Statistical Methods Assessed Coursework

URN: 6491580

April 21, 2021

Contents

Problem Statement

We are interested in analysing the sales of company X's flagship product over the 42 year period between January 1979 to December 2020. We have the total sale values of the product per month, which gives us 504 months worth of data points, or time series.

It is suggested that we model the time series data with a stochastic process of the form: $X_t = m_t + S_t + Z_t$, where

- X_t is our sales time series data, each point denoted by little **x**: $x_1, x_2, \cdots, x_{504}$.
- m_t is entirely non-random, denotes the **trend**.
- S_t is also entirely non-random, denotes **seasonality**.
- Z_t is a white noise process, **random variable**.

Trend is the component of a stochastic process that represents long-term movement in that process, denoted by m_t, it is a set of deterministic values for all values of **t**.

We assume Z_t as the **random variable** with expectation $E[Z_t] = 0$. Hence, $E[X_t] = E[m_t + Z_t] = E[m_t] + E[Z_t] = m_t + 0 = m_t$. This means that the trend values represent the mean of the stochastic process.

Question 1

a) Estimating trend using a global quadratic polynomial fit.

We can start off by visualising the raw sales data over the time period to get a better understanding of the it. We can use the global polynomial method to find an estimate for m_t denoted by $\hat{m}_t$.

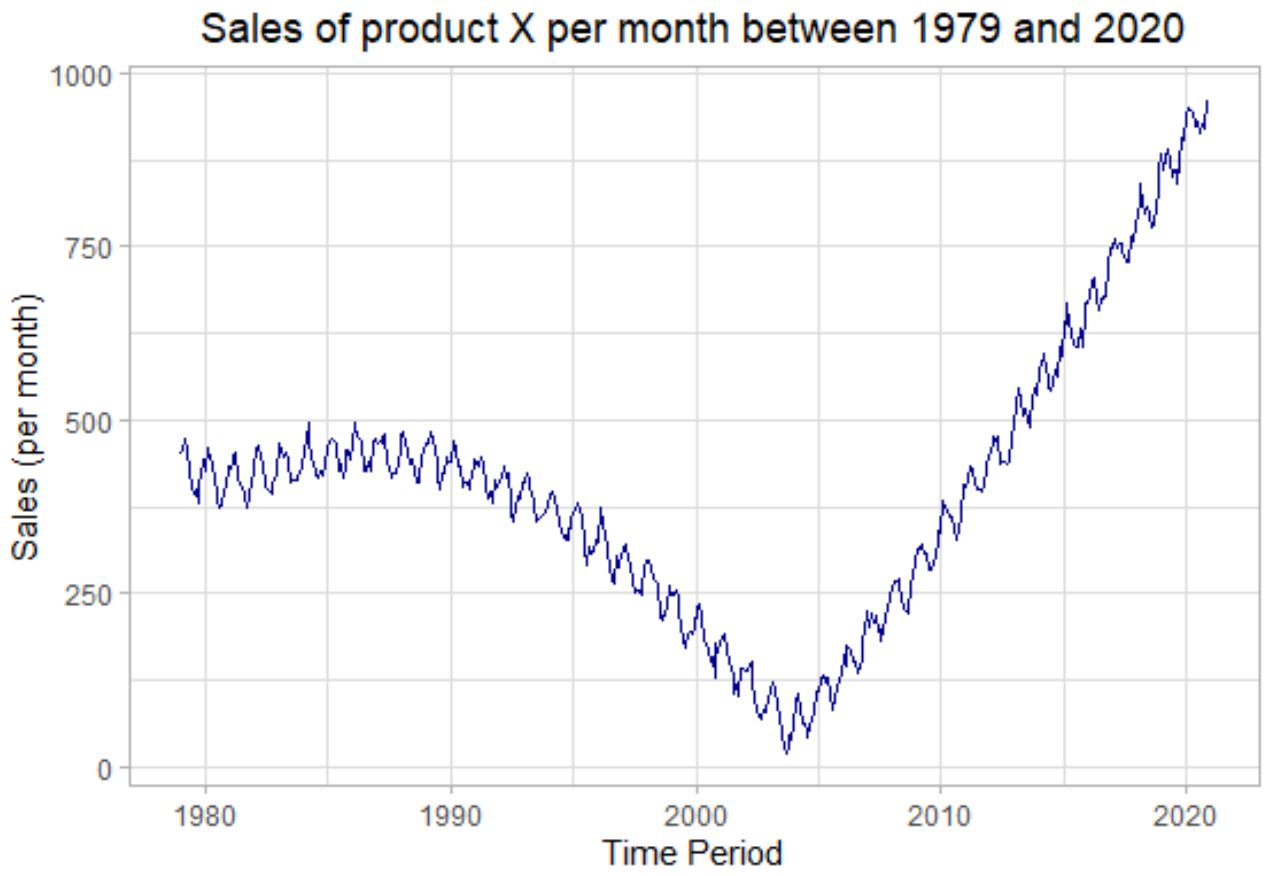


Figure 1.1: Sales data visualisation

The estimate would be of the form $\hat{m}_t = \beta_0 + \beta_1 t + \beta_2 t^2 + ..\beta_k t^k \ for \ 1 \leq t \leq n$, where n is the number of data points, in our case n = 504 sales points; k is the degree of the polynomial; and $\beta_0, \beta_1, \beta_2, \cdots \beta_k$ are the coefficients of the model, which we would need to find.

If we consider a global quadratic polynomial fit, which has degree $k = 2$ then the trend model will take the form $m_t = \beta_0 + \beta_1 t + \beta_2 t^2$ and the stochastic process will be of the form $X_t = \beta_0 + \beta_1 t + \beta_2 t^2 + Z_t$.

