

MATH1318 Time Series Analysis

Assignment 1

Joyal Joy Madeckal - s3860476

Introduction

Time series is a series of data in the order of time. The order of the data is very important in a time series. This assignment mainly focuses on interacting with time series data with the following objectives:

- Descriptive Analysis of given time series data (closing price of a share)
- To find the best fitting model among linear, quadratic, cosine, cyclical or seasonal models
- Use model building strategy for building the model including descriptive analysis, model formulation, model fitting, model diagnosis.
- Forecast the values for next 5 time periods.

Model Building

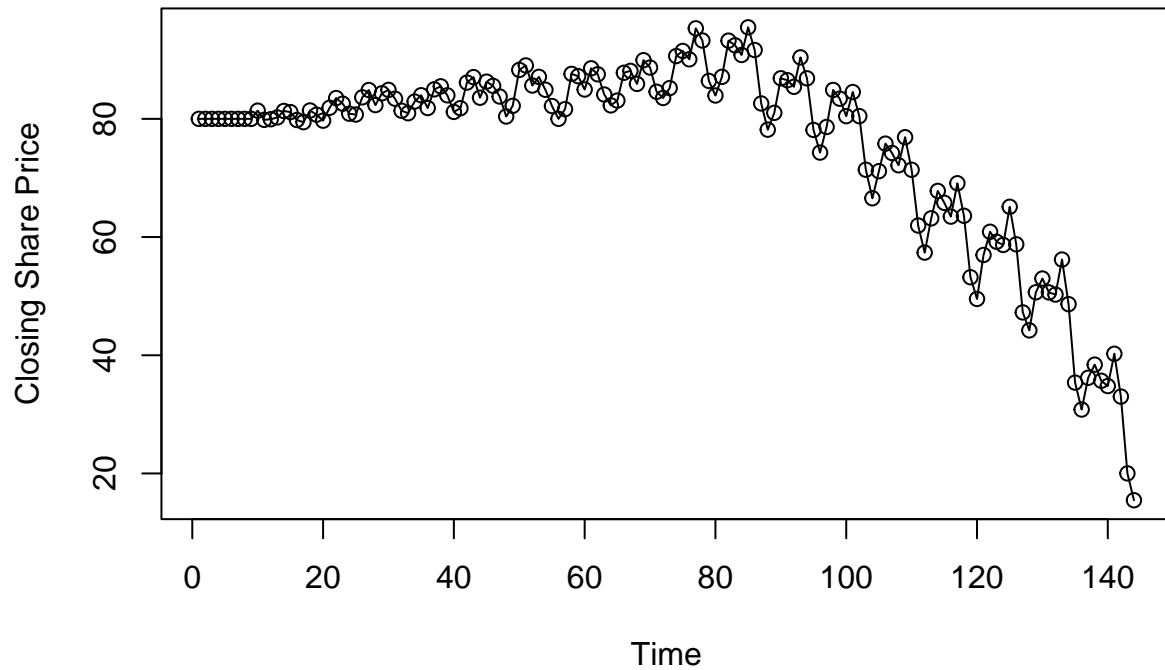
Descriptive Analysis

- Loading the required packages and data.
- Visualizing the time series

```
library(TSA)
# Loading the data
share_price <- read.csv("assignment1Data2022.csv", col.names = c("Time", "Share_Price"))
# Converting data to time series object
share_price_ts <- ts(share_price$Share_Price, start = 1, end = 144)

# Creating a function for plotting the time series data as it will be used repetitively
# for all the models.
plot_share_prices <- function(plot_header){
  plot(share_price_ts, xlab = "Time", ylab = "Closing Share Price",
       main = plot_header, type = "o")
}
# Plotting the time series
plot_share_prices("Time series plot for closing share prices")
```

Time series plot for closing share prices



The time series plot looks like there is some repeating pattern but the pattern is going downwards and variance is increasing. We can clearly see that there is an “M” pattern repeating in the time series.

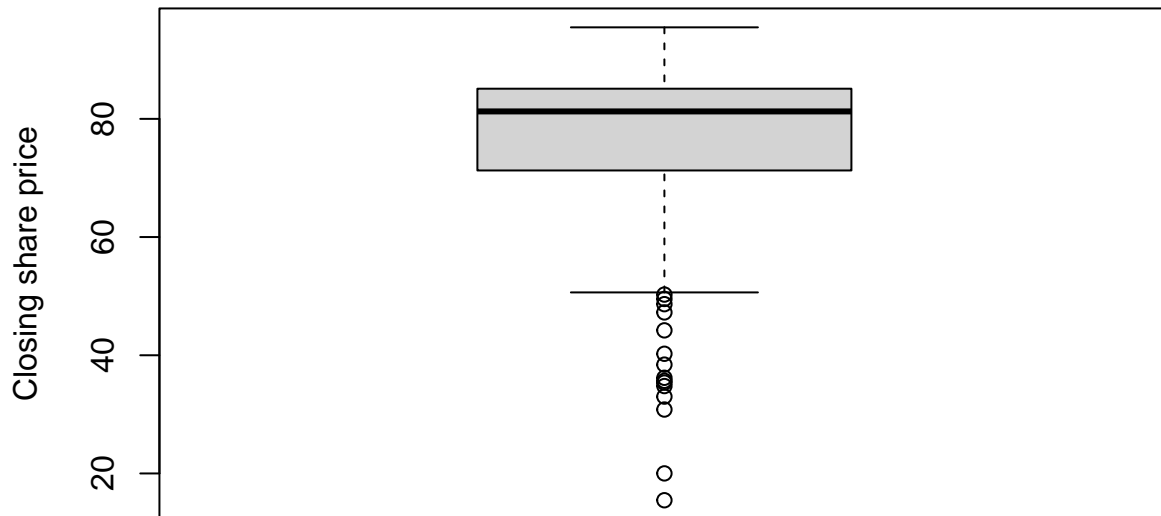
Now, we will see some of the descriptive statistics of the time series.

```
summary(share_price_ts)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  15.47   71.33   81.25   75.41   85.05   95.48
```

```
boxplot(share_price_ts, main = "Share price data distribution",
        ylab = "Closing share price")
```

Share price data distribution



- The mean of the data is less than the median which indicates that the data is left skewed.
- The boxplot says that mostly data is concentrate between 70 and 80.

We will analyse the pattern based on the following key points.

