

PSTAT 174/274

FINAL PROJECT

**Time Series Analysis of Credit Suisse Liquid
Alternative Beta Monthly Index**

Authors:

Emily LU
Enbo ZHOU
Vanessa HO
Xingyu CHEN

Supervisor:

Professor Giri GOPALAN

December 6th, 2019

University of California, Santa Barbara

Abstract

Many data in finance are time series, for example stock price, fund price and etc. These financial asset prices are very complicated and subject to many factors. In this project, we exploit the analytical tools from the course: Time Series Analysis, to analyze real-world financial data. We picked Credit Suisse Liquid Alternative Beta Index's monthly net asset values for example. We did some exploratory analysis and data transformation at first. Afterwards, we completed a time domain analysis SARIMA model to fit our data and make predictions. We also completed a frequency analysis to investigate the cycle characteristics of the time series. Finally, we completed an attempt of the GARCH model to fit the data. The time series analyses reveal some essential characteristics of the hedge fund index and make a decent prediction. To get a better performance, we may use daily data or use other models, like GARCH and VAR, in the future.

Contents

1	Introduction	1
2	Data Description	1
2.1	Data Exploration	1
2.2	Decomposition	3
3	Data Transformation	4
3.1	Box-Cox Transformation	4
3.2	Remove Seasonality and Trend	5
4	Model Identification and Estimation	7
4.1	Preliminary Model Identification	7
5	Model Selection	9
5.1	Model Estimation	9
6	Diagnostics	10
6.1	Normality	10
6.2	Independence (Serial Correlation) Checking	11
6.3	Constant Variance Checking	11
7	Forecasting	14
8	GARCH Model	16
9	Spectral Analysis	22
10	Concluding Remarks	23
11	Contribution Statement	24
	Appendix: Additional R Code	25
	Reference	31

1 Introduction

The objective of this project is demonstrate the application of Time Series Analysis towards quantitative finance. In particular, our goal is to identify trends, seasonal variations and correlation using time series analyses, and ultimately generate investment signals or filters based on inference or predictions. To do so, our approach will be to forecast and predict future values, simulate financial time series, and make relationship inferences. We aim to generate a model for real world application, especially to help predict the expectancy of various investment methods for fairly new investors in the industry. The data we will be using to do our time series analyses is Credit Suisse Liquid Alternative Beta Index's monthly net asset values from January 1998 to December 2019.

2 Data Description

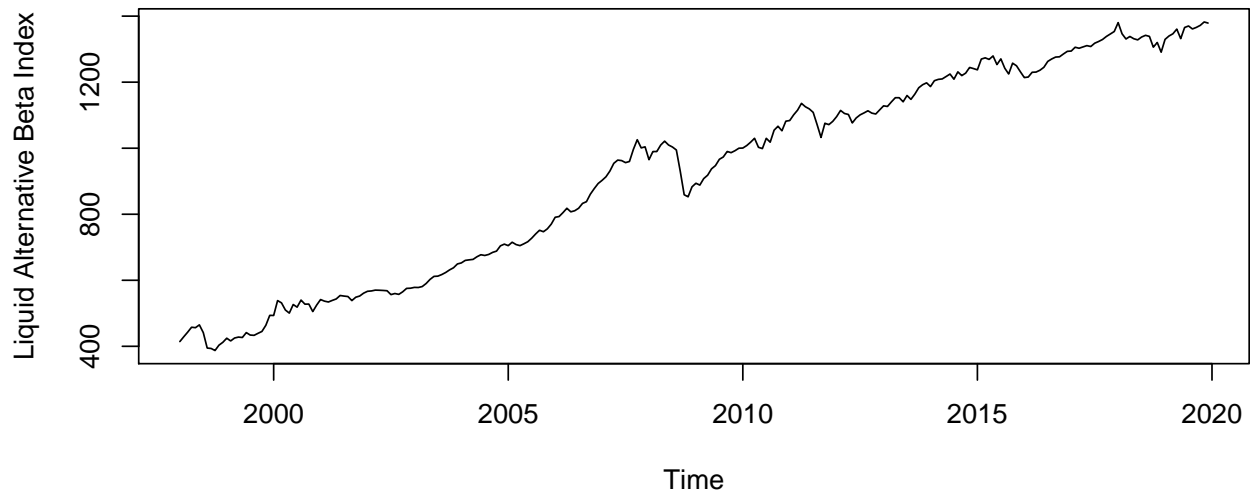
Credit Suisse' Liquid Alternative Beta Index ("LAB") is an index that reflects the returns of the individual LAB strategy indices - Long/Short, Event Driven, Global Strategies, Merger Arbitrage and Managed Futures – scaled accordingly to their respective strategy weights in the Credit Suisse Hedge Fund Index. For our purpose, we will be analyzing the monthly net asset values of Credit Suisse's Liquid Alternative Beta Index. To gather our data, we registered for free on Credit Suisse Hedge Fund Index's official website and downloaded a CSV file of the Liquid Alternative Beta monthly index values.

Subindex of LAB	Index Description
Event Driven Liquid Index	Reflects the returns of hedge funds represented by the EventDriven sector of Credit Suisse Hedge Fund Index on liquid securities
Global Strategies Liquid Index	Reflects the return of all hedge fund strategies on liquid securities not defined as Long/Short or Event Driven within the Credit Suisse Hedge Fund Index
Managed Futures Liquid Index	Reflects the Managed Futures strategy by using a pre-defined quantitative method to invest in equities, fixed income, commodities, currencies, etc. . .
Merger Arbitrage Liquid Index	Reflects the Merger Arbitrage strategy by using a pre-defined quantitative method to invest in diversified and broadly representative set of announced merger deals.

2.1 Data Exploration

With a total of 265 observations, our original dataset contains 2 variables: date on the monthly bases from January 1998 to December 2019 and monthly Liquid Alternative Beta Index from Credit Suisse. We decided to split the dataset into training and testing dataset and to cut the last 12 observations as our testing data for later comparison with our forecasted values. Thus, there are 253 training data and 12 testing data. For our preliminary data exploration, we transformed the Liquid Alternative Beta Index data into time series format and plotted it below with all the 265 observations.

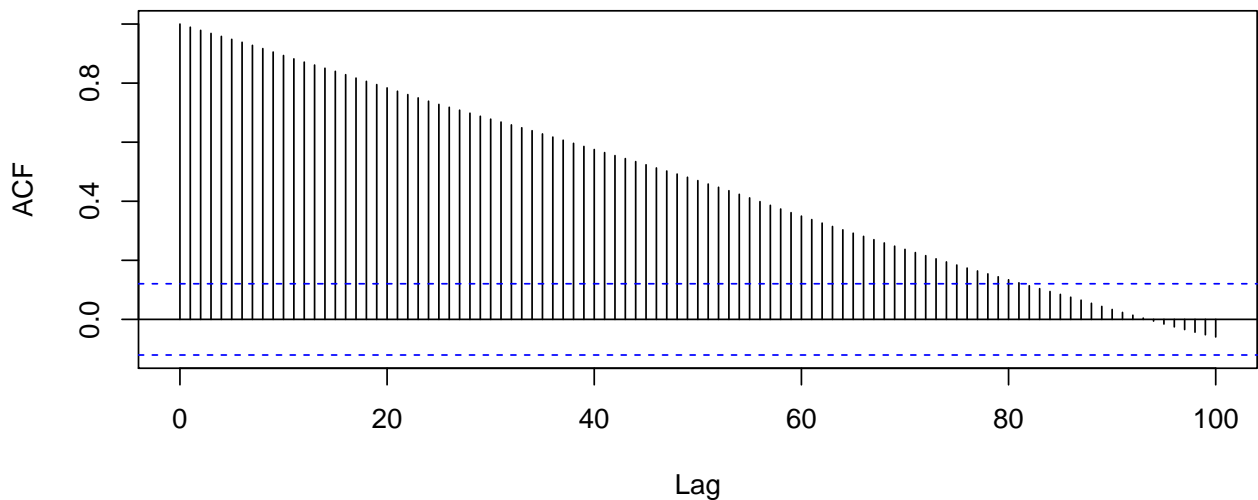
Liquid Alternative Beta Index Price Series



(Figure 2.1) Long Liquid Alternative Beta Index from Jan. 1998 to Dec. 2019

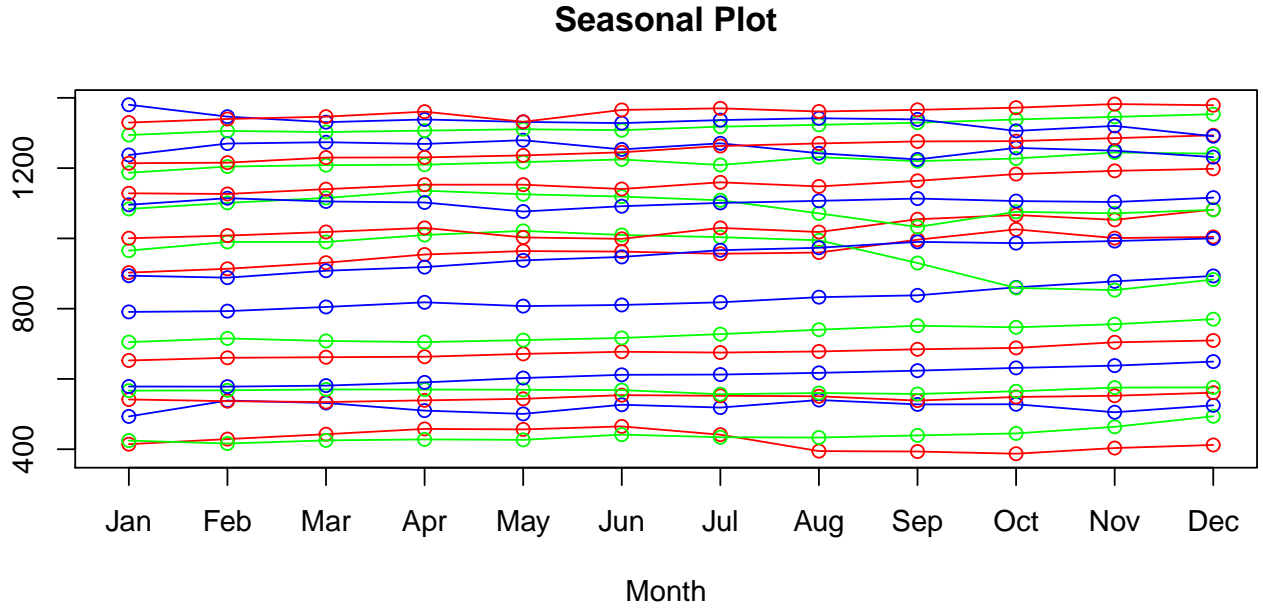
From Figure 2.1, we can see an upward trend over time, indicating the increase in index values. Besides, from the ACF Plot below, we can also tell that the data is not stationary, since the decay of ACF Plot is very slow.

ACF of Raw Data



(Figure 2.2) ACF Plot

From Figure 2.1 and Figure 2.2, the seasonality is not captured clearly, so we drew a seasonal plot as shown in Figure 2.3 below.

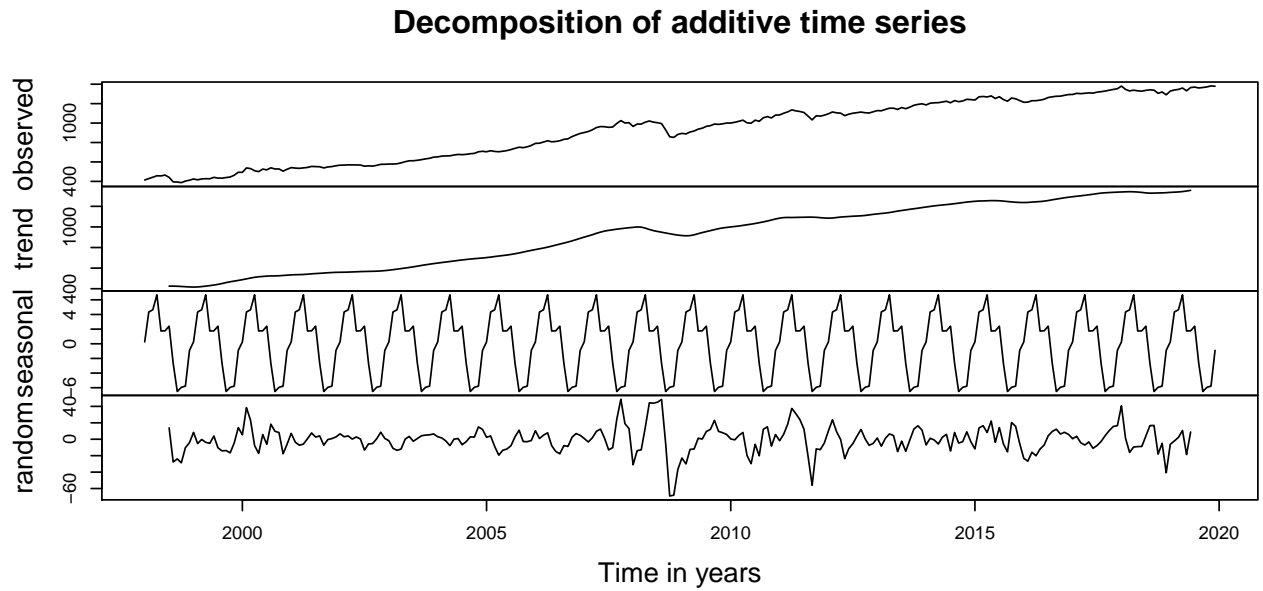


(Figure 2.3) Seasonal Plot

From Figure 2.3, the seasonality is not very obvious, so we need to build a more accurate plot to claim the existence of seasonality and trend for the data. Therefore, we will make a decomposition model in the next section.