To get the estimates for our coefficients we use the least squares method where we minimize the sum of squares:

$$S\left(\beta_0, \beta_1, \beta_2\right) = \sum_{t=1}^{504} \left(x_t - \beta_0 - \beta_1 t - \beta_2 t^2\right)^2 \tag{1.1}$$

We differentiate $S\left(\beta_0, \beta_1, \beta_2\right)$ with respect to β_0, β_1 *and* β_2 to get the following

$$\begin{aligned} \frac{dS}{d\beta_0} &= \sum_{t=1}^{504} \frac{d}{d\beta_0} \left(x_t - \beta_0 - \beta_1 t - \beta_2 t^2\right)^2 \\ &= \sum_{t=1}^{504} -2\left(x_t - \beta_0 - \beta_1 t - \beta_2 t^2\right) = -2\sum_{t=1}^{504} \left(x_t - \beta_0 - \beta_1 - t - \beta_2 t^2\right) \end{aligned} \tag{1.2}$$

$$\begin{aligned} \frac{dS}{d\beta_1} &= \sum_{t=1}^{504} \frac{d}{d\beta_1} \left(x_t - \beta_0 - \beta_1 t - \beta_2 t^2\right)^2 \\ &= \sum_{t=1}^{504} -2t\left(x_t - \beta_0 - \beta_1 t - \beta_2 t^2\right) = -2\sum_{t=1}^{504} t\left(x_t - \beta_0 - \beta_1 t - \beta_2 t^2\right) \end{aligned} \tag{1.3}$$

$$\begin{aligned} \frac{dS}{d\beta_2} &= \sum_{t=1}^{504} \frac{d}{d\beta_2} \left(x_t - \beta_0 - \beta_1 t - \beta_2 t^2\right)^2 \\ &= \sum_{t=1}^{504} -2t^2\left(x_t - \beta_0 - \beta_1 t - \beta_2 t^2\right) = -2\sum_{t=1}^{504} t^2\left(x_t - \beta_0 - \beta_1 t - \beta_2 t^2\right) \end{aligned} \tag{1.4}$$

To find the optimal values of the coefficients that minimize the sum of squares and show that they fit the 'best' global quadratic, we set the differentiated equations to 0 and add hats to them to show that they are estimates. So $\hat{\beta}_0,\ \hat{\beta}_1$ *and* $\hat{\beta}_2$ must satisfy the normal equations:

$$\begin{aligned} -2\sum_{t=1}^{504} \left(x_t - \hat{\beta}_0 - \hat{\beta}_1 t - \hat{\beta}_2 t^2\right) &= 0 \\ -2\sum_{t=1}^{504} t\left(x_t - \hat{\beta}_0 - \hat{\beta}_1 t - \hat{\beta}_2 t^2\right) &= 0 \\ -2\sum_{t=1}^{504} t^2\left(x_t - \hat{\beta}_0 - \hat{\beta}_1 t - \hat{\beta}_2 t^2\right) &= 0 \end{aligned} \tag{1.5}$$

We simplify these to

$$\begin{aligned} \sum_{t=1}^{504} x_t &= \hat{\beta}_0 \sum_{t=1}^{504} 1 + \hat{\beta}_1 \sum_{t=1}^{504} t + \hat{\beta}_2 \sum_{t=1}^{504} t^2 \quad (1) \\ \sum_{t=1}^{504} t x_t &= \hat{\beta}_0 \sum_{t=1}^{504} t + \hat{\beta}_1 \sum_{t=1}^{504} t^2 + \hat{\beta}_2 \sum_{t=1}^{504} t^3 \quad (2) \\ \sum_{t=1}^{504} t^2 x_t &= \hat{\beta}_0 \sum_{t=1}^{504} t^2 + \hat{\beta}_1 \sum_{t=1}^{504} t^3 + \hat{\beta}_2 \sum_{t=1}^{504} t^4 \quad (3) \end{aligned} \tag{1.6}$$

These simultaneous equations can be represented as matrices which we can solve. Suppose

$$A = \begin{bmatrix} \sum_{t=1}^{504} 1 & \sum_{t=1}^{504} t & \sum_{t=1}^{504} t^2 \\ \sum_{t=1}^{504} t & \sum_{t=1}^{504} t^2 & \sum_{t=1}^{504} t^3 \\ \sum_{t=1}^{504} t^2 & \sum_{t=1}^{504} t^3 & \sum_{t=1}^{504} t^4 \end{bmatrix}$$

$$x = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \end{bmatrix}$$

and

$$b = \begin{bmatrix} \sum_{t=1}^{504} x_t \\ \sum_{t=1}^{504} t x_t \\ \sum_{t=1}^{504} t^2 x_t \end{bmatrix}$$

Then we have the inhomogeneous matrix equation $\mathbf{Ax} = \mathbf{b}$. In this case we can use linear algebra to find the particular solution. We take the generalised inverse of $\mathbf{A}$ to find the coefficients, hence the coefficients are found by $\mathbf{b} = \mathbf{A^{-1}x}$.

b) Fitting a Global Linear Model

Fig. 1.1 suggests that the sales data does not originate from a single trend but it might come from two or more trends. An initial guess is that the global quadratic fit is going to be a simplistic fit for the data. However, we will test the hypothesis by fitting the model and comparing it to other models.

Also, one can notice the seasonality straight away as the points oscillate evenly.

We will use R's built-in function "'lm()"' for fitting linear models, instead of the calculation in 1. The function would do the calculation faster and would estimate the coefficients and we will examine the results.

From the output from the code in the Appendix show below, we can see that the global quadratic fit for the trend is

$$\hat{m}_t = 648.9979 - 3.7935t + 0.0083t^2. \tag{1.7}$$

We can further analyse the fitted data by plotting the sales data and the fitted data.

```
>>
Call:
lm(formula = X ~ t + I(t^2))

Coefficients:
(Intercept)            t       I(t^2)
 648.997860    -3.793496     0.008355
```

Fig.1.2 shows the global quadratic fit as the red curve on the chart. The fit shows a convex trend with a minimum point at the year 2000, whereas the raw data has a trough around the year 2004 but does not look like a convex function. The chart correctly shows that there is an upward trend beyond the year 2010 but the actual trend in the blue line is growing faster than the quadratic polynomial in the red line. Suggestion that perhaps the trend estimate is not a good estimate for the sales data.

