Analysis of a Share Market Trader Investment Portfolio

Student Name: Mrwan Alhandi

Student ID: 3969393

Finished On: 22/03/2023

Libraries and functions

```
library(TSA)
```

```
##
## Attaching package: 'TSA'
```

```
## The following objects are masked from 'package:stats':
##
##     acf, arima
```

```
## The following object is masked from 'package:utils':
##
##     tar
```

```
# Plot and output all residual analysis
residual_analysis = function(model) {
  res.model = rstudent(model)
  par(mfrow=c(2,2))
  plot(y = res.model, x = as.vector(time(data_ts)),xlab = 'Time', ylab='Standardized Residual
s',type='l',main = "Standardised residuals.")
  hist(res.model,xlab='Standardized Residuals', main = "Histogram of standardised residual
s.")
  qqnorm(y=res.model, main = "QQ plot of standardised residuals.")
  qqline(y=res.model, col = 2, lwd = 1, lty = 2)
  shapiro.test(res.model)
  acf(res.model, main = "ACF of standardized residuals.")

}

# Plotting forecasting functions at the end of the document
plotting_forecast = function(string) {
  plot(ts(data_ts_freq_7), xlim = c(1,200), ylim = c(-100, 200), ylab = "y", xlab = 't', main
= string)
lines(ts(as.vector(forecasts[,1]), start = c(127,1)), col="red", type="l")
lines(ts(as.vector(forecasts[,2]), start = c(127,1)), col="blue", type="l")
lines(ts(as.vector(forecasts[,3]), start = c(127,1)), col="blue", type="l")
legend("bottomleft", lty=1, pch=1, col=c("black","blue","red"), text.width = 18,
c("Data","5% forecast limits", "Forecasts"))
}

# adding main title to residual plot
adding_main_title = function(string) {
  mtext(string,                   # Add main title
      side = 4,
      line = - 2,
      outer = TRUE)
}

# Plotting the fitted model
plotting_model = function(model,string) {
  plot(ts(fitted(model)), col = "red", lty = 2, ylab = 'y', xlab = 't', main = string, ylim =
c(-70,170))
  points(data_ts)
  legend(1,95,legend=c("Original Data","Fitted Model"), col = c("black","red"), lty = 1:2, ce
x=0.8)
}
```

Defenitions and Essentials

In time series, we build a model based on only time. Whereas In other machine learning models, we have many features/independent variables and by using some optimization algorithms such as gradient descent, its possible to find the best parameters for each feature.

A stochastic process is a collection of random variables where each random variable has a probability of occurrence. There is discrete and continuous stochastic processes. If the time variable can only include integers, then its discrete. If it can include decimals, then its continuous. Stochastic processes such as random walk, white noise, random cosine wave and moving average can be discrete.

One of the most important concepts about any time series is stationary. Stationary is that the probability laws that govern the behavior of the process do not change over time. In simple words, the distribution of certain windows of time in the data only depend on that window of time and not in the location of time. If a time series process is stationary, then this allows us to make statistical inferences about its structure.

There are five key ideas when analyzing a time series:

```
1- Trend: Does the dependent variable tend to increase or decrease as time increase?

2- Seasonality: Is there a repeating patterns for the dependent variable as time increase wit
h specific amount?

3- Changing Variance: Are there change in fluctuations as time increase?

4- Behavior: Does the data follow a specific process? such as auto regressive, white noise, m
oving average or random walk?

5- Changing point: Is there a jump up or down in the dependent variable?
```

Goal and Methodology

In this report, a deterministic trend approach to capture the trend of the analysis of a share market trader investment portfolio data and fit an appropriate model to be able to forecast. The deterministic approach cannot capture the randomness of data nor the autocorrelation because this is a consequence of having a stochastic trend in the data. To explain this further, let us understand equation (1):

$$Y_t = \mu_t + X_t \tag{1}$$

Where:

Y_t is the dependent variable at time t. (Information in the data)

u_t is the deterministic trend or the mean function at time t. (Information captured by the model)

X_t is the randomness or residuals at time t. (Information not captured by the model + uncontrollable randomness)

The goal is that the mean function capture all the five key ideas (if exists in the series) and to not leave it to the residuals without over fitting or under fitting. If The residuals contains trend, seasonality, changing variance, behavior or changing point, then the mean function does not represent the data well. If any assumptions is made, then this can be disregarded.

Randomness cannot be identified and so we approximate it by manipulating the previous equation:

$$X_t = Y_t - \mu_t \tag{2}$$

Data Retreival

```
# reading data
data = read.csv('C:/Users/User/Desktop/Time series - assignment 1/assignment1Data2023.csv', h
eader = TRUE)
# changing column names
colnames(data) = c('day','investment')
```

```
# start of the data
head(data)
```

```
##   day investment
## 1   1   150.8031
## 2   2   152.2122
## 3   3   151.4941
## 4   4   150.6815
## 5   5   151.0155
## 6   6   151.0590
```

```
# end of the data
tail(data)
```

```
##     day investment
## 122 122  -16.29602
## 123 123  -35.55476
## 124 124  -44.62195
## 125 125  -17.30735
## 126 126   25.25656
## 127 127   43.71448
```

```
# statistical summary
summary(data)
```

```
##       day          investment
##  Min.   :  1.0   Min.   :-44.62
##  1st Qu.: 32.5   1st Qu.: 21.32
##  Median : 64.0   Median : 71.97
##  Mean   : 64.0   Mean   : 71.35
##  3rd Qu.: 95.5   3rd Qu.:135.90
##  Max.   :127.0   Max.   :156.32
```

Data preparation

In time series, outliers can describe jumps and other interesting characteristic of the series behavior. Outliers should not be removed from a time series.

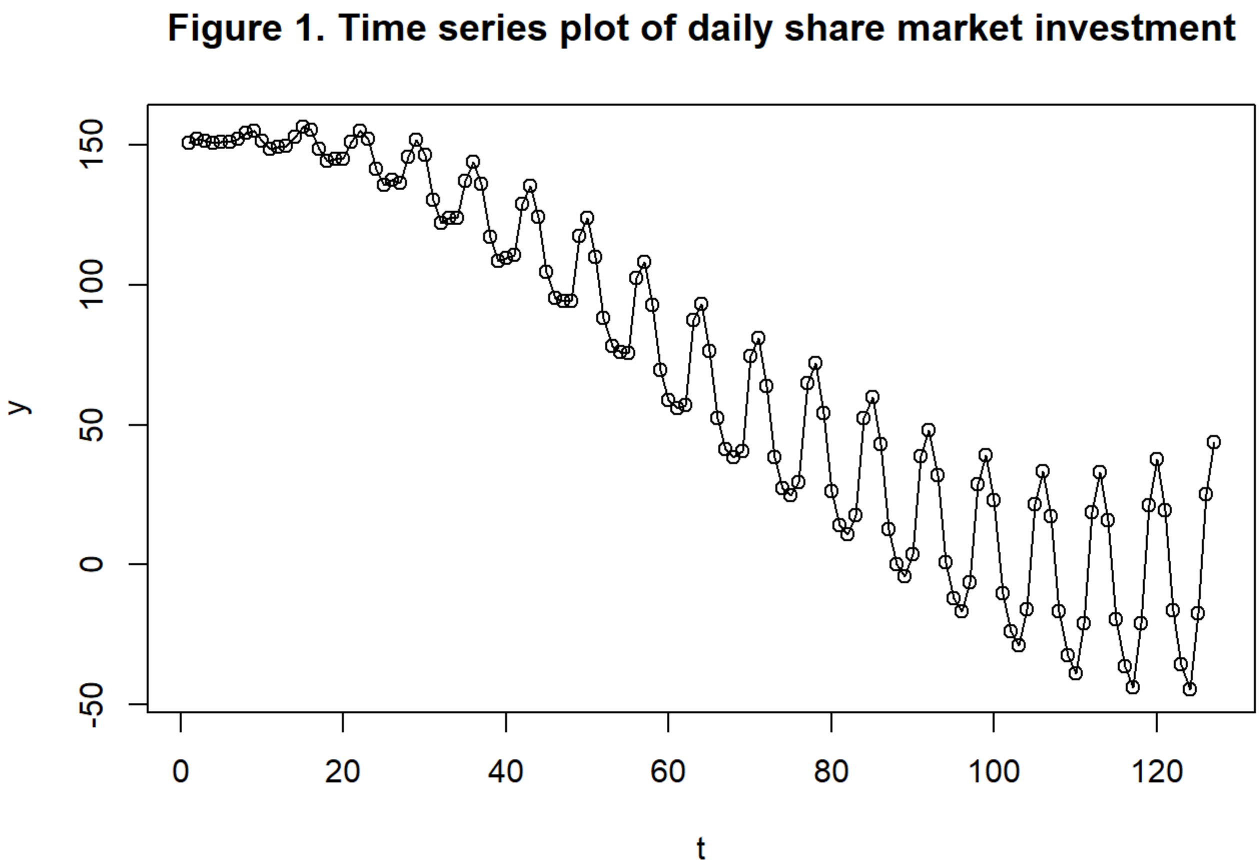
```
# Any missing values?
sum(is.na(data))
```

```
## [1] 0
```

```
# Converting data to a time series object
data_ts = ts(data[,2])
```

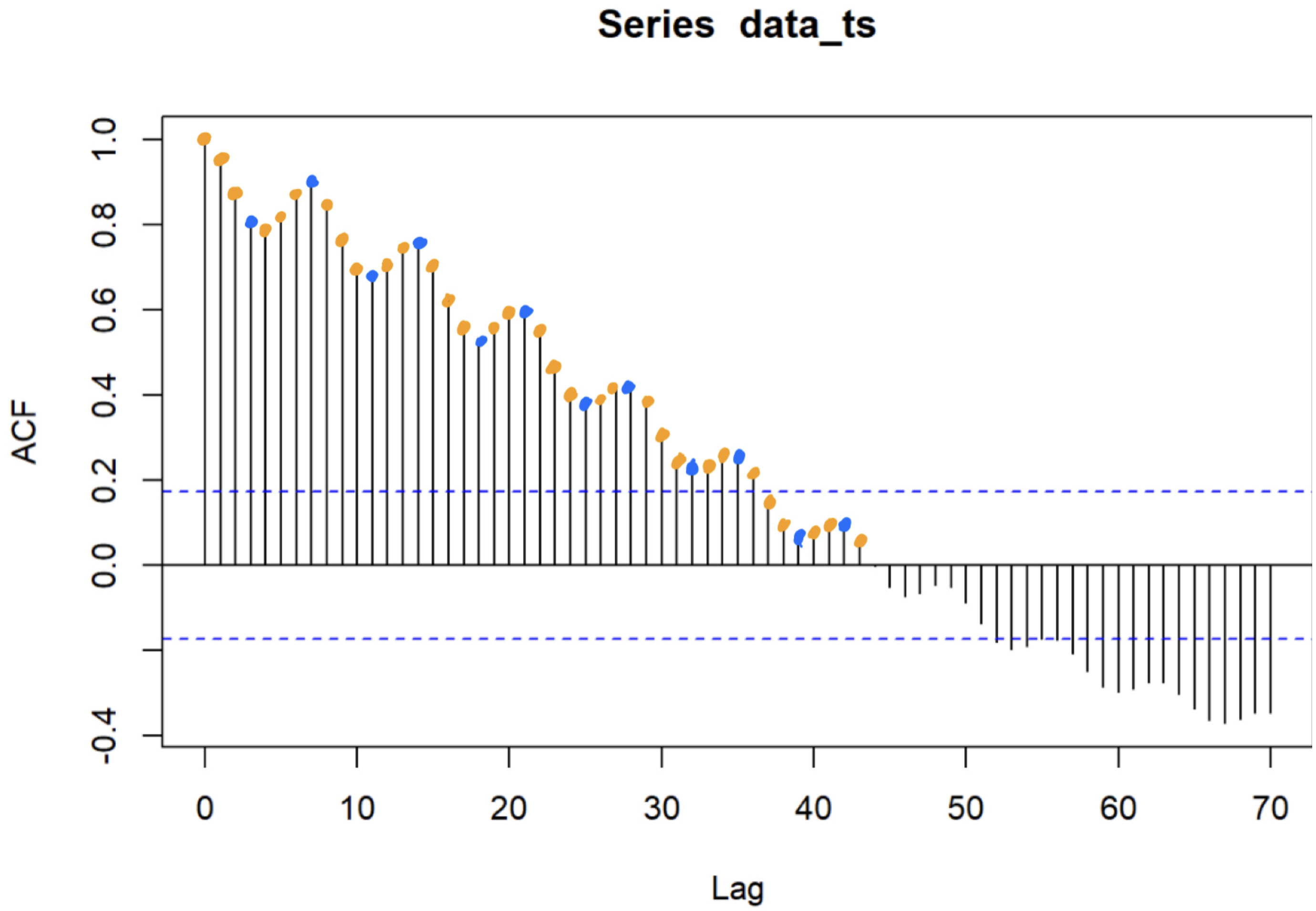
Data exploration

```
# Plotting
plot(data_ts, type = 'o',ylab = 'y',xlab = 't', main = 'Figure 1. Time series plot of daily s
hare market investment')
```



1- Trend: There is an obvious downward trend as days increase, but at the end it seems like it will increase. Perhaps like a parabolic shape.