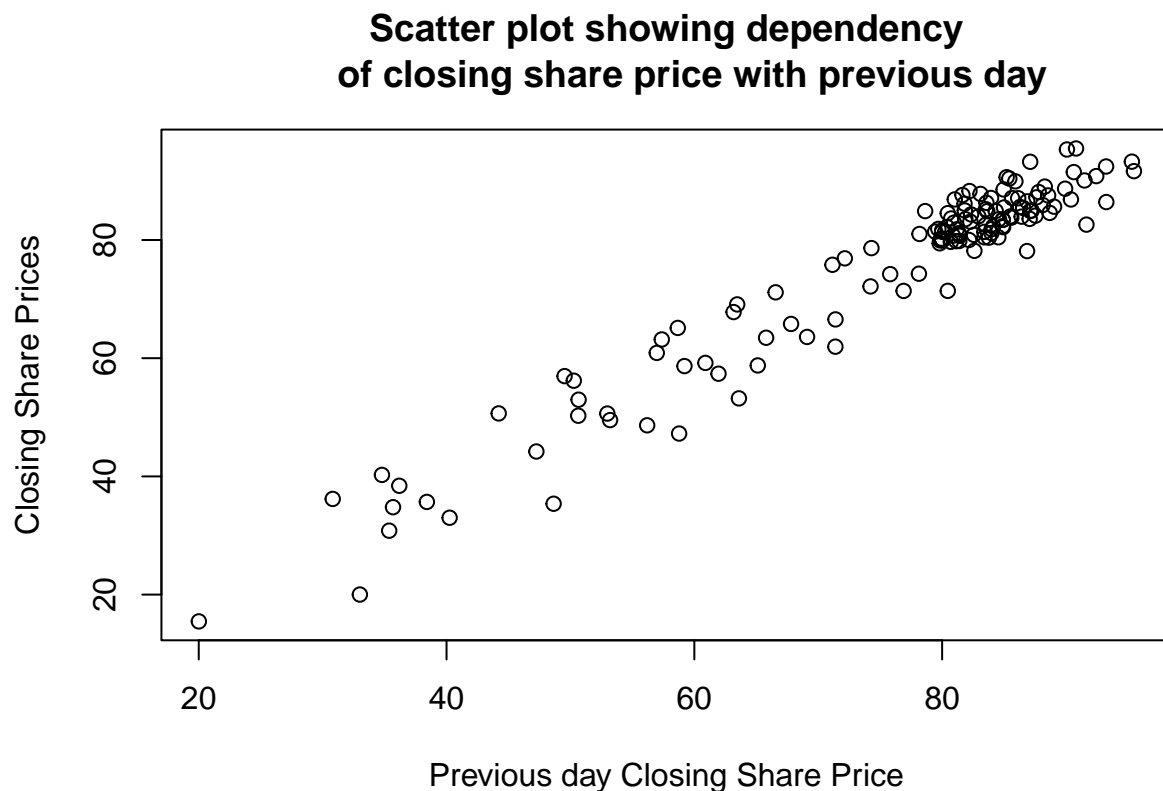
- **Trend** - We can see that there are multiple trends in the series. At the start the series is going upward and then it goes downward. But we can say that the series is predominantly going downwards since the upward slope is less when compared with the downward slope.
- **Seasonality** - We cannot find seasonality in the data. Although we can see 4 repeating patterns between time points 40 and 80, it is not following the trend afterwards. The amplitude of the pattern increases and pattern goes downwards.
- **Changing Variance** - We can say that the variance of the pattern is consistently increasing as going forward and hence we are observing changing variance (increasing) in the time series.
- **Behaviour** - There are data points close to each other as in auto regressive (AR) model and the pattern is going up and down changing the average (MA) and hence the time is having both AR and MA characteristics.
- **Change Point** - From the time series plot we are unable to see any change points. There is no point in the plot where there is a sharp change from the existing pattern.

The trading days are consecutive and hence we have to check the dependency of the share prices from the previous trading days and see if there is any stochastic trend in the series. For this analysis we check the **first lag auto correlation** of the series.

```
# Generating the first lag series
share_price_ts_first_lag <- zlag(share_price_ts)
data_range <- 2:length(share_price_ts)
# Checking the correlation with previous trading day
cor(share_price_ts_first_lag[data_range], share_price_ts[data_range])
```

```
## [1] 0.9678333
```

```
# Plotting share prices for analyzing correlation
plot(y = share_price_ts, x = share_price_ts_first_lag, ylab = "Closing Share Prices",
     xlab = "Previous day Closing Share Price", main = "Scatter plot showing dependency
     of closing share price with previous day")
```



- The correlation coefficient value 0.967 indicates there is strong positive correlation on the closing share price from the previous day.
- There are lot of data points around (80, 80) and this is because at the start of the time series we can see that the closing share prices are constant for some period.
- Stochastic trend of the data is clear from the coefficient and the scatter plot indicates the same.

Model Fitting

The analysis of the time series data and descriptive statistics is showing us that there is some curvy nature for the data with some pattern repeating. These are suggesting that we should be definitely fitting the quadratic and cyclic models.

When we start modelling of the data we may not have much idea which model will work out in our case. So, here we will try to fit number of models to the data and will analyse the model statistics. Based on the statistics and model diagnosis we will choose a suitable model for forecasting. The following models will be fitted to the time series data:

1. Linear Model
2. Quadratic Model
3. Cyclic/Seasonal Model
4. Cosine Model

If we are unable to find a suitable fit from the above models **we can try combining the models keeping the principle of parsimony in mind.**

For any model, we are interested in two parameters - **p-value** for checking if the intercept, coefficients and overall model are significant (whether model is good: p-value should be below the significance level of 0.05) and **R-squared** value which indicates how well the model is fitting the data (whether the model is useful: Value between 0.8 -0.85 can be considered as a good fit)

Another important thing is even if our time series is following stochastic trend we are trying to model it by deterministic models. So, **these models may be able to catch the series tendency but not the auto correlation structure.**

Fitting a Linear Model

Linear model follows the equation:

$\mu_t = \beta_0 + \beta_1 t$, where μ_t is the data point, β_0 is intercept, β_1 is the slope.

```
# Extracting the time from time series object
periods <- time(share_price_ts)
# Fitting linear model
linear_model <- lm(share_price_ts~periods)
summary(linear_model)

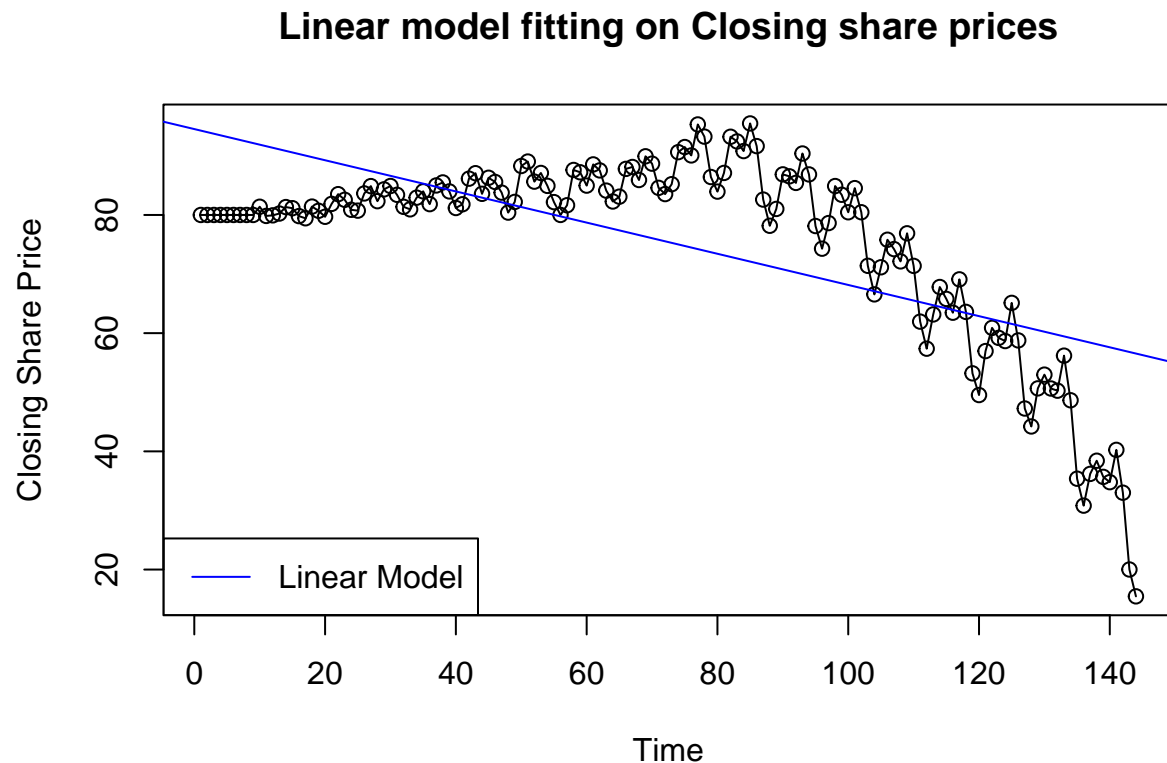
##
## Call:
## lm(formula = share_price_ts ~ periods)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -41.083  -8.949   0.231   8.896  23.370
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  94.53267    2.02418   46.70  <2e-16 ***
## periods      -0.26377    0.02422  -10.89  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.08 on 142 degrees of freedom
## Multiple R-squared:  0.4551, Adjusted R-squared:  0.4513
## F-statistic: 118.6 on 1 and 142 DF,  p-value: < 2.2e-16
```

- The p-value for the intercepts, coefficients and overall model is falling below 0.05 and hence they are all significant which indicates the model is good.

- The R-squared value is low (0.4513). So, the model is not useful for forecasting.