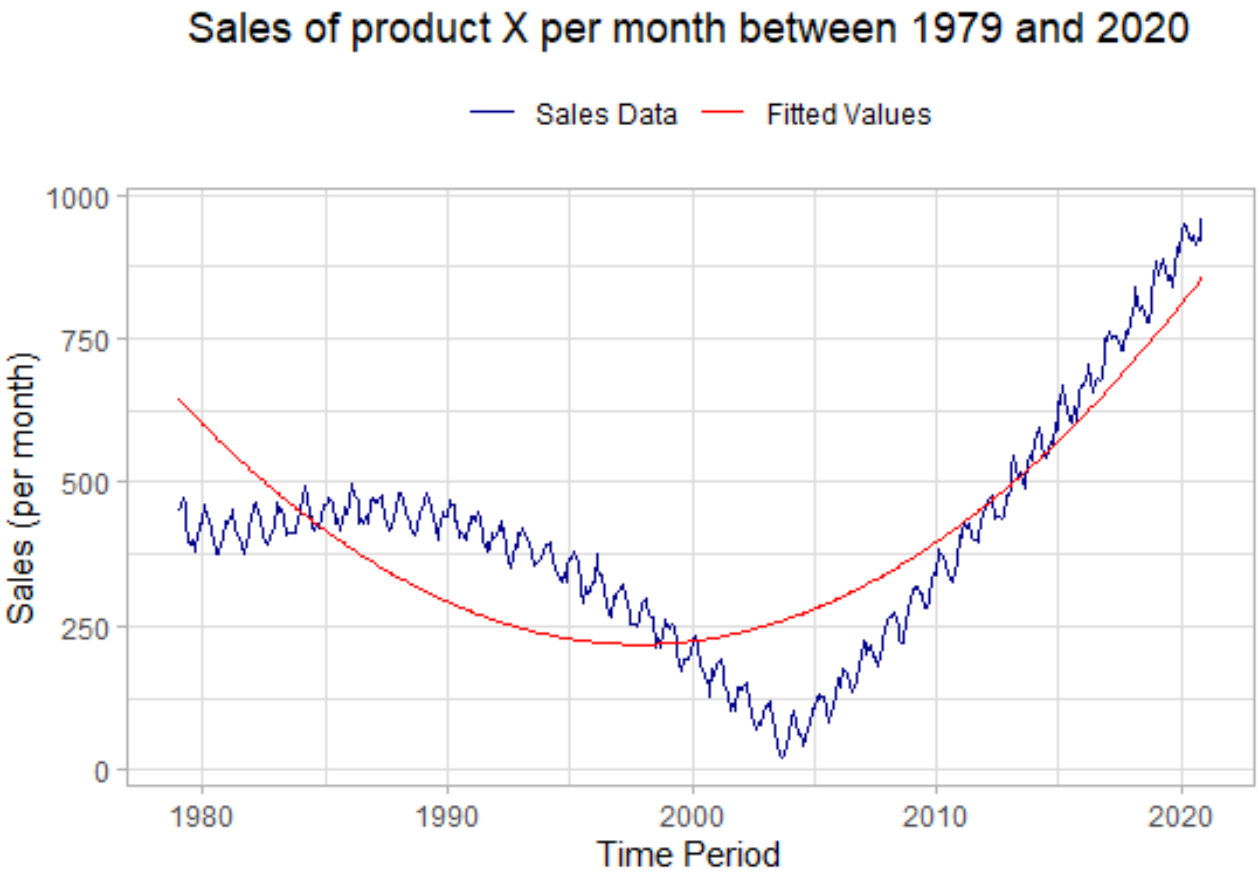


Figure 1.2: The fitted values of the global quadratic model and the raw sales data

Question 2

a) The local linear polynomial

Consider estimating the trend m_t by fitting a local linear polynomial to successive groups of 5 observations. In this case, for each value of t, fixed, the trend will be the following

$$m_{t+j} = \beta_0 + \beta_1 j, \tag{2.1}$$

where we need to estimate the coefficients $\hat{\beta}_0$ *and* $\hat{\beta}_1$. Note the **j** term shows the interval of points around a single point x_t, in this case $j \in \{-2, -1, 0, 1, 2\}$. More specifically, the trend function is for the interval $t-2 \leq x \leq t+2$ is approximated by a polynomial of degree 1.

However, we are only interested in estimating the trend m_t for singe point x_t, hence we set $j = 0$. Hence, in contrast to the global polynomial, we only need to find an estimate for $\hat{\beta}_0$, in this case the trend estimate is

$$\hat{m}_t = \hat{\beta}_0 + \hat{\beta}_1 j = \hat{\beta}_0, \tag{2.2}$$

where the result is a linear combination of the time series points.

Similarly to the global polynomial fit, we minimize the sum of squares for fixed point **t** to find the optimal values for the coefficients.

$$S = \sum_{j=-2}^{2} \left(x_{t+j} - \beta_0 - \beta_1 j\right)^2. \tag{2.3}$$

Then differentiate S with respect to β_0 and β_1 as shown

$$\begin{aligned}
\frac{dS}{d\beta_0} &= \sum_{j=-2}^{2} \frac{d}{d\beta_0} \left(x_{t+j} - \beta_0 - \beta_1 j\right)^2 \\
&= \sum_{j=-2}^{2} -2\left(x_{t+j} - \beta_0 - \beta_1 j\right) = -2 \sum_{j=-2}^{2} \left(x_{t+j} - \beta_0 - \beta_1 j\right) \\
\frac{dS}{d\beta_1} &= \sum_{j=-2}^{2} \frac{d}{d\beta_1} \left(x_{t+j} - \beta_0 - \beta_1 j\right)^2 \\
&= \sum_{j=-2}^{2} -2j\left(x_{t+j} - \beta_0 - \beta_1 j\right) = -2 \sum_{j=-2}^{2} \left(j x_{t+j} - \beta_0 j - \beta_1 j^2\right).
\end{aligned} \tag{2.4}$$

We get similar normal equations as we did in the global quadratic case. We set the equations to 0 and add hats to the coefficients to indicate that they are now estimates, we get that the two coefficients satisfy

$$\begin{aligned}
-2 \sum_{j=-2}^{2} \left(x_{t+j} - \hat{\beta}_0 - \hat{\beta}_1 j\right) &= 0 \\
-2 \sum_{j=-2}^{2} \left(j x_{t+j} - \hat{\beta}_0 j - \hat{\beta}_1 j^2\right) &= 0.
\end{aligned} \tag{2.5}$$

This can be further simplified to

$$\begin{aligned}\sum_{j=-2}^{2} x_{t+j} &= \hat{\beta}_0 \sum_{j=-2}^{2} 1 + \hat{\beta}_1 \sum_{j=-2}^{2} j \\ \sum_{j=-2}^{2} j x_{t+j} &= \hat{\beta}_0 \sum_{j=-2}^{2} j + \hat{\beta}_1 \sum_{j=-2}^{2} j^2.\end{aligned} \tag{2.6}$$

We estimate the sums $\sum_{j=-2}^{2} 1 = 5, \sum_{j=-2}^{2} j = 0, \sum_{j=-2}^{2} j^2 = 10$. Therefore, we simplify 2.6 further

$$\begin{aligned}\sum_{j=-2}^{2} x_{t+j} &= 5\hat{\beta}_0 \\ \sum_{j=-2}^{2} j x_{t+j} &= 10\hat{\beta}_1.\end{aligned} \tag{2.7}$$

The trend estimate for a single fixed point can be found using the $\hat{\beta}_0$ coefficient, which is calculated using the following function

$$\hat{m_t} = \hat{\beta}_0 = \frac{1}{5}\left(x_{t-2} + x_{t-1} + x_t + x_{t+1} + x_{t+2}\right). \tag{2.8}$$

We could calculate $\hat{\beta}_1$ but in this case we do not need to because we are only interested in a single value to estimate the trend $\hat{m_t}$ at a fixed point.