2.2 Decomposition

The decomposed model is expressed as $Y_t = m_t + s_t + S_t$, where m_t is the trend component, s_t is the seasonal component, and S_t is the stationary process.



(Figure 2.4) Decomposition Plot

From this particular plot, we can observe that there is a clear existence of seasonality with an upward

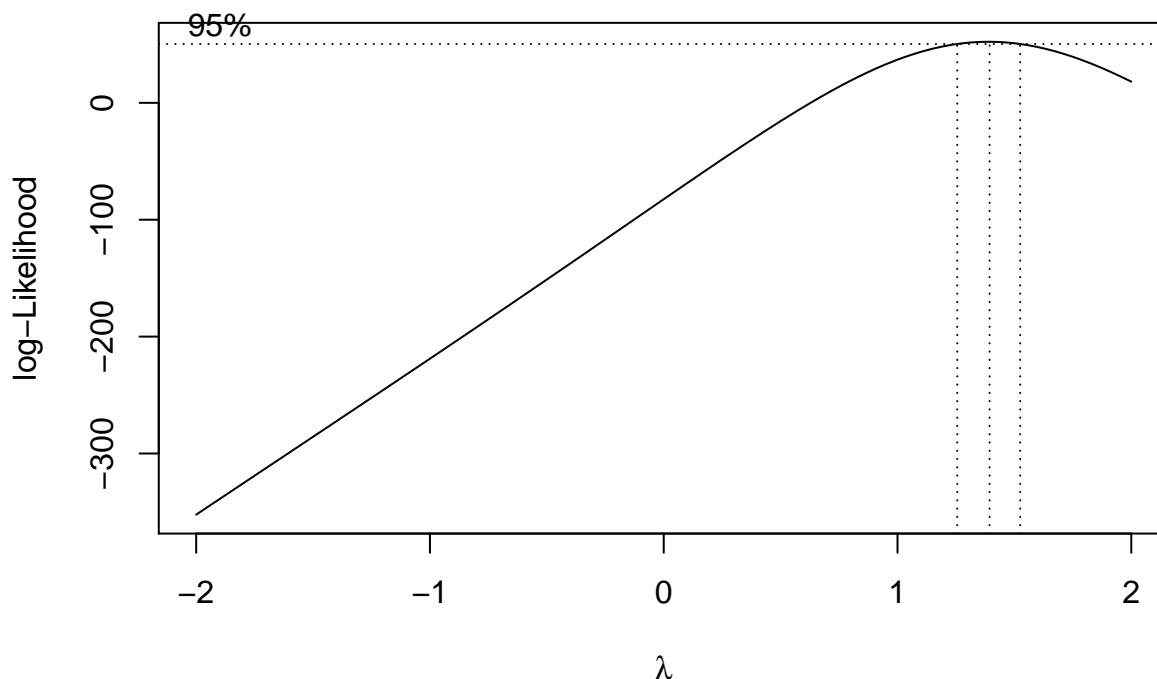
trend. However, the data does not appear to be stationary. With the data being non-stationary, it is difficult to model because the estimate of the mean and some variance would be constantly changing. Thus, before we perform the model building section, as well as data selection, we need to transform and difference the data to make the data stationary.

3 Data Transformation

To make our data stationary, we first transformed the data to stabilize variance, and then we used differencing method to remove seasonality and trend of the data.

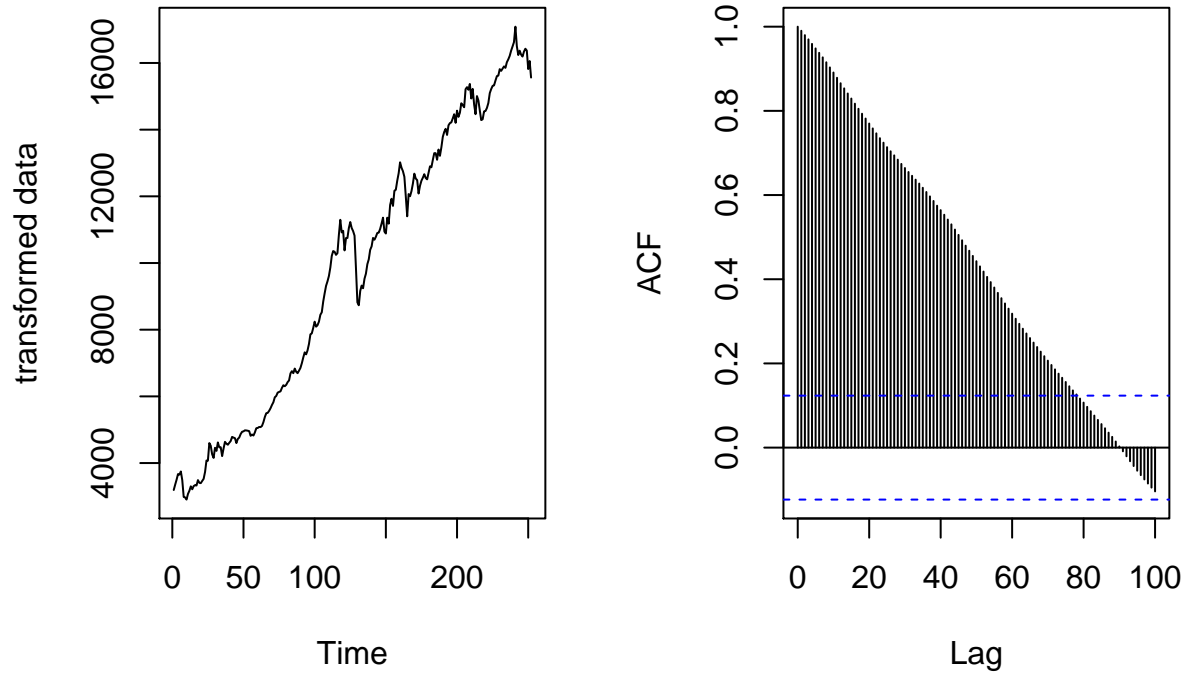
3.1 Box-Cox Transformation

We used the Box-Cox transformation, an exponent, $\lambda \in (-5, 5)$ to stabilize the variance of data and convert the data to a normally distributed one. By applying Box-Cox, we are able to assume normality and perform more tests. To find λ , we graphed a Box-Cox plot below.



(Figure 3.1.1) Box-Cox Plot

As shown in the plot above, there is a 95% confidence interval for the parameter, as seen by the two dashed vertical lines on each end. We observed that the confidence interval does not contain 0, therefore we could use the best λ ($\lambda = 1.393939$) to do the transformation $W_t = (Y_t\lambda - 1)/\lambda$ to stabilize variance.



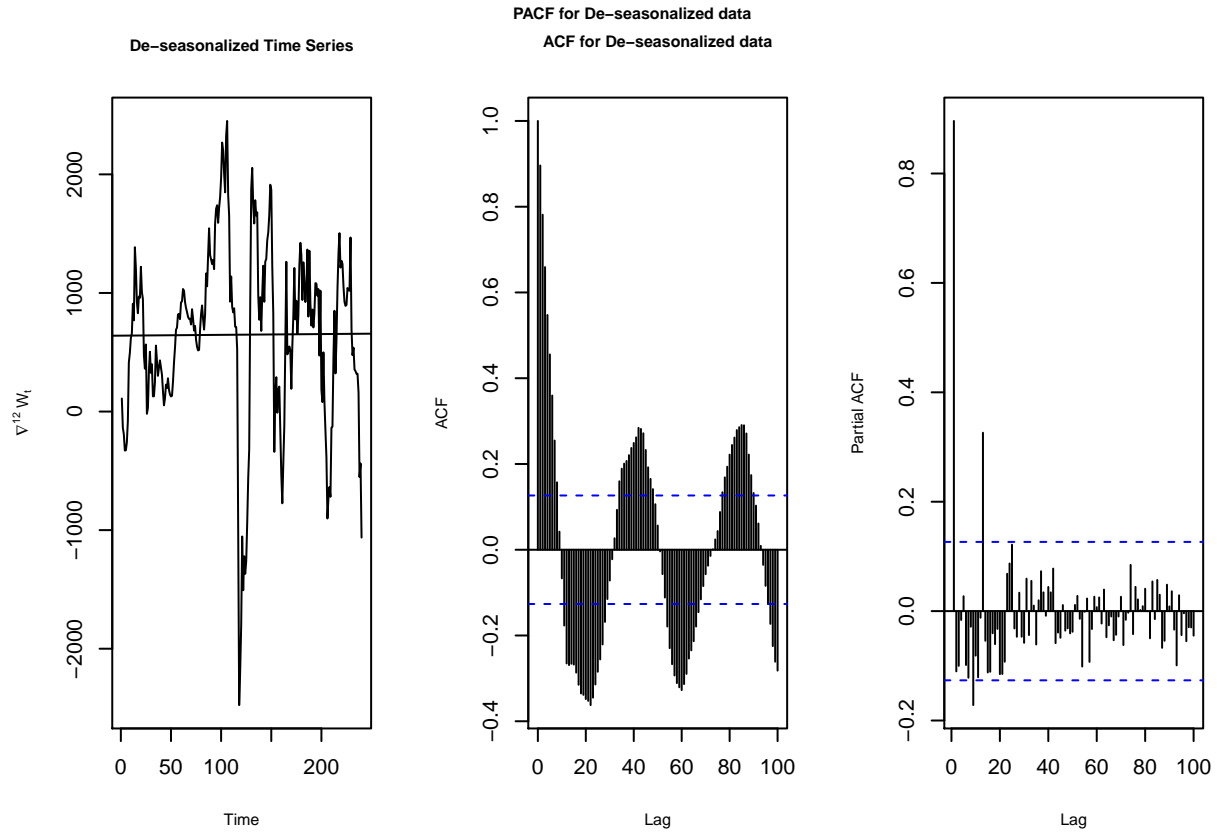
After transforming our data and plotted the ACF plot, our resulting graph above still indicates that our data is non-stationary. Thus, we then choose to perform the differencing method to stabilize our mean, which in turn de-seasonalize and detrend the data.