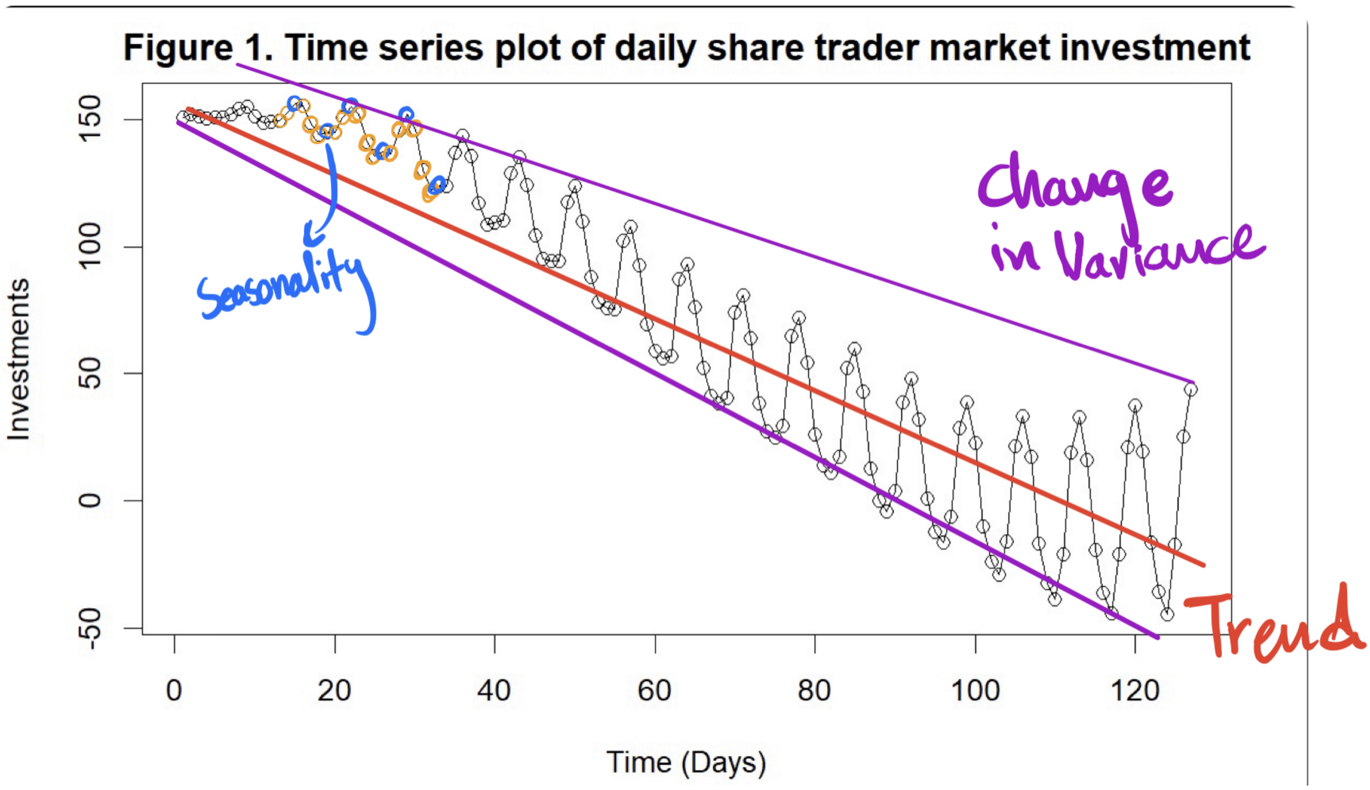
2- Seasonality: The seasonality can be observed using ACF. The frequency is 7.



3- Changing Variance: There is an increase in variance around the mean as days increase.

4- Changing point: There is no changing point.

5- Behavior: The series seems to have an auto regressive behavior. From the ACF, as lags increases, there are no significant drop. The effect of each point last longer.



Data modeling

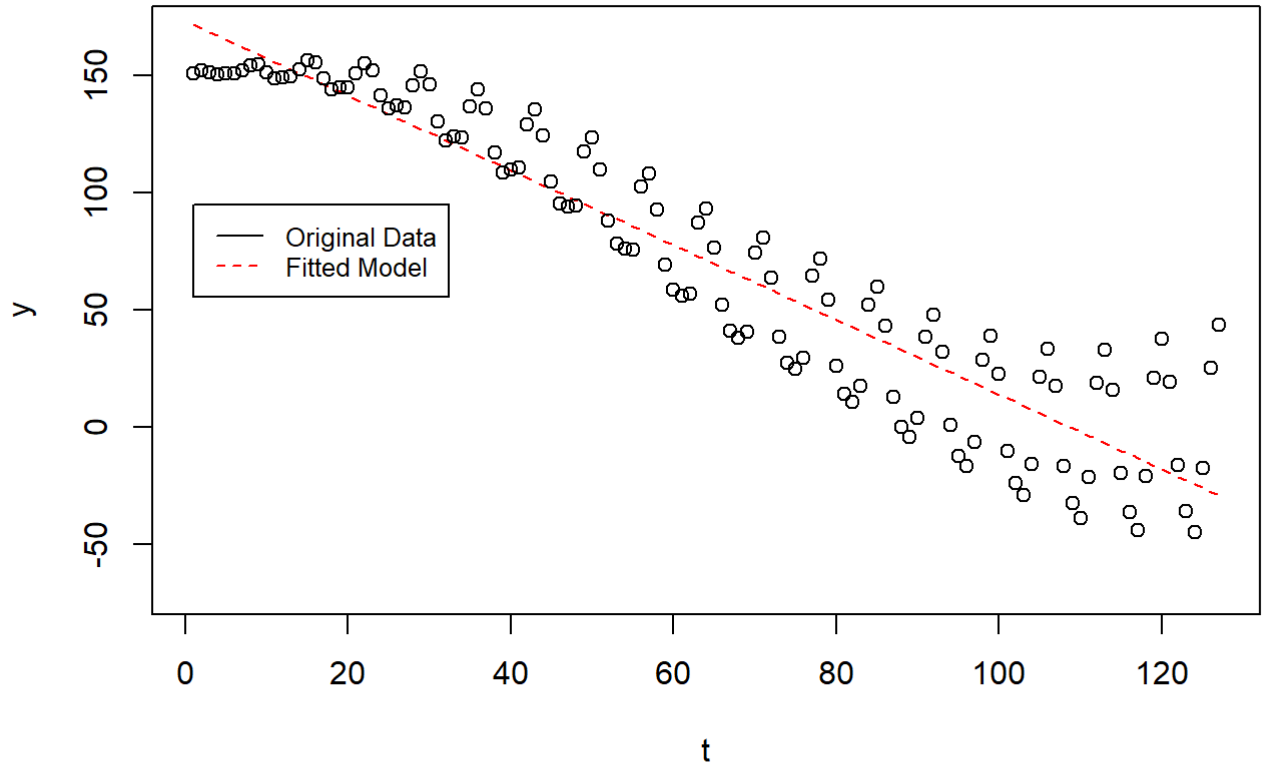
Linear model

The equation (3) represents a linear model:

$$\mu_t = \beta_0 + \beta_1 t \tag{3}$$

```
# Accessing time variable and building the linear model
t = time(data_ts)
linear_model = lm(data_ts~t)
```

Figure 2. Fitted linear model



- The linear model at least capture the trend.

```
# Model analysis
summary(linear_model)
```

```
##
## Call:
## lm(formula = data_ts ~ t)
##
## 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) 173.38533    3.85513   44.98   <2e-16 ***
## t            -1.59425    0.05227  -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
```

- Median of residuals is not around 0. The range of residuals is around 110.
- We do not reject the hypothesis of coefficients.
- Adjusted R-squared is between 0.8 and 0.9.
- Std error is low and estimate is good. t is significant.
- Overall fit is significant.
- Even though all the statistics show that it is a good model, let us see analyse the error as seen in equation (4) and see if this is the case.

$$X_t = Y_t - (\beta_0 + \beta_1 t) \tag{4}$$

```
# Residuals analysis
residual_analysis(linear_model)
adding_main_title("Figure 3-6. Linear Residuals Analysis")
```