Note that since we take an evenly spaced interval (of 5 observations) around a single point, to find the mean of the trend estimate at that point, we would not be able to use this method for the first two and last 2 observations. The iterative method would start at point $t = 3$ where we can draw the first two data points and calculate the average at that point, it is incremented once until the point $n - 2$ where we can use the last two data points, but would not be able to beyond that $n - 2$ value. Hence, the trend estimate $\hat{m_t}$ is defined for $t = 3, 4, \cdots, 502$.

b) Fitting a Local Linear Polynomial Model to a successive group of 5 observations

We expect that this method (symmetric simple moving average) would perform better than the global polynomial fit. However, the method also assumes some degree of smoothness in the trend of our observations, at least we expect smoothness around the 5 observations that help estimate a single point trend.

In fig. 2.1 the red line shows the trend of the fitted values. It looks like a better fit than the global method we saw earlier. However, from the figure we see that the fit describes the random noise as well as the trend. Therefore, there might be some degree of overfitting using this method because it follows the actual sales data extremely well. So the fit might have absorbed some random fluctuation. This can be problematic if we use this model to predict future sales values because it will introduce bias to the predicted values since the model is closely related to the sales data.

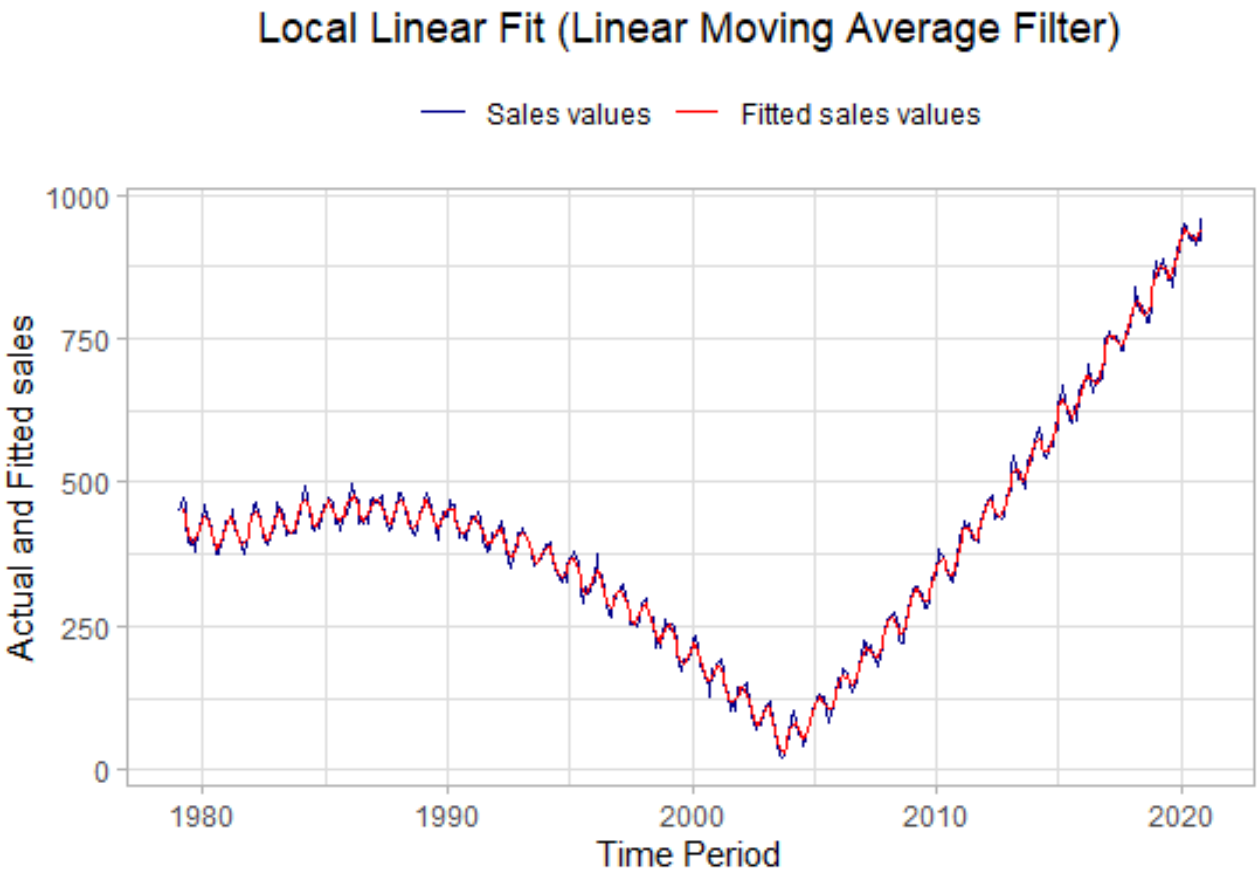


Figure 2.1: Local Linear Fit for 5 observations

Question 3

a) Model evaluation

In order to examine the goodness of fit of our model, we consider the **residual values**. The **residuals** are the deviations between the average value of our predicted or fitted value, and a particular observation from the sales data points. In other words: Residual = Observed value – Predicted value.
Let $\mathbf{e_t}$ be the residual value, then $\mathbf{e_t} \sim \mathbf{N}(\mathbf{0}, \sigma^\mathbf{2})$.