3.2 Remove Seasonality and Trend

We used the differencing method to remove seasonality and trend. After having observed the seasonality to be 12 months, we decided to start off by differencing data at lag 12 to remove seasonality. We applied the lag d difference as:

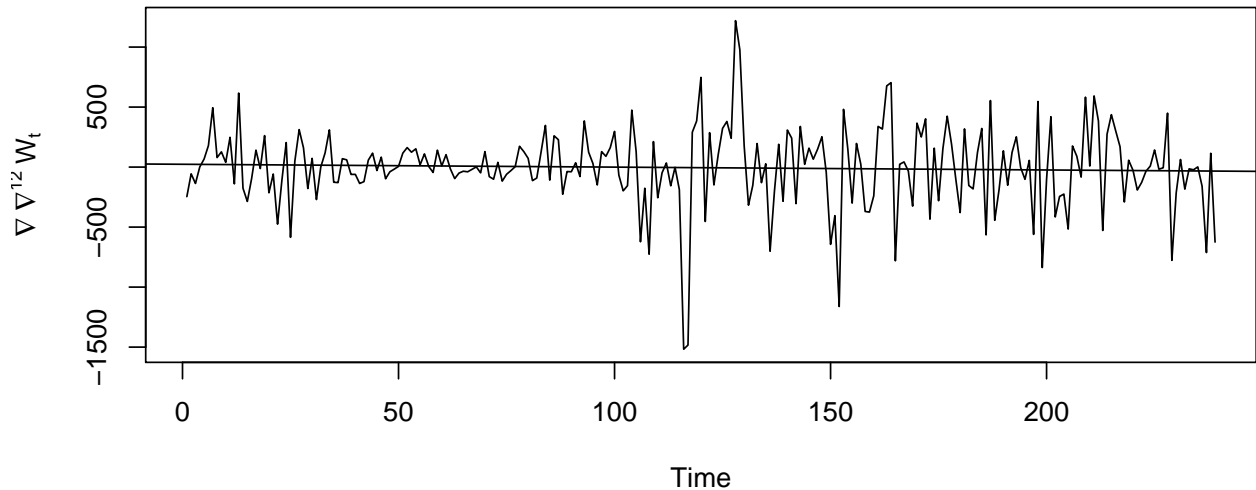
$$\nabla_{12}W_t = W_t - W_{t-12}$$

and plotted the deseasonalized data we obtained below.



Then, we need to remove the trend based on the de-seasonalized data we obtained. To do so, we difference the deseasonalized data at lag 1 and plotted the new data below.

Deseasonalized/Detrended Time Series



We obtained a variance of 112090.9. After removing the trend component, the trend line appears to be a horizontal line. If we difference it once more, the variance will go up to 202131.2, which indicates an over differencing. Therefore, we concluded that we would only need to difference the data at lag 12 in order to remove seasonality and then difference at lag 1 to remove trend. Our new

model is now

$$X_t = \nabla \nabla_{12} W_t.$$

To verify whether X_t is stationary or not, we used the Augmented Dickey-Fuller Test with the following hypotheses:

$$H_0 : X_t \text{ is non-stationary VS. } H_a : X_t \text{ is stationary.}$$

According to the test result, we obtained a p-value of 0.01. Since our p-value is less than 0.05, we reject the null hypothesis and conclude that X_t is stationary at 95% confidence level. Now, we can proceed with the model building and selection based on the stationary data.

4 Model Identification and Estimation

We fit our data into a SARIMA model since our data shows seasonality. The SARIMA model which accounts for seasonal and non-seasonal behavior is as below:

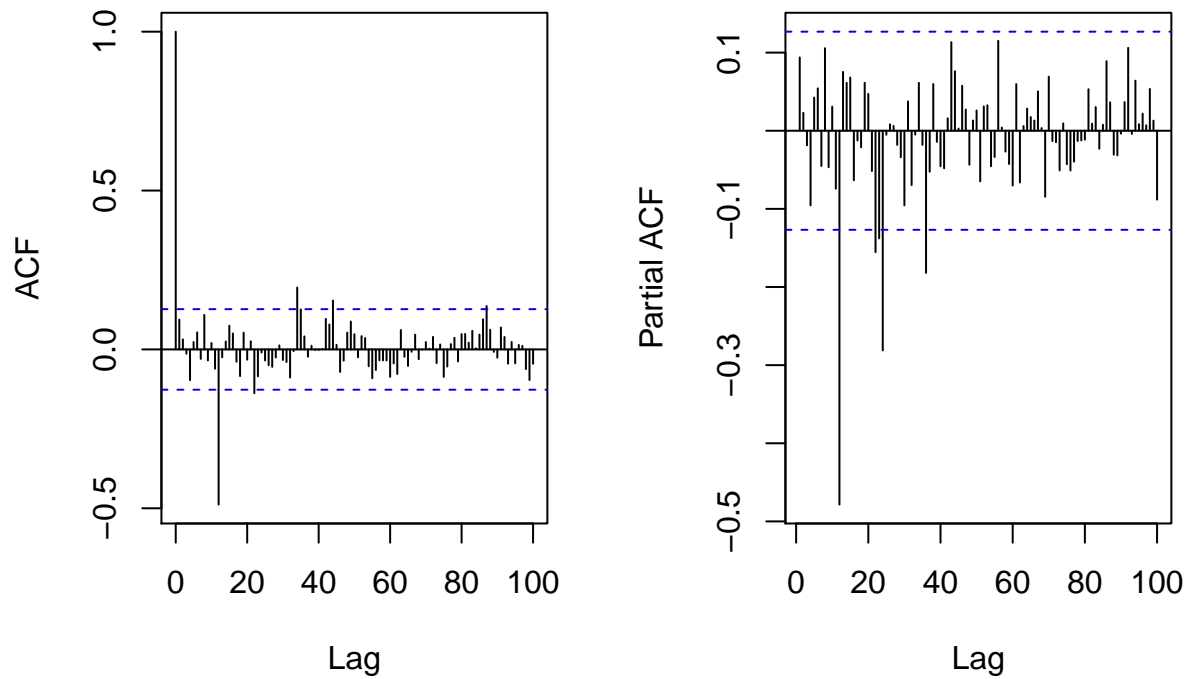
$$SARIMA(p, d, q) \times (P, D, Q)_s,$$

where p is the order of non-seasonal AR process, d is the order of non-seasonal differencing, q is the order of non-seasonal MA process, P is the order of seasonal AR process, D is the order of seasonal differencing, Q is the order of seasonal MA process, s is the seasonality period. Based on our monthly index data, we found that the seasonality period is 12, i.e. $s = 12$. First, we difference the transformed data at lag 12 to remove seasonality, then difference at lag 1 to remove trend and we found our deseasonality and detrend data to be a stationary data, so for our data, $D = 1$ and $d = 1$. Next, we need to find non-seasonal order p, q and seasonal order P, Q .

4.1 Preliminary Model Identification

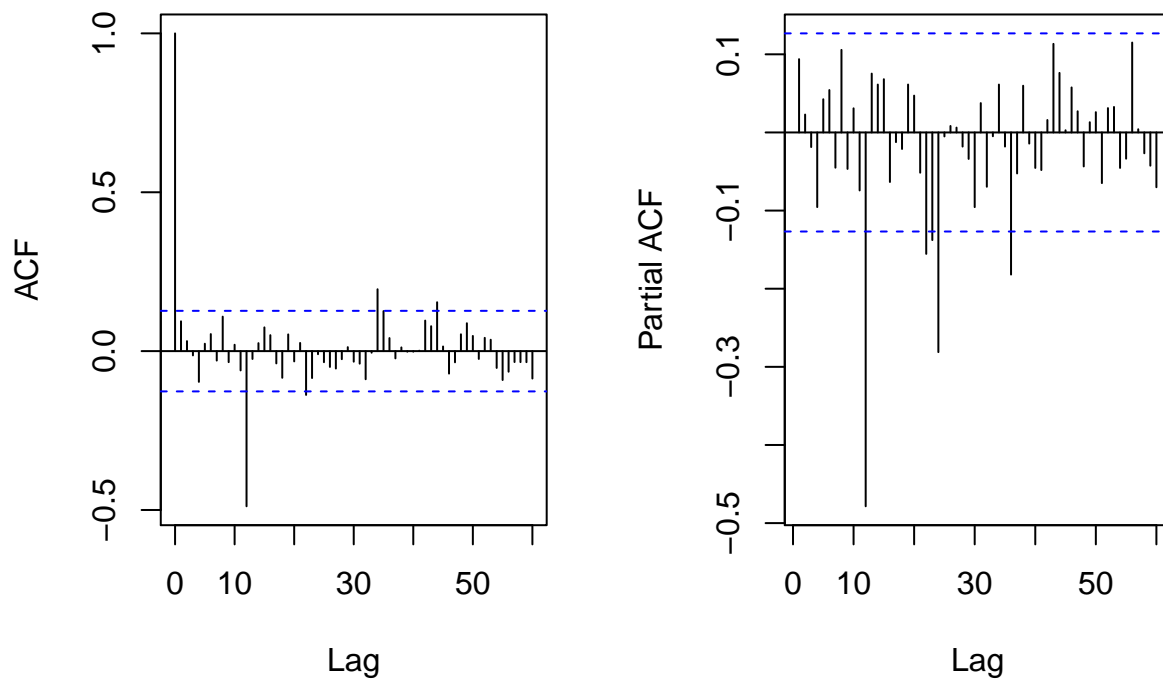
To identify the seasonal order P and Q , we checked the ACF and PACF values of the deseasonality and detrend time series, $\nabla \nabla_{12} W_t$, at seasonal lags, i.e. lag 12, 24, 36, 48, etc. Then we plotted the ACF and PACF plot below to find the seasonal orders.

PACF for Deseasonalized/Detrended data **ACF for Deseasonalized/Detrended**



As we could see from the figure above, the ACF cuts off at lag 12 ($Q = 1$), the PACF cuts off at lag 36 ($P = 3$). Thus, we conclude that $Q = 0, 1$ and $P = 0, 1, 2, 3$.

Now, to identify the non-seasonal order p and q , we used the first 11 lags of the ACF and PACF plots.



As seen in the figure above, the ACF cuts off at lag 0 ($q = 4$) and PACF cuts off at lag 0 ($p = 3$), indicating ARMA (0,0) process. Using AIC and BIC criterion, all combinations of the non-seasonal orders and seasonal orders were applied to select the best model.

5 Model Selection

We used the information criterions such as AIC and BIC to select the best model. The model with relatively small AIC and BIC values will be chosen. For each P and Q 's combination, we iterated through a loop of possible p and q values. The minimum AIC and BIC gave us a result when $P = 1$, $D = 1$, $Q = 1$, $p = 0$, $d = 1$, $q = 0$. The second minimum AIC and BIC gave us a result when $P = 0$, $D = 1$, $Q = 1$, $p = 0$, $d = 1$, $q = 0$. The third minimum AIC and BIC values gave us a result when $P = 2$, $D = 1$, $Q = 1$, $p = 0$, $d = 1$, and $q = 0$. The fourth minimum AIC and BIC values gave us the result $P = 3$, $D = 1$, $Q = 1$, $p = 0$, $d = 1$, $q = 0$. Therefore, we obtained four possible models based on our results.

We then fitted the first four possible models,

$$SARIMA(0, 1, 0) \times (1, 1, 1)_{12}$$

$$SARIMA(0, 1, 0) \times (0, 1, 1)_{12}$$

$$SARIMA(0, 1, 0) \times (2, 1, 1)_{12}$$

$$SARIMA(0, 1, 0) \times (3, 1, 1)_{12}.$$

5.1 Model Estimation

For the four models we selected above, we fit and estimate the coefficients based on the MLE method. The results for all three models are showed in Table 4.2.

Denote X_t as the transformed, de-trended and de-seasonalized data, $X_t = (Y_t 1.3939 - 1)/1.3939$, where Y_t is our original index data. Model 1: $SARIMA(0, 1, 0)(0, 1, 1)_{12}$

$$X_t = (1 \sim B^{12})Z_t, \text{ where } Z_t \sim N(0, 52886);$$

Model 2: $SARIMA(0, 1, 0)(1, 1, 1)_{12}$

$$(1 + 0.1042B)X_t = (1 \sim 0.9994B^{12})Z_t, \text{ where } Z_t \sim N(0, 51911);$$

Model3: $SARIMA(0, 1, 0)(2, 1, 1)_{12}$

$$(1 + 0.1374B + 0.0797B^2)X_t = (1 \sim 0.9262B^{12})Z_t, \text{ where } Z_t \sim N(0, 54183);$$

