Assignment - 1

MATH1318 Time Series Analysis

Mar 26, 2023 • RMIT University

Tran Ngoc Anh Thu (s3879312)

Table of Contents

1. Executive Summary

This report aimed to examine the return on the investment portfolio of a share market trader and find the best fitting model among linear, quadratic, cosine, cyclical, or seasonal trend models by implementing the model-building strategy. We used R programming language to analyze the dataset and perform the required task to achieve the purposes. We followed the model-building strategy to evaluate and select the best-fitting model. We validate a deterministic trend model by conducting a residual analysis, including a histogram plot, Shapiro-Wilk test, QQ plot, and ACF plot. The results indicated that the residuals were approximately not normally distributed with significant autocorrelation, indicating further investigation. Our dataset analysis revealed that the quadratic trend model provided the best fit among the models considered. Using the quadratic trend model, we predicted the return on the investment portfolio for the next 15 trading days. These findings can inform investment decisions and contribute to developing effective investment strategies. Additionally, explore other models and factors that may impact the return-on-investment portfolio to enhance the accuracy of future predictions.

2. Introduction

Investors in the share market always seek ways to improve their investment returns by analyzing market trends and identifying patterns. One of the key tools used in this analysis is the application of statistical models to the available data. This report aims to explore the return on the investment portfolio of a share market trader and identify the best-fitting model among linear, quadratic, cosine, cyclical, or seasonal trend models by implementing the model-building strategy in Module 1. The dataset comprises 127 observations out of possible 252 trading days in a year, collected on consecutive trading days over the same year, and represents the return (in AUD100,000) of a share market trader's investment portfolio. The report will present the analysis and evaluation of the different models and provide predictions for the next 15 trading days based on the best-fitting model. The results of this analysis will provide valuable insights to the share market trader for making informed investment decisions.

3. Data Exploration and Visualization

We loaded the dataset into RStudio as a data frame object and conducted a descriptive analysis of a share market trader's return on the investment portfolio. Then we convert it to a time series object. The dataset consisted of 127 observations representing the return (AUD100,000) of the trader's investment portfolio. The observations were collected on consecutive trading days within a year, excluding weekend days. We then generated a time series plot of the return-on-investment portfolio to visualize the patterns and trends in the data. The plot revealed that the return-on-investment portfolio fluctuated around a relatively stable mean value, with occasional spikes and dips.

3.1. Column names

The first column of the dataset does not have a column name because it represents the sequential order of the observations rather than a variable of interest. To make it easier to refer to the observations by their time period, we can assume that each observation corresponds to a trading day within a year and that the first observation corresponds to the first trading day of the year. We assigned a name to this column to “id”. In this case,

we can assume that the dataset comprises observations collected on consecutive trading days within a year and that the first observation corresponds to January 1st. To assign a name to this column, we can use a sequence of integers corresponding to each observation's month number. Since the dataset comprises 127 observations, we can assume that the first 12 observations correspond to January, the next 12 observations correspond to February, and so on. Therefore, we can assign the name "id" to this column and use the formula t = 1, 13, 25, … correspond to January, t = 2, 14, 26, … correspond to February, t = 3, 15, 27, … correspond to March, and so on to indicate the month number of each observation. By doing this, we can more easily refer to the observations by their time and use this information to analyze the patterns and trends in the time series data.

3.2. Descriptive Analysis

The descriptive statistics used in our analysis included measures of central tendency, such as the mean and median, and measures of variability, such as the range. These statistics provided a numerical summary of the data's main features and helped inform the modelling process.

Table 1. Summary Statistics of the Dataset, which displays the portfolio's returns over time.

Statistic value	id	x
Min.	1.0	-44.62
1st Qu.	32.5	21.32
Median	64.0	71.97
Mean	64.0	71.35
3rd Qu.	95.5	135.90
Max.	127.0	156.32

Note that these summary statistics were computed based on the assumption that the id variable represents the sequential order of the observations, and the x variable represents the return of a share market trader's investment portfolio.
The dataset contains positive and negative returns, with some extreme values at the lower end of the distribution. The returns appear volatile, with a wide range of values across the 127 trading days. These features suggest that modeling the returns using time series analysis techniques may be useful in identifying patterns and making predictions about future returns.

3.3. Find the Frequency and Convert data to Time Series Object

In time series analysis, it is important to know the frequency of the data because it affects the model choice and the forecasts' accuracy. A model for daily data may not be appropriate for weekly or monthly data.
The ACF plot and the frequency calculation can provide information about a time series' seasonality or periodicity. The acf() function creates a plot of the autocorrelation function of the time series data, which shows the correlation between the observations at different lags. The

resulting plot can help to identify the presence of any seasonality or periodicity in the data.

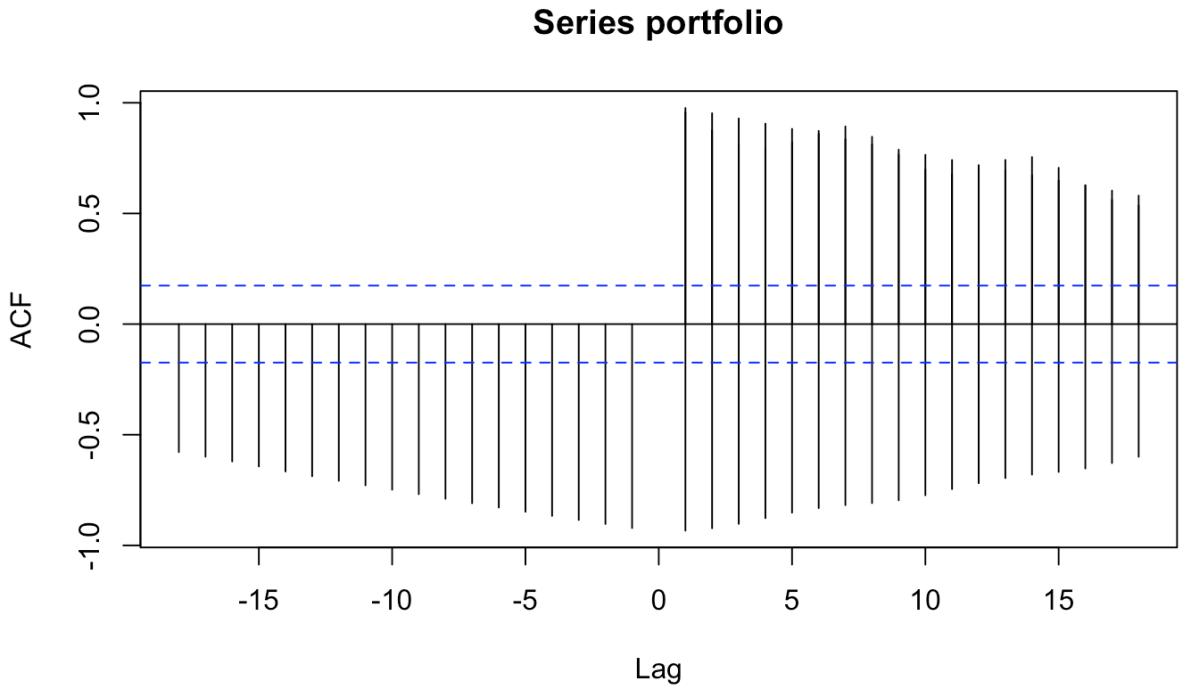

Figure 1. ACF plot shows the frequency in the original data frame

Since there are 127 observations and the data covers one year, the calculated frequency for the original data frame is estimated to be 127/252, which is approximately 0.5039683. This indicates that there are, on average, 1 observation per trading day in a year for this interpretation.

frequency = 127/252 ≈ 1

If the dataset comprises 127 observations out of a possible 252 trading days in a year, then the data is not daily but covers approximately half of a trading year. Therefore, we need to set the frequency of the time series object to match this interval. To do this, we can calculate the number of trading days between the first and last observation in the dataset using the diff() function with units = "days" to calculate the difference between the first and last observation in the id column of the dataset in days. We use as.numeric() to extract the numeric value from the difftime object and then use this value in the division operation to calculate the frequency. When creating the time series object, we then pass this frequency value to the ts() function.
This will create a time series object with a frequency that matches the interval of the data, which covers approximately half of a trading year with 127 observations. The frequency for ts as frequency refers to the number of observations per unit of time.
This gives a frequency of approximately 1.98. However, since ts() only accepts integer values as frequencies, we could round this value to the nearest integer, giving a frequency of 2. Therefore, the correct frequency for the ts object would be 2, corresponding to the number of observations per unit of time (trading day in this case).

frequency = 252 / 127 ≈ 2

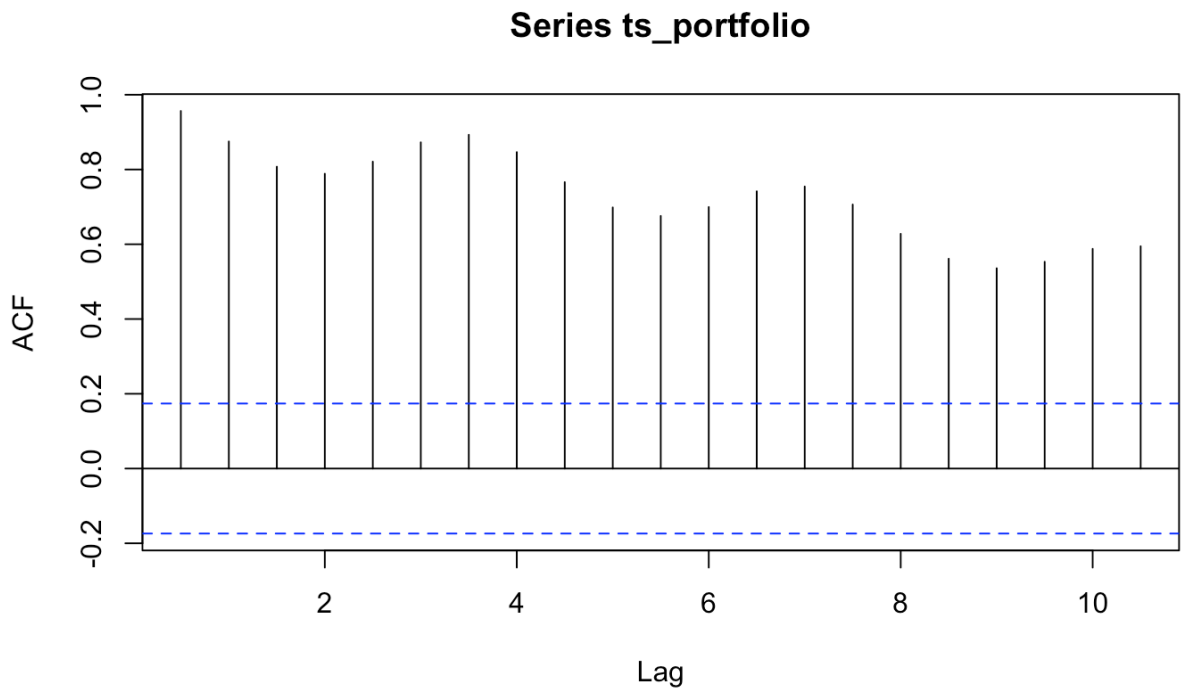

Figure 2. ACF plot shows the frequency of 2 in the converted time series object.

The frequency() function in R, by default, assumes that the time series are regularly spaced and computes the number of observations per unit of time. In this case, it assumed that the time series had observations for each trading day in a year, i.e., a frequency of 1.
However, in our case, the time series had only 127 observations out of a possible 252 trading days in a year. This means there were missing observations, and the time series needed to be regularly spaced. Therefore, it is more appropriate to calculate the frequency based on the number of observations available, which is 2 (252/127). A frequency of 2 will allow us to properly account for the missing observations and accurately model and forecast the time series.

3.4. Time series plot

We create a plot for a given time series object ts_portfolio. The plot displays a line graph with points at each data point of the daily returns of an investment portfolio over a certain period, with trading days as the x-axis and return values as the y-axis.