The residuals should follow the following properties:

- they sum to 0 $\sum_{i=1}^{n} e_t = 0$.
- the mean of the ith residual is 0. i.e. $E(e_t) = 0$.
- the variance of each residual is given by $Var[e_t] = \sigma^2 \left(1 - \frac{1}{n} - \frac{(x_t - \hat{x})^2}{S_{xx}}\right)$,

where n=504 and $S_{xx} = \sum_{i=1}^{n}(x_t - \hat{x})^2$ is the regression sum of squares.

Fig. 3.1 shows four different residual plots which will help visually analyse the fitted values against the raw data. The first plot (Fitted Values vs Residuals) is a simple scatterplot showing the residuals or errors on the y-axis and the predicted mean values for each observation on the x-axis. The plot shows a clear pattern, hence the assumptions of constant variance and linearity are not satisfied.

The second plot is a qq-plot or Quantile-Quantile plot. It shows the theoretical quantiles on the x-axis, i.e. where a residuals would be in terms of a standard normal distribution if the variance was known and the residuals come from a normal distribution. On the y-axis are the sample quantiles of the sample data points - our residuals. You can see some extreme deviations in the tails of the graph which suggests that the residuals do not follow a normal distribution.

The third plot is similar to the first plot, but the y-axis are the standardized residuals plotted against fitted values. It shows that there aren't any large outliers in the data but also confirms that the assumptions for our residuals are not met.

The fourth plot is a simple histogram of the residuals. The plot of the residuals doesn't follow a normal distribution, the bell shaped curve centered around 0 with some variability on each side.

To conclude, the simple quadratic polynomial fit is not a good fit for our Sales data and we should look for an alternative model.

In addition, from the ANOVA table in our summary output we can see that all the parameters $\hat{\beta}_0$, $\hat{\beta}_1$ *and* $\hat{\beta}_2$ are significant. Therefore, at the 5 % significance level, the hypothesis $H_0 : \beta_0 = \beta_1 = \beta_2 = 0$ is rejected in favour of the alternative hypothesis $H_1 : \beta_i \neq 0$ for i $\in 0, 1, 2$. That is, we conclude that the model is not a good fit.

```
summary(global_fit)

Call:
lm(formula = X ~ t + I(t^2))
```

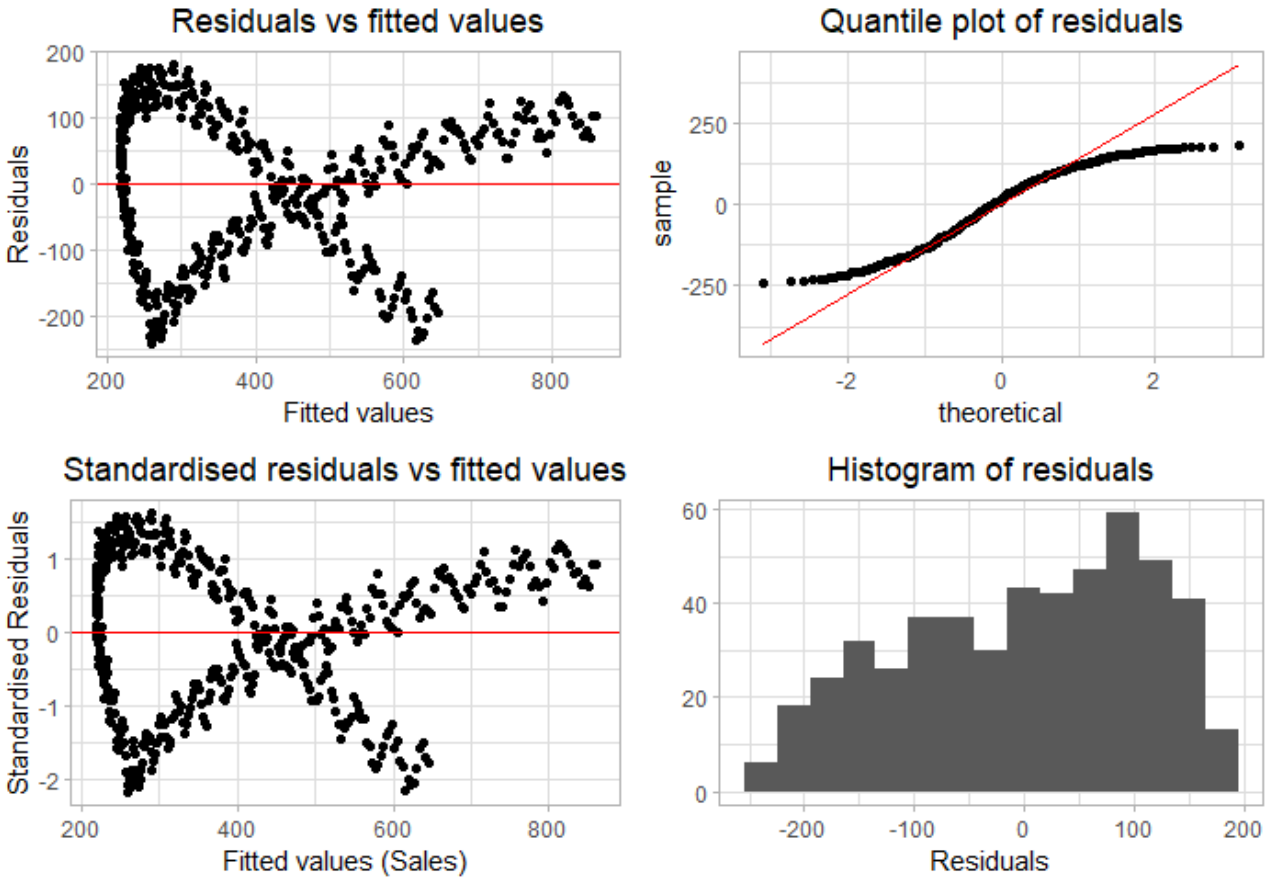


Figure 3.1: Evaluating the Global Linear Trend Model using different residual plots

```
Residuals:
    Min      1Q  Median      3Q     Max
-239.35  -92.28   13.16   93.35  180.30

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.490e+02  1.492e+01   43.50   <2e-16 ***
t           -3.793e+00  1.364e-01  -27.80   <2e-16 ***
I(t^2)       8.355e-03  2.616e-04   31.93   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 111.2 on 501 degrees of freedom
Multiple R-squared:  0.7013,Adjusted R-squared:  0.7001
F-statistic: 588.2 on 2 and 501 DF,  p-value: < 2.2e-16
```

b) Local Linear Polynomial Model Evaluation

The model seems to be a better fit for our data but we will still want to analyse it further. From fig.3.2 we see that the residual plots from the local linear fit method are better than the ones in (3.a). The assumptions made about the residuals that they are independent and normally distributed in section (3.a) look like they have been met.