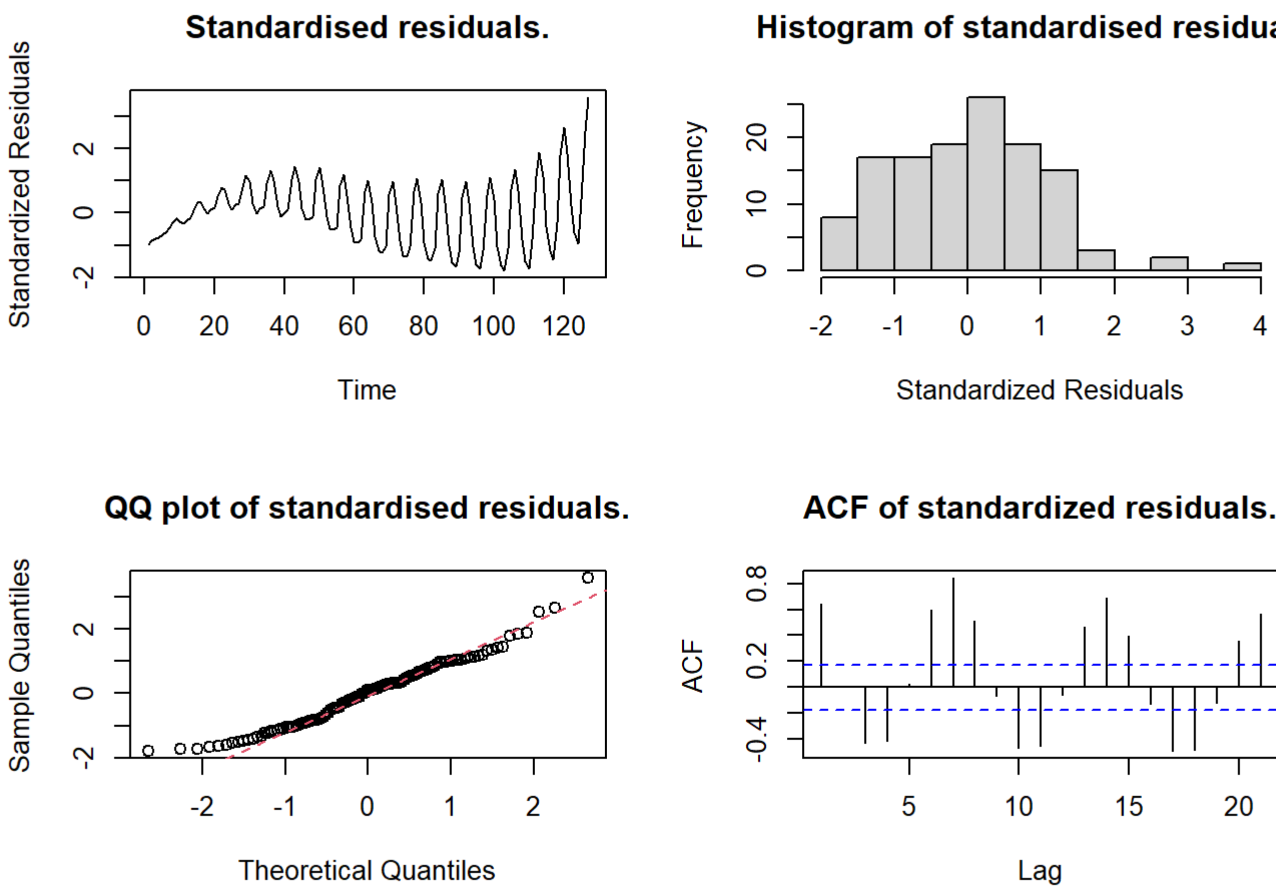


Figure 3-6. Linear Residuals Analysis

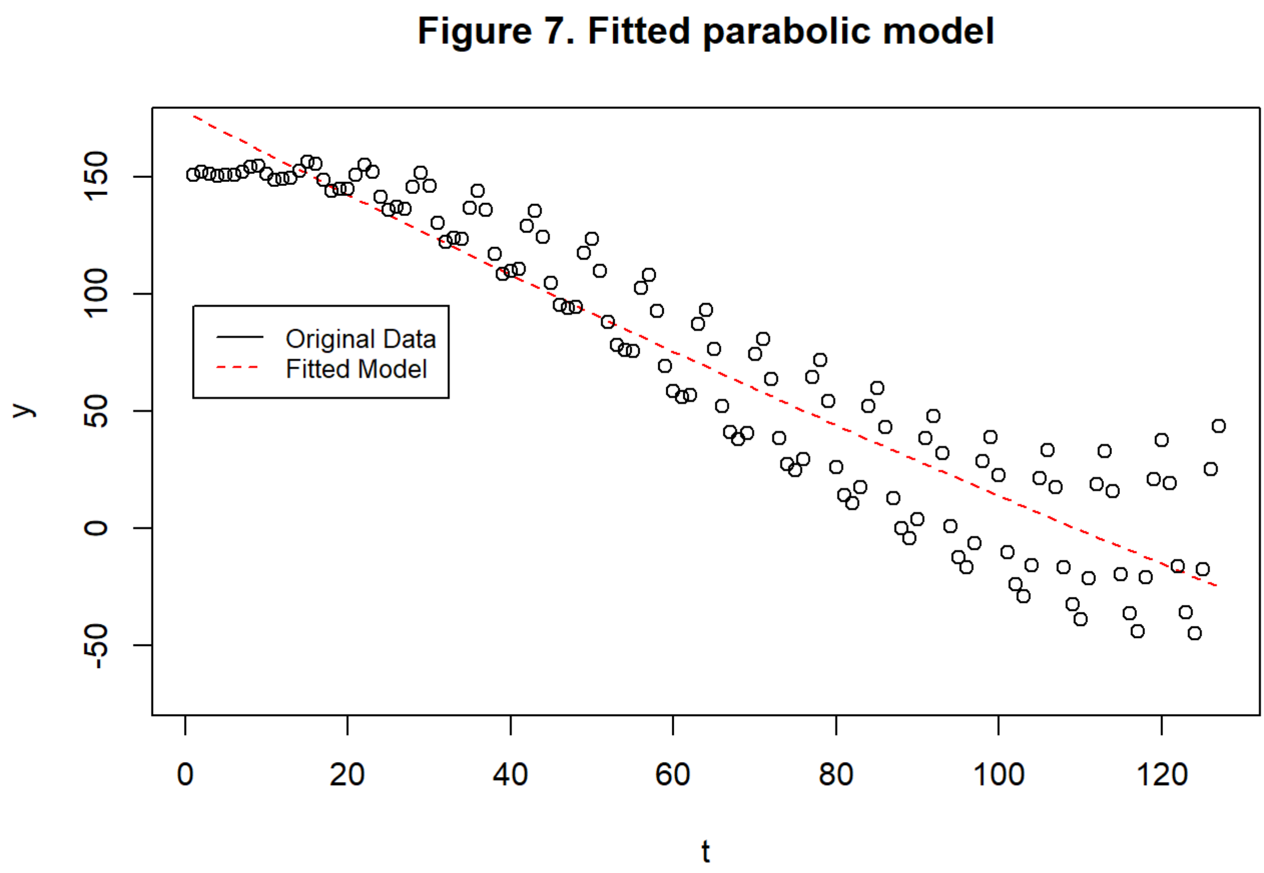
- Standardized residuals from linear model: The residuals are capturing the trend, behavior,changing variance and seasonality. This means the linear model is bad because the residuals are not random.
- Histogram of standardized residuals: right-skewed distribution which does not indicate normality.
- QQ plot of standardized residuals: Some points at the two ends are not close to the line.
- ACF of standardized residuals: Some lines are crossing the confidence interval. This means that the model is not capturing the autocorrelation.
- Sharpiro wilk: p-value < 0.05 which indicates that residuals distribution is not random/normal.

A parabolic model might do better, since there seems to be a downward then upward shape? Equation (5) can describe a parabolic shape.

Parabolic model

$$\mu_t = \beta_0 + \beta_1 t + \beta_2 t^2 \tag{5}$$

```
# Assigning variables and building the linear model
t2 = t^2
parabolic_model = lm(data_ts~t2+t)
```



- There does not seem a significant difference visually from graph between linear and parabolic.

```
##
## Call:
## lm(formula = data_ts ~ t2 + t)
##
## 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) 177.915494   5.838752  30.471  < 2e-16 ***
## t2            0.001646   0.001594   1.033    0.304
## t            -1.804958   0.210586  -8.571 3.43e-14 ***
## ---
## 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
```

- Median is better than linear since it is closer to 0. Slight improvement in the range of residuals.
- t2 is insignificant and it estimate is low.
- No improvement in R-squared.
- t2 seems to have no improvement, but let us check the residuals analysis to confirm:

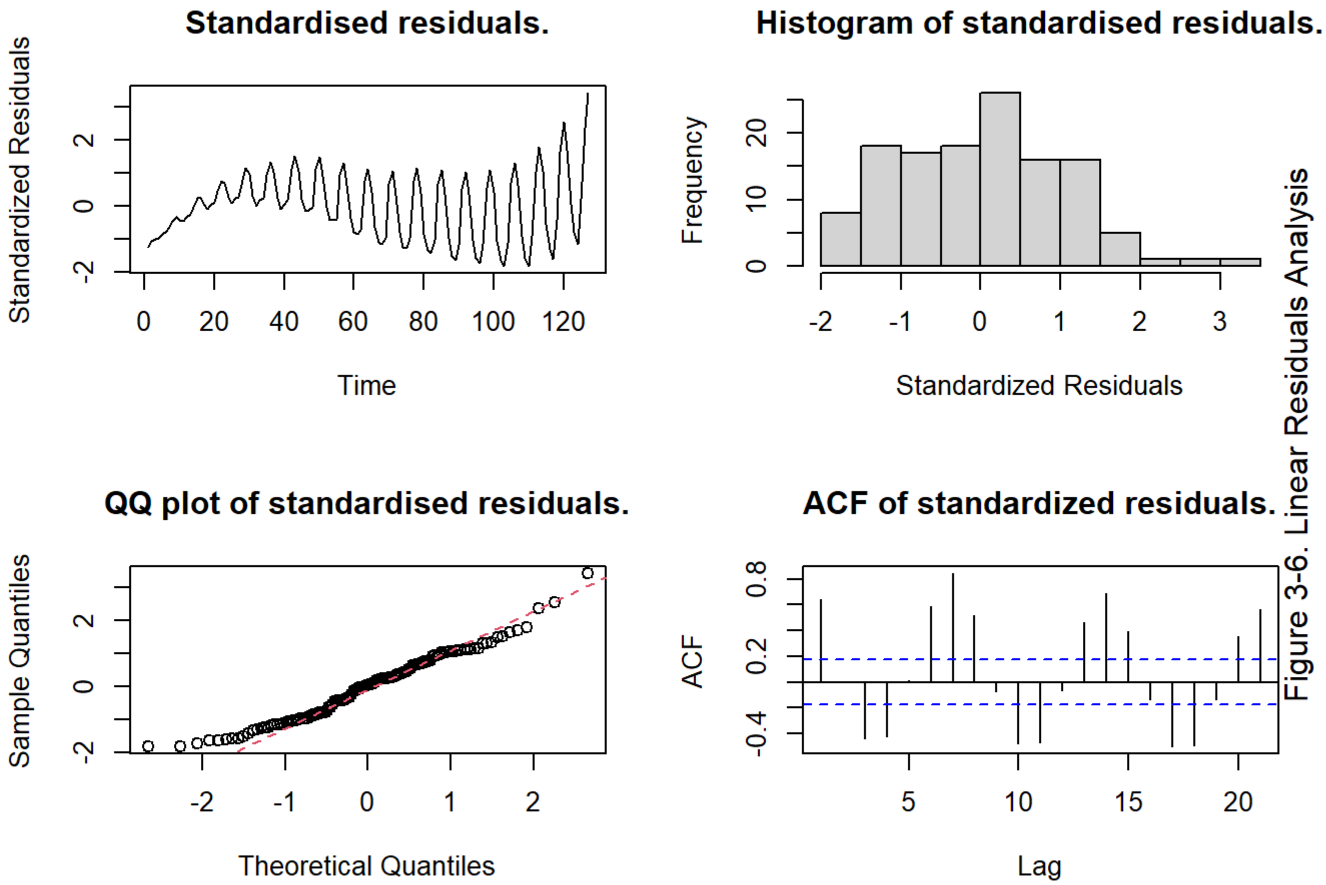


Figure 3-6. Linear Residuals Analysis

- As expected, no improvement in adding t2

But, what happens as the power increases? Equation (6) can describe this.

High Polynomial fit

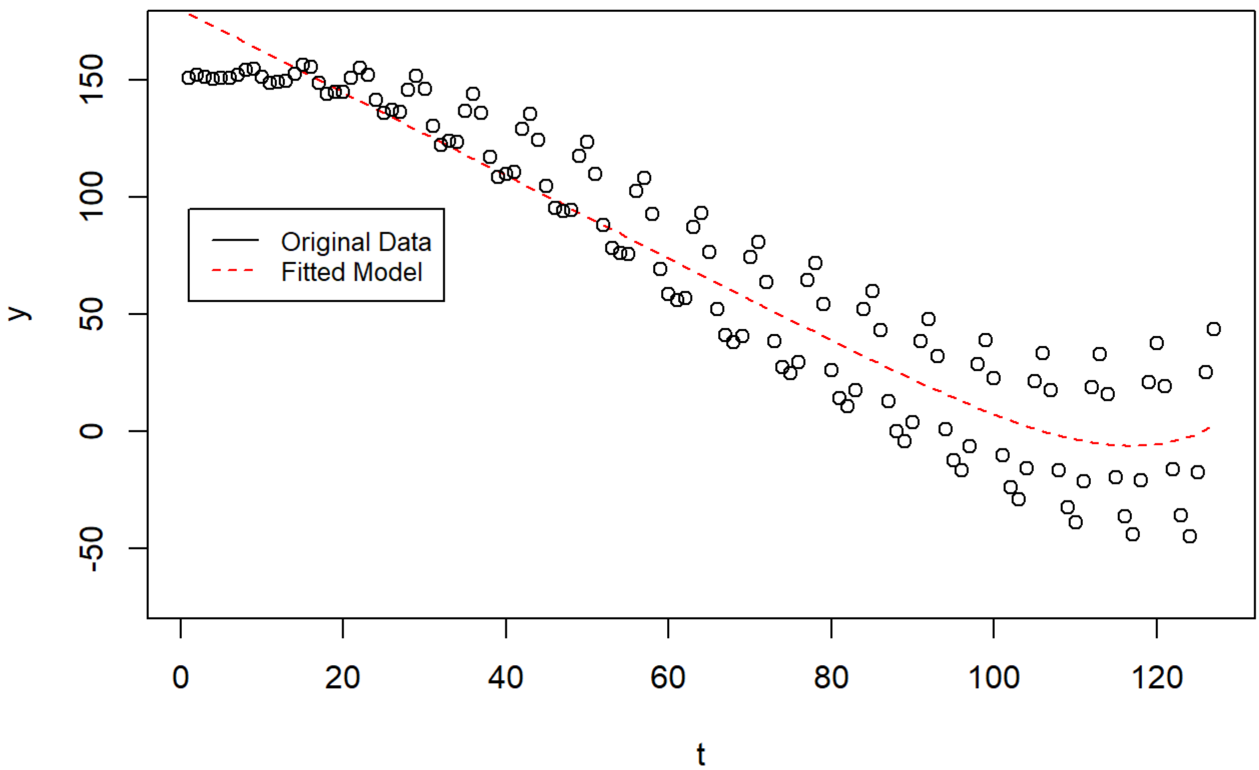
$$\mu_t = \beta_0 + \beta_1 t + \beta_2 t^{\inf} \tag{6}$$

```
# Assigning variable and building model
t10 = t^10
poly_10 = lm(data_ts~t10+t)
```

```
##
## Call:
## lm(formula = data_ts ~ t10 + t)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.743 -16.216  -1.324  18.691  42.898
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.803e+02  4.002e+00  45.044  < 2e-16 ***
## t10          4.385e-20  1.069e-20   4.101  7.4e-05 ***
## t           -1.773e+00  6.575e-02 -26.963  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.35 on 124 degrees of freedom
## Multiple R-squared:  0.8957, Adjusted R-squared:  0.894
## F-statistic: 532.4 on 2 and 124 DF,  p-value: < 2.2e-16
```

- Median of residuals became slightly worse and the range of them is approximately the same.
- t10 is insignificant and it is estimate is very low.

