regression model for y and x with a time interval of one minute (intraday)

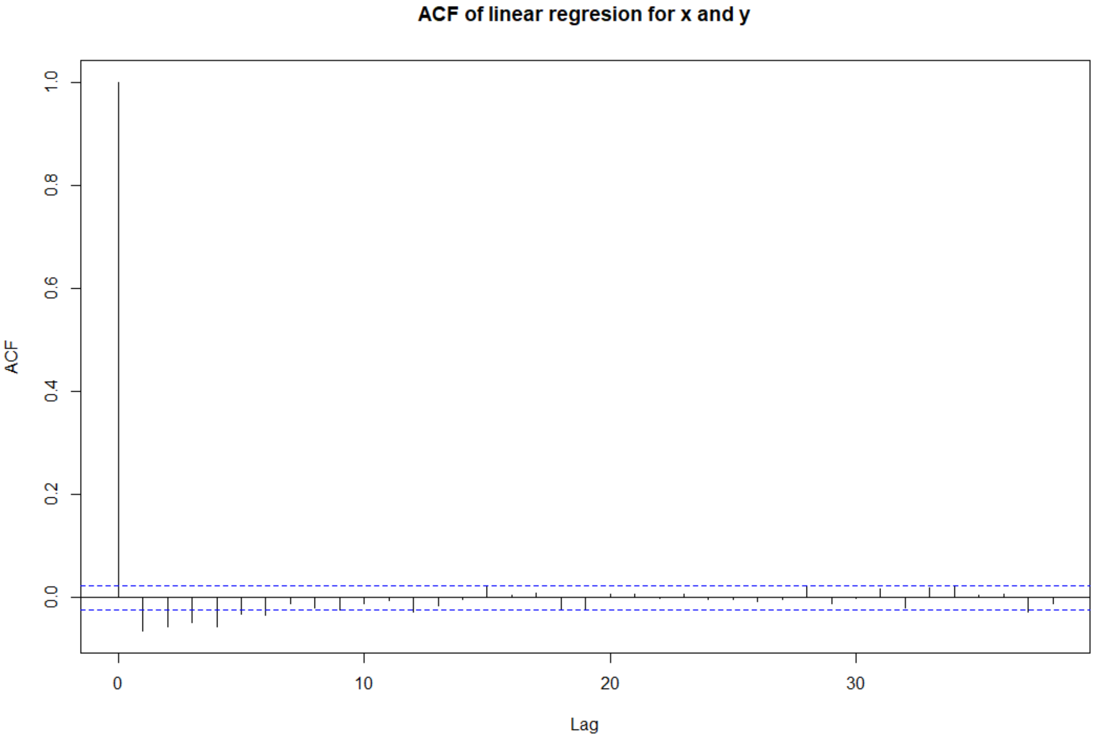

Figure 7: Autocorrelation function of linear regression for difference vectors y and s over a lag of up to 38

From figures 6 and 7 we observe that the ACF of the residuals of the difference vectors implies faster decay, less significance, and stationary behaviour.

IV. Analysis of regression techniques

Let us fit an ARIMA (1,0,1) model to the difference vectors x and y and use it to compare the performance of the regression with time series errors and the simple regression:

```
> model3=arima(y_n, order=c(1,0,1),xreg=x_n,include.mean=F)
> model3

Call:
arima(x = y_n, order = c(1, 0, 1), xreg = x_n, include.mean = F)

Coefficients:
         ar1      ma1     x_n
      0.8204  -0.9359  0.7203
s.e.  0.0127   0.0083  0.0177

sigma^2 estimated as 8.389e-08:  log likelihood = 47499.05,  aic = -94990.11
> rsq=(sum(y_n^2)-sum(model3$residuals^2))/sum(y_n^2)
> rsq
[1] 0.180411
```

Hence, the ARIMA (1,0,1) model is given by:

$$y_t = c + 0.8204y_{t-1} - 0.9359e_{t-1} + 0.7203x_t \qquad (8)$$

One can compare the performance of simple regression and the regression with time series errors by conducting a t-test or an F-test and check whether the difference in the prediction errors is statistically significant. We see that the R-squared value of the ARIMA (1, 0, 1) model is 18.04% which is higher than that of the simple regression (15.09%). A larger R-squared value for ARIMA (1, 0, 1) implies that the model explains more variance in the dependent variable. However, it is important to keep in mind that R-squared only measures the proportion of explained variance, and does not necessarily mean that the model is a good predictor.