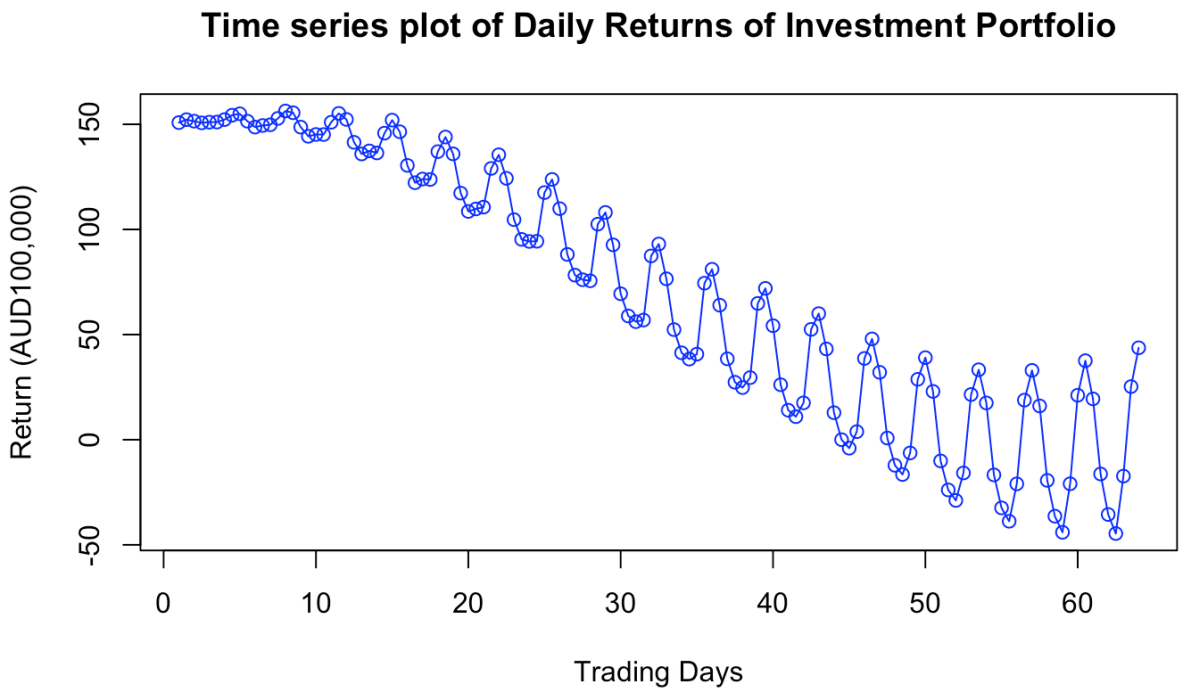

Figure 3. Time series plot of Daily Returns of Investment Portfolio

Trend - The time series plot shows a clear downward tendency, indicating a decreasing trend in the daily returns of the investment portfolio over time.
Seasonality - There are some repeating patterns in the data, which indicate seasonality. However, we can only conclusively determine the seasonality with more information about the underlying data-generating process.
Changing variance - The fluctuations in the data appear to be getting larger consecutively, indicating changing variance.
Behaviour - The plot shows up and down-movement, indicating average moving behaviour. There also seems to be some autocorrelation between successive points, indicating an autoregressive behaviour.
Changepoint - The time series plot appears to have a vague change point.
We must conclude which trend models are appropriate with more information about the underlying data-generating process.

3.5. Scatter plot

The scatter plot shows the relationship between the current year's investment return and the previous year's return. The x-axis represents the previous year's return, while the y-axis represents the current year's return. The correlation between the two variables can be determined by calculating the Pearson correlation coefficient, denoted as r below, which measures the strength and direction of the linear relationship between two variables.

$$r = cov(X,Y) / (sd(X) * sd(Y))$$

Where cov(X,Y) is the covariance between X and Y, sd(X) is the standard deviation of X, and sd(Y) is the standard deviation of Y.

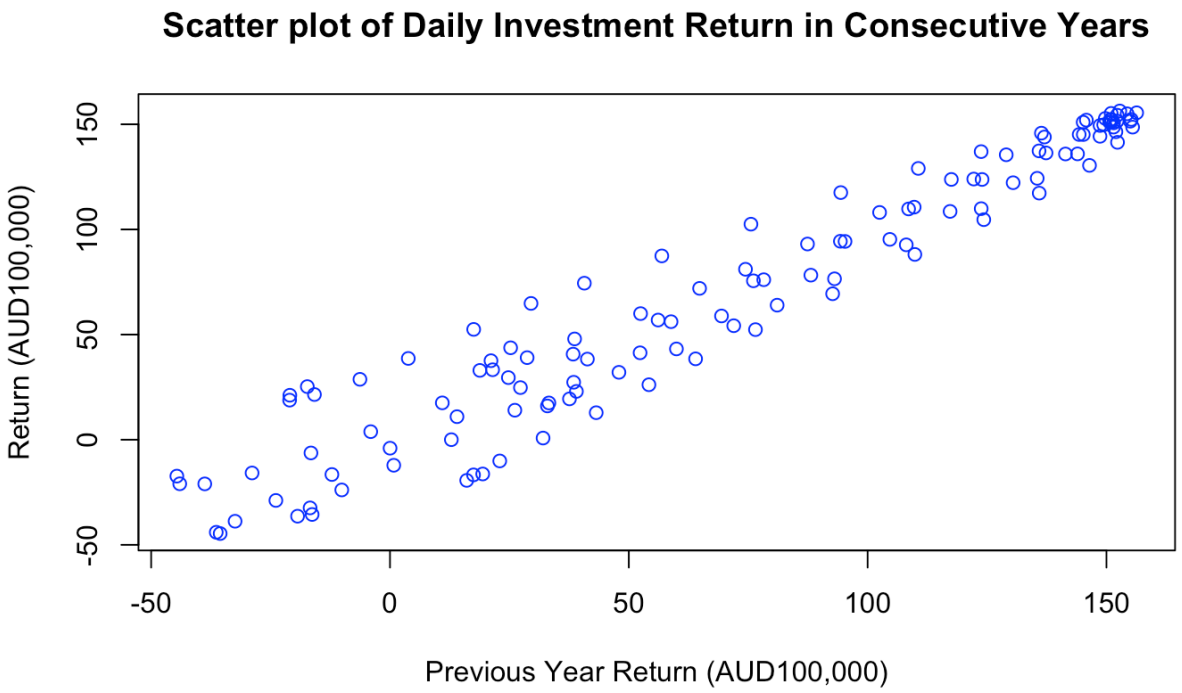

Figure 4. Scatter plot of Daily Investment Return in Consecutive Years

The blue points on the scatter plot form a roughly straight line sloping upwards from left to right, then the two variables have a positive correlation. The data correlation value of 0.9635593 indicates a strong positive correlation between the current year's investment return and the previous year's investment return. A value of 1 would indicate a perfect positive correlation, while a value of -1 would indicate a perfect negative correlation. In this case, a value of 0.9635593 suggests a strong relationship between the two variables, with a high degree of predictability in the current year's investment return based on the previous year's return.
The correlation coefficient measures the strength of the linear relationship between two variables. However, it does not provide information about the nature of the relationship, the direction of causality, or whether the relationship is significant. Therefore, more is needed to rely solely on the correlation coefficient to determine the model to predict. Instead, it is important to conduct further analysis and consider additional factors, such as the data distribution, the nature of the variables, and potential confounding factors, before selecting an appropriate prediction model.

4. Methodology

To determine the best-fitting model among the linear, quadratic, cosine, cyclical, or seasonal trend models, we implemented the following steps.

4.1. Deterministic Versus Stochastic Trends

One way is to use statistical tests like the Augmented Dickey-Fuller (ADF) test. The ADF test is a statistical test used to determine whether a time series is stationary. In this case, the test statistic (Dickey-Fuller) is -0.22725, and the lag order is 5, which suggests that the series may have a deterministic trend. However, the p-value obtained from the test is 0.99, which is greater than the

printed p-value (usually 0.05 or 0.01). This means we cannot reject the null hypothesis that the time series is non-stationary. Therefore, based on this result, we should not conclude that the time series has a deterministic trend.
Another way to determine if a time series has a deterministic trend is to plot the data and visually inspect if there is a consistent trend over time, which we did above in figure 1, clearly indicating the presence of a trend.
When a time series with a deterministic trend is modeled, the resulting model will be stationary. Therefore, our time series data exhibit a deterministic trend and is considered trend-stationary.

4.2. Decomposition

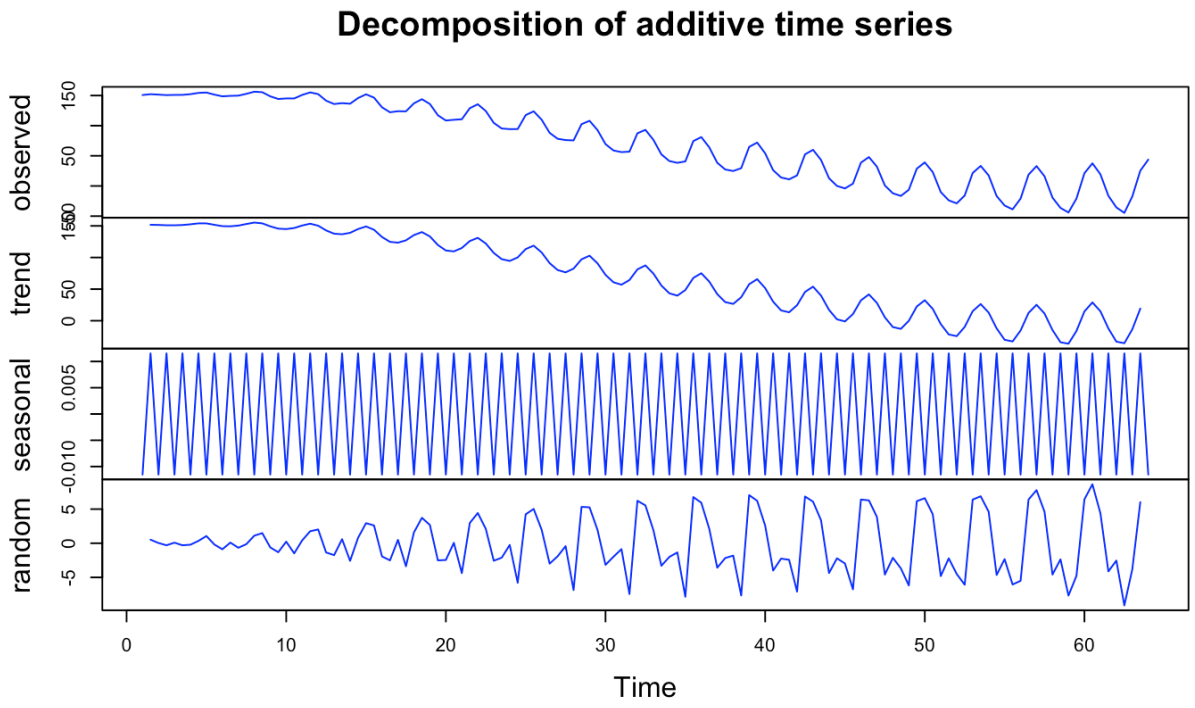

Figure 5. Decomposition graph shows the observed, trend, seasonal, and random components.

The plot allows us to visualize the individual components and how they contribute to the time series. By decomposing the time series, we can gain insights into the underlying patterns and trends in the data, which can help inform our choice of forecasting models and methods. For example, if we observe a clear seasonal pattern in the data, we may use a seasonal forecasting method to make accurate predictions. As visual inspection in figure 1 noted earlier, the time series plot of the dataset shows a general upward trend with fluctuations and seasonality. This suggests that a deterministic trend model is appropriate.