Model 4: $SARIMA(0, 1, 0)(3, 1, 1)_{12}$

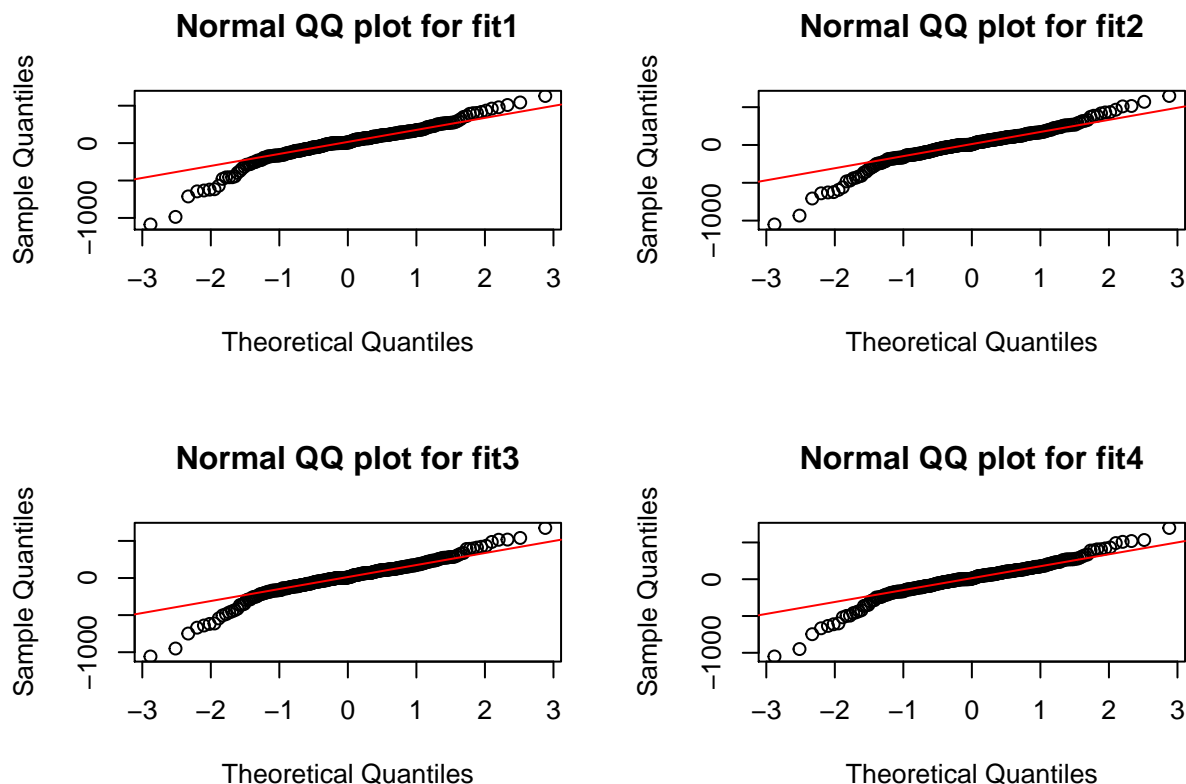
$$(1 + 0.1214B + 0.0634B^2 \sim 0.0301B^3)X_t = (1 \sim 0.9516B^{12})Z_t, \text{ where } Z_t \sim N(0, 53536);$$

6 Diagnostics

Now, we will validate the assumptions of the four models, which include normality, independence and constant variance of errors.

6.1 Normality

We used Q-Q plots to check whether residuals of our models are normally distributed as shown in the figure below. In the Q-Q plots, residuals of all models do not appear to be normally distributed.



We also performed the Shapiro-Wilk Test at $\alpha = 0.05$ to further test the normality assumption.

H_0 : residuals are normally distributed vs. H_a : residuals are not normally distributed

The results are shown in Table 5.1:

(Table 5.1) Shapiro-Wilk Test

	W Stat	p-value
Model 1	9.2342e-01	4.122e-10
Model 2	9.2892e-01	1.21e-09
Model 3	9.3052e-01	1.673e-09
Model 4	9.2973e+04	1.425e-09

According to Table 5.1, all p-values are smaller than $\alpha = 0.05$, therefore we reject the null hypothesis that the residuals are normal. These results might have occurred due to the hidden components within our data that we are unable to detect.

6.2 Independence (Serial Correlation) Checking

Next, we performed the Ljung-Box Test and the Box-Pierce Test at $\alpha = 0.05$ to check whether the residuals in our model are serial uncorrelated with their own lagged values. The results are presented in Table 5.2:

(Table 5.2) Ljung-Box and Box-Pierce Tests Results

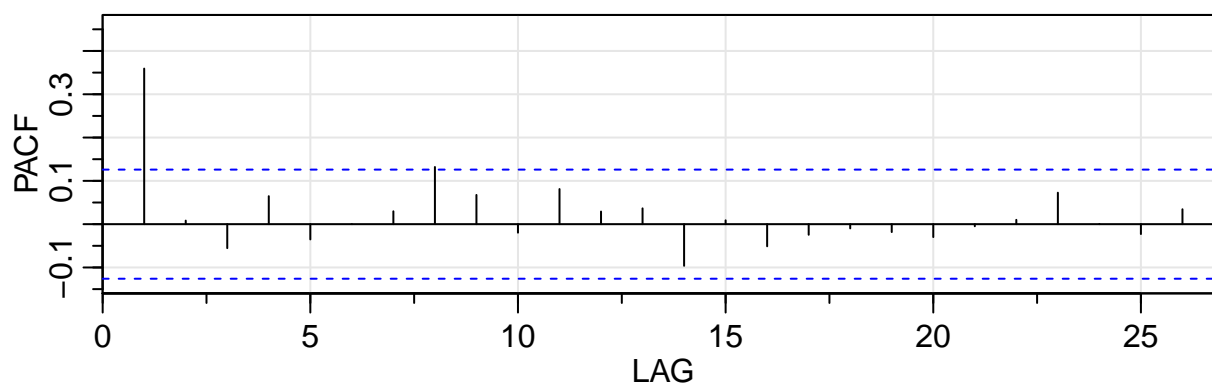
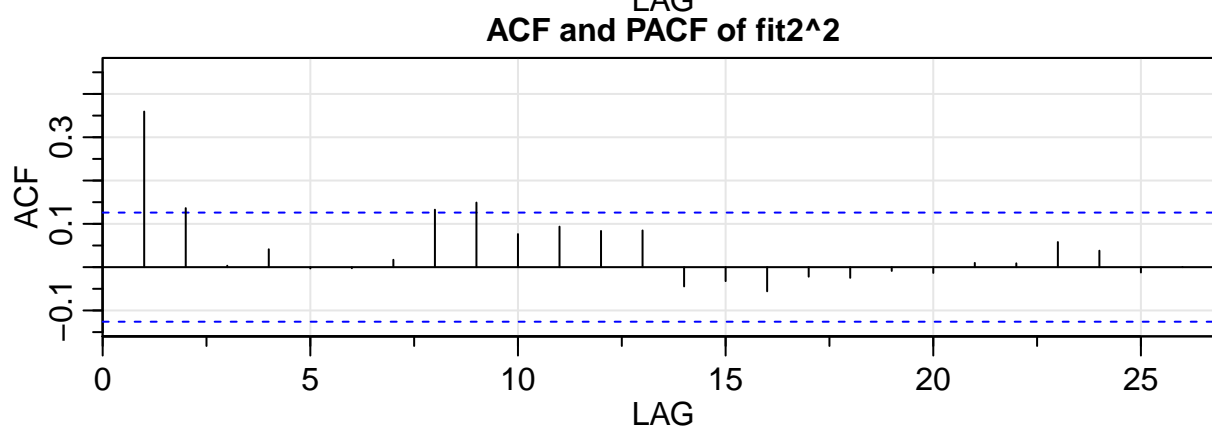
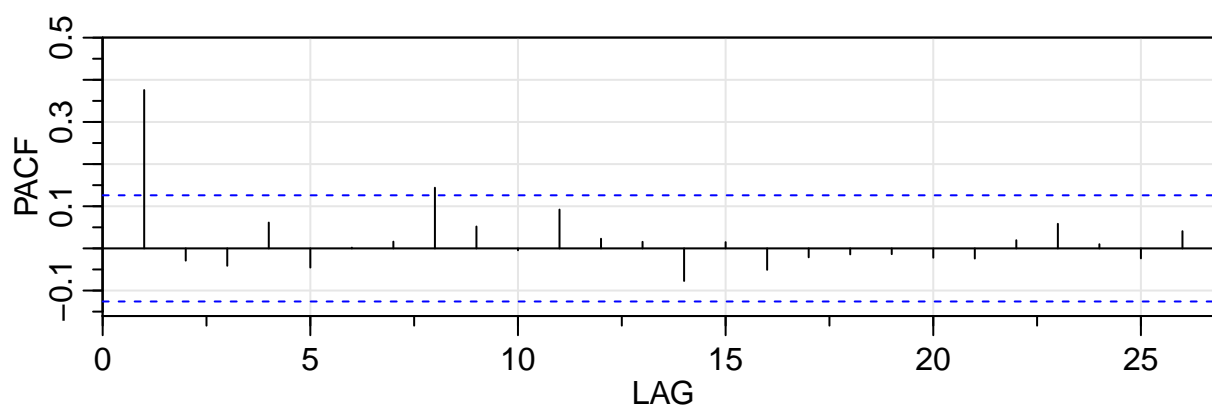
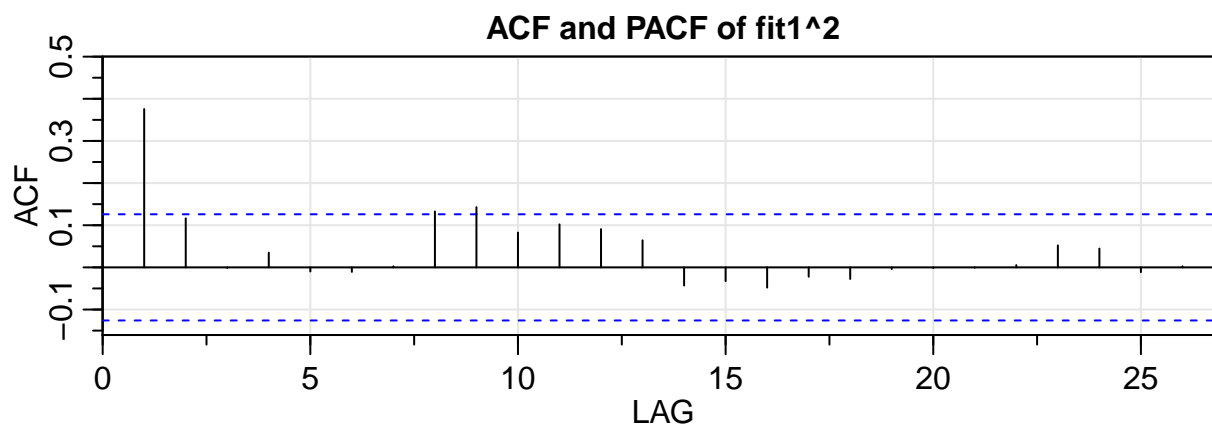
	Ljung-Box	Box-Pierce
Model 1	0.3745	0.3773
Model 2	0.4293	0.4320
Model 3	0.4939	0.4965
Model 4	0.5120	0.5145