Figure 12. Fitted High polynomial



- The interesting thing is that it captures the end behavior by shifting upwards?

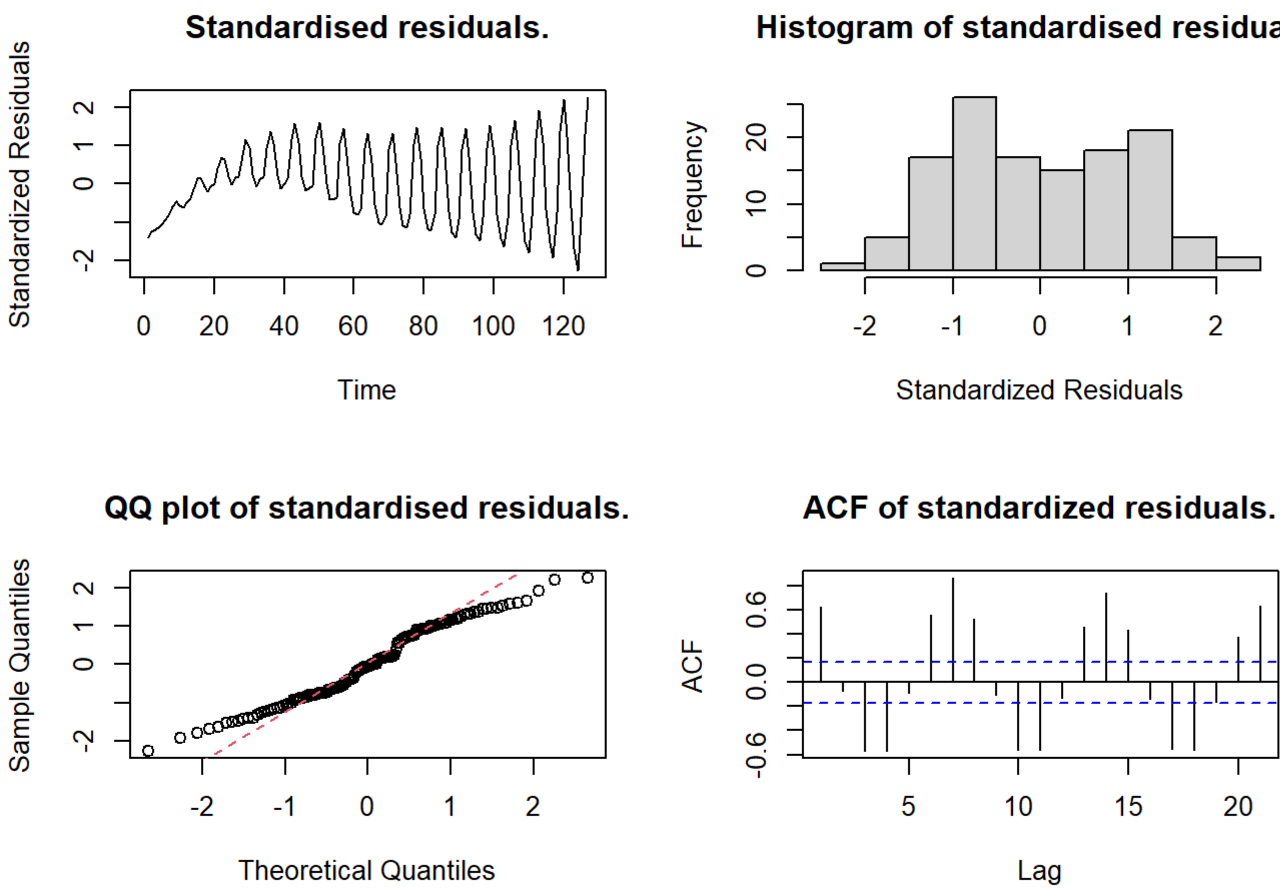


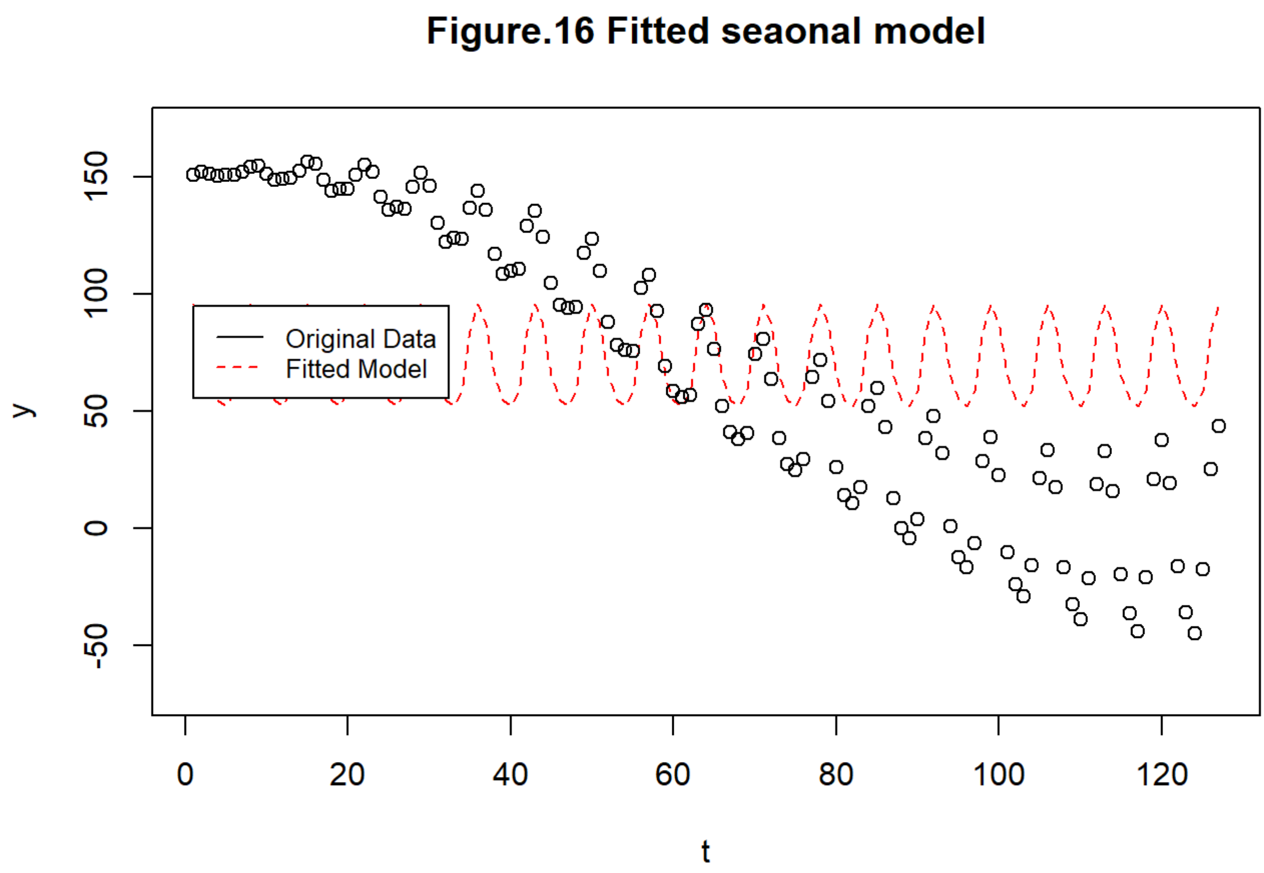
Figure 13-15. High polynomial Residuals Analysis

- Normality decreased and the model is not better.

Let us now capture the seasonality and see how a seasonal model do:

Seasonal model - Frequency = 7

```
# Assigning variables and building model
data_ts_freq_7 = ts(data[,2], frequency = 7)
seasons.= season(data_ts_freq_7)
seasonal_model = lm(data_ts_freq_7~seasons.-1)
```




```
##
## Call:
## lm(formula = data_ts_freq_7 ~ seasons. - 1)
##
## Residuals:
##      Min       1Q  Median      3Q     Max
## -96.865 -56.530  -2.747  58.007  98.773
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## seasons.Monday       95.81      14.17   6.761 5.27e-10 ***
## seasons.Tuesday      87.23      14.56   5.991 2.24e-08 ***
## seasons.Wednesday    65.06      14.56   4.469 1.80e-05 ***
## seasons.Thursday     54.72      14.56   3.758 0.000266 ***
## seasons.Friday       52.24      14.56   3.588 0.000483 ***
## seasons.Saturday     58.54      14.56   4.021 0.000102 ***
## seasons.Sunday       84.50      14.56   5.804 5.39e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 61.77 on 120 degrees of freedom
## Multiple R-squared:  0.5979, Adjusted R-squared:  0.5744
## F-statistic: 25.49 on 7 and 120 DF,  p-value: < 2.2e-16
```

- Median and the range of residuals are worse.
- Coefficients seems to be significant
- Adjusted R-squared way lower.