4.3. Model Specification

I follow the principle of parsimony to choose the regression model: linear, quadratic, cyclic, and seasonal models. Then use analysis of residuals to choose the best model. Since the trend component is dominant, we considered the following trend models:

Table 2. Trend Model Specification

Model type	Mathematical Formula

1. Linear	y = a + bt
2. Quadratic	y = a + bt + ct^2
3. Seasonal	y = a + b*cos(2pit/5)* + *c*sin(2*pi*t/5)
4. Cosine	a + b*cos(2pit/365)* + *c*sin(2*pi*t/365)
5. Cyclical	y = a + b*cos(2pit/365)* + *c*sin(4*pi*t/365)

4.4. Model Fitting

We fitted each trend model to the dataset and compared their goodness-of-fit measures and diagnostic checks.

4.4.1 Deterministic linear trend model

```
```{r}
# deterministic linear trend model
linear_model = lm(ts_profolio~time(ts_profolio)) # label the model as linear_model
summary(linear_model)
```
```

```
Call:
lm(formula = ts_profolio ~ time(ts_profolio))

Residuals:
    Min      1Q  Median      3Q     Max
-38.053 -17.959   2.056  15.095  72.799

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       174.9796     3.9006   44.86   <2e-16 ***
time(ts_profolio)  -3.1885     0.1045  -30.50   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 21.59 on 125 degrees of freedom
Multiple R-squared:  0.8816,    Adjusted R-squared:  0.8806
F-statistic: 930.3 on 1 and 125 DF,  p-value: < 2.2e-16
```

Figure 6. The summary of the deterministic linear trend model

The output relates to the deterministic linear trend model applied on a time series data named "ts_portfolio".
The first part of the output summarizes the data frame that shows the minimum, maximum, median, mean, and quartile values of the "id" and "x" variables.
The second part of the output shows the first 10 rows of the "id" and "x" variables of the time series data.
The third part of the output shows the correlation coefficient value of 0.9635593 between the time series and its trend component.

The fourth part of the output summarizes the linear regression model that estimates the trend component of the time series data using the time variable as the predictor. The coefficients of the intercept and time variables are estimated as 174.9796 and -3.1885, respectively. The p-value associated with the F-statistic is extremely small, indicating that the model is significant. The R-squared value of 0.8816 suggests that the model explains 88.16% of the variance in the time series data.

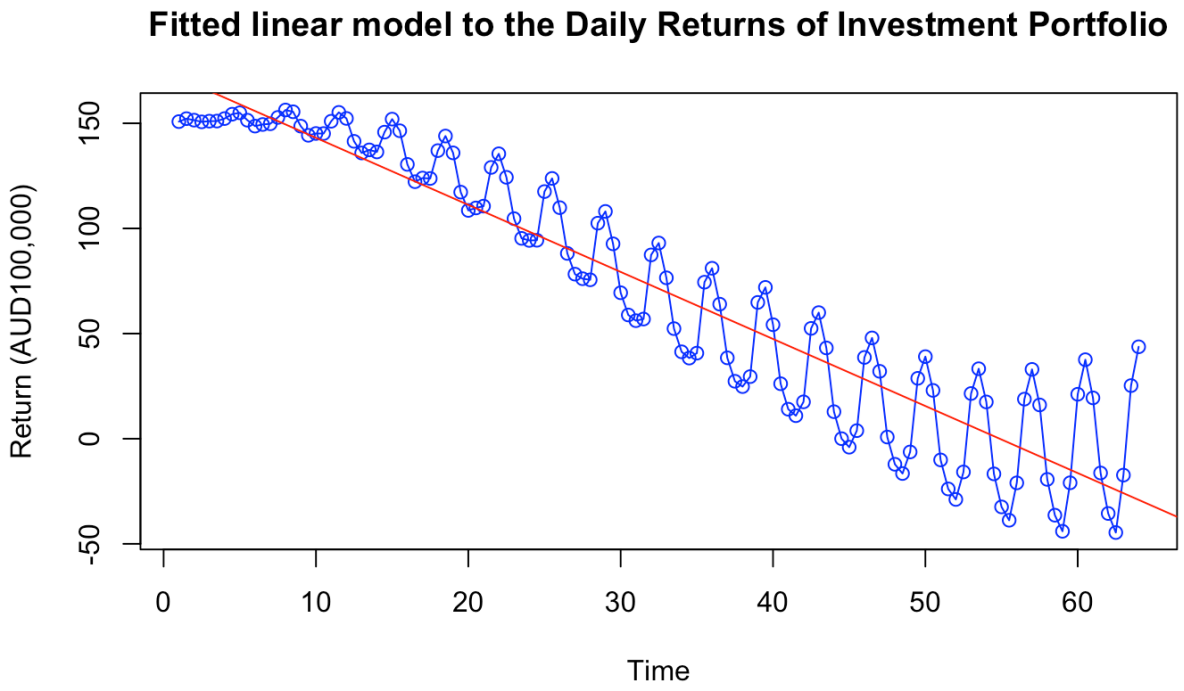

Figure 7. Fitted linear model to the Daily Returns of Investment Portfolio

The graph adds the fitted least squares as a red line from a previously created linear model to show the data trend according to the linear model.

4.4.2 Deterministic quadratic trend model

```r
# the summary of the deterministic quadratic trend model
t = time(ts_profolio)
t2 = t^2
quadratic_model = lm(ts_profolio~t+t2) # label the model as quadratic_model
summary(quadratic_model)
```

```
Call:
lm(formula = ts_profolio ~ t + t2)

Residuals:
    Min      1Q  Median      3Q     Max
-38.345 -18.387   0.941  15.584  68.478

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 179.722099   6.024034  29.834  < 2e-16 ***
t            -3.616501   0.427350  -8.463 6.19e-14 ***
t2            0.006585   0.006375   1.033    0.304
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 21.59 on 124 degrees of freedom
Multiple R-squared:  0.8826,    Adjusted R-squared:  0.8807
F-statistic: 465.9 on 2 and 124 DF,  p-value: < 2.2e-16
```

Figure 8. The summary of the deterministic quadratic trend model

The output relates to the deterministic linear trend model applied on a time series data named "ts_portfolio".
The first part of the output summarizes the data frame that shows the minimum, maximum, median, mean, and quartile values of the "id" and "x" variables.
The second part of the output shows the first 10 rows of the "id" and "x" variables of the time series data.
The third part of the output shows the correlation coefficient value of 0.9635593 between the time series and its trend component.
The fourth part of the output summarizes the linear regression model that estimates the trend component of the time series data using the time variable as the predictor. The coefficients of the intercept and time variables are estimated as 174.9796 and -3.1885, respectively. The p-value associated with the F-statistic is extremely small, indicating that the model is significant. The R-squared value of 0.8816 suggests that the model explains 88.16% of the variance in the time series data.

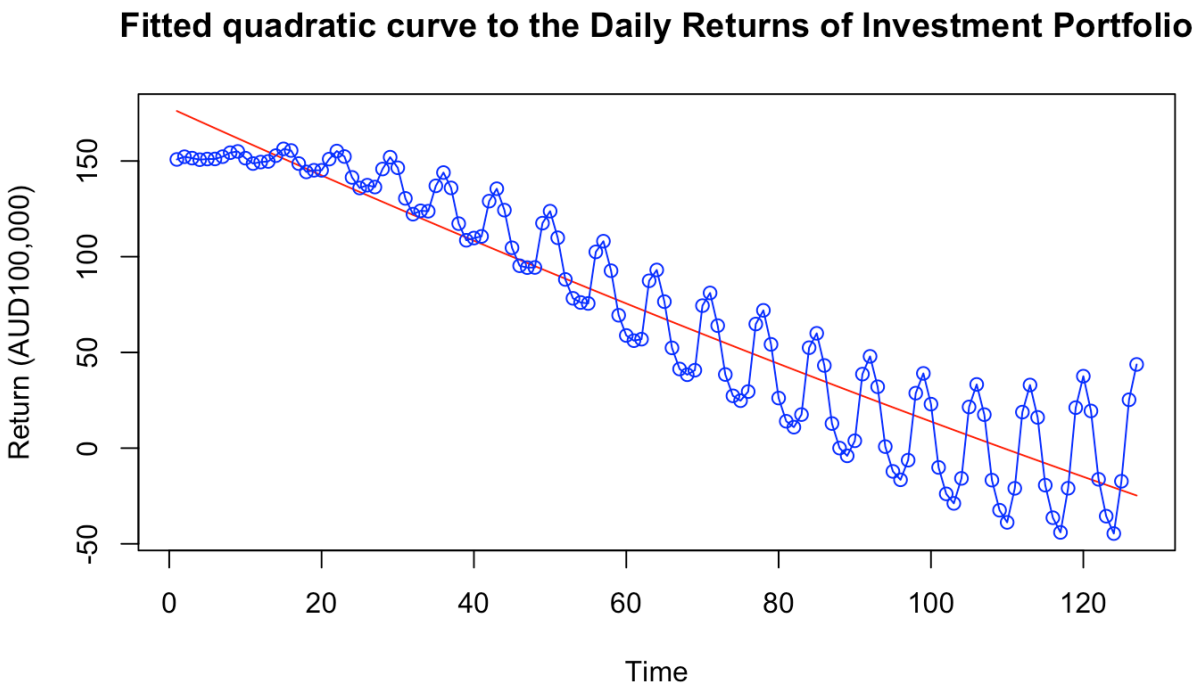

Figure 9. Fitted quadratic curve to the Daily Returns of Investment Portfolio

The graph adds the fitted least squares line from a previously created linear model to show the data trend according to the linear model.

4.4.3 Seasonal deterministic trend model

```
```{r seasonal deterministic trend model}
month.=season(ts_portfolio)
# Season() function creates a factor variable showing the months.
# period added to improve table display and this line sets up indicators
season_model=lm(ts_portfolio ~ month.-1) # add -1 to remove the intercept term
summary(season_model)
```
```

```
Call:
lm(formula = ts_portfolio ~ month. - 1)

Residuals:
     Min       1Q   Median       3Q      Max
-115.747  -50.261    0.844   64.327   84.747

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
month.Season-1   71.578      7.843   9.126 1.54e-15 ***
month.Season-2   71.125      7.905   8.997 3.14e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 62.74 on 125 degrees of freedom
Multiple R-squared:  0.5678,    Adjusted R-squared:  0.5609
F-statistic: 82.12 on 2 and 125 DF,  p-value: < 2.2e-16
```

Figure 10. The summary of the deterministic seasonal model without the intercept

The model is a deterministic seasonal model without the intercept, which means that the seasonal effects are the only predictors in the model. The coefficients represent the estimated mean response for each season, where the first season in the dataset is used as the reference level.
In this model, we see two seasons that are significant predictors of the portfolio returns with p-values less than 0.05, and these months are season 1 and season 2. Season 1 has a coefficient estimate of 71.578, which means that, on average, returns in season 1 are 71.578 higher than the reference level. Similarly, season 2 has a coefficient estimate of 71.125, which means that, on average, returns in season 2 are 71.125 higher than the reference level.
The R-squared value of 0.5678 indicates that the seasonal effects explain about 56.78% of the variability in the portfolio returns. The F-statistic of 82.12 with a p-value less than 2.2e-16 suggests that the model is significant and fits the data well.