From plot 1 (Residuals vs fitted values) and plot 3 (Standardized residuals vs fitted values), we can see that the residuals are scattered randomly against the fitted values in both cases, (somewhat) meeting the assumption that $e_t \sim N(0, \sigma^2)$.

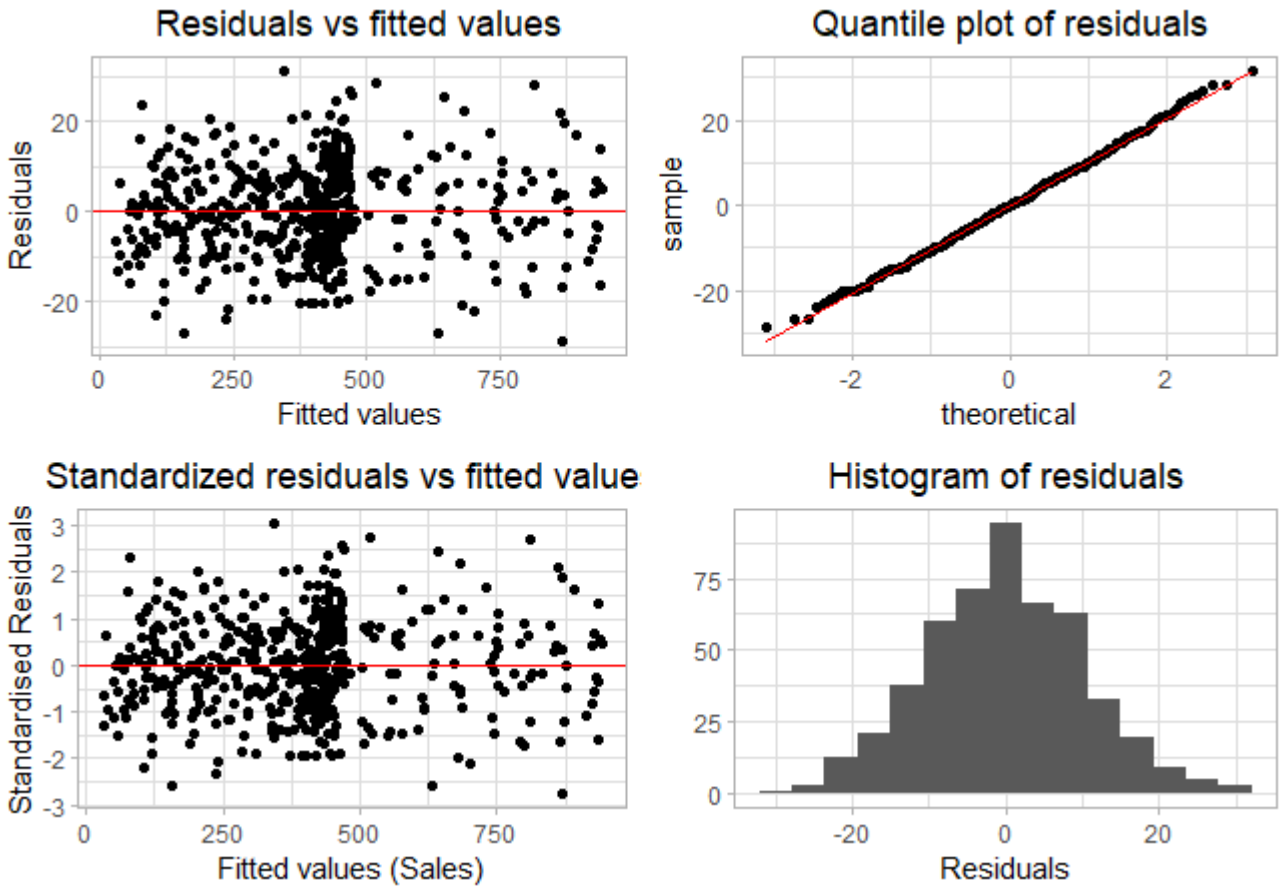


Figure 3.2: Residual plots for the local linear fit model evaluation

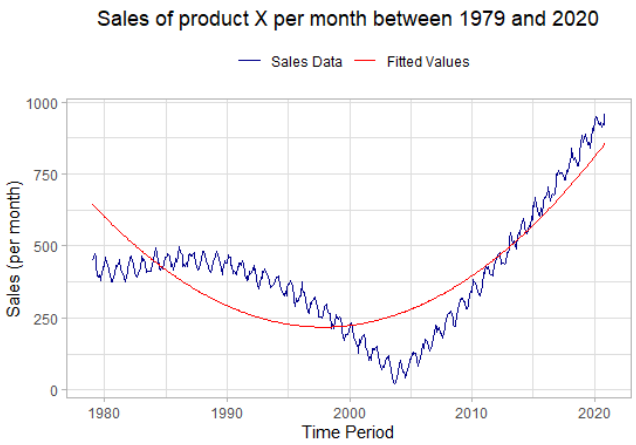
Plot 1 shows a significant reduction in the residual values from 200 and -200 using the global quadratic method to 20 and -20 using the local linear one.

Furthermore plot 3 shows some residuals which can be potential outliers because they lie beyond the values of 2 and -2 but the sales at those points could also be affected by seasonality which could explain the wide spread.

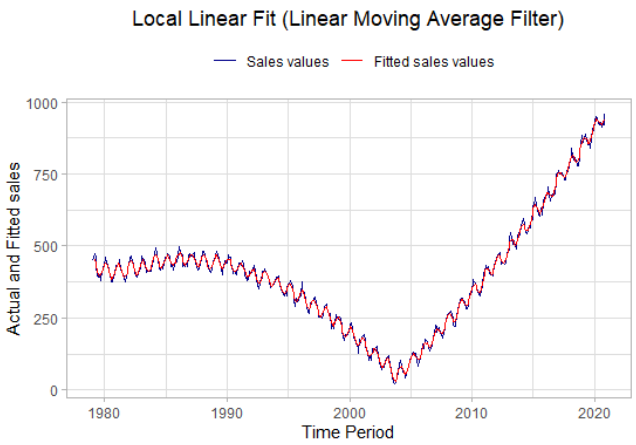
We should note that the residuals below the Predicted Sales of 500, are a lot more than the residuals beyond that sales point because of the shape of the raw sales data. It is slightly quadratic and the sales value only increases beyond 500 after the year 2010. Hence, plots 1 and 3 shows a good spread of residuals.