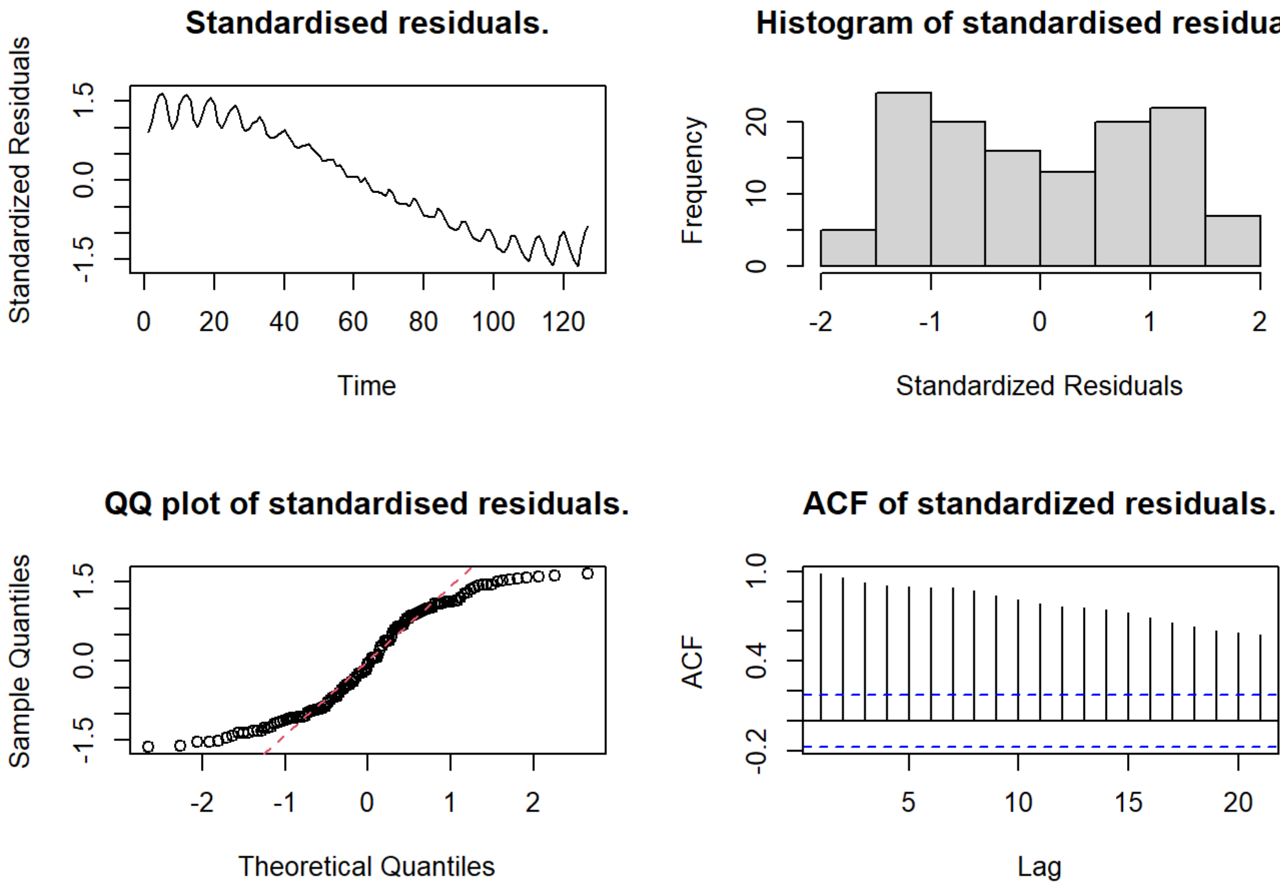


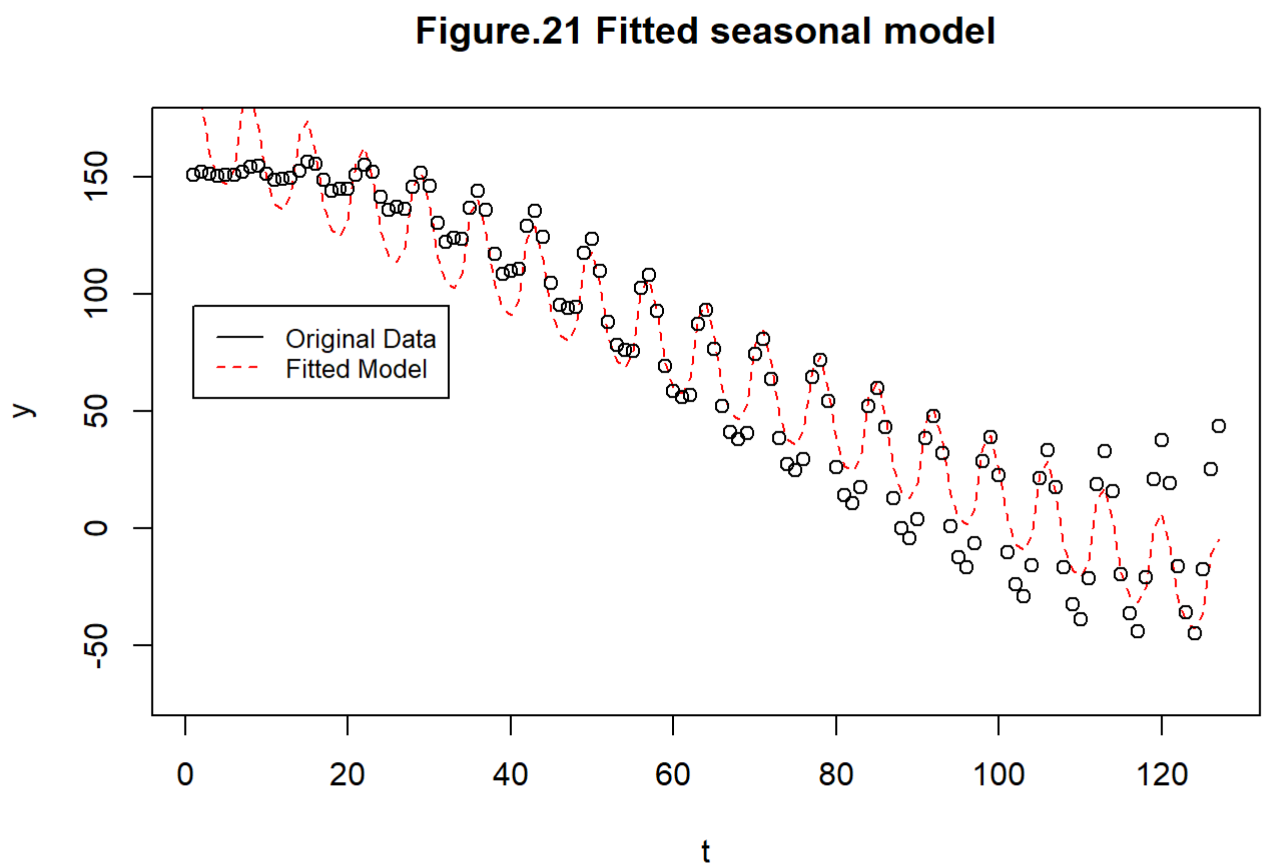
Figure 17-20. High polynomial Residuals Analysis

- The model does not capture autocorrelation and residuals are not normal at all from the test and graphs.
- The only good thing about this model is that it is capturing some of the seasonality/frequencies and residuals are started to become white noise.

What if we combine linear t term and seasonal to capture the downward trend?

Linear + Seasonal

```
# Assigning variables and building model
t = time(data_ts_freq_7)
seasons_linear.= season(data_ts_freq_7)
seasonal_linear_model = lm(data_ts_freq_7~seasons_linear.+t-1)
```



- Captured the trend and seasonality.

```
##
## Call:
## lm(formula = data_ts_freq_7 ~ seasons_linear. + t - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -45.517  -8.844  -1.459   9.071  48.404
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## seasons_linear.Monday    207.4871     4.0745   50.92   <2e-16 ***
## seasons_linear.Tuesday   194.9136     4.0971   47.57   <2e-16 ***
## seasons_linear.Wednesday 174.3387     4.1169   42.35   <2e-16 ***
## seasons_linear.Thursday  165.5947     4.1369   40.03   <2e-16 ***
## seasons_linear.Friday    164.7126     4.1571   39.62   <2e-16 ***
## seasons_linear.Saturday  172.6072     4.1775   41.32   <2e-16 ***
## seasons_linear.Sunday    200.1644     4.1981   47.68   <2e-16 ***
## t                        -11.1672     0.2422  -46.11   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.28 on 119 degrees of freedom
## Multiple R-squared:  0.9787, Adjusted R-squared:  0.9773
## F-statistic:   683 on 8 and 119 DF,  p-value: < 2.2e-16
```

- Median and range of residuals better than seasonal but still not good.
- All coefficients estimate are good and they are significant.

- Std Error are low which is good.
- High adjusted R-squared. Is the model overfitting? Let us analyse the residuals:

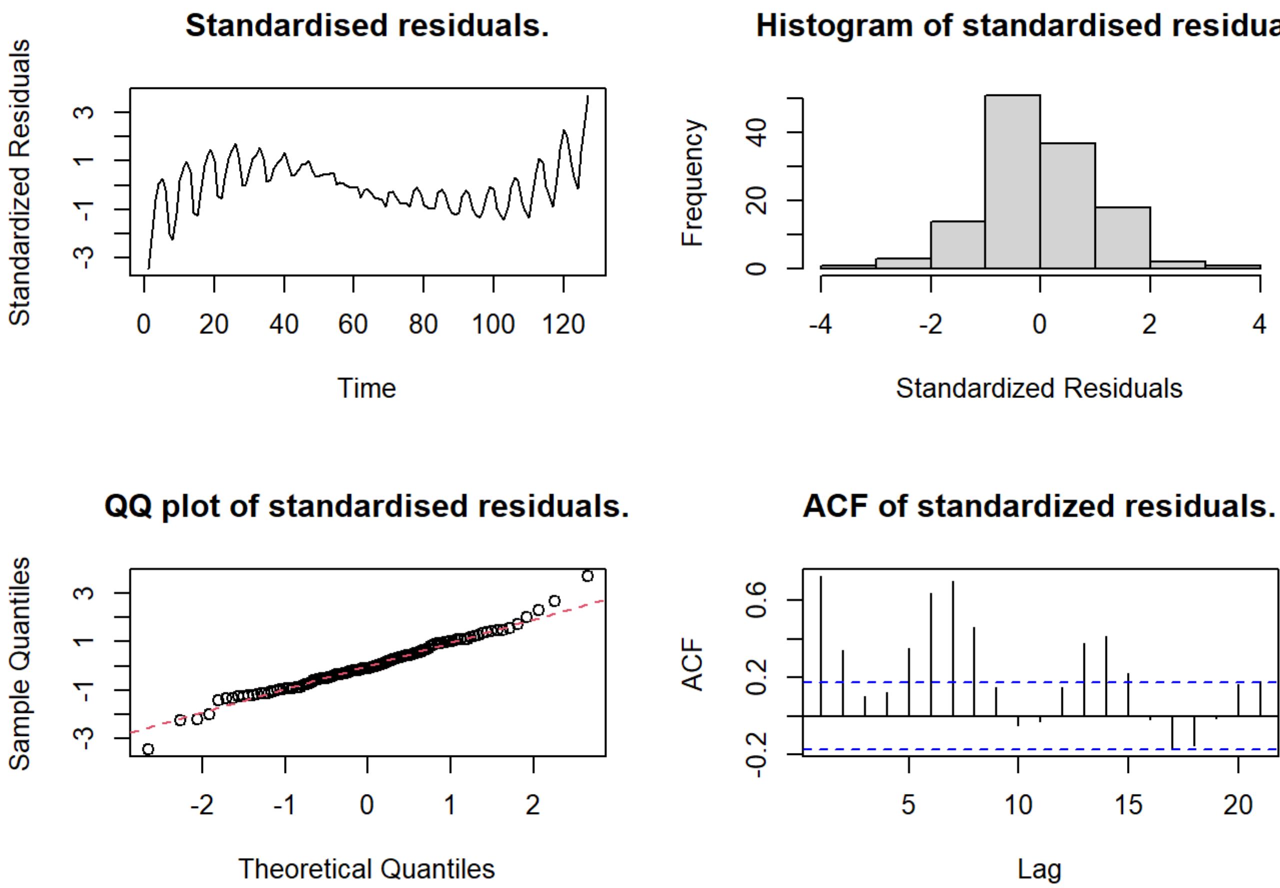


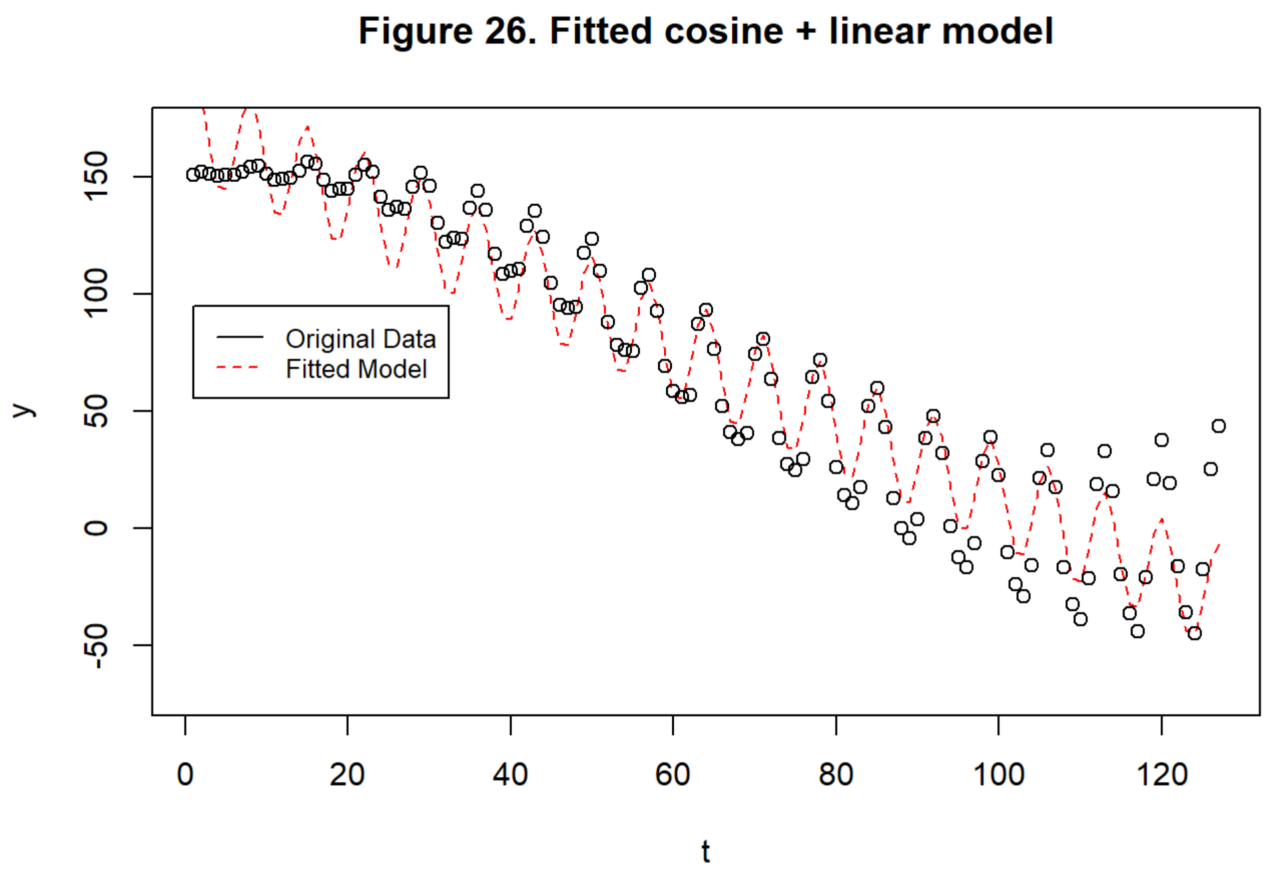
Figure 22-25. Seasonal Residuals Analysis

- As expected, the model is capturing better trend and overall behavior of dataset. Therefore, residuals started to behave like white noise.
- Histogram shows normality.
- Most of points are on the line in the QQ plot which indicate normality.
- Shapiro wilk test p value is very close to 0.05.