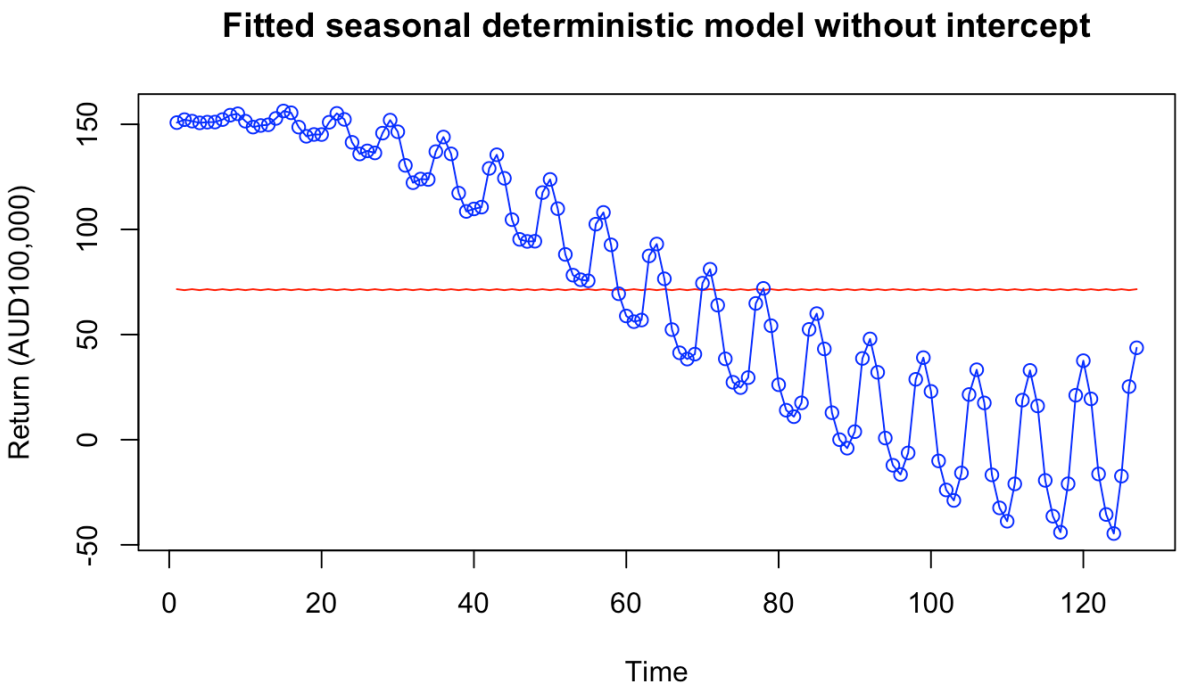

Figure 11. Fitted Seasonal curve to the Daily Returns of Investment Portfolio

The graph shows the fitted seasonal deterministic model and the original time series data. We set the limits for the y-axis, which are calculated based on the minimum and maximum values of the fitted model and the original time series data. The red-fitted model line is on top of the original blue data.

4.4.4 Cosine Trend Model

```
```{r cosine trend model}
cosine_model <- lm(ts_portfolio ~ cos(2*pi*time(ts_portfolio)) + sin(2*pi*time(ts_portfolio)))
summary(cosine_model)
```
```

```
Call:
lm(formula = ts_portfolio ~ cos(2 * pi * time(ts_portfolio)) +
    sin(2 * pi * time(ts_portfolio)))

Residuals:
     Min       1Q   Median       3Q      Max
-115.985  -50.739    0.571   64.473   84.953

Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                     7.134e+01  5.593e+00  12.756   <2e-16 ***
cos(2 * pi * time(ts_portfolio)) -2.798e-02  6.786e+00  -0.004    0.997
sin(2 * pi * time(ts_portfolio)) -3.038e+13  4.593e+14  -0.066    0.947
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 63 on 124 degrees of freedom
Multiple R-squared:  4.85e-05,  Adjusted R-squared:  -0.01608
F-statistic: 0.003007 on 2 and 124 DF,  p-value: 0.997
```

Figure 12. The summary of the cosine model

This output summarizes a cosine trend model fitted to time series data. The output includes the data summary, the time series data, the results of the augmented Dickey-Fuller test, and several regression models.

The time series data consists of 127 observations with a frequency of 2. The Augmented Dickey-Fuller test results indicate that the time series is non-stationary.

The first regression model is a linear regression with the time variable as a predictor, showing that time is a significant predictor of the time series. The second regression model includes a quadratic time term and shows that time significantly predicts the time series.

The third model includes a seasonal component with the month as the predictor, which suggests a seasonal pattern in the data.

The fourth model is a seasonal-trend decomposition using Loess (STL) with the robust option, but it produces an error message.

The last is a cosine trend model that includes the cosine and sine terms of the 2*pi*time variable. However, this model is insignificant as the cosine and sine terms have p-values close to 1. This suggests that a cosine trend is not an appropriate model for our time series data.

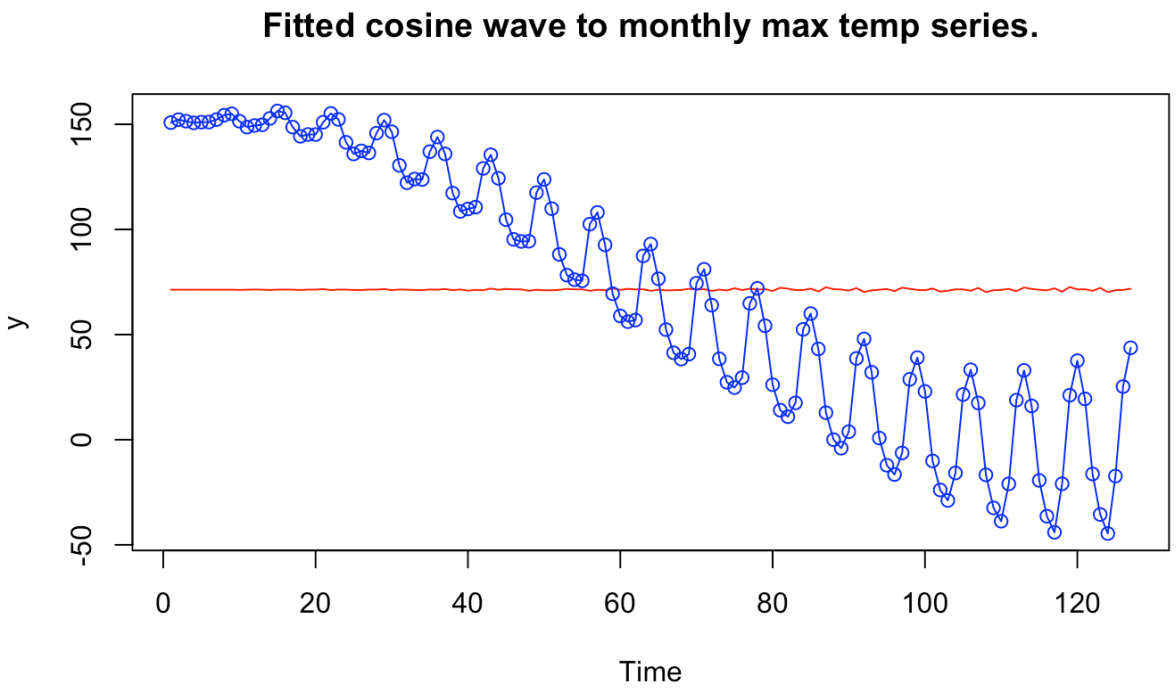

Figure 13. Fitted Cosine curve to the Daily Returns of Investment Portfolio

The graph shows a plot of the fitted values of a cosine model along with the original time series. We set the limits of the y-axis based on the minimum and maximum values of both the fitted values and the original time series. The red-fitted model line is on top of the original blue data.

4.4.5 Cyclical Trend Model

```
```{r Cyclical Trends model}
cyclical_model <- lm(ts_portfolio ~ sin(2*pi*time(ts_portfolio)) + cos(2*pi*time(ts_portfolio)))
summary(cyclical_model)
```
```

```
Call:
lm(formula = ts_portfolio ~ sin(2 * pi * time(ts_portfolio)) +
    cos(2 * pi * time(ts_portfolio)))

Residuals:
     Min       1Q   Median       3Q      Max
-115.985  -50.739    0.571   64.473   84.953

Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                     7.134e+01  5.593e+00  12.756   <2e-16 ***
sin(2 * pi * time(ts_portfolio)) -3.038e+13  4.593e+14  -0.066    0.947
cos(2 * pi * time(ts_portfolio)) -2.798e-02  6.786e+00  -0.004    0.997
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 63 on 124 degrees of freedom
Multiple R-squared:  4.85e-05,  Adjusted R-squared:  -0.01608
F-statistic: 0.003007 on 2 and 124 DF,  p-value: 0.997
```

Figure 14. The summary of the Cyclical trend model

This model uses a cyclical trend to predict the values of a time series variable. The formula used in the model includes two sine and cosine functions of time that represent a repeating pattern over a fixed interval. The model's coefficients indicate each term's estimated effect on the variable's predicted values.

The summary output shows the estimates for the intercept and the coefficients for the sine and cosine terms. The p-values for both coefficients are insignificant, indicating that neither of these terms has a statistically significant effect on the predicted variable. The adjusted R-squared value is negative, which suggests that the model is not a good fit for the data. The F-statistic and associated p-value also suggest that the model does not significantly improve over a null model with no predictors. Finally, the residuals indicate that the model has a high error level, with a large range between the minimum and maximum values.

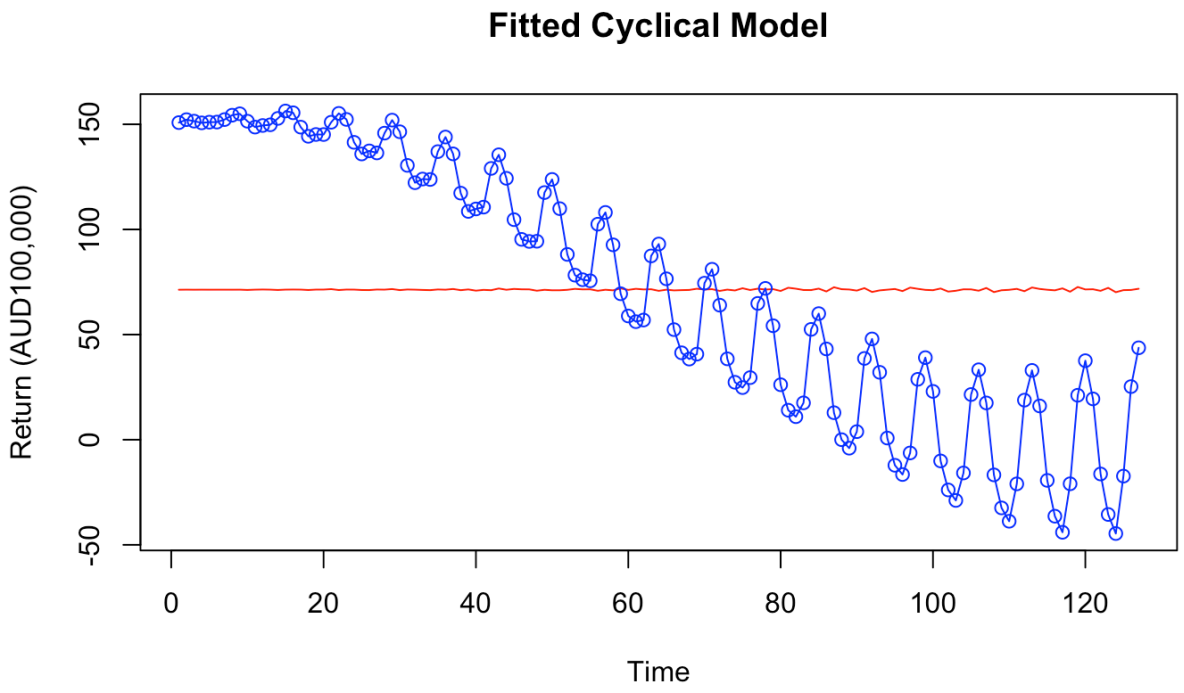

Figure 15. The Fitted Cyclical curve to the Daily Returns of Investment Portfolio

The plot shows the fitted values from a cyclical trends model (in red) and the actual portfolio returns (in blue) over time. We set the limits of the y-axis based on the minimum and maximum values of both the fitted values and the original time series. The red-fitted model line is on top of the original blue data.
Comparing the red and blue lines, we can see that the model's fitted values are relatively flat and do not capture the cyclical patterns in the actual portfolio returns. Therefore, the model may not be a good fit for the data.

Table 3. Comparison of Models

5 different models were tested on the data, and their respective RSS and R-squared values was computed.

Model	R_squared	RSS
1. Linear	8.82E-01	58289.78
2. Quadratic	8.83E-01	57792.56
3. Seasonal	5.68E-01	492111.1
4. Cosine	4.85E-05	492093.74
5. Cyclical	4.85E-05	492093.74

4.5. Diagnostics Checking

When assessing the performance of the models, we considered both the R-squared values, the residual diagnostics, and other diagnostic checking. A good R-squared value for time series data is typically in the 80% -90% range to avoid overfitting, but this is not a strict rule. It is important to consider all other residual diagnostics and p-values as well. Therefore, we used a combination of R-squared values and residual diagnostics to evaluate the models and determine the best fit.