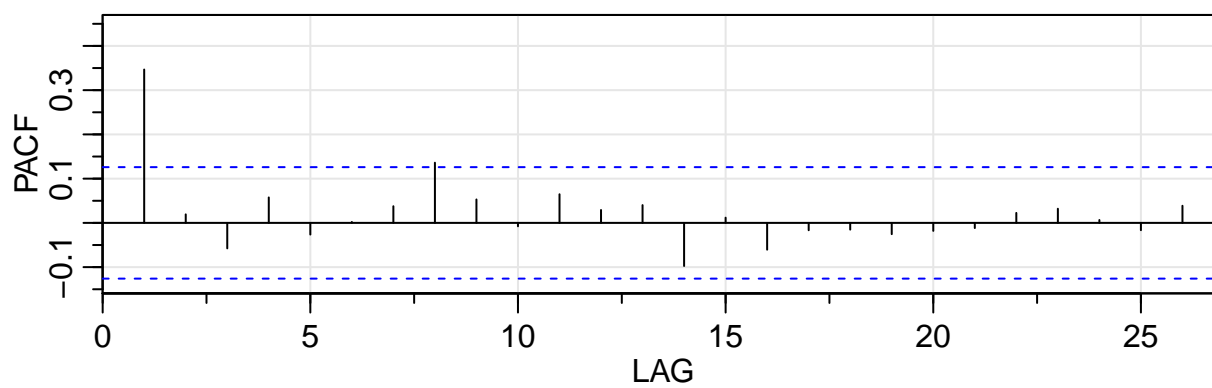
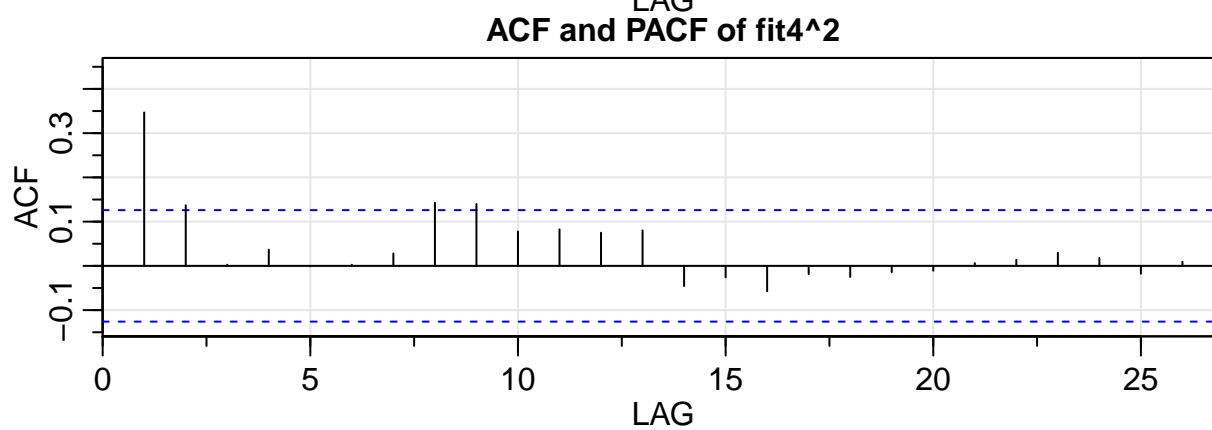
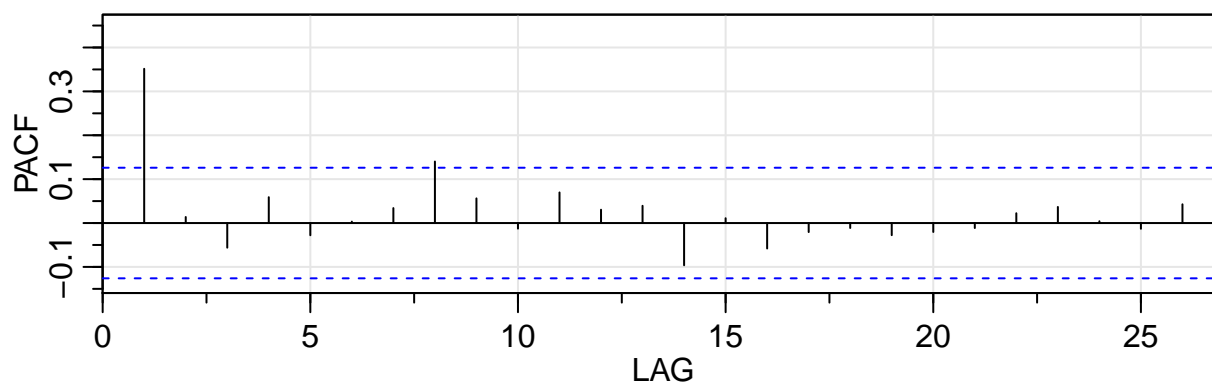
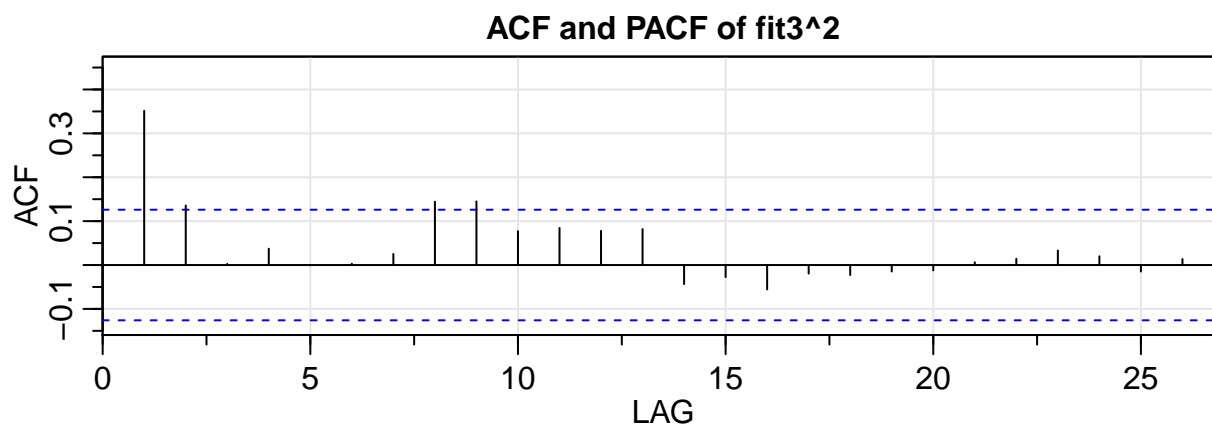
The results in Table 5.2 shows that all p-values are larger than $\alpha = 0.05$, hence we fail to reject the null hypothesis and concluded that the residuals are serially uncorrelated.

6.3 Constant Variance Checking

To check the assumption of homoskedasticity, we use the ACF and PACF plots of the squared residuals in our models. If the ACF and PACF values lie within 95% of the White Noise limit (blue dashed lines), then we can conclude that our model fitting has constant variance.

As seen from the figures below, the ACF and PACF plots of squared residuals in these four models do lie within the 95% of the White Noise Limits (blue dashed lines). Therefore, our models have constant variance.





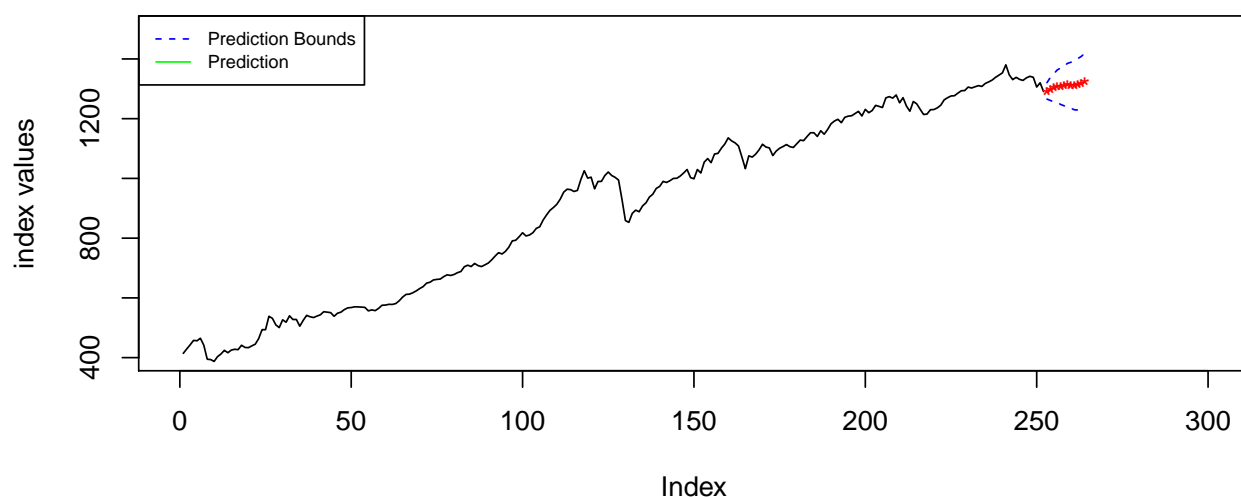
Finally, we chose model 1 and the final model is:

$$SARIMA(0, 1, 0)(0, 1, 1)_{12}X_t = (1 \sim B^{12})Z_t, \text{ where } Z_t \sim N(0, 52886).$$

7 Forecasting

Since the goal of our project is to predict the future index values of Credit Suisse Liquid Alternative Beta Index, we will now forecast future 12 months returns by minimizing the mean absolute forecast errors, from January 2019 to December 2019 at monthly intervals.

Forecasting based on data for the next twelve points

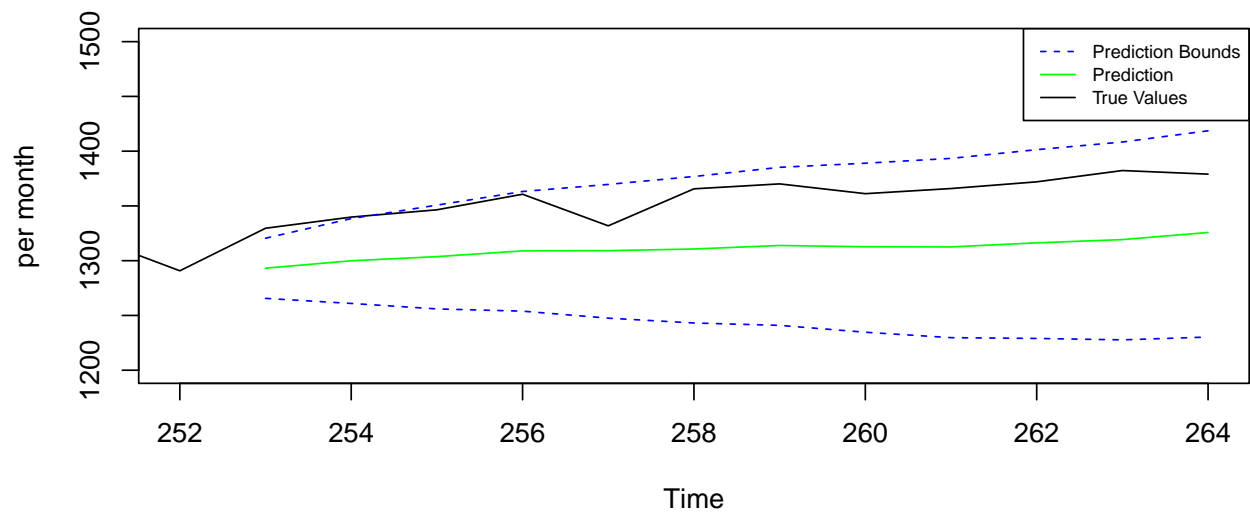


(Figure 2.1) Long Liquid Alternative Beta Index from Jan. 1998 to Dec. 2019

Figure 7.1 shows the twelve forecasted index values with a 95% confidence interval.

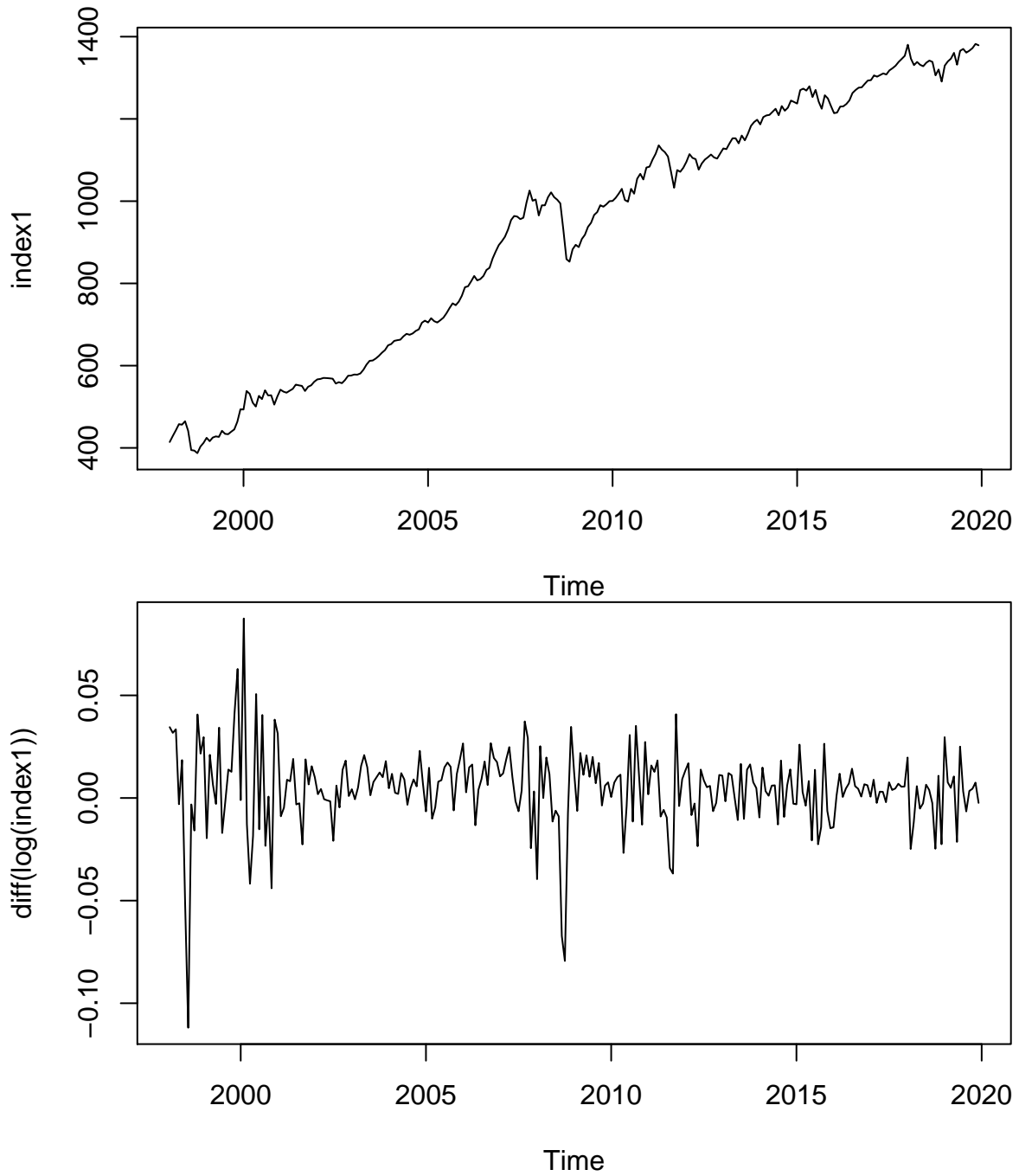
We could now construct a plot to compare the real data with our predictions. In Figure 7.2 below, we could check to see if the pattern of our predicted values is similar to the original data within our upper and lower confidence interval boundaries.