Will a cosine model + linear model residuals do better in capturing normality and less autocorrelation?

Cosine + linear model

```
# Assigning variables and building model
harmonic. = harmonic(data_ts_freq_7,1)
cosine_lin = lm(data_ts_freq_7~harmonic.+t)
```




```
##
## Call:
## lm(formula = data_ts_freq_7 ~ harmonic. + t)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -43.446 -10.293  -0.176   8.217  50.397
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            182.8039     2.7589  66.259   <2e-16 ***
## harmonic.cos(2*pi*t)   22.6080     1.8021  12.546   <2e-16 ***
## harmonic.sin(2*pi*t)   -0.5800     1.8177  -0.319     0.75
## t                      -11.1629     0.2444 -45.666   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.42 on 123 degrees of freedom
## Multiple R-squared:  0.9481, Adjusted R-squared:  0.9468
## F-statistic: 748.4 on 3 and 123 DF,  p-value: < 2.2e-16
```

- Median is close to 0 but the range still not that good.
- Coefficients are all significant except the sin term.
- Std error are all small.
- VERY small change in R squared.

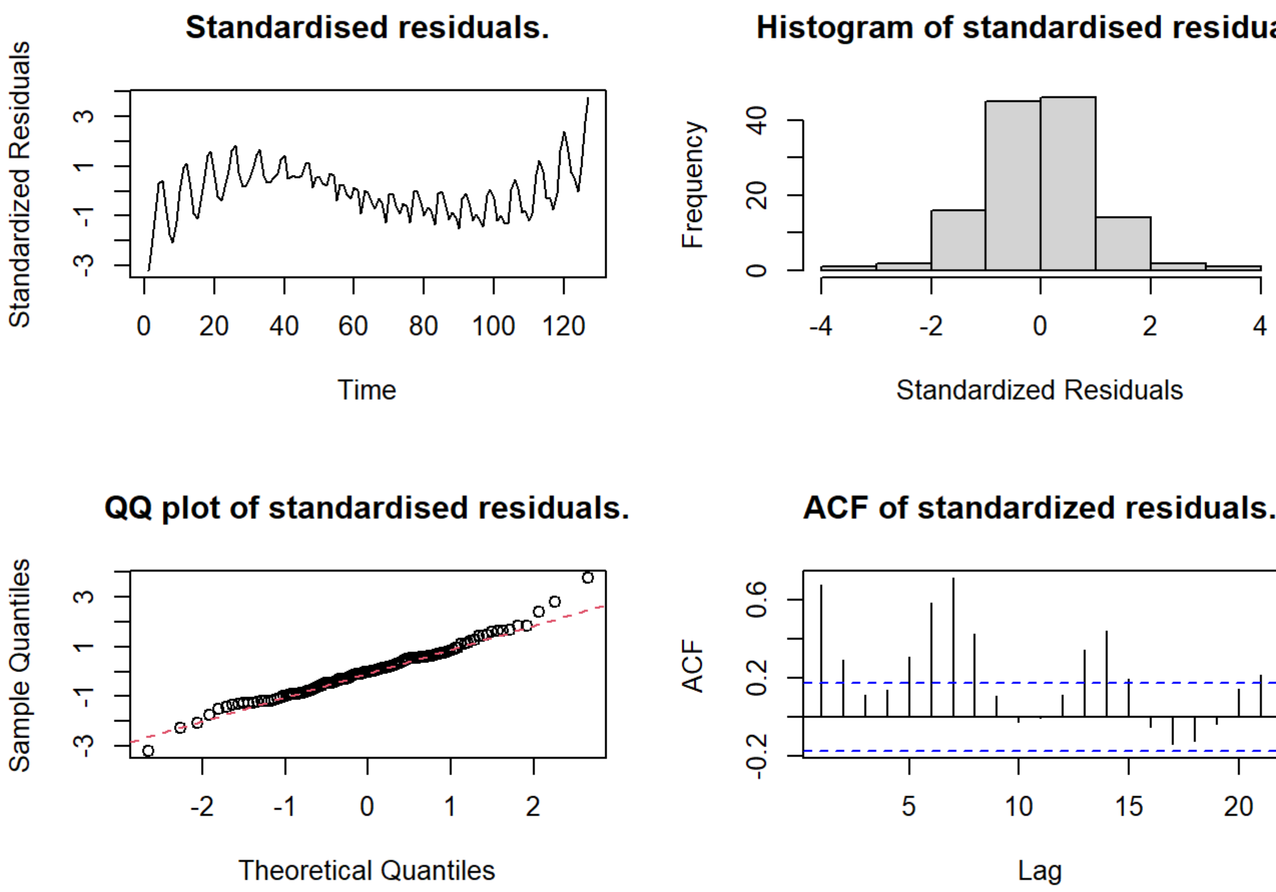


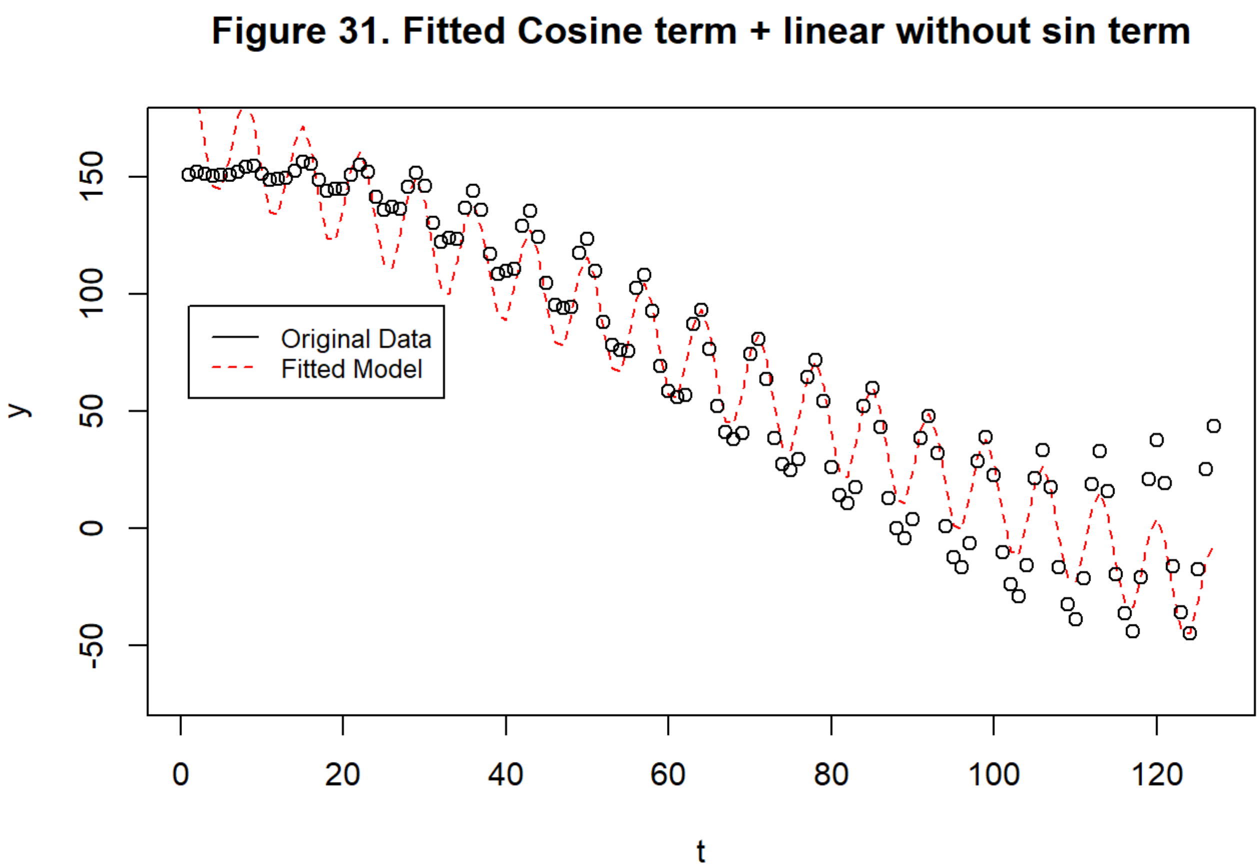
Figure 27-30. Cosine + linear Residuals Analysis

- The white noise behavior became more obvious!
- Shapiro wilk p-value of cosine + linear model > season + linear model. Even in plots, cosine + linear residuals seems to have more normality.

Since the sin term was insignificant, does removing it will improve overall model or decrease residuals AC?

Removed sin term + linear

```
# Assigning variables and building model
cosine_term = harmonic.[,'cos(2*pi*t)']
cosine_lin_1 = lm(data_ts_freq_7~cosine_term+t)
```



- Does not seems there is a change in plot but let us analyse the summary and residuals.

```
##
## Call:
## lm(formula = data_ts_freq_7 ~ cosine_term + t)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.418 -10.463   0.227   8.291  50.369
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 182.7728     2.7472   66.53   <2e-16 ***
## cosine_term  22.6080     1.7955   12.59   <2e-16 ***
## t           -11.1598     0.2434  -45.86   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.36 on 124 degrees of freedom
## Multiple R-squared:  0.948,  Adjusted R-squared:  0.9472
## F-statistic:  1131 on 2 and 124 DF,  p-value: < 2.2e-16
```

- No improvement in residuals median and range.
- R-squared improved by VERY unnoticeable amount.