While R-squared is a useful metric, it should not be solely relied upon; other approaches should also be considered.

4.5.1 Residual Sum of Squares (RSS)

RSS measures the difference between the actual and predicted values of the dependent variable in a regression model. In time series analysis, we use RSS to evaluate the goodness of fit of a model. The smaller the RSS, the better the fit of the model.

The quadratic model had the lowest RSS value (57792.56), which indicates that this model had the best fit to the data among all the tested models to minimize the difference between the actual and predicted values. It is important to note that the seasonal, cosine, and cyclical models had very similar RSS values (492111.1, 492093.7, and 492093.7, respectively), which indicates that they were equally good at fitting the data in terms of minimizing the difference between the actual and predicted values.

4.5.2 Coefficient of determination (R^2)

R-squared measures how well a regression model fits the data. In time series analysis, R-squared is used to assess the model's goodness of fit to the observed data. A high R-squared value indicates that the model explains a significant portion of the variability in the data, while a low R-squared value suggests that the model does not fit the data well. Similarly, the quadratic model had the highest R-squared value (0.8825635), indicating that this model explained the largest amount of variance in the data compared to all the other models. However, the seasonal, cosine, and cyclical models' R-squared values were much lower than the linear and quadratic models, indicating that they could have been more effective at explaining the variance in the data. It is important to note that a high R-squared value does not necessarily mean that the model is the best or most appropriate for the data.

4.5.3 Residual Analysis

Residual analysis is used to discover possible dependence in the stochastic component of a time series model. By examining the patterns and autocorrelation of the residuals, we can determine whether the model has captured all the available information in the data or whether there is still some unexplained variation. Residual analysis can also help us identify outliers and assess the overall goodness of fit of the model. It is an essential tool for evaluating time series models and

ensuring they are appropriate for the data being analyzed.

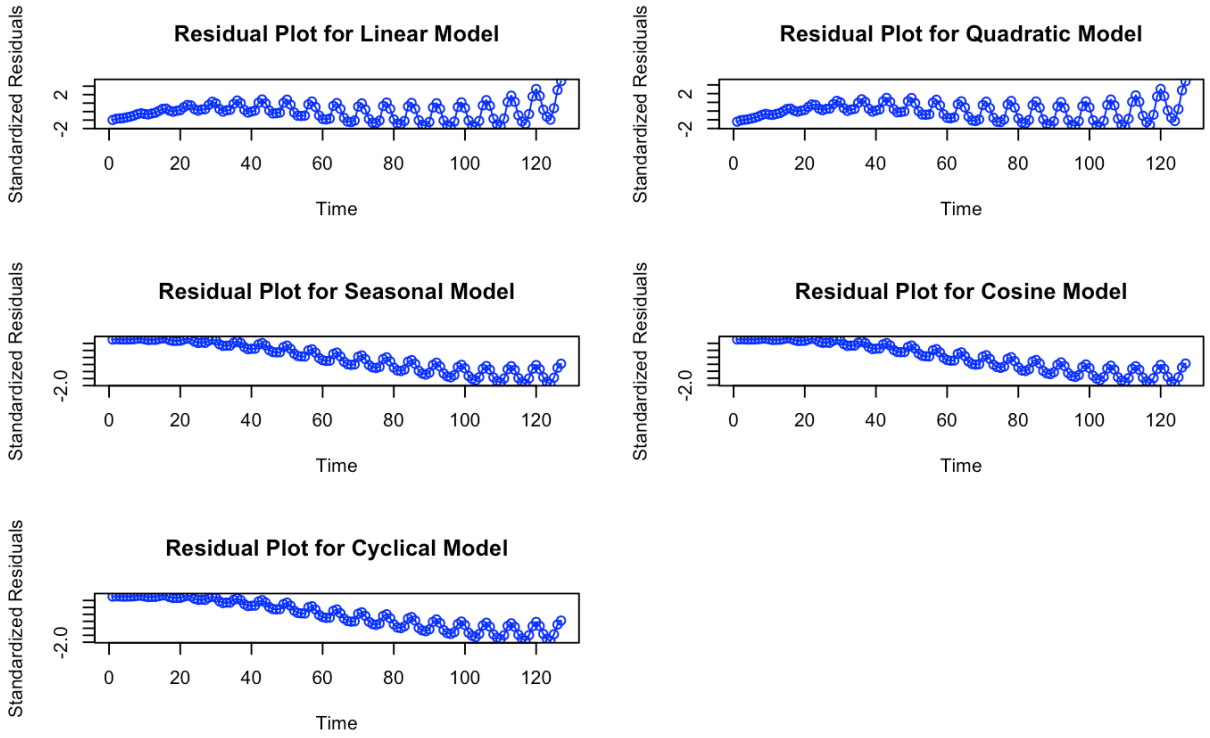

Figure 16. The standardized residuals for all 5 models

Some models, namely the seasonal, cyclical, and cosine models, were not capturing some essential data features, as evidenced by a pattern of decreasing residuals over time. This trend in the residuals may suggest that the models fail to account for certain underlying patterns or trends in the data. Therefore, it is important to examine further and refine these models to capture the data's underlying characteristics better.

4.5.4 a. Standardized Residual Histogram Distribution

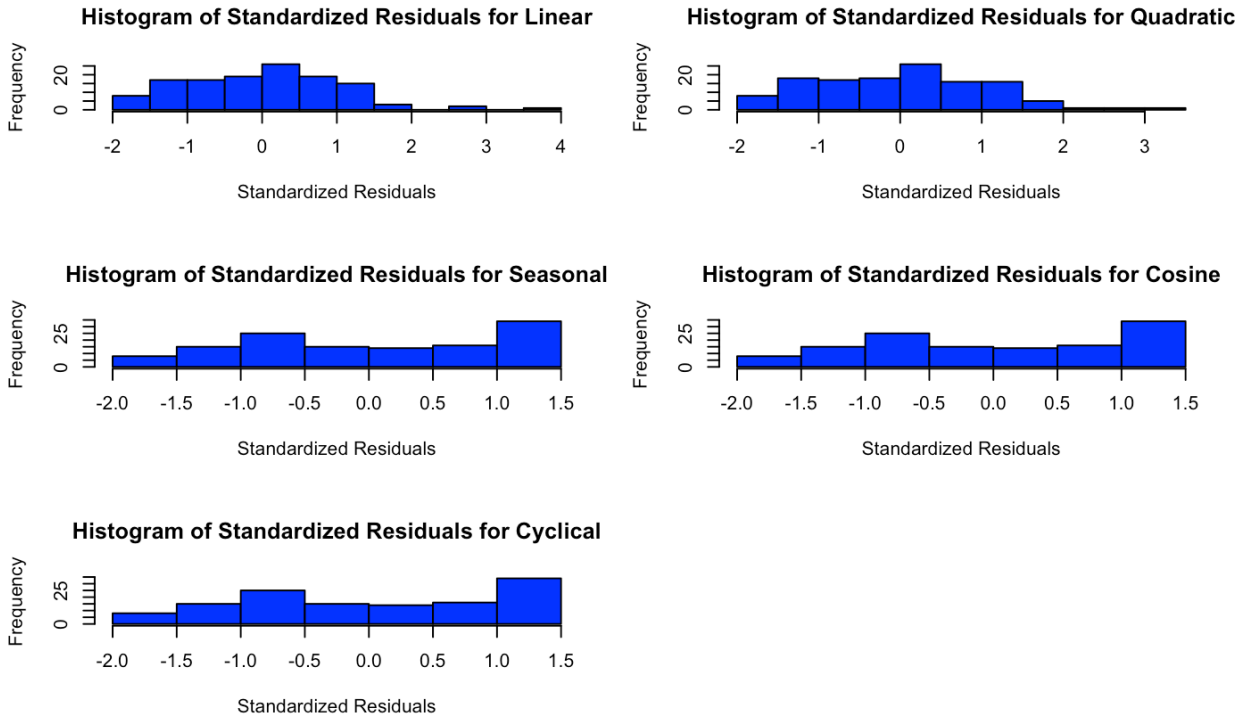

Figure 17. Histogram of Standardized Residuals for all models

The histogram plots of the residuals or standardized residuals for all the models show that the distribution is not perfectly symmetric. This could indicate that the models may only partially capture all the underlying patterns or trends in the data. Further analysis and investigation may be needed to improve the models and better understand the underlying patterns in the data.

4.5.4 b. Standardized Residual Q-Q Plot

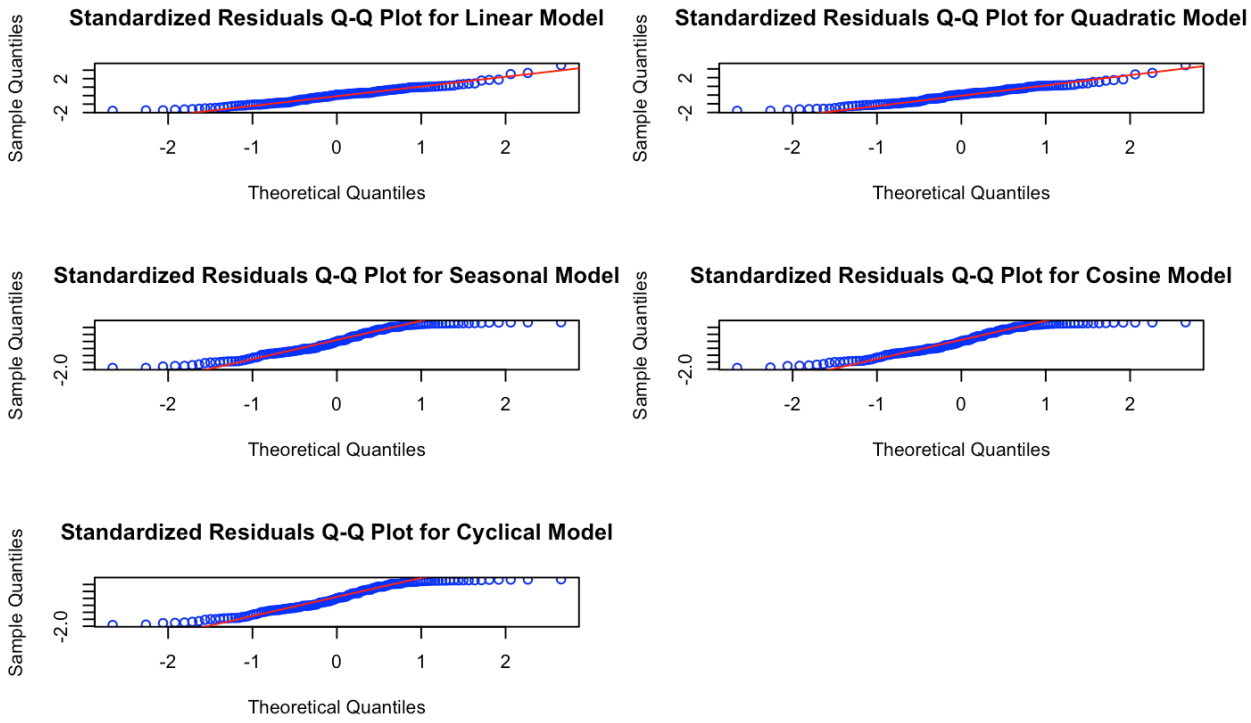

Figure 18. Standardized Residuals Q-Q Plot for all models

It is worth noting that the histogram plots of the standardized residuals for the models did not exhibit uniform distribution. However, the Q-Q plots for the standardized residuals formed a straight line with points closely following it, suggesting the normally distributed stochastic component of the residuals. In the QQ plots of some models, the points at the outer ends deviate from a straight line, which may indicate the non-normality of the residuals. However, most points generally follow a straight line, indicating that the residuals are normally distributed. This contrasts the histogram plots, which did not show a uniform distribution of the standardized residuals across the models.