The qq-plot in plot 2, and the histogram in plot 4 show that the residuals in the local linear fit are coming from a normal distribution because there are no tails in the qq-plot and the histogram is centered at 0 and evenly spread assimilating a bell-shaped curve.

To sum up, from fig. 3.3a and 3.3b we see that the local linear fit is a better fit for predicting the trend of the company's sales data. For this reason, we will use the local linear fit to conduct further analysis to estimate the seasonality of the data in the next section.



(a) 1.a) Global linear model



(b) 2.b) Local linear model

Figure 3.3: Comparing the global linear and local linear models

Question 4

The **seasonal period** is the smallest positive integer such that a seasonal pattern recurs every **s** steps apart. The difference between seasonality and trend is that seasonality has a recurring pattern. For example, for the sales data we might have the seasonal period, **s=12** since we have monthly sales data.

We will use the time series $\mathbf{y_t}$ where $\mathbf{y_t} = \mathbf{x_t} - \hat{\mathbf{m}}_\mathbf{t}$ to calculate the seasonality estimates. The model becomes

$$Y_t = S_t + Z_t,$$

where the S_t describes the seasonal component and Z_t is a white noise process. We will look at seasonality estimates with period 12.

a) Obtaining the seasonality estimates

Assume that we have constant values within season and hence

$$S_1 = S_{s+1} = \cdots = S_{(k-1)s+1}, S_2 = S_{s+2} = \cdots = S_{(k-1)s+2}, \cdots, S_{12} = S_{s+12} = \cdots = S_{(k-1)s+12}. \tag{4.1}$$

This means that $n = ks = 504$ so that we have k=42 complete cycles of seasons. So $X_1, X_{s+1}, X_{2s+1}, \cdots, X_{(k-1)s+1}$ all have the same cycle, $X_2, X_{s+2}, X_{2s+2}, \cdots, X_{(k-1)s+2}$ all have the same cycle but different to the previous one, etc.

The model is simple because we have that $s = 12 < n = 504$. We write the stochastic process as

$$X_{js+1} = S_i + Y_{js+1} \; for \; i \in 1, 2, \cdots, 12, \; j \in 0, 1, \cdots, 41. \tag{4.2}$$

To find the optimal seasonality estimates, we need to minimize the sum of squares

$$\begin{aligned}
S &= \sum_{i=1}^{ks} (x_i - S_i)^2 \\
&= \sum_{j=0}^{k-1} (x_{js+1} - S_1)^2 + \sum_{j=0}^{k-1} (x_{js+2} - S_2)^2 + \\
&+ \sum_{j=0}^{k-1} (x_{js+3} - S_3)^2 + \sum_{j=0}^{k-1} (x_{js+4} - S_4)^2 + \\
&+ \sum_{j=0}^{k-1} (x_{js+5} - S_5)^2 + \sum_{j=0}^{k-1} (x_{js+6} - S_6)^2 + \\
&+ \sum_{j=0}^{k-1} (x_{js+7} - S_7)^2 + \sum_{j=0}^{k-1} (x_{js+8} - S_8)^2 + \\
&+ \sum_{j=0}^{k-1} (x_{js+9} - S_9)^2 + \sum_{j=0}^{k-1} (x_{js+10} - S_{10})^2 + \\
&+ \sum_{j=0}^{k-1} (x_{js+11} - S_{11})^2 + \sum_{j=0}^{k-1} (x_{js+12} - S_{12})^2 .
\end{aligned} \tag{4.3}$$

Differentiate $\mathbf{S}$ with respect to $S_1, S_2, \cdots, S_{12}$ and set the differentials to 0 to get the normal equations. We can see that S_1 is only contained in the first sum of squares, S_1 is in the second sum of squares and so on. The differentials are

$$\begin{aligned}
\frac{dS}{dS_1} &= \sum_{j=0}^{k-1} 2\left(x_{js+1} - S_1\right)(-1) \\
\frac{dS}{dS_2} &= \sum_{j=0}^{k-1} 2\left(x_{js+2} - S_2\right)(-1) \\
\frac{dS}{dS_3} &= \sum_{j=0}^{k-1} 2\left(x_{js+3} - S_3\right)(-1) \\
\frac{dS}{dS_4} &= \sum_{j=0}^{k-1} 2\left(x_{js+4} - S_4\right)(-1) \\
\frac{dS}{dS_5} &= \sum_{j=0}^{k-1} 2\left(x_{js+5} - S_5\right)(-1) \\
\frac{dS}{dS_6} &= \sum_{j=0}^{k-1} 2\left(x_{js+6} - S_6\right)(-1) \\
\frac{dS}{dS_7} &= \sum_{j=0}^{k-1} 2\left(x_{js+7} - S_7\right)(-1) \\
\frac{dS}{dS_8} &= \sum_{j=0}^{k-1} 2\left(x_{js+8} - S_8\right)(-1) \\
\frac{dS}{dS_9} &= \sum_{j=0}^{k-1} 2\left(x_{js+9} - S_9\right)(-1) \\
\frac{dS}{dS_{10}} &= \sum_{j=0}^{k-1} 2\left(x_{js+10} - S_{10}\right)(-1) \\
\frac{dS}{dS_{11}} &= \sum_{j=0}^{k-1} 2\left(x_{js+11} - S_{11}\right)(-1) \\
\frac{dS}{dS_{12}} &= \sum_{j=0}^{k-1} 2\left(x_{js+12} - S_{12}\right)(-1).
\end{aligned} \tag{4.4}$$