We will plot the linear model on top of the time series plot to visualize the model.

```
plot_share_prices("Linear model fitting on Closing share prices")
# Adding the linear model
abline(linear_model, col = "blue")
legend("bottomleft", legend = "Linear Model", col = "blue", lty=1)
```



- The blue line is the fitted linear model for the time series data and we can see that the fit is not good at all.
- The linear model is unable to capture the curvy behaviour of the data.

Fitting a Quadratic Model

Quadratic model follows the equation:

$\mu_t = \beta_0 + \beta_1 t + \beta_2 t^2$, where μ_t is the data point, β_0 is intercept, β_1 corresponding to linear trend and β_2 corresponding to quadratic trend

```
periods2 <- periods^2
quadratic_model <- lm(share_price_ts~periods+periods2)
summary(quadratic_model)
```

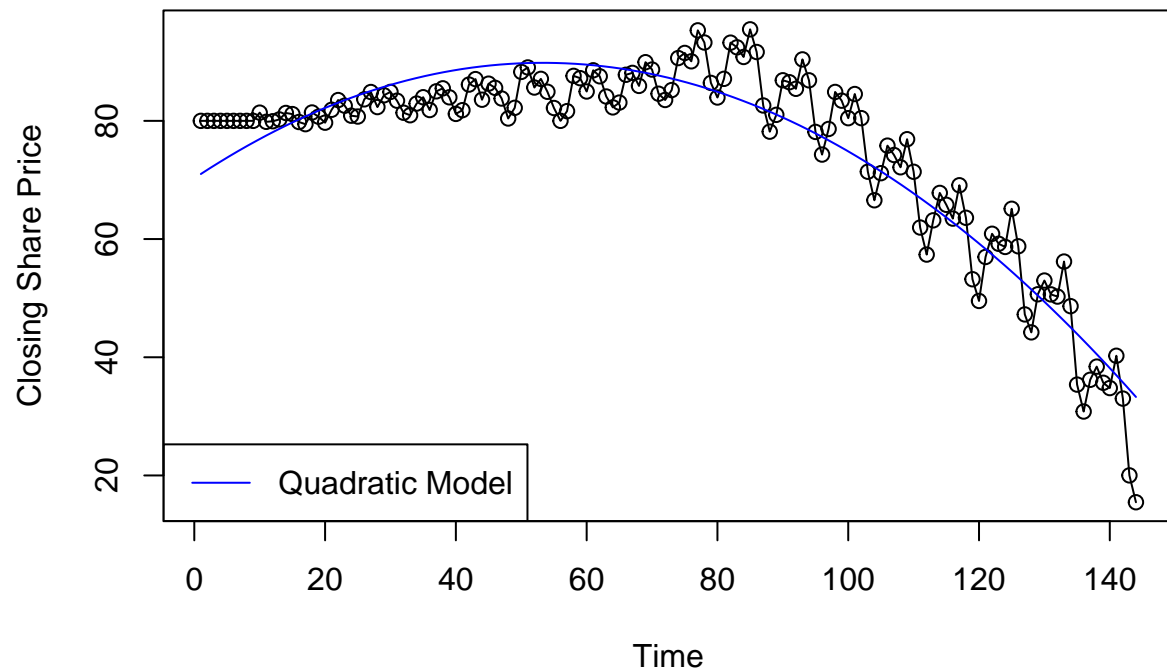
```
##
## Call:
## lm(formula = share_price_ts ~ periods + periods2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.8176  -3.5322  -0.7553   3.6988  12.5656
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 70.2777995  1.4284180   49.20  <2e-16 ***
## periods      0.7330056  0.0454811   16.12  <2e-16 ***
## periods2     -0.0068743  0.0003038  -22.62  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.634 on 141 degrees of freedom
## Multiple R-squared:  0.8823, Adjusted R-squared:  0.8807
## F-statistic: 528.6 on 2 and 141 DF,  p-value: < 2.2e-16
```

- The p-value for the intercepts, coefficients and overall model is falling below 0.05 and hence they are all significant which indicates the model is good.
- The R-squared value is high (0.8823). So, the model is useful for forecasting but, as we are looking for R squared values in the range 0.8-0.85 there can be a chance the model can be slightly over fitting.

We will try to plot the quadratic model on top of the time series plot to visualize the model.

```
plot_share_prices("Quadratic model fitting on Closing share prices")
# Adding the quadratic model
curve(quadratic_model$coefficients[1]+x*quadratic_model$coefficients[2]+
      x^2*quadratic_model$coefficients[3], from = 1,
      to = 144, add = TRUE, col = "blue")
legend("bottomleft", legend = "Quadratic Model", col = "blue", lty=1)
```

Quadratic model fitting on Closing share prices



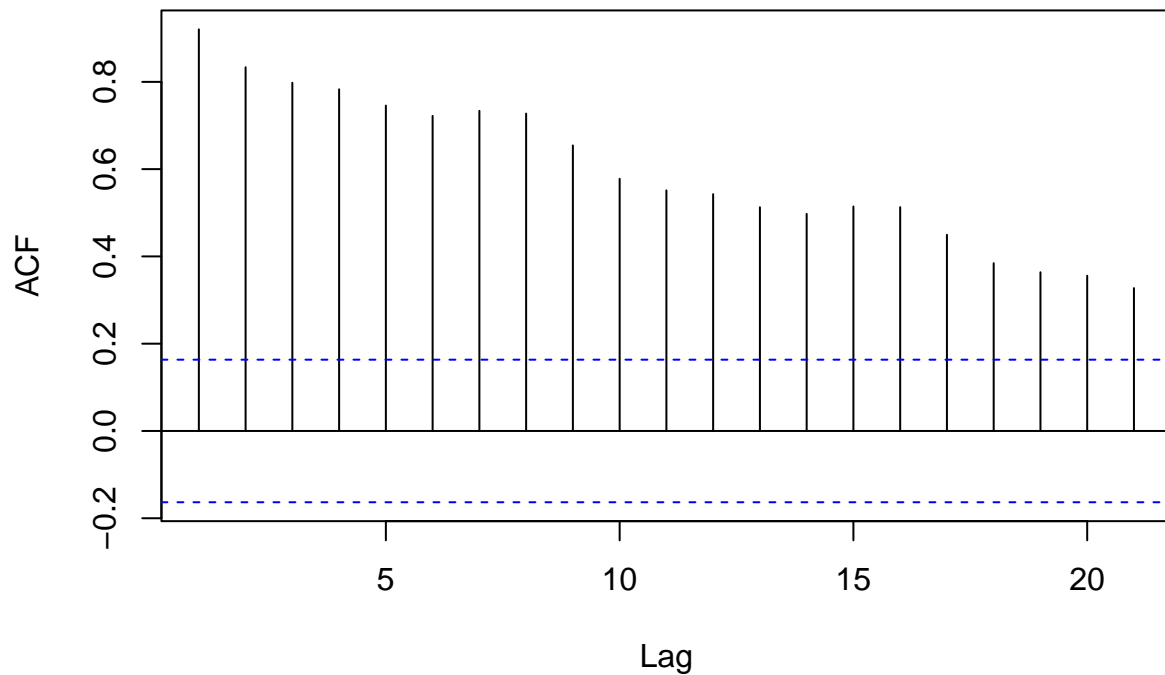
- The blue line is the fitted quadratic model for the time series data and we can see that the fit is good when compared to the linear model.
- The overall skeleton of the data is almost captured by the quadratic model - the curvy behaviour.