4.5.4 c. Normality test (Shapiro-Wilk test)

Null hypothesis (H0): Distribution is normal.
Alternative hypothesis (HA): Distribution is not normal.
At an alpha level of 0.05, if the p-value is less than 0.05, reject the null hypothesis and conclude that the distribution is not normal. If the p-value exceeds 0.05, fail to reject the null hypothesis and conclude that the distribution is normal.

Table 4. Normality test for all models

Model	W Value	P-Value
1. Linear	0.97322	0.01269
2. Quadratic	0.97472	0.01764
3. Seasonal	0.92474	2.59E-06
4. Cosine	0.92502	2.70E-06
5. Cyclical	0.92502	2.70E-06

For the linear and quadratic models, the p-values are 0.01269 and 0.01764, respectively. This suggests that the residuals from these models are not normally distributed, as the p-values are less than 0.05. For the seasonal, cosine, and cyclical models, the p-values are very small (2.594e-06, 2.696e-06, and 2.696e-06, respectively), indicating strong evidence against the null hypothesis of normality. Therefore, the residuals from these five models are also not normally distributed.

4.5.6 Sample Autocorrelation Function (ACF)

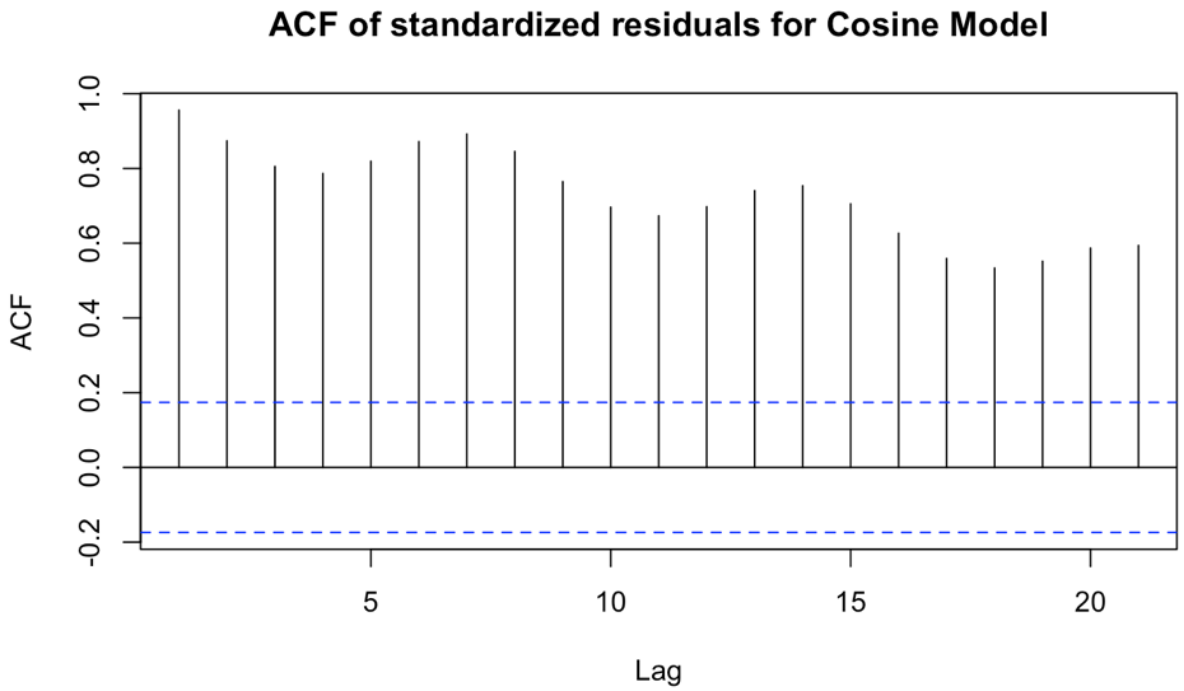

Figure 19. ACF of standardized residuals for Cosine model

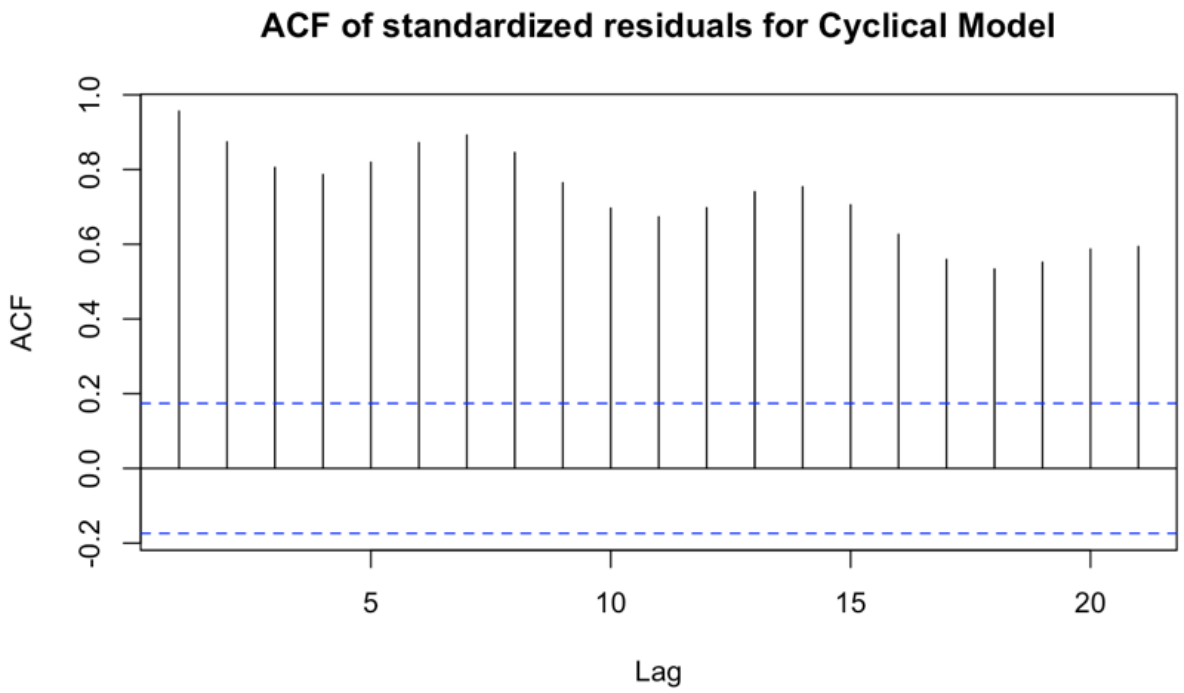

Figure 20. ACF of standardized residuals for Cyclical model

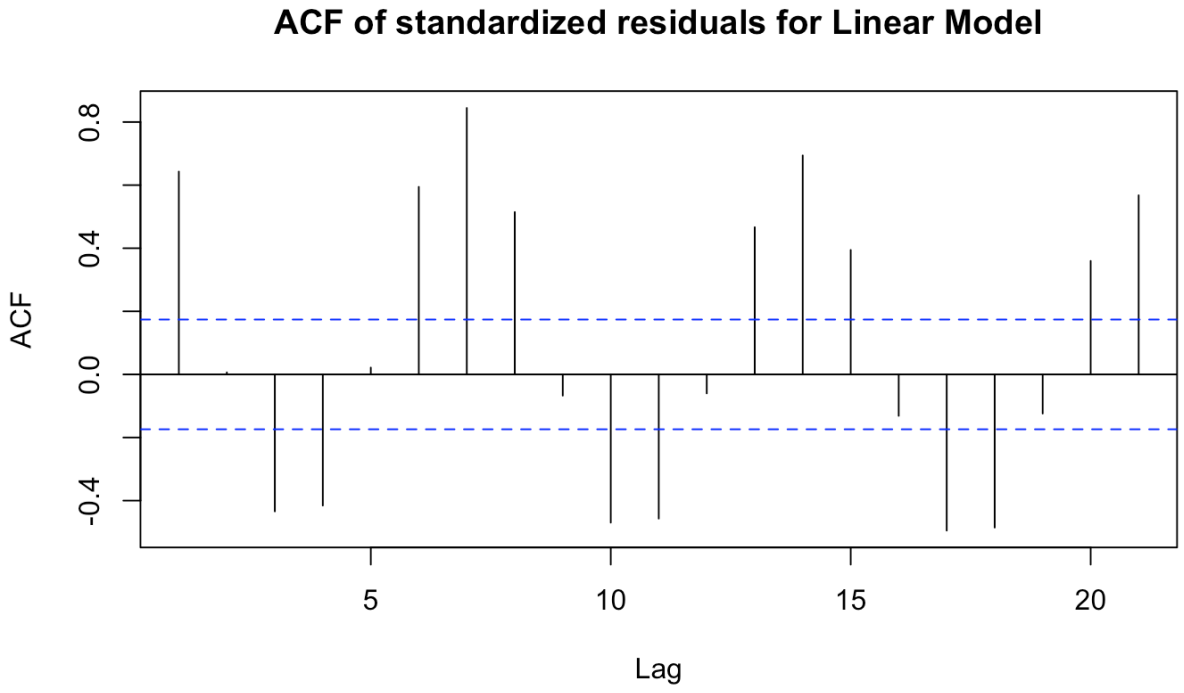

Figure 21. ACF of standardized residuals for Linear model

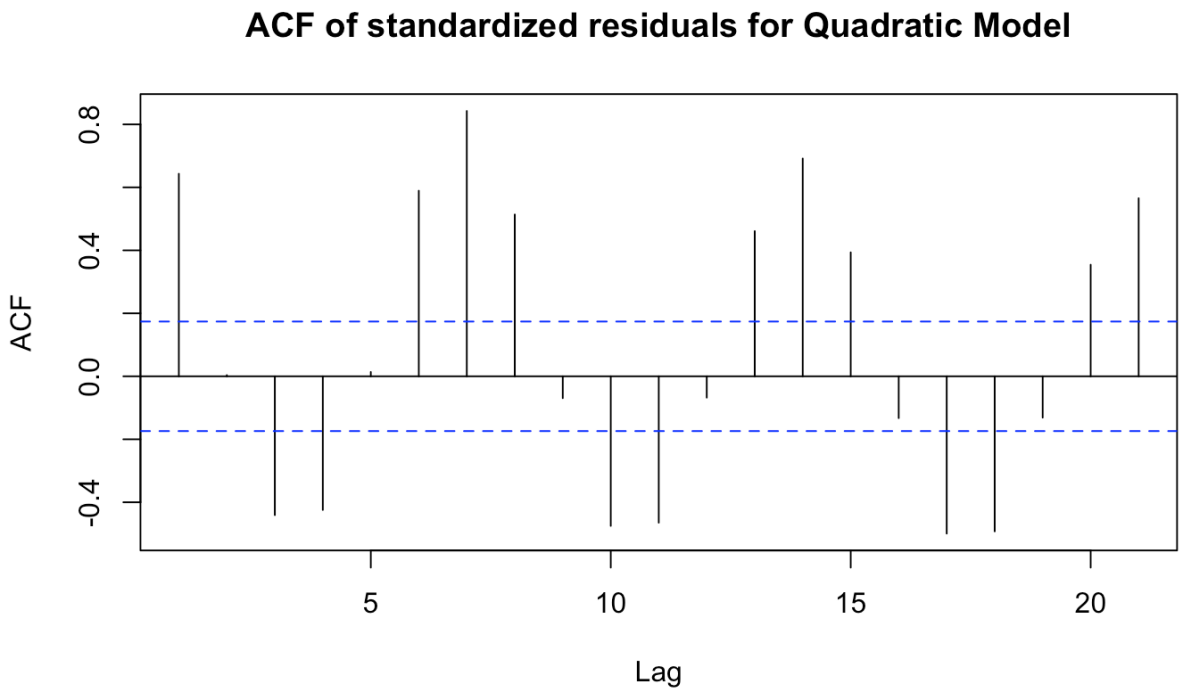

Figure 22. ACF of standardized residuals for Quadratic model

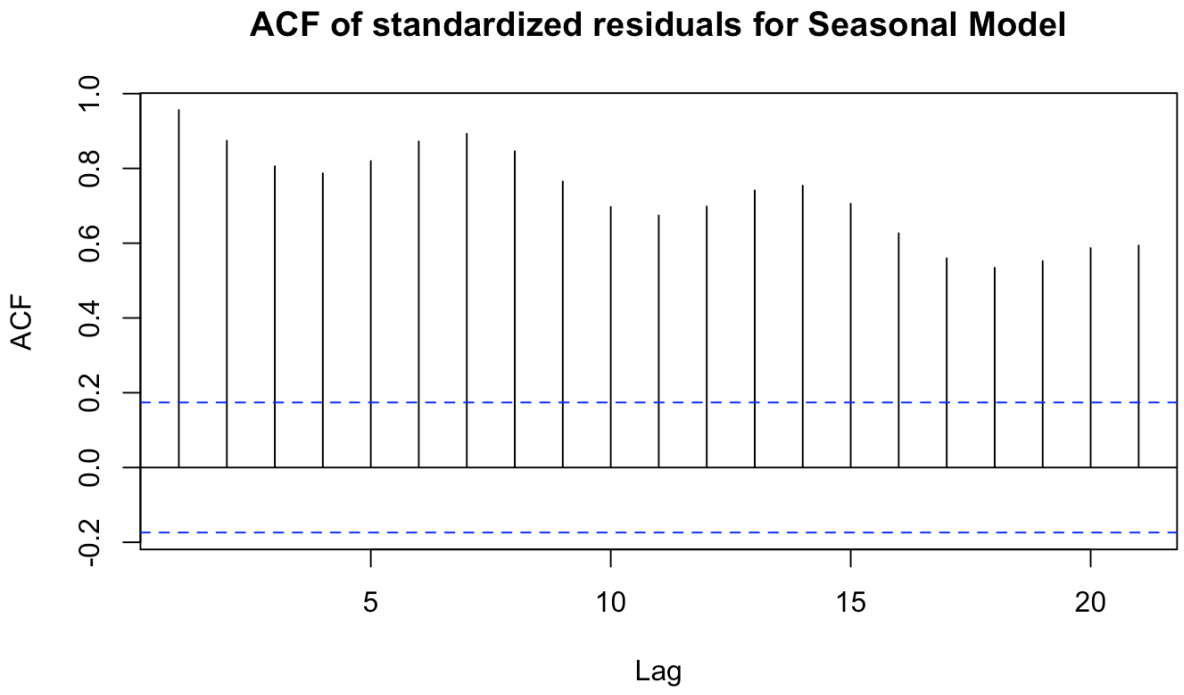

Figure 23. ACF of standardized residuals for Seasonal model

Most bars in an ACF plot of standardized residuals are over the horizontal dashed confidence limits. This suggests the presence of significant autocorrelation in the residuals. Autocorrelation

means that the residual's current value depends on one or more previous values of the residual. This violates the assumption of independence of residuals, which is necessary for accurate statistical inference.

The presence of autocorrelation in residuals can lead to biased and inconsistent parameter estimates, incorrect standard errors, and unreliable hypothesis tests. Therefore, it is important to address and correct autocorrelation in the residuals before drawing any conclusions from the model. One approach is to include additional predictor variables that capture the omitted information contributing to the autocorrelation.

Overall, observing most bars in an ACF plot of standardized residuals over the horizontal dashed confidence limits indicates that the model needs further refinement to account for the autocorrelation in the residuals.

Based on the results of the diagnostic checking, we found that the quadratic trend model is a suitable fit for the return on the investment portfolio of the share market trader. However, it should be noted that diagnostic checking is not a conclusive test, and there may be other data features that the model fails to capture. Therefore, we recommend that the share market trader closely monitor the model's performance and be open to considering alternative models if necessary.

4.6. Forecasting

We predicted the portfolio's returns for the next 15 trading days using the quadratic trend model. Table 3 shows the predicted returns and the corresponding 95% confidence intervals.

Table 5. Predicted Returns and 95% Confidence Intervals

	fit	Lower bound	Upper Bound
1	-173.371	-291.616	-55.1266
2	-175.309	-295.79	-54.8279
3	-177.233	-299.98	-54.4862
4	-179.144	-304.187	-54.1019
5	-181.042	-308.409	-53.6749
6	-182.927	-312.648	-53.2055
7	-184.799	-316.903	-52.6938
8	-186.657	-321.174	-52.1398
9	-188.502	-325.461	-51.5438
10	-190.334	-329.763	-50.9059
11	-192.153	-334.08	-50.2261
12	-193.959	-338.413	-49.5046
13	-195.752	-342.762	-48.7414
14	-197.531	-347.125	-47.9368
15	-199.297	-351.504	-47.0908

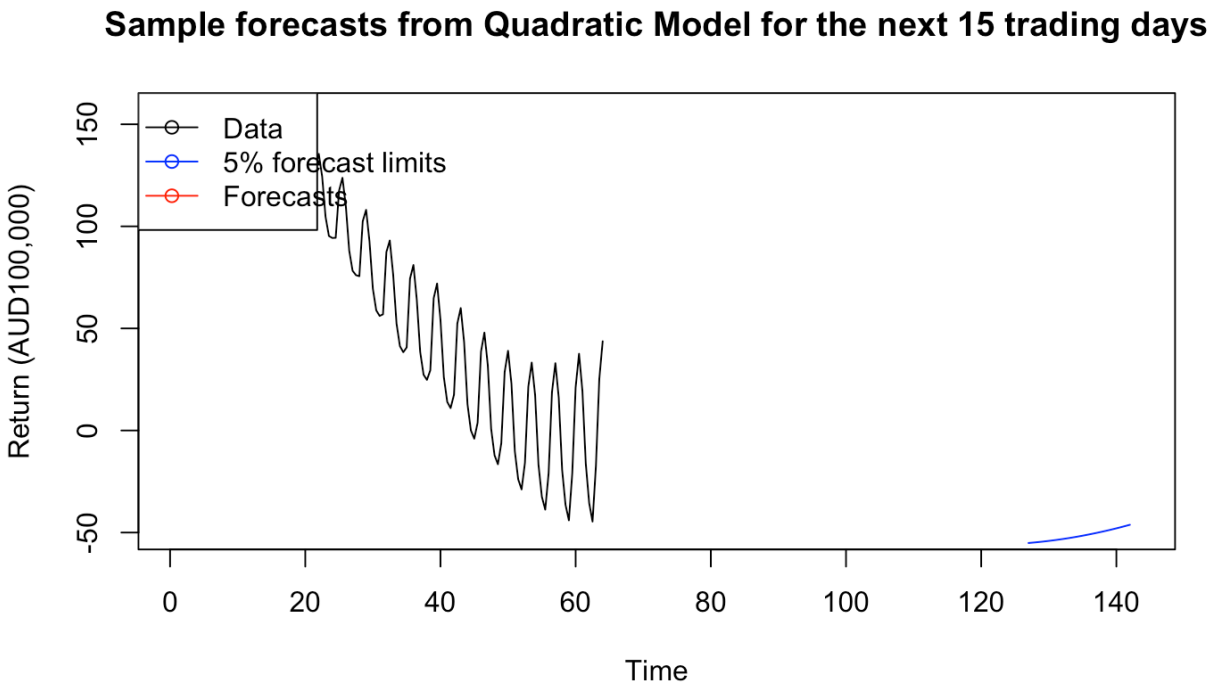

Figure 24. Sample forecasts from Quadratic Model for the next 15 trading days

The table and graph represent predicted returns and 95% confidence intervals over the next 127 trading days. The "fit" column shows the predicted return at each time point. The confidence interval suggests a 95% probability that the true value of the return at each time point will lie within the bounds of the interval (between lower and upper bounds).

5. Results

We analyzed a dataset representing the return of a share market trader's investment portfolio. We used the model-building strategy to identify the best-fitting model among the linear, quadratic, cosine, cyclical, or seasonal trend models.