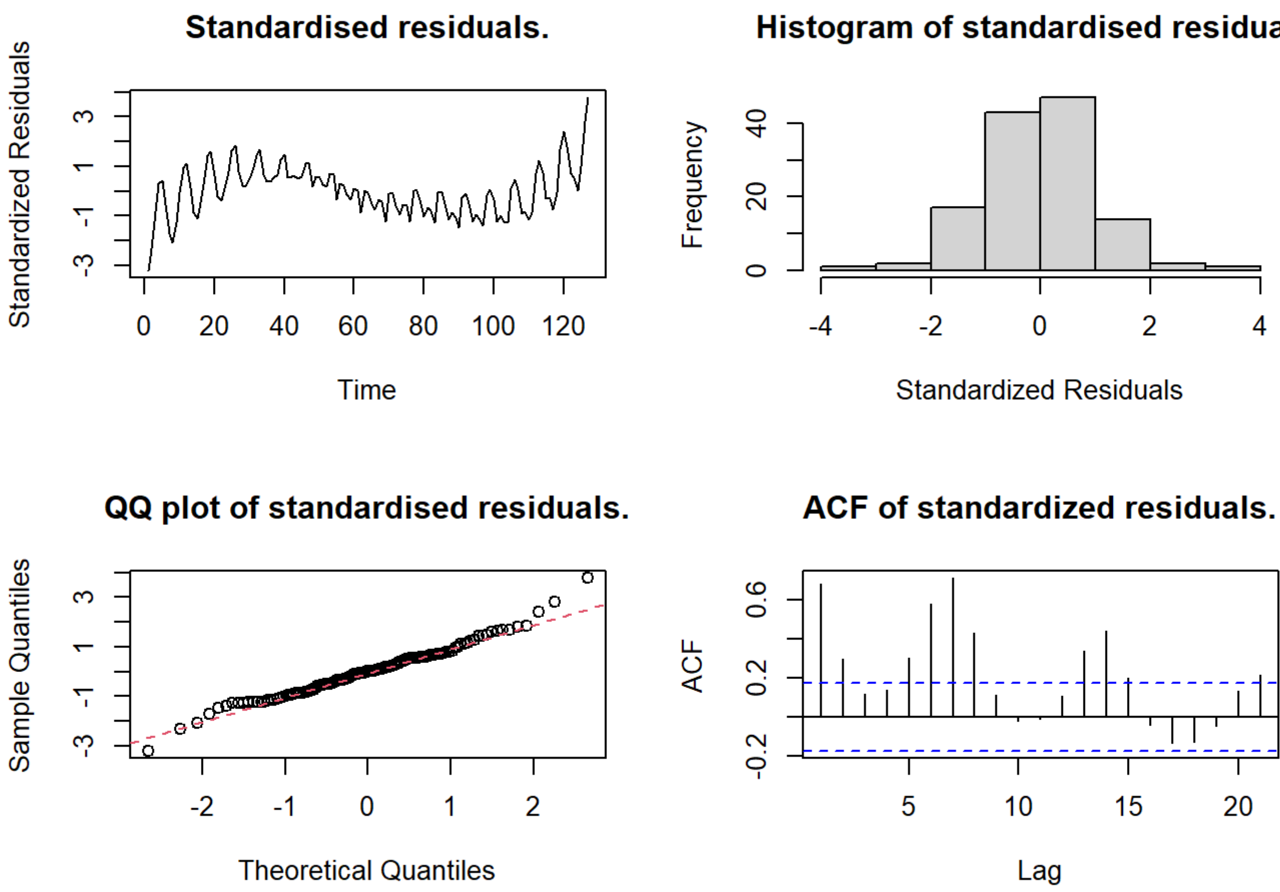
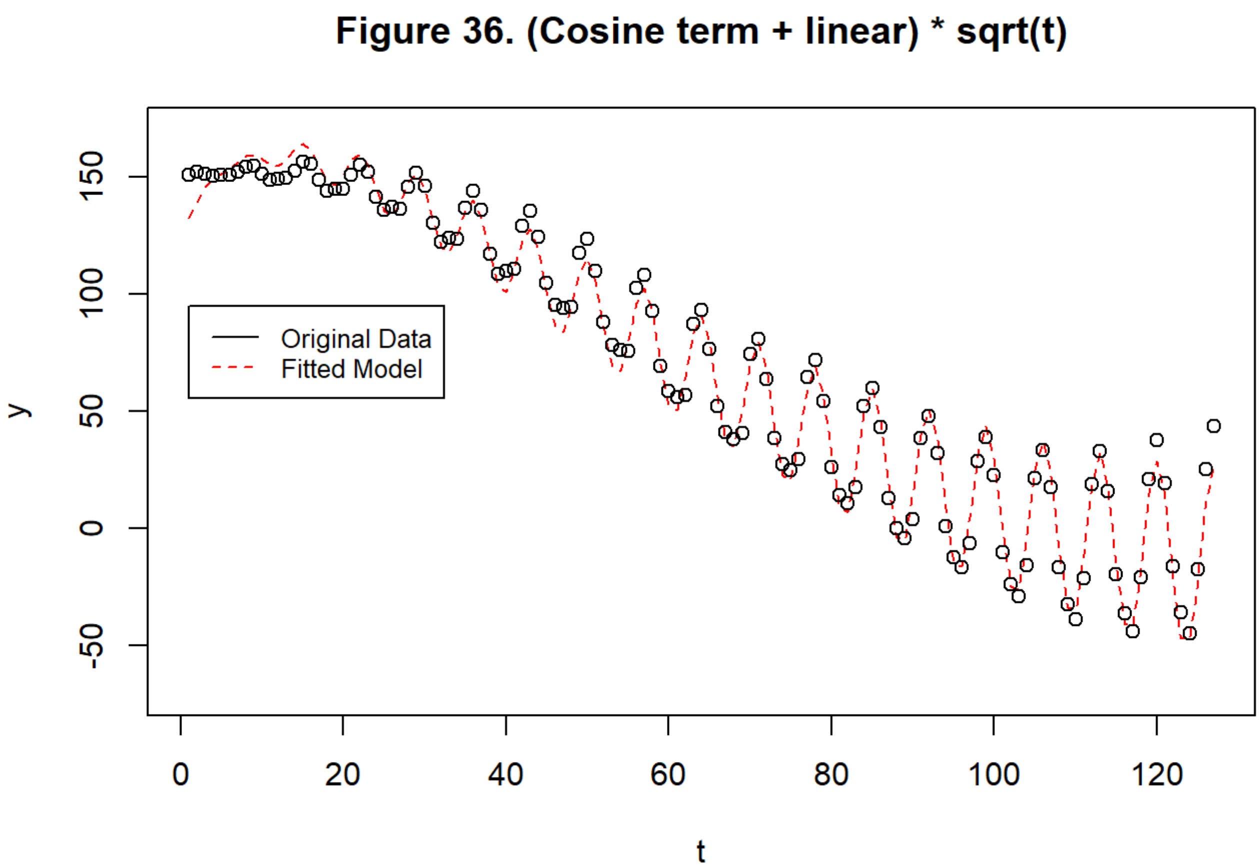


Figure 32-35. cosine + t without sin Residuals Analysis

- The p-value decreased in Shapiro wilk by inconsiderable amount. From principle of parsimony, this is better.
- The model is not capturing the start behavior. What if we multiplied cos amplitude by sqrt(t)?

(Cos + t) * sqrt(t)

```
# Assigning variables and building model
tsqrt = sqrt(t)
cosine_lin_sqrt = lm(data_ts_freq_7~((cosine_term+t)*tsqrt))
```




```
##
## Call:
## lm(formula = data_ts_freq_7 ~ ((cosine_term + t) * tsqrt))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.1895  -4.3627   0.1583   4.0276  18.2347
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -39.5759    13.9799  -2.831  0.00544 **
## cosine_term        -15.9939     2.6361  -6.067 1.53e-08 ***
## t                 -118.4723     6.5308 -18.140  < 2e-16 ***
## tsqrt              281.0517    17.1598  16.378  < 2e-16 ***
## cosine_term:tsqrt   12.8557     0.8333  15.428  < 2e-16 ***
## t:tsqrt             12.7030     0.7808  16.270  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.115 on 121 degrees of freedom
## Multiple R-squared:  0.9908, Adjusted R-squared:  0.9904
## F-statistic:  2608 on 5 and 121 DF,  p-value: < 2.2e-16
```

- Median very close to 0 and the range is way smaller!
- All coefficients are significant and their std error is low.
- Very high R which might tell us that the model is overfitting. But let us analyse the residuals:

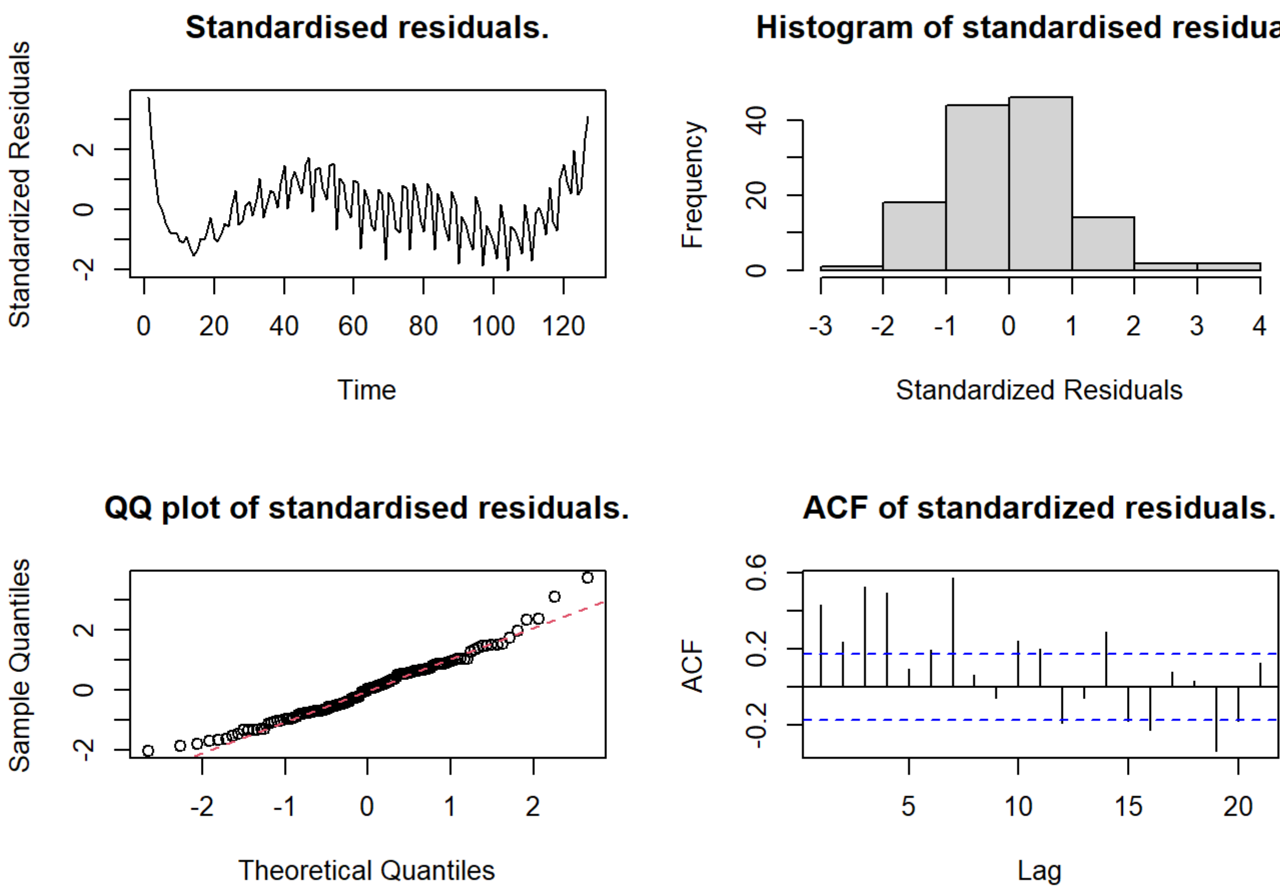


Figure 37-40. (cosine + t) * sqrt(t) Residuals Analysis

- Residuals seems to be fully white noise.
- Histogram and QQ plot indicate normality but Shapiro wilk p value is low.
- ACF of residuals does not indicate auto regressive which means the model is capturing most of AF.

Conclusion and predictions

- There are two models that are good in terms of residuals analysis which are lm(formula = data_ts_freq_7 ~ ((cosine_term + t) * tsqrt)) and lm(formula = data_ts_freq_7 ~ cosine_term + t) but these seems to over fit the data and might just copy the behavior.
- The other models does not over fit but most of the analysis such as trend, seasonality and ACF are captured in residuals.

Predictions of chosen model

lm(formula = data_ts_freq_7 ~ ((cosine_term + t) * tsqrt))

- This is the chosen model because of the followings:
  - Cosine_term captures the seasonality of the data.
  - t term captures the negative trend.
  - By multiplying with tsqrt, the model captures the change in variance or the change of waves as time increases.
  - The model analysis shows a good range in residual and R squared values.
  - The residuals analysis shows that the residuals are white noise. The normality is there by using all relevant testing.