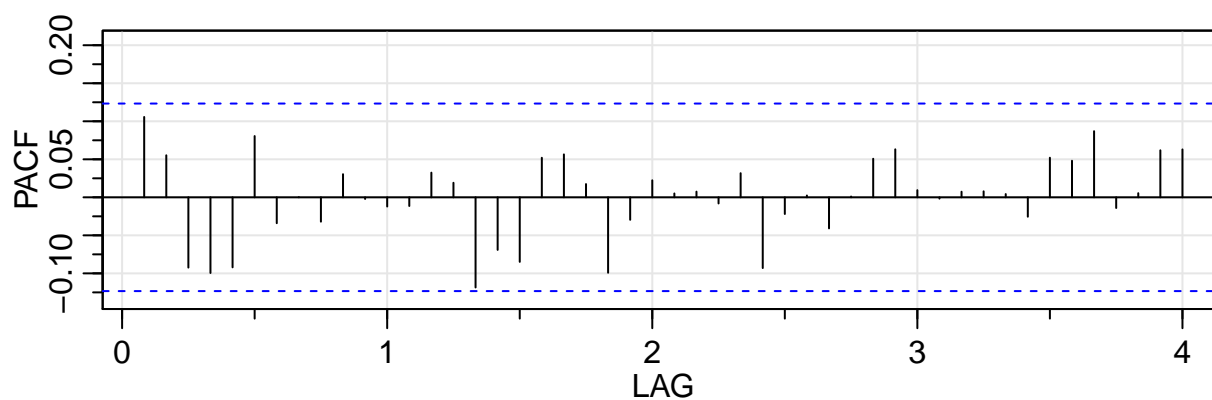
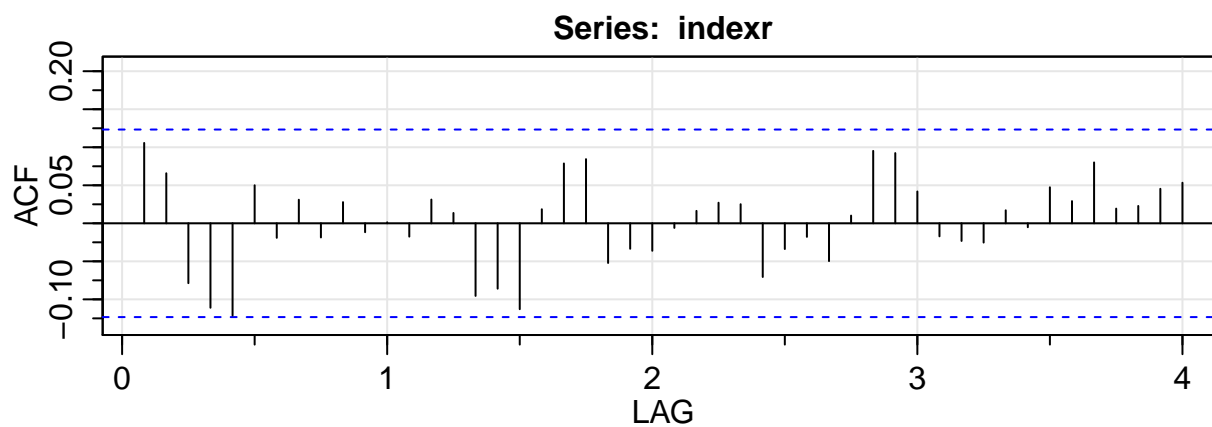
Comparison between Observed Values and Forecasted Values



(Figure 7.2) A Comparison of True Values and Forecasted Values

8 GARCH Model





##		ACF	PACF
##	[1,]	0.11	0.11
##	[2,]	0.07	0.06
##	[3,]	-0.08	-0.09
##	[4,]	-0.11	-0.10
##	[5,]	-0.12	-0.09
##	[6,]	0.05	0.08
##	[7,]	-0.02	-0.03
##	[8,]	0.03	0.00
##	[9,]	-0.02	-0.03
##	[10,]	0.03	0.03
##	[11,]	-0.01	0.00
##	[12,]	0.00	-0.01
##	[13,]	-0.02	-0.01
##	[14,]	0.03	0.03
##	[15,]	0.01	0.02
##	[16,]	-0.10	-0.12
##	[17,]	-0.09	-0.07
##	[18,]	-0.11	-0.08
##	[19,]	0.02	0.05
##	[20,]	0.08	0.06
##	[21,]	0.08	0.02
##	[22,]	-0.05	-0.10
##	[23,]	-0.03	-0.03

```

## [24,] -0.04  0.02
## [25,] -0.01  0.01
## [26,]  0.02  0.01
## [27,]  0.03 -0.01
## [28,]  0.03  0.03
## [29,] -0.07 -0.09
## [30,] -0.03 -0.02
## [31,] -0.02  0.00
## [32,] -0.05 -0.04
## [33,]  0.01  0.00
## [34,]  0.10  0.05
## [35,]  0.09  0.06
## [36,]  0.04  0.01
## [37,] -0.02  0.00
## [38,] -0.02  0.01
## [39,] -0.03  0.01
## [40,]  0.02  0.00
## [41,] -0.01 -0.03
## [42,]  0.05  0.05
## [43,]  0.03  0.05
## [44,]  0.08  0.09
## [45,]  0.02 -0.01
## [46,]  0.02  0.01
## [47,]  0.05  0.06
## [48,]  0.05  0.06

##
## Series Initialization:
## ARMA Model:          arma
## Formula Mean:        ~ arma(1, 1)
## GARCH Model:         garch
## Formula Variance:    ~ garch(1, 1)
## ARMA Order:          1 1
## Max ARMA Order:      1
## GARCH Order:         1 1
## Max GARCH Order:     1
## Maximum Order:       1
## Conditional Dist:    std
## h.start:             2
## llh.start:           1
## Length of Series:    263
## Recursion Init:      mci
## Series Scale:        0.01946846
##
## Parameter Initialization:
## Initial Parameters:   $params
## Limits of Transformations: $U, $V
## Which Parameters are Fixed? $includes

```

```

## Parameter Matrix:
##           U           V      params includes
##   mu      -2.34861316   2.348613  0.2356789      TRUE
##   ar1      -0.99999999   1.000000  0.2671342      TRUE
##   ma1      -0.99999999   1.000000 -0.1593542      TRUE
##   omega    0.00000100 100.000000  0.1000000      TRUE
##   alpha1   0.00000001  1.000000  0.1000000      TRUE
##   gamma1  -0.99999999   1.000000  0.1000000     FALSE
##   beta1    0.00000001  1.000000  0.8000000      TRUE
##   delta    0.00000000  2.000000  2.0000000     FALSE
##   skew     0.10000000 10.000000  1.0000000     FALSE
##   shape    1.00000000 10.000000  4.0000000      TRUE
## Index List of Parameters to be Optimized:
##   mu      ar1      ma1  omega alpha1  beta1  shape
##     1       2       3     4       5       7     10
## Persistence:                0.9
##
##
## --- START OF TRACE ---
## Selected Algorithm: nlminb
##
## R coded nlminb Solver:
##
## 0:      318.92779: 0.235679 0.267134 -0.159354 0.100000 0.100000 0.800000 4.00000
## 1:      316.24165: 0.235487 0.261680 -0.164141 0.0793790 0.102765 0.786031 3.99969
## 2:      311.93976: 0.234207 0.224770 -0.196843 0.0318460 0.180645 0.775240 4.00019
## 3:      311.31116: 0.234100 0.223879 -0.197461 0.0444300 0.186290 0.778436 4.00039
## 4:      311.00540: 0.233695 0.220697 -0.199683 0.0422294 0.191495 0.766112 4.00051
## 5:      310.47631: 0.232769 0.216587 -0.202078 0.0545758 0.209353 0.748867 4.00122
## 6:      309.78347: 0.231205 0.231715 -0.188726 0.0534408 0.243108 0.708422 4.00255
## 7:      309.37683: 0.227435 0.201052 -0.221802 0.0702474 0.253184 0.685371 4.00729
## 8:      308.92865: 0.227278 0.225476 -0.204135 0.0531716 0.296325 0.672511 4.00888
## 9:      308.74682: 0.227133 0.224480 -0.205064 0.0634777 0.300980 0.675608 4.00918
## 10:     308.65259: 0.226128 0.228389 -0.207602 0.0606438 0.306305 0.667849 4.01097
## 11:     308.52486: 0.223404 0.238011 -0.213859 0.0671293 0.319379 0.660453 4.01600
## 12:     307.14105: 0.100224 0.667985 -0.679901 0.116739 0.427228 0.496729 4.20744
## 13:     306.98656: 0.0823778 0.694881 -0.699170 0.124879 0.443312 0.490738 4.22526
## 14:     306.98324: 0.0853738 0.713794 -0.693531 0.111750 0.449129 0.492489 4.22508
## 15:     306.82184: 0.0843063 0.703796 -0.696924 0.115886 0.452813 0.497740 4.22521
## 16:     306.79447: 0.0839671 0.698622 -0.697766 0.105798 0.459445 0.501612 4.22498
## 17:     306.75451: 0.0857456 0.697351 -0.690720 0.108451 0.463746 0.508865 4.22306
## 18:     306.73109: 0.0874206 0.690437 -0.684411 0.102366 0.465374 0.510907 4.22022
## 19:     306.71363: 0.0911272 0.677290 -0.670389 0.100721 0.469932 0.519440 4.21339
## 20:     306.70336: 0.0937944 0.666082 -0.656376 0.0951299 0.475457 0.519429 4.19761
## 21:     306.69749: 0.0968148 0.657579 -0.650649 0.0965015 0.474050 0.521267 4.17562
## 22:     306.69693: 0.0967421 0.657541 -0.650409 0.0970816 0.474498 0.521531 4.17554
## 23:     306.69658: 0.0966584 0.657559 -0.650123 0.0965984 0.474906 0.521166 4.17539
## 24:     306.69604: 0.0965481 0.657787 -0.650203 0.0971115 0.475590 0.521210 4.17395