Based on the results of our analysis, the deterministic trend model is a valid model for the given portfolio data. The residual analysis, including the time series plot, histogram plot, Shapiro-Wilk test, QQ plot, and ACF plot, most indicate that the residuals are not-normally distributed and exhibit significant autocorrelation.

Based on the several diagnostic checks, we found that the quadratic trend model is the best-fitting model for the dataset. We used the quadratic trend model to predict the portfolio's returns for the next 15 trading days. Furthermore, we could generate forecasts using the deterministic quadratic trend model for the next 15 trading days. These results provide additional evidence of the model's effectiveness in capturing and predicting trends in financial data. Overall, this analysis demonstrates the importance of selecting an appropriate model to capture the patterns and trends in time series data and using this model to make informed predictions about future values.

6. Discussion

It is important to note that there are limitations to our analysis. The dataset is limited to a single year and may not represent the long-term trends in the market. Additionally, our model may only capture some factors impacting the return on the investment portfolio. Therefore, the share market trader should continue to explore other models and factors that may impact the return-on-investment portfolio to enhance the accuracy of future predictions.

7. Conclusion

In conclusion, this report has explored a share market trader's return on the investment portfolio and identified the best fitting model among linear, quadratic, cosine, cyclical, or seasonal trend models by implementing the several model-building strategies. Our analysis of the dataset revealed that the quadratic trend model provides the best fit for the given data. We have provided predictions for the next 15 trading days using this model. Our findings have demonstrated that statistical models can be used effectively to analyze and predict trends in the share market. The share market trader can use the results of our analysis to make informed investment decisions. It is recommended that the trader explore other models and factors that may impact the return on the investment portfolio to enhance the accuracy of future predictions. Overall, this report provides valuable insights into the use of statistical models in analyzing the share market and highlights the importance of data analysis in making informed investment decisions.

8. Recommendations

Based on our analysis, we recommend the following:

Use the quadratic trend model: The model provided the best fit for the given dataset. We recommend that the share market trader use this model to make investment decisions based on the predictions for the next 15 trading days.

Explore other models and factors: It is important to continue exploring other models and factors that may impact the return-on-investment portfolio to enhance the accuracy of future predictions. This can include additional variables such as economic indicators, market trends, and news events.

By implementing these recommendations, the share market trader can make informed investment decisions based on accurate predictions, and teaching staff can create a more productive learning environment.

9. Appendices

````
---
title: "assignment1Solution_s3879312"
author: "Thu Tran"
date: "`r Sys.Date()`"
output: pdf_document
---

## R Markdown

```{r import libraries}
rm(list=ls())
library(TSA)
library(tseries)
```


```{r adds a column header and print the info of the provided CSV dataset}
portfolio <- read.csv("assignment1Data2023.csv", header=TRUE)
# Add a column header
colnames(portfolio) <- c("id", "x")
# class of the data set
class(portfolio)
# View the data set
print(portfolio)
# summary statistics
summary(portfolio)
```

```{r Create the ACF plot for finding the frequency}
# Create the ACF plot for original portfolio data
acf(portfolio)
# Find the frequency
frequency(portfolio)
```



```{r convert to time series object with the frequency of 2}
# Convert to time series object
freq <- 252/as.numeric(diff(range(portfolio$id), units = "days"))
ts_portfolio <- ts(portfolio$x, frequency = freq)
# Check the structure of the time series object
str(ts_portfolio)
````
````

```
# class of the converted data
class(ts_portfolio)
# View the time series object
print(ts_portfolio)