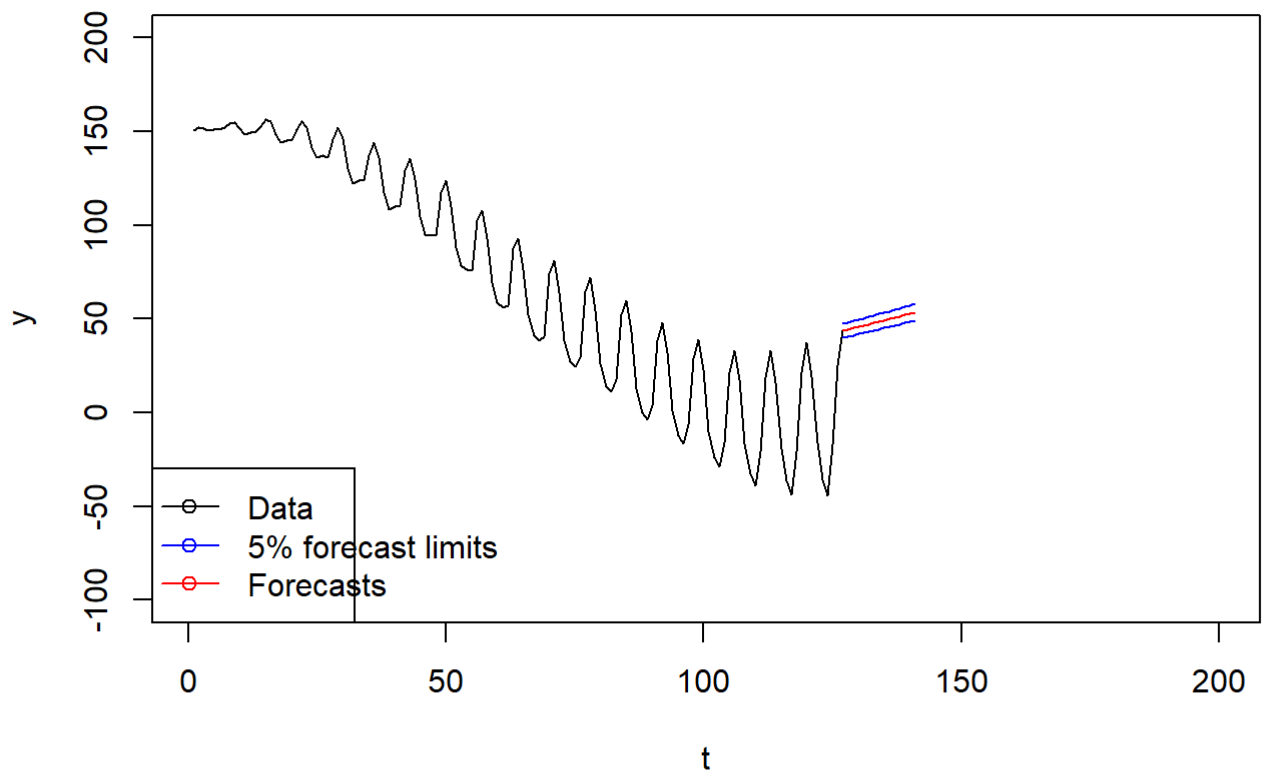
```
# Predicting
h = 15
new = data.frame(t = seq((length(t)+1), (length(t)+h), 1), cosine_term = seq((length(cosine_t
erm)+1), (length(cosine_term)+h), 1), tsqrt = seq((length(tsqrt)+1), (length(tsqrt)+h), 1))
forecasts = predict(cosine_lin_sqrt, new, interval = "prediction")/10000
forecasts
```

```
##         fit      lwr      upr
## 1  43.74769 40.09073 47.40466
## 2  44.41921 40.70594 48.13248
## 3  45.09584 41.32584 48.86584
## 4  45.77758 41.95042 49.60474
## 5  46.46443 42.57968 50.34919
## 6  47.15640 43.21362 51.09918
## 7  47.85347 43.85224 51.85471
## 8  48.55566 44.49554 52.61578
## 9  49.26296 45.14352 53.38239
## 10 49.97537 45.79618 54.15455
## 11 50.69289 46.45353 54.93225
## 12 51.41552 47.11555 55.71550
## 13 52.14327 47.78226 56.50428
## 14 52.87613 48.45365 57.29861
## 15 53.61410 49.12972 58.09848
```

- The predictions seems to increase to + infinity and that does not capture the cyclic behavior.
- The resulted values were huge and must divided by 10000. The reason behind this is unknown.

```
# Plotting
plotting_forecast("Figure 41. Forecasting return trading using (cosine_term + t) * tsqrt")
```

Figure 41. Forecasting return trading using (cosine_term + t) * tsqrt

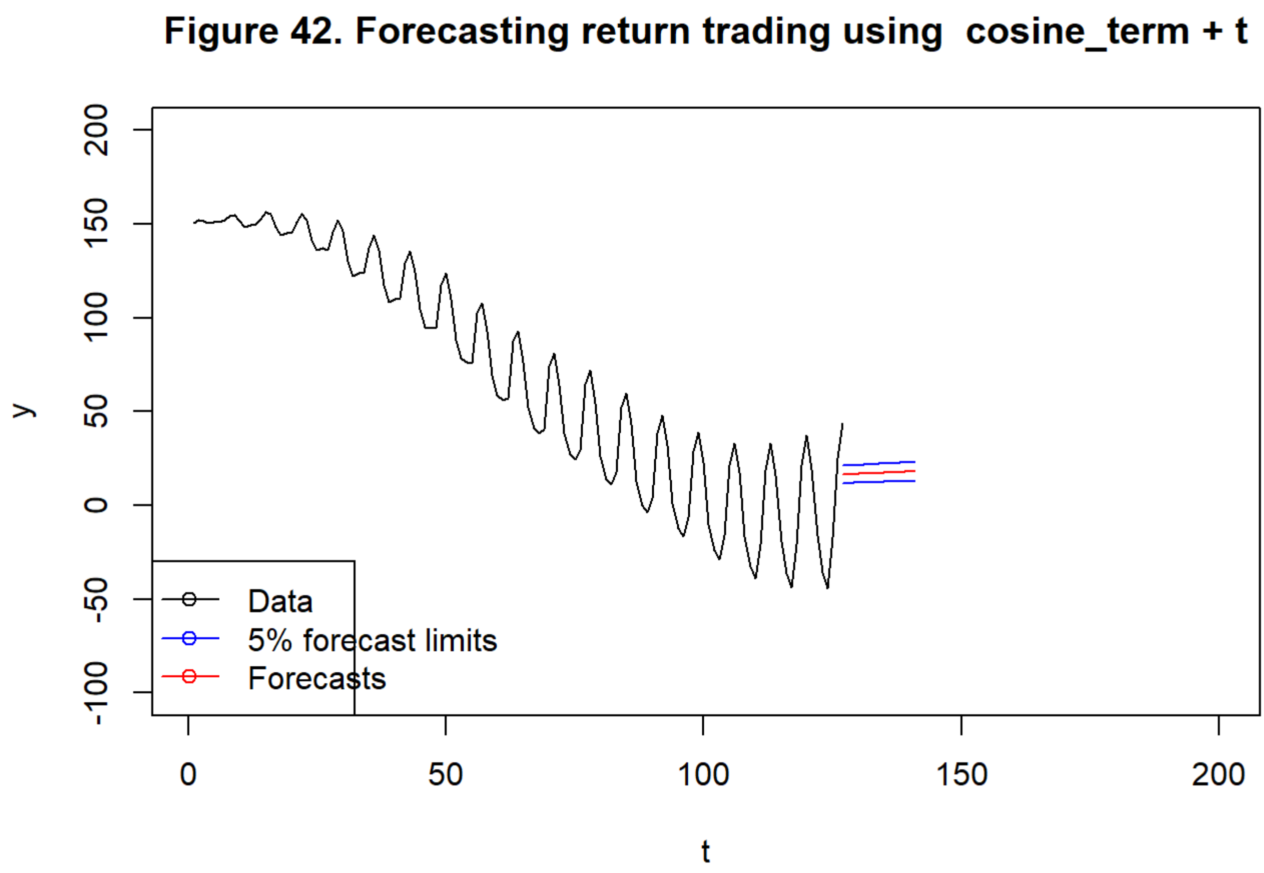


Predictions of two other models that are not chosen

lm(formula = data_ts_freq_7 ~ cosine_term + t)

```
##         fit      lwr      upr
## 1  16.48140 11.88847 21.07433
## 2  16.59588 11.96716 21.22460
## 3  16.71036 12.04585 21.37488
## 4  16.82485 12.12454 21.52515
## 5  16.93933 12.20322 21.67543
## 6  17.05381 12.28191 21.82571
## 7  17.16829 12.36059 21.97599
## 8  17.28277 12.43927 22.12627
## 9  17.39725 12.51796 22.27655
## 10 17.51174 12.59664 22.42684
## 11 17.62622 12.67532 22.57712
## 12 17.74070 12.75400 22.72741
## 13 17.85518 12.83267 22.87769
## 14 17.96966 12.91135 23.02798
## 15 18.08415 12.99003 23.17826
```

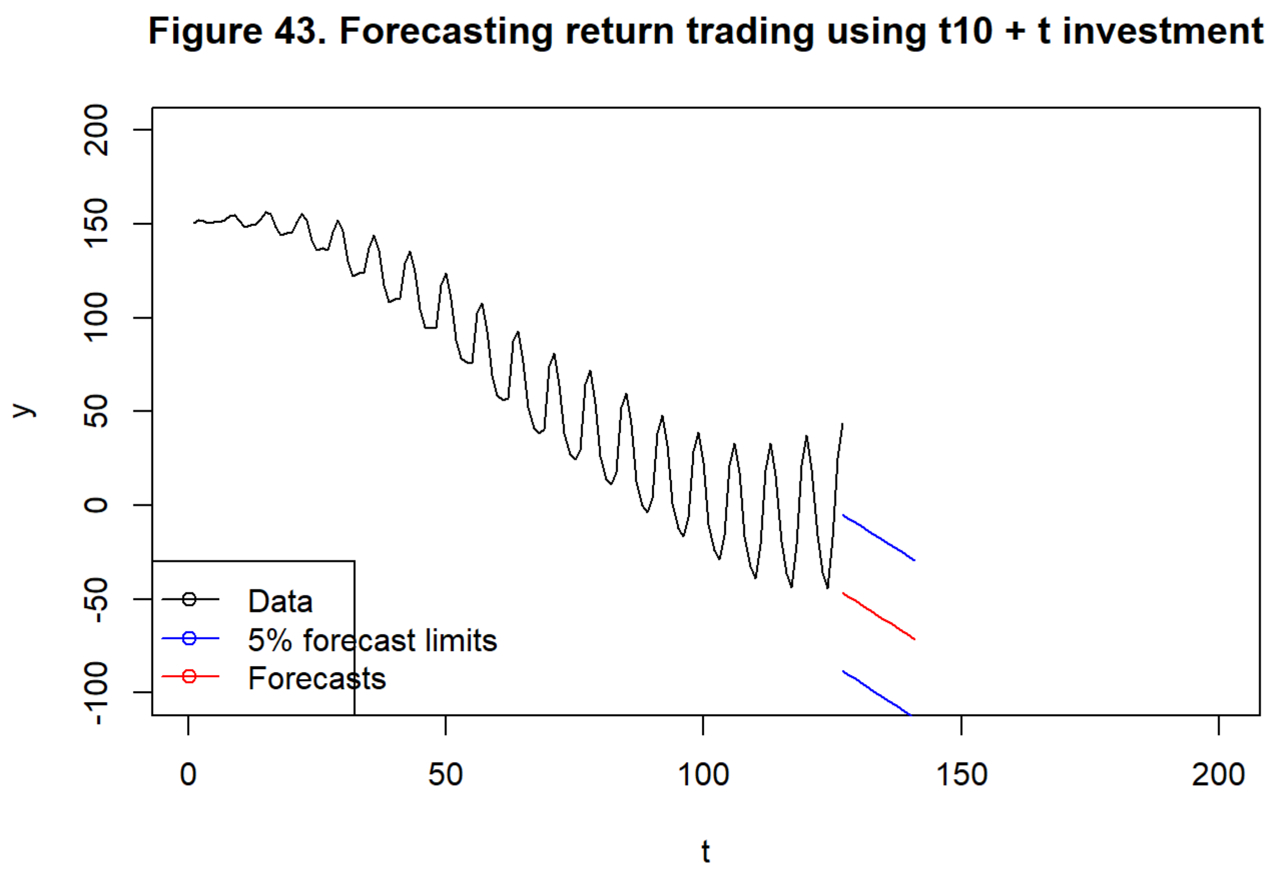
- The same issues of previous model predictions are represented here.



lm(formula = data_ts ~ t10 + t)

```
##         fit       lwr        upr
## 1  -46.65567  -88.28267  -5.028677
## 2  -48.42859  -90.08637  -6.770821
## 3  -50.20151  -91.89045  -8.512582
## 4  -51.97443  -93.69491 -10.253959
## 5  -53.74735  -95.49975 -11.994955
## 6  -55.52027  -97.30497 -13.735570
## 7  -57.29319  -99.11058 -15.475805
## 8  -59.06611 -100.91656 -17.215661
## 9  -60.83903 -102.72292 -18.955138
## 10 -62.61195 -104.52966 -20.694238
## 11 -64.38487 -106.33677 -22.432961
## 12 -66.15779 -108.14426 -24.171309
## 13 -67.93071 -109.95213 -25.909283
## 14 -69.70363 -111.76037 -27.646882
## 15 -71.47654 -113.56898 -29.384109
```

- Same issues. The model is approaching - infinity.



References

- MATH1318 Time Series Analysis notes prepared by Dr. Haydar Demirhan.
- Time Series Analysis with application in R by Cryer and Chan.