Fitting a Cyclic/Seasonal model

For fitting the model, we require the frequency of the data which is unavailable. We can use the Auto Correlation Function (ACF) plot of the time series to find the frequency of the data.

```
# Plotting ACF for time series data  
acf(share_price_ts, main = "ACF plot for Closing share price")
```


ACF plot for Closing share price



- From the plot, we can take the frequency of the time series data as number of lines between the peaks in the wave pattern. Here, we can see that we can take the frequency of the data as 8 as there are 8 lines between the peaks.
- Now, again we have to form a time series object considering the frequency as 8.

```
share_price_ts_freq <- ts(share_price$Share_Price,start = c(1, 1), end = c(18,8),
                          frequency = 8)
# Extracting the seasonal pattern of the data
season. <- season(share_price_ts_freq)

# Forming the seasonal model excluding the intercept
seasonal_model <- lm(share_price_ts_freq~season.-1)
summary(seasonal_model)
```

```
##
## Call:
## lm(formula = share_price_ts_freq ~ season. - 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -52.855  -3.670   6.255  10.167  16.053
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## season.Season-1    74.442      3.842   19.38  <2e-16 ***
```

```
## season.Season-2    78.373      3.842    20.40    <2e-16 ***
## season.Season-3    77.792      3.842    20.25    <2e-16 ***
## season.Season-4    76.109      3.842    19.81    <2e-16 ***
## season.Season-5    79.811      3.842    20.77    <2e-16 ***
## season.Season-6    77.199      3.842    20.09    <2e-16 ***
## season.Season-7    71.226      3.842    18.54    <2e-16 ***
## season.Season-8    68.322      3.842    17.79    <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.3 on 136 degrees of freedom
## Multiple R-squared:  0.9578, Adjusted R-squared:  0.9554
## F-statistic: 386.2 on 8 and 136 DF,  p-value: < 2.2e-16
```

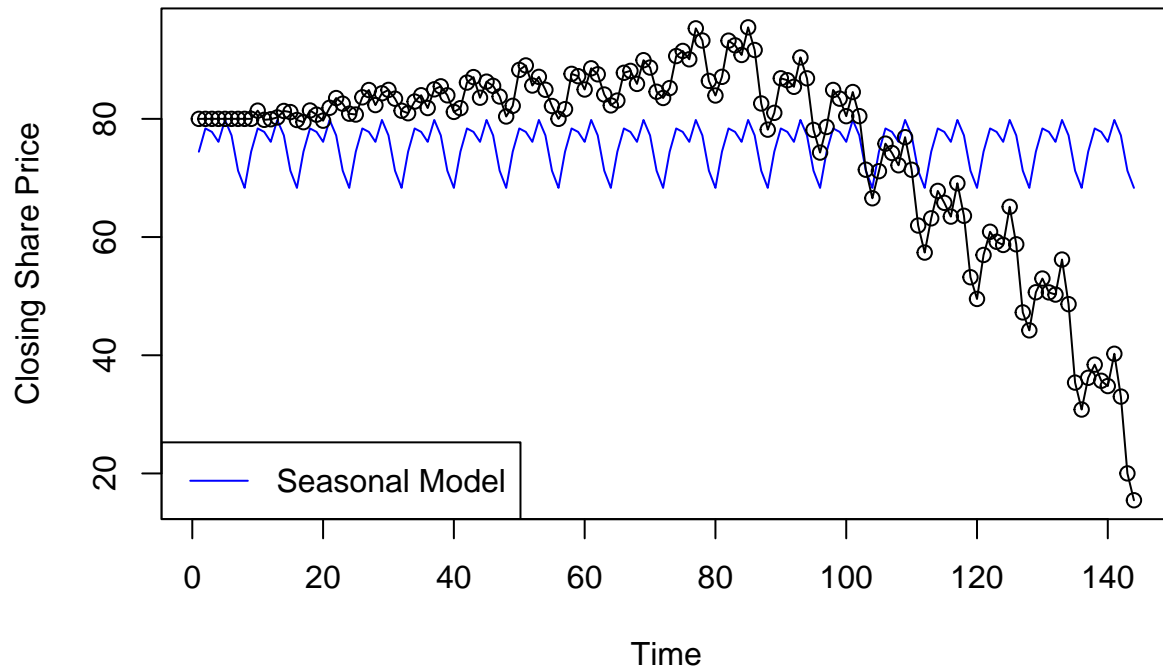
- Here for the model we are not considering the intercepts since we want the raw values for the coefficients. Otherwise we might get the values relative to the first season.
- The p-value for the all coefficients and overall model is falling below 0.05 and hence they are all significant which indicates the model is good.
- The R-squared value is very high (0.9554). The model here is too over fitting. If we are considering the model for forecasting it might be repeating the values which can lead to faulty forecasting.

We will try to plot the seasonal model on top of the time series plot to visualize the model.

```
# Function for plotting model on data
plot_model_on_data <- function(model, name){
  plot(ts(fitted(model)), xlab = "Time", ylab = "Closing Share Price",
        main = paste(name, "model fitting on Closing share prices"),
        ylim = c(min(c(fitted(model), as.vector(share_price_ts_freq))),
                  max(c(fitted(model), as.vector(share_price_ts_freq)))
        ), col = "blue" )
  lines(as.vector(share_price_ts), type="o")
  legend("bottomleft", legend = paste(name, "Model"), col = "blue", lty=1)
}

plot_model_on_data(seasonal_model, "Seasonal")
```

Seasonal model fitting on Closing share prices



- The plot is a flat and straight plot. Lot of values are not caught by the model.
- The major reason why the model is having a very high R-squared can be explained by the pattern of the model. The pattern in the model is almost the same pattern we are seeing in the original time series data.
- The plot is having pattern structure similar to that in the time series plot.
- The curvy nature of the data is not captured by the plot. So, the plot may not be a good fit for forecasting.

Fitting a Cosine model

Cosine model follows the following equation:

$$\mu_t = \beta_0 + \beta_1 \cos(2\pi ft) + \beta_2 \sin(2\pi ft), \text{ where } f \text{ is the frequency of the curve}$$

```
har. <- harmonic(share_price_ts_freq)
data <- data.frame(share_price_ts_freq, har.)
# Fitting the model
cosine_model <- lm(share_price_ts_freq ~ cos.2.pi.t. + sin.2.pi.t., data = data)
summary(cosine_model)
```

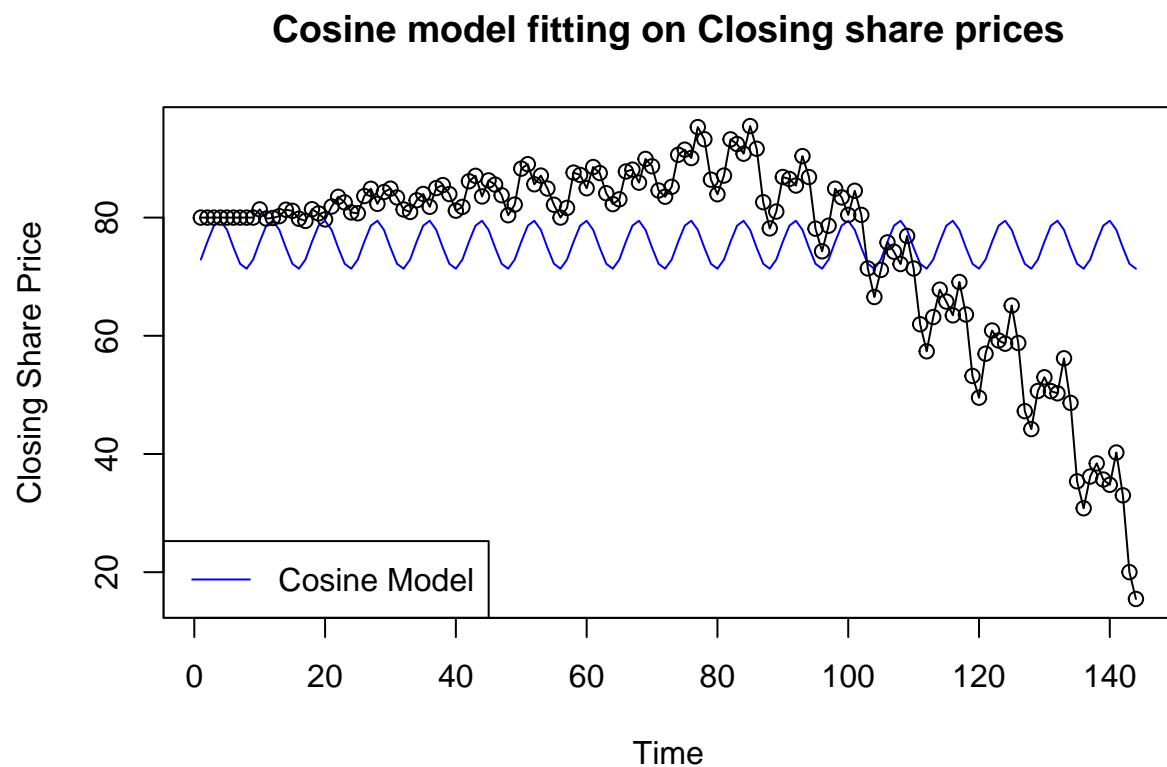
```
##
## Call:
## lm(formula = share_price_ts_freq ~ cos.2.pi.t. + sin.2.pi.t.,
##     data = data)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -55.886  -4.509   6.495  10.263  18.348
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   75.409      1.347   55.986 <2e-16 ***
## cos.2.pi.t.  -2.511      1.905   -1.318  0.1895
## sin.2.pi.t.   3.225      1.905    1.693  0.0926 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.16 on 141 degrees of freedom
## Multiple R-squared:  0.03163,    Adjusted R-squared:  0.01789
## F-statistic: 2.302 on 2 and 141 DF,  p-value: 0.1038
```

- The p-values are all well above the significance level 0.05.
- The R-squared value is very low (0.01789) indicating the model is not useful at all.
- The model is not good and not capturing any of the data points as per the model statistics.