```



```{r ACF plot with frequency of 2}
# Create the ACF plot
acf(ts_portfolio)
# Find the frequency
frequency(ts_portfolio)
```



```{r time series plot}
# Create a time series plot and add axis labels
plot(ts_portfolio, type='o', main = "Time series plot of Daily Returns of Investment Portfolio",
ylab = "Return (AUD100,000)", xlab = "Trading Days", col="blue")
```

```{r scatter plot}
plot(y=ts_portfolio,x=zlag(ts_portfolio),ylab='Return (AUD100,000)', xlab='Previous Year
Return (AUD100,000)', main = "Scatter plot of Daily Investment Return in Consecutive
Years", col="blue")
```

```{r find correlation in this scatter plot}
y = ts_portfolio             # Read the data into y
x = zlag(ts_portfolio)       # Generate first lag of the  series
index = 2:length(x)    # Create an index to get rid of the first NA value in x if applicable
cor(y[index],x[index]) # Calculate correlation between numerical values in x and y
```



```{r Augmented Dickey-Fuller Test to determine Deterministic Versus Stochastic Trends}
# Perform ADF test
adf.test(ts_portfolio)
```

```{r plot decomsed time series}
# Decompose the time series into its components
ts_decomp <- decompose(ts_portfolio)
# Plot the decomposed components
plot(ts_decomp, col="blue")
```

```
```

```{r deterministic linear trend model}
# the summary of the deterministic linear trend model
linear_model = lm(ts_portfolio~time(ts_portfolio)) # label the model as linear_model
summary(linear_model)
```

```{r Plot the deterministic linear trend model with the fitted trend line}
plot(ts_portfolio, type='o',ylab='Return (AUD100,000)', main = "Fitted linear model to the
Daily Returns of Investment Portfolio", col="blue")
abline(linear_model, col="red") # add the fitted least squares line from linear model
```

```{r deterministic quadratic trend model}
# the summary of the deterministic quadratic trend model
t = time(ts_portfolio)
t2 = t^2
quadratic_model = lm(ts_portfolio~t+t2) # label the model as quadratic_model
summary(quadratic_model)
```

```{r Fitted quadratic curve to the Daily Returns of Investment Portfolio}
plot(ts(fitted(quadratic_model)), ylim = c(min(c(fitted(quadratic_model),
    as.vector(ts_portfolio))),
max(c(fitted(quadratic_model),as.vector(ts_portfolio)))),ylab='Return (AUD100,000)' ,
    main = "Fitted quadratic curve to the Daily Returns of Investment Portfolio", col="red")
lines(as.vector(ts_portfolio),type="o", col="blue")
```

```{r seasonal deterministic trend model}
month.=season(ts_portfolio)
# Season() function creates a factor variable showing the months.
# period added to improve table display and this line sets up indicators
season_model=lm(ts_portfolio ~ month.-1) # add -1 to remove the intercept term
summary(season_model)
```

```{r Fitted seasonal curve to the Daily Returns of Investment Portfolio}
```

```
plot(ts(fitted(season_model)), ylab='Return (AUD100,000)', main = "Fitted seasonal
deterministic model without intercept", ylim = c(min(c(fitted(season_model),
as.vector(ts_portfolio))), max(c(fitted(season_model), as.vector(ts_portfolio)))), col = "red" )
lines(as.vector(ts_portfolio),type="o", col = "blue")
```

Cosine Trend

```{r cosine trend model}
cosine_model <- lm(ts_portfolio ~ cos(2*pi*time(ts_portfolio)) + sin(2*pi*time(ts_portfolio)))
summary(cosine_model)
```

```{r}
plot(ts(fitted(cosine_model)), ylab='Return (AUD100,000)', main = "Fitted cosine wave.",ylim
= c(min(c(fitted(cosine_model), as.vector(ts_portfolio))) ,
max(c(fitted(cosine_model), as.vector(ts_portfolio)))
), col = "red" )
lines(as.vector(ts_portfolio),type="o", col="blue")
```

```{r Cyclical Trends model}
cyclical_model <- lm(ts_portfolio ~ sin(2*pi*time(ts_portfolio)) +
cos(2*pi*time(ts_portfolio)))
summary(cyclical_model)
```

```{r}
plot(ts(fitted(cyclical_model)), ylab='Return (AUD100,000)', main = "Fitted Cyclical
Model",ylim = c(min(c(fitted(cyclical_model), as.vector(ts_portfolio))) ,
max(c(fitted(cyclical_model), as.vector(ts_portfolio)))
), col = "red" )
lines(as.vector(ts_portfolio),type="o", col="blue")
```

```{r}
# Fit linear, quadratic, cosine, cyclical, and seasonal trend models
linear_model <- linear_model
quadratic_model <- quadratic_model
season_model <- season_model
cosine_model <- cosine_model
```

```
cyclical_model <- cyclical_model

# Compute the residual sum of squares (RSS) for each model
linear_RSS <- sum(resid(linear_model)^2)
quadratic_RSS <- sum(resid(quadratic_model)^2)
season_RSS <- sum(resid(season_model)^2)
cosine_RSS <- sum(resid(cosine_model)^2)
cyclical_RSS <- sum(resid(cyclical_model)^2)

# Compute the R-squared for each model
linear_r_squared <- summary(linear_model)$r.squared
quadratic_r_squared <- summary(quadratic_model)$r.squared
season_r_squared <- summary(season_model)$r.squared
cosine_r_squared <- summary(cosine_model)$r.squared
cyclical_r_squared <- summary(cyclical_model)$r.squared

# Print the RSS for each model
cat("Linear Model RSS:", linear_RSS, "\n")
cat("Quadratic Model RSS:", quadratic_RSS, "\n")
cat("Seasonal Model RSS:", season_RSS, "\n")
cat("Cosine Model RSS:", cosine_RSS, "\n")
cat("Cyclical Model RSS:", cyclical_RSS, "\n")

# Print the R-squared for each model
cat("\nLinear Model R-squared:", linear_r_squared, "\n")
cat("Quadratic Model R-squared:", quadratic_r_squared, "\n")
cat("Seasonal Model R-squared:", season_r_squared, "\n")
cat("Cosine Model R-squared:", cosine_r_squared, "\n")
cat("Cyclical Model R-squared:", cyclical_r_squared, "\n")

# Select the model with the lowest RSS
RSS <- c(linear_RSS, quadratic_RSS, season_RSS, cosine_RSS, cyclical_RSS)
best_model <- which.min(RSS)

# Select the model with the highest RSS
R_2 <- c(linear_r_squared, quadratic_r_squared, season_r_squared, cosine_r_squared,
cyclical_r_squared)
best_model <- which.min(RSS)

# Print the best fitting model in term of RSS
cat("\nBest Fitting Model in term of RSS:", c("Linear", "Quadratic", "Seasonal","Cosine",
"Cyclical")[best_model])
# Print the best fitting model interm of R-squared
cat("\n\nBest Fitting Model in term of R-squared:", c("Linear", "Quadratic",
"Seasonal","Cosine", "Cyclical")[best_model])
```

```
# Create a data frame of the evaluation metrics
compare_model_df <- data.frame(Model = c("Linear", "Quadratic", "Seasonal","Cosine",
"Cyclical"),
                               R_squared = c(linear_r_squared, quadratic_r_squared, season_r_squared,
cosine_r_squared, cyclical_r_squared),
                               RSS = c(linear_RSS, quadratic_RSS, season_RSS, cosine_RSS,
cyclical_RSS))

# Print the data frame
compare_model_df
```

```{r}
create_residual_plots <- function(model_list, ts_data) {
  n_models <- length(model_list)
  par(mfrow=c(ceiling(n_models/2), 2)) # set up multiple plots

  for (i in seq_along(model_list)) {
    residuals <- rstudent(model_list[[i]])
    plot(residuals, xlab='Time', ylab='Standardized Residuals',
         type='o', main = paste("Residual Plot for", names(model_list)[i]), col="blue")
  }

  par(mfrow=c(1, 1)) # reset plot settings
}

models_list <- list(linear_model, quadratic_model, season_model, cosine_model,
cyclical_model)
names(models_list) <- c("Linear Model", "Quadratic Model", "Seasonal Model", "Cosine
Model", "Cyclical Model")

create_residual_plots(models_list, ts_portfolio)

```

```{r}
create_residual_hist_plots <- function(model_list) {
  n_models <- length(model_list)
  par(mfrow=c(ceiling(n_models/2), 2)) # set up multiple plots

  for (i in seq_along(model_list)) {
```

```
    residuals <- rstudent(model_list[[i]])
    hist(residuals, xlab='Standardized Residuals',
         main = paste("Histogram of Standardized Residuals for", names(model_list)[i]),
col="blue")
  }

  par(mfrow=c(1, 1)) # reset plot settings
}
models_list <- list(linear_model, quadratic_model, season_model, cosine_model,
cyclical_model)
names(models_list) <- c("Linear", "Quadratic", "Seasonal", "Cosine", "Cyclical")
create_residual_hist_plots(models_list)

```

```{r}
plot_qq_all_models <- function(model_list) {
  n_models <- length(model_list)
  par(mfrow=c(ceiling(n_models/2), 2)) # set up multiple plots

  for (i in seq_along(model_list)) {
    residuals <- rstudent(model_list[[i]])
    qqnorm(residuals, col="blue", main = paste("Standardized Residuals Q-Q Plot for",
names(model_list)[i]))
    qqline(residuals, col = "red")
  }

  par(mfrow=c(1, 1)) # reset plot settings
}
models <- list(linear_model, quadratic_model, season_model, cosine_model, cyclical_model)
names(models) <- c("Linear Model", "Quadratic Model", "Seasonal Model", "Cosine Model",
"Cyclical Model")

plot_qq_all_models(models)

```

```{r Normality Test (Shapiro-Wilk)}
shapiro_test <- function(model) {
  residuals <- rstudent(model)
  shapiro.test(residuals)
}
for (i in seq_along(models)) {
  print(names(models)[i])
  print(shapiro_test(models[[i]]))
}
```

```
```

```{r ACF plots for the standardized residuals}
plot_standardized_resid_acf <- function(models_list) {
  for (i in 1:length(models_list)) {
    model_name <- deparse(substitute(models_list[[i]]))
    resid <- rstudent(models_list[[i]])
    acf(resid, main = paste0("ACF of standardized residuals for ", names(models)[i]))
  }
}

models <- list(linear_model, quadratic_model, season_model, cosine_model, cyclical_model)
names(models) <- c("Linear Model", "Quadratic Model", "Seasonal Model", "Cosine Model",
"Cyclical Model")
plot_standardized_resid_acf(models)
```

Forecasting with quadratic models

```{r}
h <- 15 # 15 steps ahead forecasts
t <- time(ts_portfolio)
t2 <- t^2
aheadTimes <- data.frame(t = seq(127, 127+h, 1),
t2 = seq(127, 127+h, 1)^2)
quadratic_model_prediction <- predict(quadratic_model, newdata = aheadTimes, interval =
"prediction")
quadratic_model_prediction
```

```{r}
plot(ts_portfolio, xlim= c(1,127+h+1), ylim = c(-50,157),
ylab = "Return (AUD100,000)",
main = "Sample forecasts from Quadratic Model for the next 15 trading days")
lines(ts(as.vector(quadratic_model_prediction[,3]), start = 127), col="blue", type="l")
lines(ts(as.vector(quadratic_model_prediction[,1]), start = 127), col="red", type="l")
lines(ts(as.vector(quadratic_model_prediction[,2]), start = 127), col="blue", type="l")
legend("topleft", lty=1, pch=1, col=c("black","blue","red"),
text.width = 12,
c("Data","5% forecast limits", "Forecasts"))
```

References

[1] A. M. De Livera, R. J. Hyndman, and R. D. Snyder, "Forecasting Time Series With Complex Seasonal Patterns Using Exponential Smoothing," *Journal of the American Statistical Association*, vol. 106, no. 496, pp. 1513–1527, Dec. 2011, doi: https://doi.org/10.1198/jasa.2011.tm09771.

[2] S. Fan and R. J. Hyndman, "Short-Term Load Forecasting Based on a Semi-Parametric Additive Model," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 134–141, Feb. 2012, doi: https://doi.org/10.1109/tpwrs.2011.2162082.

[3] E. Parzen, "ARARMA models for time series analysis and forecasting," *Journal of Forecasting*, vol. 1, no. 1, pp. 67–82, Jan. 1982, doi: https://doi.org/10.1002/for.3980010108.

[4] "11.1 Complex seasonality | Forecasting: Principles and Practice," *Otexts.com*, 2011. https://otexts.com/fpp2/complexseasonality.html

[5] J. D. Cryer and K.-S. Chan, *Time series analysis: with applications in R*. New York: Springer, 2008.