```

```

## 25:      306.68622: 0.0922252 0.671529 -0.662651 0.100397 0.486344 0.513964 4.08841
## 26:      306.68314: 0.0896375 0.681238 -0.674681 0.0965532 0.493747 0.522209 4.00078
## 27:      306.68167: 0.0965816 0.658282 -0.651956 0.100682 0.503426 0.515182 3.92268
## 28:      306.67756: 0.0942917 0.665898 -0.658986 0.100473 0.500610 0.513775 3.99766
## 29:      306.67745: 0.0937320 0.667814 -0.661203 0.100479 0.501745 0.513575 3.98769
## 30:      306.67745: 0.0938664 0.667316 -0.660660 0.100468 0.502208 0.513445 3.98729
## 31:      306.67745: 0.0938807 0.667281 -0.660665 0.100519 0.502244 0.513353 3.98745
## 32:      306.67745: 0.0938710 0.667312 -0.660692 0.100515 0.502258 0.513351 3.98747
##
## Final Estimate of the Negative LLH:
## LLH: -729.2689      norm LLH: -2.772886
##      mu      ar1      ma1      omega      alpha1
## 1.827525e-03 6.673116e-01 -6.606922e-01 3.809735e-05 5.022584e-01
##      beta1      shape
## 5.133512e-01 3.987471e+00
##
## R-optimhess Difference Approximated Hessian Matrix:
##      mu      ar1      ma1      omega
## mu -1.761476e+07 -8.567897e+04 11786.095668 1.830752e+07
## ar1 -8.567897e+04 -8.492358e+02 -357.887411 4.631606e+04
## ma1 1.178610e+04 -3.578874e+02 -433.746109 -6.250039e+04
## omega 1.830752e+07 4.631606e+04 -62500.385511 -9.942927e+09
## alpha1 -2.620610e+03 -3.258860e+00 14.715053 -7.246673e+05
## beta1 -2.605049e+03 1.226886e+01 29.190800 -1.982172e+06
## shape -4.084037e+02 2.496816e-01 3.612938 -1.083811e+05
##      alpha1      beta1      shape
## mu -2620.61045 -2.605049e+03 -4.084037e+02
## ar1 -3.25886 1.226886e+01 2.496816e-01
## ma1 14.71505 2.919080e+01 3.612938e+00
## omega -724667.28438 -1.982172e+06 -1.083811e+05
## alpha1 -128.58661 -2.240047e+02 -1.447423e+01
## beta1 -224.00473 -5.288357e+02 -2.923380e+01
## shape -14.47423 -2.923380e+01 -2.707330e+00
## attr("time")
## Time difference of 0.04267621 secs
##
## --- END OF TRACE ---
##
##
## Time to Estimate Parameters:
## Time difference of 0.133553 secs
##
## Title:
## GARCH Modelling
##
## Call:
## garchFit(formula = ~arma(1, 1) + garch(1, 1), data = indexr,
## cond.dist = "std")

```

```

##
## Mean and Variance Equation:
## data ~ arma(1, 1) + garch(1, 1)
## <environment: 0x7f9f844f98c8>
## [data = indexr]
##
## Conditional Distribution:
## std
##
## Coefficient(s):
##      mu      ar1      ma1      omega      alpha1
## 1.8275e-03 6.6731e-01 -6.6069e-01 3.8097e-05 5.0226e-01
##      beta1      shape
## 5.1335e-01 3.9875e+00
##
## Std. Errors:
## based on Hessian
##
## Error Analysis:
##      Estimate Std. Error t value Pr(>|t|)
## mu      1.828e-03 1.090e-03 1.677 0.093484 .
## ar1      6.673e-01 1.932e-01 3.454 0.000552 ***
## ma1     -6.607e-01 1.958e-01 -3.374 0.000742 ***
## omega    3.810e-05 2.243e-05 1.698 0.089478 .
## alpha1   5.023e-01 2.023e-01 2.483 0.013020 *
## beta1    5.134e-01 1.435e-01 3.578 0.000347 ***
## shape    3.987e+00 1.050e+00 3.798 0.000146 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
## 729.2689      normalized: 2.772886
##
## Description:
## Tue Dec 10 23:28:55 2019 by user:
##
##
## Standardised Residuals Tests:
##
##      Statistic p-Value
## Jarque-Bera Test R Chi^2 101.6502 0
## Shapiro-Wilk Test R W 0.9528717 1.622902e-07
## Ljung-Box Test R Q(10) 12.90013 0.2293095
## Ljung-Box Test R Q(15) 15.7065 0.4018216
## Ljung-Box Test R Q(20) 18.97753 0.5232867
## Ljung-Box Test R^2 Q(10) 5.544401 0.8519795
## Ljung-Box Test R^2 Q(15) 9.411908 0.8550146
## Ljung-Box Test R^2 Q(20) 15.6798 0.7362799
## LM Arch Test R TR^2 5.28755 0.9476785

```

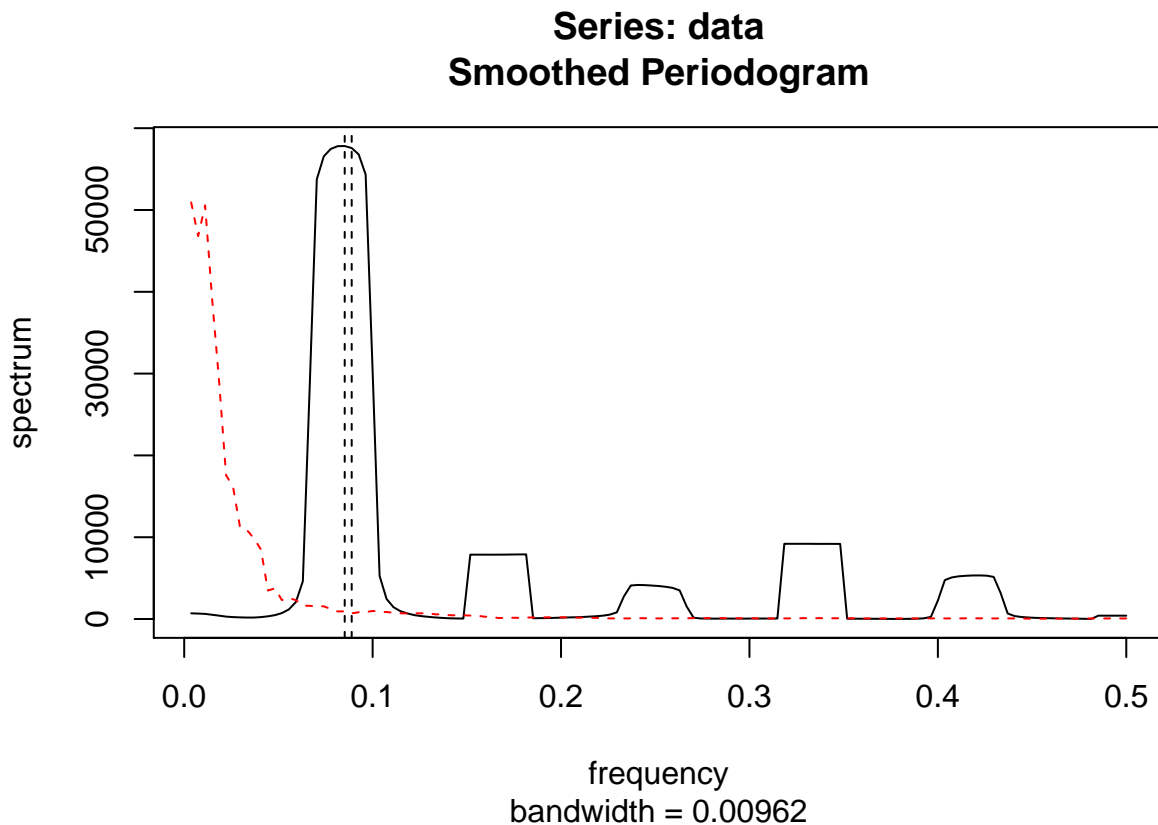


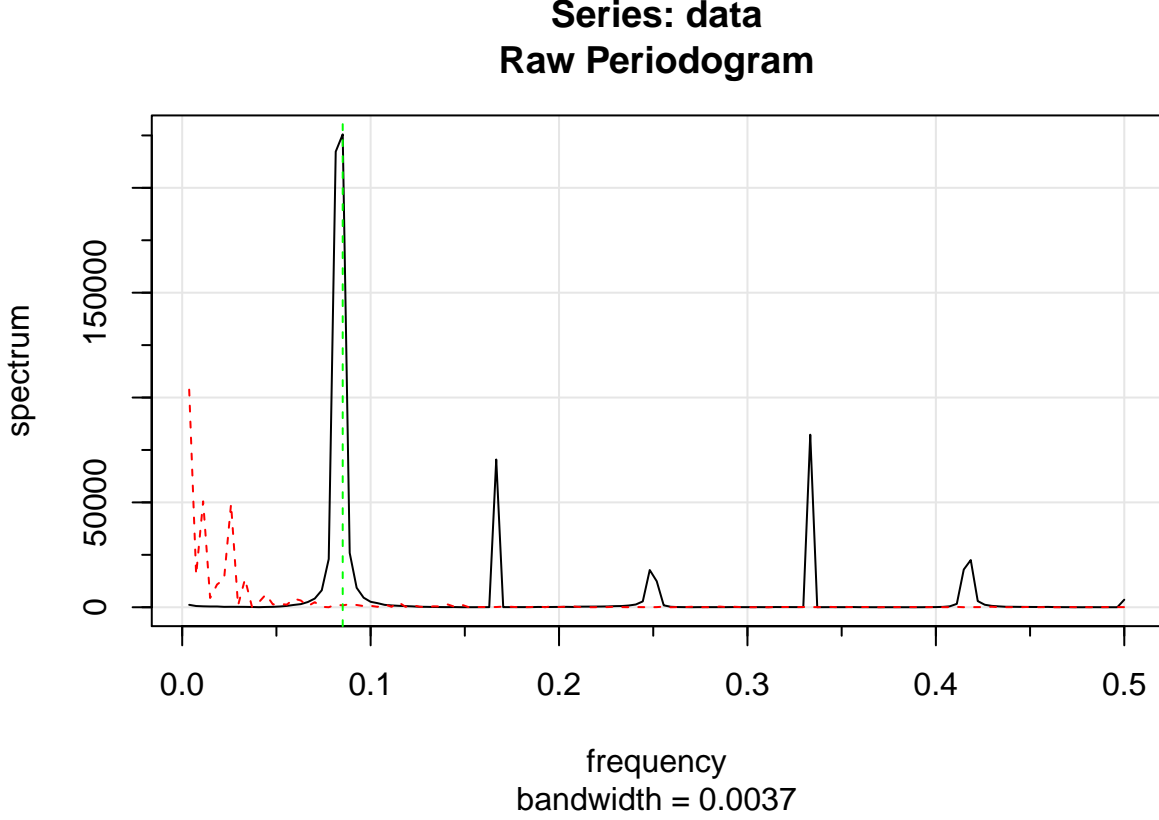
```
##
## Information Criterion Statistics:
##      AIC      BIC      SIC      HQIC
## -5.492539 -5.397463 -5.493908 -5.454330
```

9 Spectral Analysis

An important factor many investors would consider before investing in an asset is on the cyclical behavior of the asset. Cyclical assets are known to follow systematic changes according to the overall economy. These type of assets are ones where investors would avoid during a recession and invest in during economic strength. Therefore, before we could consider whether Credit Suisse Liquid Alternative Beta Index (LAB) as a potential investment, we must analyze its cyclical behavior by performing a spectral analysis.

First, we will find the predominant frequency by plotting a periodogram and compute the 95% Confidence Intervals for our identified periods.





From raw periodogram of LAB above, the green dashed line indicates the first dominant period which is $\frac{1}{0.08518519} \approx 11.74$. Therefore we could conclude that the major peak occurs approximately at a 12 time periods which means one year for our data and our 95% confidence interval for the spectrum $f_s(1/11)$ is $[3731.079, 14522.070]$. Given that the dominant cyclical behavior occurs yearly, we could conclude that the Liquid Alternative Beta Index is cyclical to an extent depending on which scale one is comparing to. For instance, long-term investors may not be as inclined to invest in this index since it is susceptible to annual economic changes.

10 Concluding Remarks

Price analysis and prediction are very important in finance area. For this project, we were interested in exploring and forecasting the Long Alternative Beta Index values from Credit Suisse. By the help of time series analysis, we conduct time domain analysis and frequency analysis. Using SARIMA, we identified the model parameters, select and estimated the model, and predicted future price with the estimated model. Our analyses exhibits a good result. We also conducted an initial GARCH model analysis and a spectral analysis to add another dimension to the time series analysis. We identified the dominant frequency in the time series, which is quite important for financial analysis. The analysis results of the index show that time series is powerful especially for financial data analysis. Our findings in this project may provide some insights for hedge fund price prediction.

11 Contribution Statement

For this project, Emily and Vanessa worked on the compiling and formatting of the project, introduction, data description, and spectral analysis. Xingyu and Enbo both worked on the abstract, time domain analyses, other analyses, and conclusion. Additonally, all members of the group proofread the final report, cross-checked, and added additional analyses to each other work.