We will try to plot the cosine model on top of the time series plot to visualize the model.

```
plot_model_on_data(cosine_model, "Cosine")
```



- The plot is a flat and straight plot. Lot of values are not caught by the model similar to that of the seasonal model.

- The plot is confirming the statistics we obtained.
- The curvy nature of the data is not captured by the plot. So, the plot is not a good fit for forecasting.

Model Comparison using Statistics

Table 1: Model comparison table

Model	R_squared	p_values
Linear	0.45130	All below 0.05
Quadratic	0.88070	All below 0.05
Seasonal	0.95540	All below 0.05
Cosine	0.01789	All above 0.05

- We can rule out linear and cosine models from the R-squared values itself as they are very low and not at all useful for modeling the data.
- When we compare quadratic and seasonal, we can see that seasonal model is too over fitting whereas quadratic is also little over fitting. But compared to seasonal quadratic is better.
- The nature of the fitted plots of quadratic and seasonal also suggests quadratic follows the curve very well but not the pattern.
- The advantage with seasonal is it is able to replicate almost the same pattern as present in the time series plot.

We will try to **combine quadratic and seasonal** model and try to see the fitted model. This is having a lot of parameters involved (parsimony) but as we are trying to see the model and analyse the details of the model.

Fitting combined Quadratic and Seasonal model

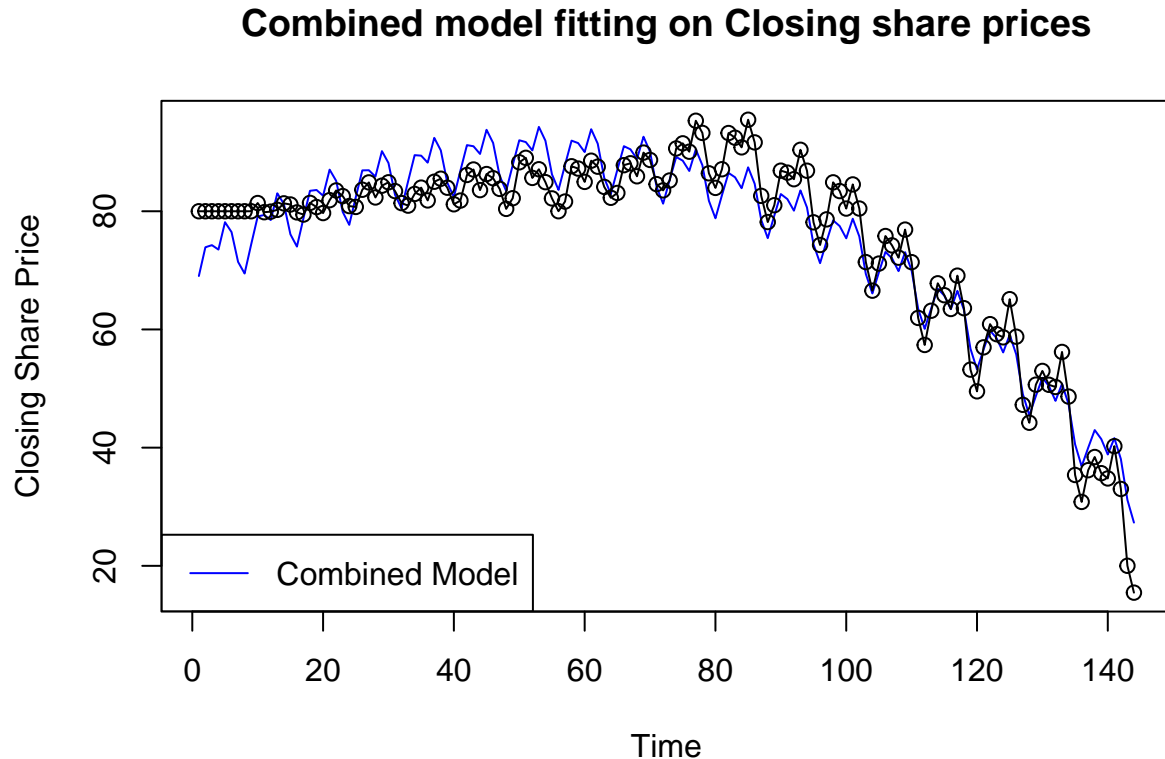
```
quad_seasonal_model <- lm(share_price_ts_freq~season.+periods+periods2-1)
summary(quad_seasonal_model)
```

```
##
## Call:
## lm(formula = share_price_ts_freq ~ season. + periods + periods2 -
##      1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.8532  -3.7024   0.2489   3.3269  10.9561
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## season.Season-1  68.3164881   1.5482844   44.12  <2e-16 ***
## season.Season-2  72.4677062   1.5530511   46.66  <2e-16 ***
## season.Season-3  72.1213720   1.5575753   46.30  <2e-16 ***
## season.Season-4  70.6863629   1.5618582   45.26  <2e-16 ***
## season.Season-5  74.6497810   1.5659013   47.67  <2e-16 ***
## season.Season-6  72.3139431   1.5697064   46.07  <2e-16 ***
## season.Season-7  66.6302852   1.5732758   42.35  <2e-16 ***
```

```
## season.Season-8 64.0294028 1.5766120 40.61 <2e-16 ***
## periods        0.7342655 0.0375483 19.55 <2e-16 ***
## periods2       -0.0068694 0.0002508 -27.39 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.651 on 134 degrees of freedom
## Multiple R-squared:  0.9966, Adjusted R-squared:  0.9964
## F-statistic: 3947 on 10 and 134 DF, p-value: < 2.2e-16
```

- Now all the p-values tell us the model is significant since all are well below the 0.05 significance level.
- The R-squared value is 0.9964 which is over fitting too much. So, there is no way we can use this model. It will just repeat the pattern as it has learned too much.

```
plot_model_on_data(quad_seasonal_model, "Combined")
```



From the plot we can understand that this is one of the best fit to the time series data. But, unfortunately as the model has learned too much, there is no use in using this model to do forecasting. We have just tried to see if the combined models can give us some good fit models.

Model Diagnostics

As the fitting of the models are done now we have to validate the models and finalise on the model which is best for making the predictions. We will follow the steps below for the validation of the models.

For the diagnostics part we are not considering the combined model since its too over fitting and a lot of parameters were involved.

We are plotting all the four plots for the different models together so the comparison of the plots is easier.