Hence we shall split our data into two data sets where one is training data and the other is testing data like so:

```
> df_train <- df[row.names(df) %in% 1:5671, ]
>
> df_test <- df[row.names(df) %in% (5671+1):nrow(df),]
> x_train <- df_train[,c(1)]
> x_test <- df_test[,c(1)]
> y_train <- df_train[,c(2)]
> y_test <- df_test[,c(2)]
> model1_train <- lm(formula = y_train ~ x_train, data = df_train)
> model3_train <- arima(x = y_train, order = c(1, 0, 1),
                        xreg = x_train, include.mean = F)
```

We have separated the first 80% of the data frame from the rest of the data frame to create a set of training data to create two new models based off the old models we already have to see how good they are at predicting the other 20% of their own model:

```
> model1_prediction <- predict(model1_train, n.ahead = length(y_test),
                                newxreg = x_test)
> model3_prediction <- predict(model3_train, n.ahead = length(y_test),
                                newxreg = x_test)
> mae(model1_prediction, y_test)
[1] 0.0002289542
> mae(model3_prediction$pred, y_test)
[1] 0.000210203
> mse(model1_prediction, y_test)
[1] 1.130715e-07
> mse(model3_prediction$pred, y_test)
[1] 8.653581e-08
> AIC(model1_train)
[1] -76087.26
> AIC(model3_train)
[1] -76284.91
```

From these metrics that we have calculated, we can see that the ARIMA (1, 0, 1) model achieves a lower MAE (mean absolute error), MSE (mean squared error), and Akaike Information Criterion compared to simple regression. MAE is a measure of the average magnitude of the error, calculated as the average of the absolute differences between the predicted and actual values. A lower MAE indicates that the model's predictions are closer to the true values. MSE is a measure of the average squared differences between the predicted and actual values so a lower MSE indicates that the model's predictions are closer to the true values. AIC is used as a model selection criterion and balances the goodness of fit of a model with the number of parameters used. A lower AIC value indicates that the model has a good fit to the data while using fewer parameters. Hence, by all these metrics, we can conclude that the ARIMA (1, 0, 1) model is better for prediction.

V. Conclusion

In conclusion, we have gathered a substantial amount of strong evidence to suggest that regression with time series errors gives better prediction results than simple regression with regards to the data set *sp5may.dat*. The results of the regression with time series errors showed a consistently lower mean squared error and mean absolute error compared to the simple regression, indicating a better fit to the data. Hence we accept our hypothesis that we introduced in the introduction of this report. The improvements in the prediction results can be attributed to the ability of the time series errors model to capture the underlying patterns and trends in the data set.

Regression with time series errors gives better prediction results than simple regression because it accounts for the temporal dependence in the data. Time series data often exhibit patterns and trends that are not captured by simple regression models. Regression with time series errors incorporates these patterns and trends by modeling the errors in the regression as a time series, allowing for a more accurate and robust prediction. This is particularly useful when there is a clear temporal structure in the data, such as seasonality or autocorrelation. By taking these dependencies into account, the time series errors regression model provides a better fit to the data and a more accurate prediction compared to simple regression models.

Whilst performing this analysis, we notice that the implementation of the time series errors regression showed significant computational efficiency, making it a feasible option for large-scale data analysis. Another important outcome of this analysis is that we see that the methodology used in section 2.9 of Ruey S. Tsay's book 'Analysis of Financial Time Series' can be applied to many further high-frequency data sets.