-signed by Emily Lu, Enbo Zhou, Vanessa Ho, and Xingyu Chen

Appendix: Additional R Code

```
# data = read.csv("C:/Users/11351/Desktop/index.csv",header=T)
# head(data,3)
#
# # Data Exploratory Analysis
#
# index1=ts(data[,2],start=1998,f=12)
# index=ts(data[,2])
# ts.plot(index1,main = "Raw data",ylab="Liquid Alternative Beta Index")
#
# acf(index,lag.max=100,na.action = na.pass,main="")
# pacf(index,lag.max=100,na.action=na.pass,main="")
# title(main="PACF for raw data", line=-1,outer=TRUE)
#
# # Seasonal Plot and Decomposition plot
# library(forecast)
# seasonplot(index1,12,col=rainbow(3),year.labels=FALSE,main="Seasonal Plot")
# decom <- decompose(ts(data[,2],start=1998,f=12),type=c("additive"))
# plot(decom, xlab="Time in years")
#
# nlength=length(index)-12
# Transindex=index[1:nlength]
# Transindex1=index1[1:nlength]
# acf(Transindex,main="ACF for Training data",na.action = na.pass)
# pacf(Transindex,na.action=na.pass,main="")
# title(main="PACF for Training data", line=-1,outer=TRUE)
#
# # Data Transformation
# # First Step:Stabilize Variance
#
# library(MASS)
# length(Transindex)
# bxTransform <- boxcox(Transindex~as.numeric(1:length(Transindex)))
# lambda = bxTransform$x[which(bxTransform$y == max(bxTransform$y))]
# lambda
#
# par(mfrow=c(1,2))
# trans_index=(Transindex^lambda-1)/lambda
# trans_index1=(Transindex1^lambda-1)/lambda
# ts.plot(trans_index,ylab="transformed data")
# acf(trans_index,lag.max=100,na.action = na.pass,main='')
#
# pacf(trans_index,lag.max=100,na.action=na.pass)
# title(main="PACF for Transformed data", line=-1,outer=TRUE)
#
```

```

# # remove Seasonality
# diffindex = diff(trans_index, 12)
# ts.plot(diffindex,main = "De-seasonalized Time Series",ylab=expression(nabla^{12}~W[t]))
# abline(lm(diffindex~as.numeric (1:length(diffindex))))
#
# acf(diffindex,lag.max=100,main="ACF for De-seasonalized data",na.action = na.pass)
# pacf(diffindex,lag.max=100,na.action=na.pass,main="")
# title(main="PACF for De-seasonalized data", line=-1,outer=TRUE)
#
# # remove Seasonality and Trend
# diffindex_s<-diff(diffindex, lag=1)
# ts.plot(diffindex_s,main = "De-seasonalized/De-trended Time Series",
#         ylab = expression(nabla~nabla^{12}~W[t]))
# abline(lm(diffindex_s~as.numeric(1:length(diffindex_s))))
# acf(diffindex_s,lag.max=100,main="ACF for De-seasonalized/De-trended data",na.action = na.pa
# pacf(diffindex_s,lag.max=100,na.action=na.pass,main="")
# title(main="PACF for De-seasonalized/De-trended data", line=-1,outer=TRUE)
#
# par(mfrow=c(1,2))
# diffindex = diff(trans_index, 12)
# ts.plot(diffindex,main = "De-seasonalized Time Series",ylab=expression(nabla^{12}~W[t]))
# abline(lm(diffindex~as.numeric (1:length(diffindex))))
# diffindex_s<-diff(diffindex, lag=1)
# ts.plot(diffindex_s,main = "De-seasonalized/De-trended Time Series",
#         ylab = expression(nabla~nabla^{12}~W[t]))
# abline(lm(diffindex_s~as.numeric(1:length(diffindex_s))))
#
# diffindex13<-diff(diffindex_s, lag=1)
# tsdisplay(diffindex13,lag.max=100)
# diffindex14<-diff(diffindex13, lag=1)
# tsdisplay(diffindex14,lag.max=100)
#
# # Check for overdifferencing or not
# var(diffindex_s)
# var(diffindex13)
# var(diffindex14)
#
# # Check unit root using ADF test
# library(tseries)
# adf.test(diffindex_s)
#
#
# # Model Identification
#
# # Find P, Q, P=3, Q=1
# par(mfrow=c(1,2))
# acf(diffindex_s,lag.max=60,na.action = na.pass,main="")

```

```

# pacf(diffindex_s, lag.max=60, na.action=na.pass, main="")
#
# # find p, q by looking at ACF and PACF plots of the zoomed plot p=0, q=0
# par(mfrow=c(1,2))
# acf(diffindex_s, lag.max=12, main="", na.action = na.pass)
# pacf(diffindex_s, lag.max=12, na.action=na.pass, main="")
#
# library(qpcR)
# AICc1<-matrix(NA, nr = 4, nc =2)
# for (p in 0:3){
#   for (q in 0:1){
#     AICc1[p+1,q+1]<-AICc(arima(trans_index, order=c(0,1,0),
#                                     seasonal= list(order=c(p,1,q), period=12), method = "ML"))
#   }
# }
#
# AICc1<-as.data.frame(AICc1)
# rownames(AICc1)<-c("P=0", "P=1", "P=2", "P=3")
# colnames(AICc1)<-c("Q=0", "Q=1")
# AICc1
# (AICc1==min(AICc1))
#
# BICc1<-matrix(NA, nr = 4, nc =2)
# for (p in 0:3){
#   for (q in 0:1){
#     BICc1[p+1,q+1]<-BIC(arima(trans_index, order=c(0,1,0),
#                                     seasonal= list(order=c(p,1,q), period=12), method = "ML"))
#   }
# }
#
# BICc1<-as.data.frame(AICc1)
# rownames(BICc1)<-c("P=0", "P=1", "P=2", "P=3")
# colnames(BICc1)<-c("Q=0", "Q=1")
# BICc1
# (BICc1==min(BICc1))
#
# # Fit and Estimation based on MLE method
# # MODEL 4 : p =0 and q = 0, P=3, Q=1
# fit4 <- arima(trans_index, order=c(0,1,0),
#               seasonal= list(order=c(3,1,1), period=12), method="ML" )
# summary(fit4)
#
# # Fit and Estimation based on MLE method
# # MODEL 2 : p =0 and q = 0, P=1, Q=1
# fit2 <- arima(trans_index, order=c(0,1,0),
#               seasonal= list(order=c(1,1,1), period=12), method="ML" )
# summary(fit2)

```

```

#
# # Fit and Estimation based on MLE method
# # MODEL 3 :  $p = 0$  and  $q = 0$ ,  $P=2$ ,  $Q=1$ 
# fit3 <- arima(trans_index,order=c(0,1,0),
#             seasonal= list(order=c(2,1,1),period=12),method="ML" )
# summary(fit3)
#
# # Fit and Estimation based on MLE method
# # MODEL 1 :  $p = 0$  and  $q = 0$ ,  $P=0$ ,  $Q=1$ 
# fit1 <- arima(trans_index,order=c(0,1,0),
#             seasonal= list(order=c(0,1,1),period=12),method="ML" )
# summary(fit1)
#
# # Check the normality of the residuals for both models
# par(mfrow=c(2,2))
# qqnorm(residuals(fit1),main="Normal QQ plot for fit1")
# qqline(residuals(fit1),col="red")
# qqnorm(residuals(fit2),main="Normal QQ plot for fit2")
# qqline(residuals(fit2),col="red")
# qqnorm(residuals(fit3),main="Normal QQ plot for fit3")
# qqline(residuals(fit3),col="red")
# qqnorm(residuals(fit4),main="Normal QQ plot for fit4")
# qqline(residuals(fit4),col="red")
#
# hist(residuals(fit1),col="blue")
# hist(residuals(fit2),col="blue")
# hist(residuals(fit3),col="blue")
# hist(residuals(fit4),col="blue")
#
# shapiro.test(residuals(fit1))
# shapiro.test(residuals(fit2))
# shapiro.test(residuals(fit3))
# shapiro.test(residuals(fit4))
#
# # Test for independence of residuals
# Box.test(residuals(fit1), type="Ljung")
# Box.test(residuals(fit2), type="Ljung")
# Box.test(residuals(fit3), type="Ljung")
# Box.test(residuals(fit4), type="Ljung")
# Box.test(residuals(fit1), type="Box-Pierce")
# Box.test(residuals(fit2), type="Box-Pierce")
# Box.test(residuals(fit3), type="Box-Pierce")
# Box.test(residuals(fit4), type="Box-Pierce")
#
# # Heteroscedasticity Checking
# par(mfrow=c(4,2))
# library(astsa)

```

```

# acf2(residuals(fit1)^2,main = "ACF and PACF of fit1^2")
# acf2(residuals(fit2)^2,main = "ACF and PACF of fit2^2")
# acf2(residuals(fit3)^2,main = "ACF and PACF of fit3^2")
# acf2(residuals(fit4)^2,main = "ACF and PACF of fit4^2")
#
# pred.try=predict(fit1,n.ahead = 12)
#
# U.try<-(lambda*(pred.try$pred+1.96*pred.try$se)+1)^(1/lambda)
# L.try<-(lambda*(pred.try$pred-1.96*pred.try$se)+1)^(1/lambda)
# plot(index[1:nlength],xlim=c(0,300),type="l",ylab="index values",
#       main = expression("Forecasting based on data for the next twelve points"),ylim=c(400,1500))
# newpred=(lambda*(pred.try$pred)+1)^(1/lambda)
# lines((length(index[1:nlength])+1):(length(index[1:nlength])+12),newpred, col = "green")
# lines((length(index[1:nlength])+1):(length(index[1:nlength])+12),U.try,col = "blue",lty = 2)
# lines((length(index[1:nlength])+1):(length(index[1:nlength])+12),L.try,col = "blue",lty = 2)
# points((length(index[1:nlength])+1):(length(index[1:nlength])+12),newpred, col = "red", pch = 1)
# legend("topleft", c("Prediction Bounds", "Prediction"),
#       col = c("blue", "green"), lty= c(2,1), cex = 0.7)
#
# ts.plot(index, xlim = c(length(index)-12,length(index)),
#       main = expression("Comparison between Obsevered Values and Forcasted Values"),
#       ylab = 'per month',ylim=c(1200,1500))
# newpred=(lambda*(pred.try$pred)+1)^(1/lambda)
# lines((length(index[1:nlength])+1):(length(index[1:nlength])+12),newpred, col = "green")
# lines((length(index[1:nlength])+1):(length(index[1:nlength])+12),U.try,col = "blue",lty = "d")
# lines((length(index[1:nlength])+1):(length(index[1:nlength])+12),L.try,col = "blue",lty = "d")
# legend("topright", c("Prediction Bounds", "Prediction", "True Values"),
#       col = c("blue", "green", "black"), lty= c(2, 1, 1), cex = 0.7)

# library("astsa")
# library(fGarch)
# index1=ts(data[,2],start=1998,f=12)
# index=ts(data[,2])
# par(mfrow = c(3,1))
# plot(index1)
# plot(diff(log(index1)))
#
# indexr = diff(log(index1))
# acf2(indexr) # exhibits some autocorrelation (not shown)
#
# # GARCH fit
# summary(indexr.g <- garchFit(~arma(1,1)+garch(1,1), data=indexr, cond.dist='std'))

# Spectral Analysis
# par(mfrow = c(2,1))
# sp <- spec.pgram(data, taper = 0, kernel('daniell', 4), log = 'no')
# head(n=2, order(decreasing = TRUE, sp$spec)) # predominant frequencies

```



```

# abline(v = sp$freq[c(24,23)], lty=2)
# sortedSpecs <- sort(sp$spec, decreasing = TRUE)[c(1,2,4)]
# sortedSpecs <- sort(sp$spec, decreasing = TRUE)
# CI <- function(peakSpec) {
#   u <- qchisq(0.025, sp$df)
#   l <- qchisq(0.975, sp$df)
#   c((2*peakSpec)/l, (2*peakSpec)/u)
# }
#
# CI(sortedSpecs[1])
# CI(sortedSpecs[2])
# CI(sortedSpecs[3])
# abline(v=sortedSpecs[1], lty=2, col = 'green')
# abline(v=CI(sortedSpecs[2]), lty=2, col='blue')
#
# p1 <- sp$freq[sp$spec == sortedSpecs[1]]
# p2 <- sp$freq[sp$spec == sortedSpecs[2]]
# p3 <- sp$freq[sp$spec == sortedSpecs[3]]
#
# # cat('Cycles are occuring at', 1/p1, 1/p2, 1/p3, 'years.')
#
# data.per = muspec(data, log="no")
# m = data.per$freq[which.max(data.per$spec)] # Major peak
# abline(v=m, lty=2, col = 'green')

```

Time Series Analysis and Its Applications with R Examples by R. H. Shumway and D. S. Stoffer
Credit Suisse Liquid Alternative Beta Index Data from <https://lab.credit-suisse.com/#/en/login>