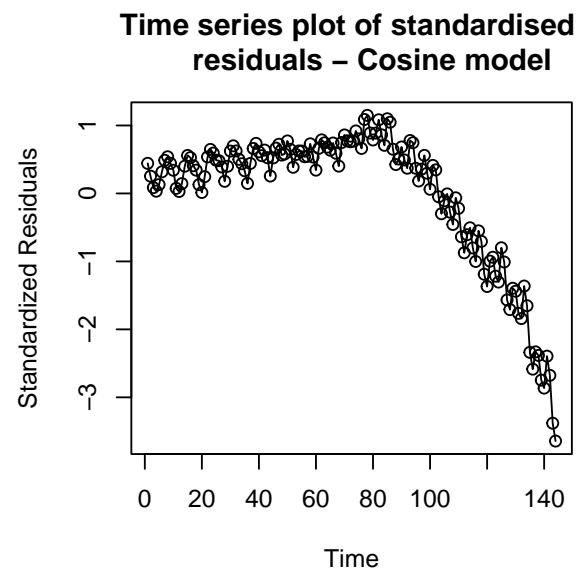
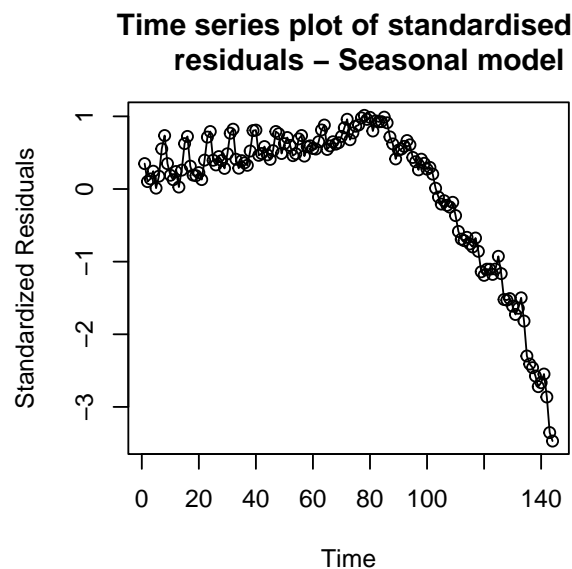
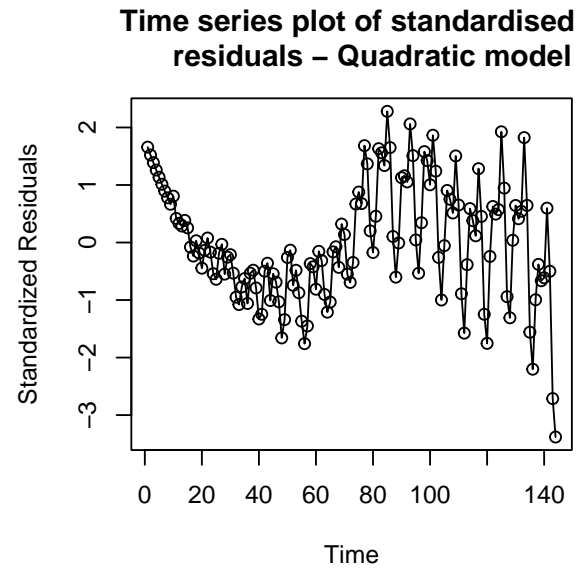
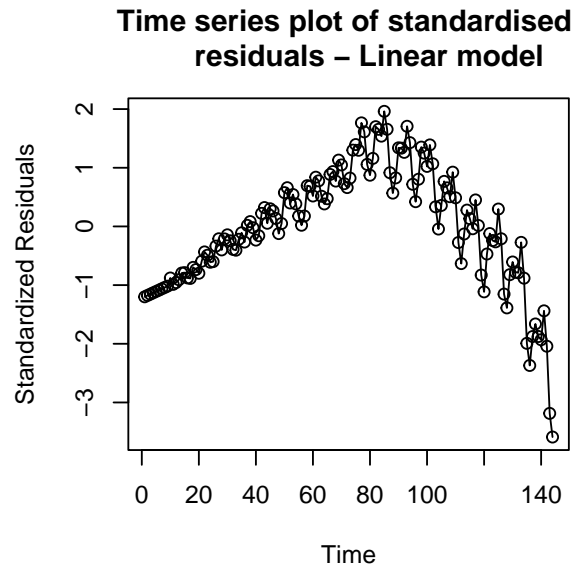
1. Time series plot of Standardized Residuals

We need model with a totally random pattern for the standard residuals plot. If the plot is not resembling white noise then it indicates that the complete information is not captured by the model and some information is lost in the residuals.

```
# Creating multiple plots.
par(mfrow=c(2,2))

plot_std_residuals <- function(model, name){
  plot(y=rstudent(model),x=as.vector(time(share_price_ts)), xlab='Time',
       ylab='Standardized Residuals',type='o', main = paste("Time series plot of standardised
       residuals -", name,"model"))
}

plot_std_residuals(linear_model, "Linear")
plot_std_residuals(quadratic_model, "Quadratic")
plot_std_residuals(seasonal_model, "Seasonal")
plot_std_residuals(cosine_model, "Cosine")
```



- From the plot, comparing with the original data time series plot we can see that except quadratic all others have plot similarity to the original time series plot.
- For the quadratic plot we can say that it is kind of random and is having less information when compared with the other models.

2. Histogram Analysis

For the best model we are hoping to see a symmetrical structure for the standardised residuals.

```
par(mfrow=c(2,2))

plot_hist_std_res <- function(model, name) {
  hist(rstudent(model), xlab='Standardized Residuals', ylab = "Frequency",
  main = paste("Histogram of standardised
```

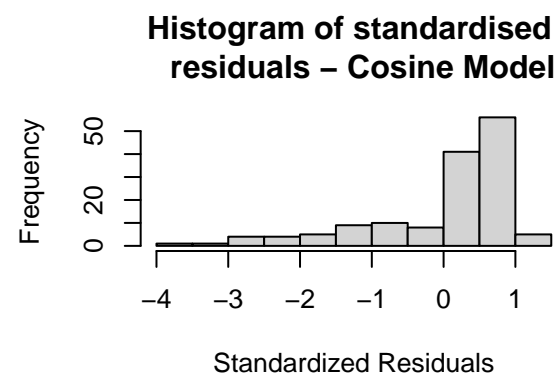
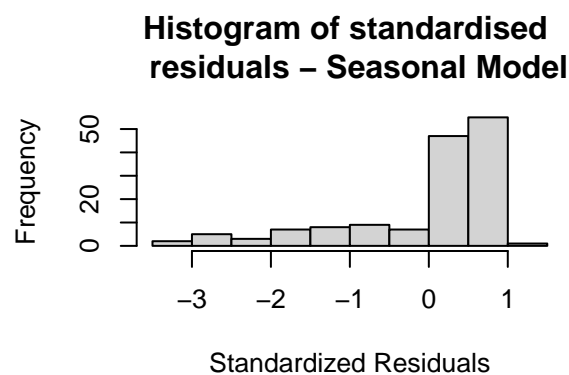
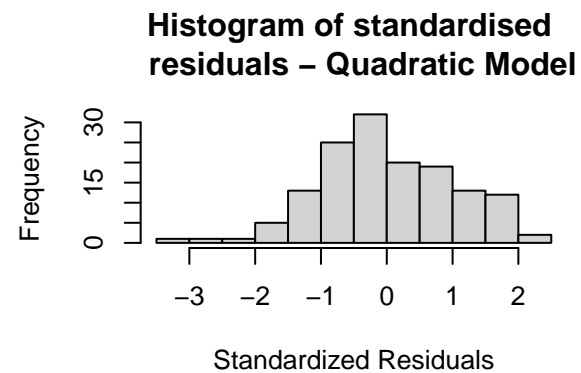
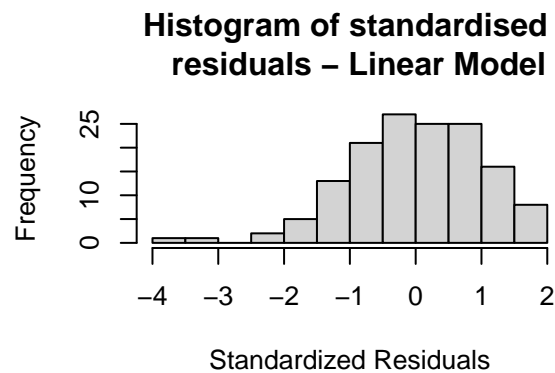


```

    residuals ~", name,"Model"))
}

plot_hist_std_res(linear_model, "Linear")
plot_hist_std_res(quadratic_model, "Quadratic")
plot_hist_std_res(seasonal_model, "Seasonal")
plot_hist_std_res(cosine_model, "Cosine")

```



- Standard residuals for all the models fall in the range -4 to +4 which is good.
- First look itself we can understand that the histogram is most symmetric for quadratic model compared with other models.
- For the cosine and seasonal model we can see huge left skewness and not at all symmetric.
- For linear model, we have slight left skewness, but a lot of values are concentrated on the right side and hence symmetry is less when compared with quadratic model.

3. Q-Q Plot of Standardized Residuals

The models for which the standardized residuals are following a normal distribution will be a good fitting model. The normality of the residuals are checked using Q-Q plot.

```

par(mfrow=c(2,2))

plot_qq_std_res <- function(model, name){
  std_res = rstudent(model)
  qqnorm(std_res, main = paste("QQ plot of standardised residuals

```

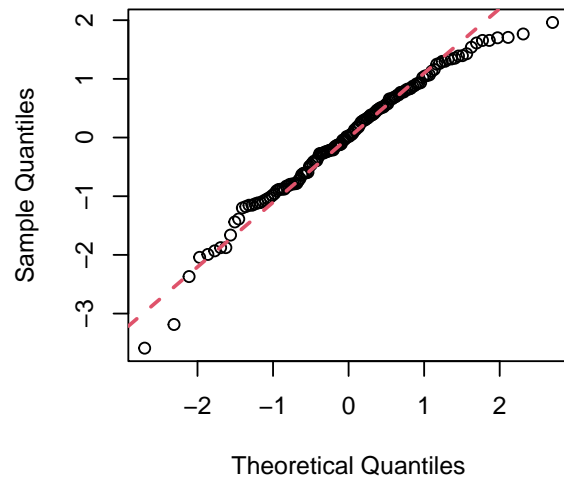
```

    "-", name,"model"))
  qqline(std_res, col = 2, lwd = 2, lty = 2)
}

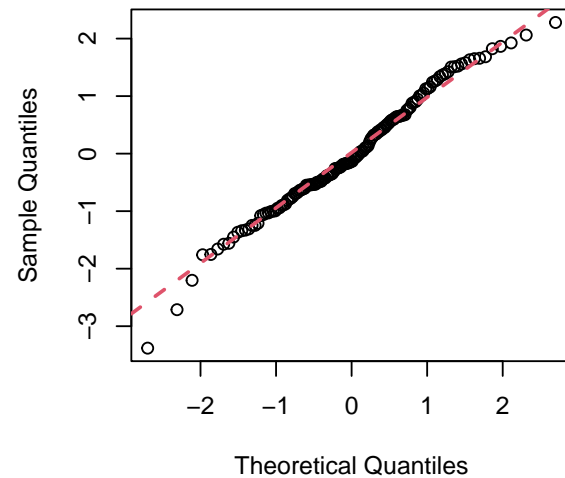
plot_qq_std_res(linear_model, "Linear")
plot_qq_std_res(quadratic_model, "Quadratic")
plot_qq_std_res(seasonal_model, "Seasonal")
plot_qq_std_res(cosine_model, "Cosine")

```

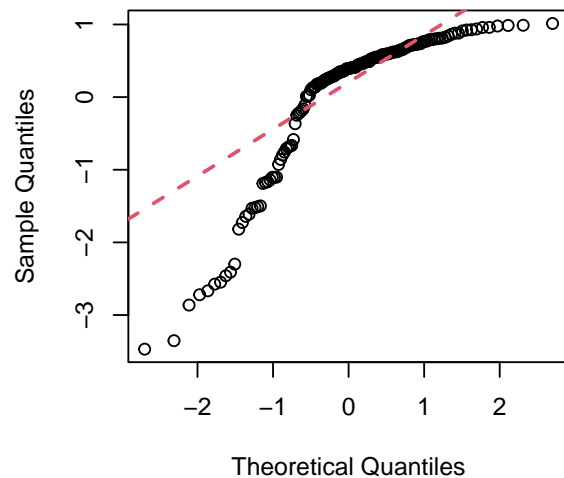
**QQ plot of standardised residuals
– Linear model**



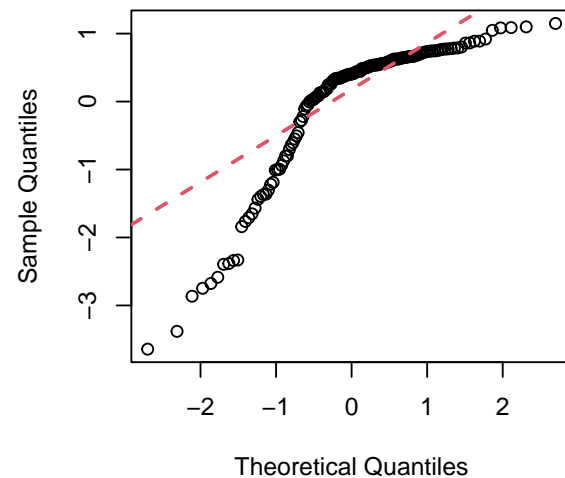
**QQ plot of standardised residuals
– Quadratic model**



**QQ plot of standardised residuals
– Seasonal model**



**QQ plot of standardised residuals
– Cosine model**



- The plots clearly show that there is significant deviation from normality for the residuals from seasonal and cosine models at the beginning of the plots.
- When we compare linear and quadratic we can see that normality is more followed by the residuals from quadratic model. At the end, we can see deviation from normality for the residuals from the

linear model.

4. Shapiro - Wilk Test

The test can be used for ensuring normality for the standardized residuals. For the test we go with the following null hypothesis:

H_0 : Data is distributed normally.

So, we need to get a value higher than the significance level 0.05 to not to reject the null hypotheses.

```
# For linear model
shapiro.test(rstudent(linear_model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rstudent(linear_model)
## W = 0.97653, p-value = 0.01413
```

```
# For quadratic model
shapiro.test(rstudent(quadratic_model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rstudent(quadratic_model)
## W = 0.98771, p-value = 0.2324
```

```
# For Seasonal model
shapiro.test(rstudent(seasonal_model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rstudent(seasonal_model)
## W = 0.79197, p-value = 5.03e-13
```

```
# For Cosine model
shapiro.test(rstudent(cosine_model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rstudent(cosine_model)
## W = 0.79165, p-value = 4.907e-13
```

- From the shapiro wilk test results we can see that only for quadratic model we have a value greater than 0.05.
- So, the test concludes that only residuals from quadratic model follows normal distribution

5. ACF Plot

The auto correlation function plot for the residuals of different models tells us how much information is lost by the model in the residuals. We will be having the confidence levels shown in the ACF plot. if no information is lost all of the lines should be coming inside the confidence levels. The more the number of lines crossing the confidence levels, more the information lost.

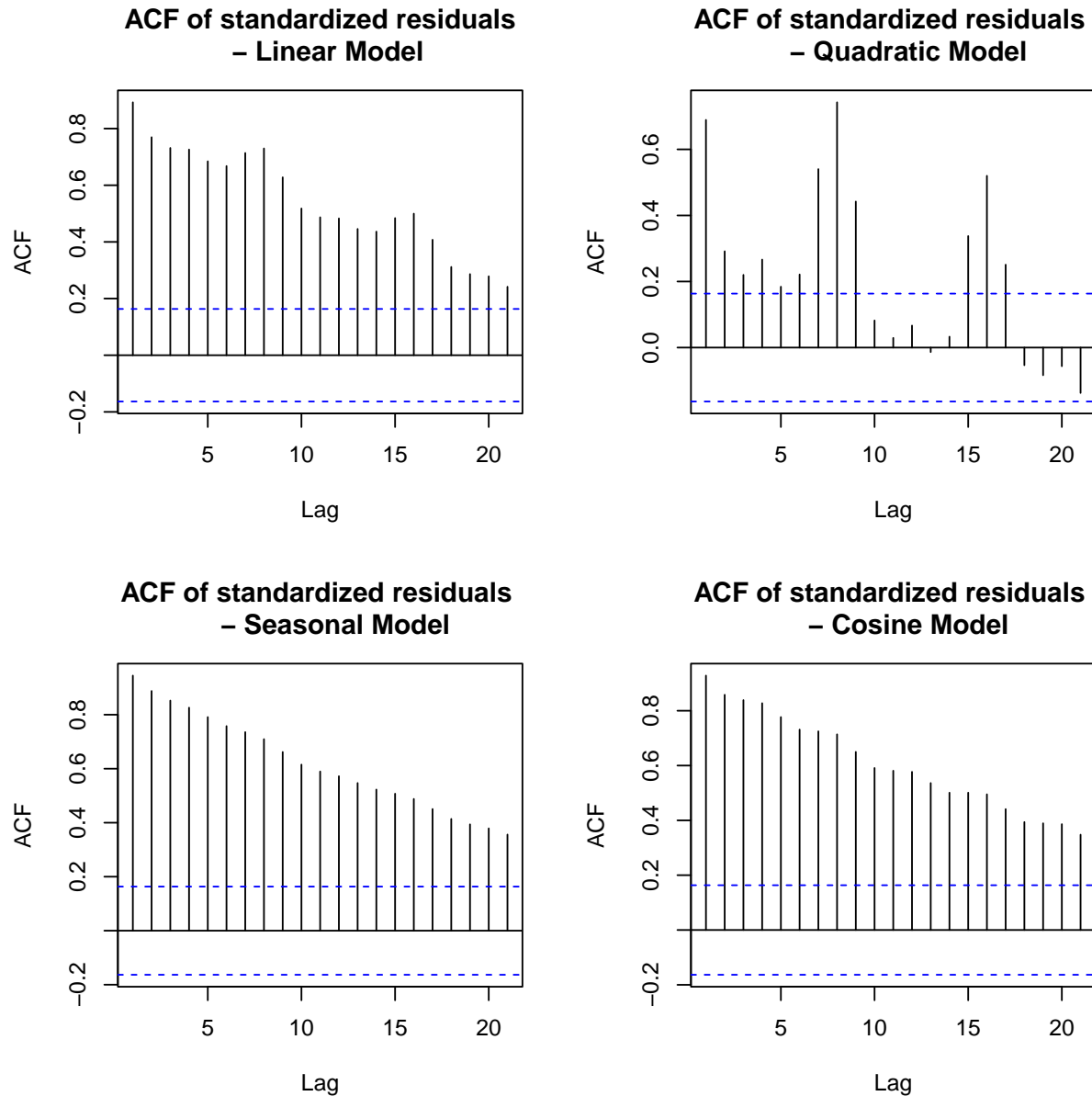
```
par(mfrow=c(2,2))

acf(rstudent(linear_model), main = "ACF of standardized residuals
- Linear Model")

acf(rstudent(quadratic_model), main = "ACF of standardized residuals
- Quadratic Model")

acf(rstudent(seasonal_model), main = "ACF of standardized residuals
- Seasonal Model")

acf(rstudent(cosine_model), main = "ACF of standardized residuals
- Cosine Model")
```



- Compared with all the models we can see that the number of lines crossing the confidence levels is very less for the quadratic model. So, among all the models quadratic model was able to catch more information of the data.
- For all the models except quadratic model we can see that the auto correlation at different lags are all crossing the confidence boundary which indicates that all those models were able to catch very few information about the time series data.

Results

- The model fitting and model diagnostics performed indicates that we have to go with quadratic model for the forecasting of the time series data.
- Quadratic model is the best model for this use case when compared with other models. But, we cannot say quadratic model will be the best one because there might be lot of other models which can give

far better results than the quadratic model.

```
forecast_periods <- data.frame( periods = seq(145, 149, 1), periods2 = seq(145, 149, 1)^2)
forecast_data <- predict(quadratic_model, newdata = forecast_periods,
                        interval = "prediction")
print(forecast_data)
```

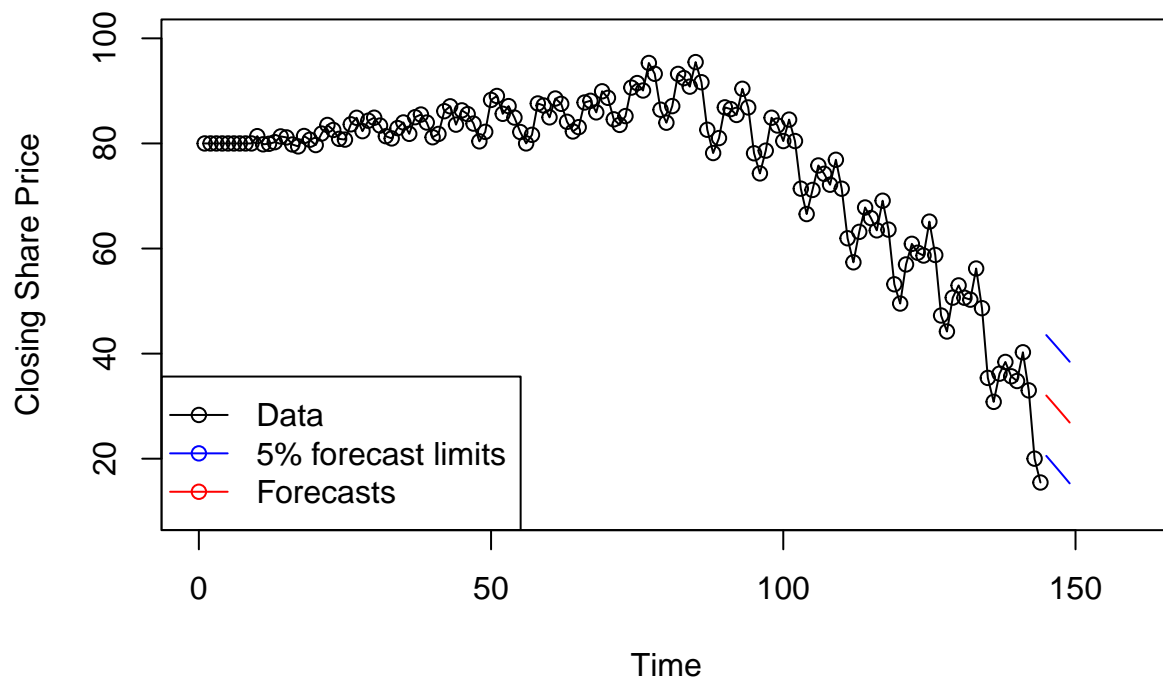
```
##      fit      lwr      upr
## 1 32.03113 20.53974 43.52253
## 2 30.76371 19.25266 42.27477
## 3 29.48255 17.95092 41.01417
## 4 28.18763 16.63449 39.74077
## 5 26.87896 15.30335 38.45458
```

The fit column gives us the prediction made by the models and the other columns are covering the 95% confidence interval levels.

Now, we will try to plot the forecast data onto the original time series plot.

```
# Plotting the forecast data on the original time series plot.
plot(share_price_ts, type = "o", xlim = c(0, 160), ylim = c(10, 100),
     main = "Forecast Data on Time Series plot", ylab = "Closing Share Price")
lines(ts(as.vector(forecast_data[,3]), start = 145), col="blue", type="l")
lines(ts(as.vector(forecast_data[,1]), start = 145), col="red", type="l")
lines(ts(as.vector(forecast_data[,2]), start = 145), col="blue", type="l")
legend("bottomleft", lty=1, pch=1, col=c("black", "blue", "red"),
      c("Data", "5% forecast limits", "Forecasts"))
```

Forecast Data on Time Series plot



- The forecast data here gives a possible range of values where the actual values may lie in.

Conclusion

- Forecasting data is just one of the steps which happen to be showcasing all the efforts done in the other steps
- There are a lot of models available and we have tried the modeling strategy with a handful of models.
- Based on the complexity of the time series data, we can even combine multiple models for finding a best fit for the data.
- For the data given, quadratic model happen to be the best fit. Even though we did try combining multiple models to see the results, we got a model which has learnt so much.
- An over fitted model can give faulty forecasts and hence its better not to consider models having R-squared values more that 0.9.
- Model building strategy has to be followed for every modeling problem.
- Model diagnostic analysis plays a key role in determining the model to be chosen for forecasting.

References

- Haydar Demirhan, 2022, ‘Week 1 & 2 materials’, viewed 20th to 27th March 2022, file:///C:/RMIT/Semester%201%202022/Time%20Series/Module%201/Module%201%20-%20Online%20Notes.html
- Jonathan D. Cryer, Kung-Sik Chan 2008, *Time Series Analysis With Applications in R*, Springer, USA