The normal equations are

$$
\begin{aligned}
-2\sum_{j=0}^{k-1}\left(x_{js+1}-\hat{S_1}\right)&=0\\
-2\sum_{j=0}^{k-1}\left(x_{js+2}-\hat{S_2}\right)&=0\\
-2\sum_{j=0}^{k-1}\left(x_{js+3}-\hat{S_3}\right)&=0\\
-2\sum_{j=0}^{k-1}\left(x_{js+4}-\hat{S_4}\right)&=0\\
-2\sum_{j=0}^{k-1}\left(x_{js+5}-\hat{S_5}\right)&=0\\
-2\sum_{j=0}^{k-1}\left(x_{js+6}-\hat{S_6}\right)&=0\\
-2\sum_{j=0}^{k-1}\left(x_{js+7}-\hat{S_7}\right)&=0\\
-2\sum_{j=0}^{k-1}\left(x_{js+8}-\hat{S_8}\right)&=0\\
-2\sum_{j=0}^{k-1}\left(x_{js+9}-\hat{S_9}\right)&=0\\
-2\sum_{j=0}^{k-1}\left(x_{js+10}-\hat{S_{10}}\right)&=0\\
-2\sum_{j=0}^{k-1}\left(x_{js+11}-\hat{S_{11}}\right)&=0\\
-2\sum_{j=0}^{k-1}\left(x_{js+12}-\hat{S_{12}}\right)&=0.
\end{aligned}
\tag{4.5}
$$

The estimates are

$$\begin{aligned}
\hat{S_1}\sum_{j=0}^{k-1}1 &= \sum_{j=0}^{k-1}x_{js+1}\\
\hat{S_2}\sum_{j=0}^{k-1}1 &= \sum_{j=0}^{k-1}x_{js+2}\\
\hat{S_3}\sum_{j=0}^{k-1}1 &= \sum_{j=0}^{k-1}x_{js+3}\\
\hat{S_4}\sum_{j=0}^{k-1}1 &= \sum_{j=0}^{k-1}x_{js+4}\\
\hat{S_5}\sum_{j=0}^{k-1}1 &= \sum_{j=0}^{k-1}x_{js+5}\\
\hat{S_6}\sum_{j=0}^{k-1}1 &= \sum_{j=0}^{k-1}x_{js+6}\\
\hat{S_7}\sum_{j=0}^{k-1}1 &= \sum_{j=0}^{k-1}x_{js+7}\\
\hat{S_8}\sum_{j=0}^{k-1}1 &= \sum_{j=0}^{k-1}x_{js+8}\\
\hat{S_9}\sum_{j=0}^{k-1}1 &= \sum_{j=0}^{k-1}x_{js+9}\\
\hat{S_{10}}\sum_{j=0}^{k-1}1 &= \sum_{j=0}^{k-1}x_{js+10}\\
\hat{S_{11}}\sum_{j=0}^{k-1}1 &= \sum_{j=0}^{k-1}x_{js+11}\\
\hat{S_{12}}\sum_{j=0}^{k-1}1 &= \sum_{j=0}^{k-1}x_{js+12}.
\end{aligned} \tag{4.6}$$

We know that $\sum_{j=0}^{k-1} 1 = 42$. So we have that the estimates for the seasonal components with period 12 are:

$$\begin{aligned}
\hat{S}_1 &= \frac{1}{42}\sum_{j=0}^{k-1} x_{js+1} \\
\hat{S}_2 &= \frac{1}{42}\sum_{j=0}^{k-1} x_{js+2} \\
\hat{S}_3 &= \frac{1}{42}\sum_{j=0}^{k-1} x_{js+3} \\
\hat{S}_4 &= \frac{1}{42}\sum_{j=0}^{k-1} x_{js+4} \\
\hat{S}_5 &= \frac{1}{42}\sum_{j=0}^{k-1} x_{js+5} \\
\hat{S}_6 &= \frac{1}{42}\sum_{j=0}^{k-1} x_{js+6} \\
\hat{S}_7 &= \frac{1}{42}\sum_{j=0}^{k-1} x_{js+7} \\
\hat{S}_8 &= \frac{1}{42}\sum_{j=0}^{k-1} x_{js+8} \\
\hat{S}_9 &= \frac{1}{42}\sum_{j=0}^{k-1} x_{js+9} \\
\hat{S_{10}} &= \frac{1}{42}\sum_{j=0}^{k-1} x_{js+10} \\
\hat{S_{11}} &= \frac{1}{42}\sum_{j=0}^{k-1} x_{js+11} \\
\hat{S_{12}} &= \frac{1}{42}\sum_{j=0}^{k-1} x_{js+12},
\end{aligned} \tag{4.7}$$

which means that the best seasonal component estimates are the averages of all the time points in a season **i**.

b) Calculating the seasonal estimates

We can create the seasonal averages for a local linear trend by excluding the first two and last two sales values from the seasonal estimate calculation because those values are set as **NA**.

```
> ## define a vector that contains the local linear fit residuals,
> ## excluding the first and last two data points as they are NA
> X <- df$local_residual[3:502]

> ## create an indicator value
> ind <- rep(1:12,42)
## repeat values from 1 to 12, 42 times - create a vector of 504 indicator values

> ## Exclude the first two and last two data points from our indicator vector
> ind <- ind[3:502]

> ## take the average over all values with the same indicator variable
> avr_values <- c( mean(X[ind == 1]), mean(X[ind == 2]), mean(X[ind == 3]), mean(X[ind == 4]),
mean(X[ind ==5]), mean(X[ind == 6]), mean(X[ind == 7]), mean(X[ind == 8]), mean(X[ind == 9]),
mean(X[ind == 10]), mean(X[ind == 11]), mean(X[ind == 12]))
```

```
> ## Create seasonal estimates by repeating the averages k=42 number of times
> sfit <- rep(avr_values, 42)

> df$seasonal_fit <- sfit

> head(df,5)

  global_fit sales       time  residual linear_polynomial local_residual local_std_res seasonal_fit
1   645.2127   451 1979-01-01 -194.2127                NA             NA            NA     2.239024
2   641.4443   456 1979-02-01 -185.4443                NA             NA            NA     5.502439
3   637.6926   474 1979-03-01 -163.6926             453.8           20.2     1.9586434     9.714286
4   633.9576   460 1979-04-01 -173.9576             444.0           16.0     1.5514007     6.671429
5   630.2393   428 1979-05-01 -202.2393             431.6           -3.6    -0.3490652     2.295238
   seas_res Seas_std_res local_and_seasonal
1        NA           NA                NA
2        NA           NA                NA
3 10.485714    1.2143975          463.5143
4  9.328571    1.0803836          450.6714
5 -5.895238   -0.6827539          433.8952
```

c) Examine the goodness of fit

We can now examine the goodness of for the seasonal estimates by plotting the values.

Fig. 4.1 shows that the seasonality estimates model the peaks and troughs of the residual data quite well. But

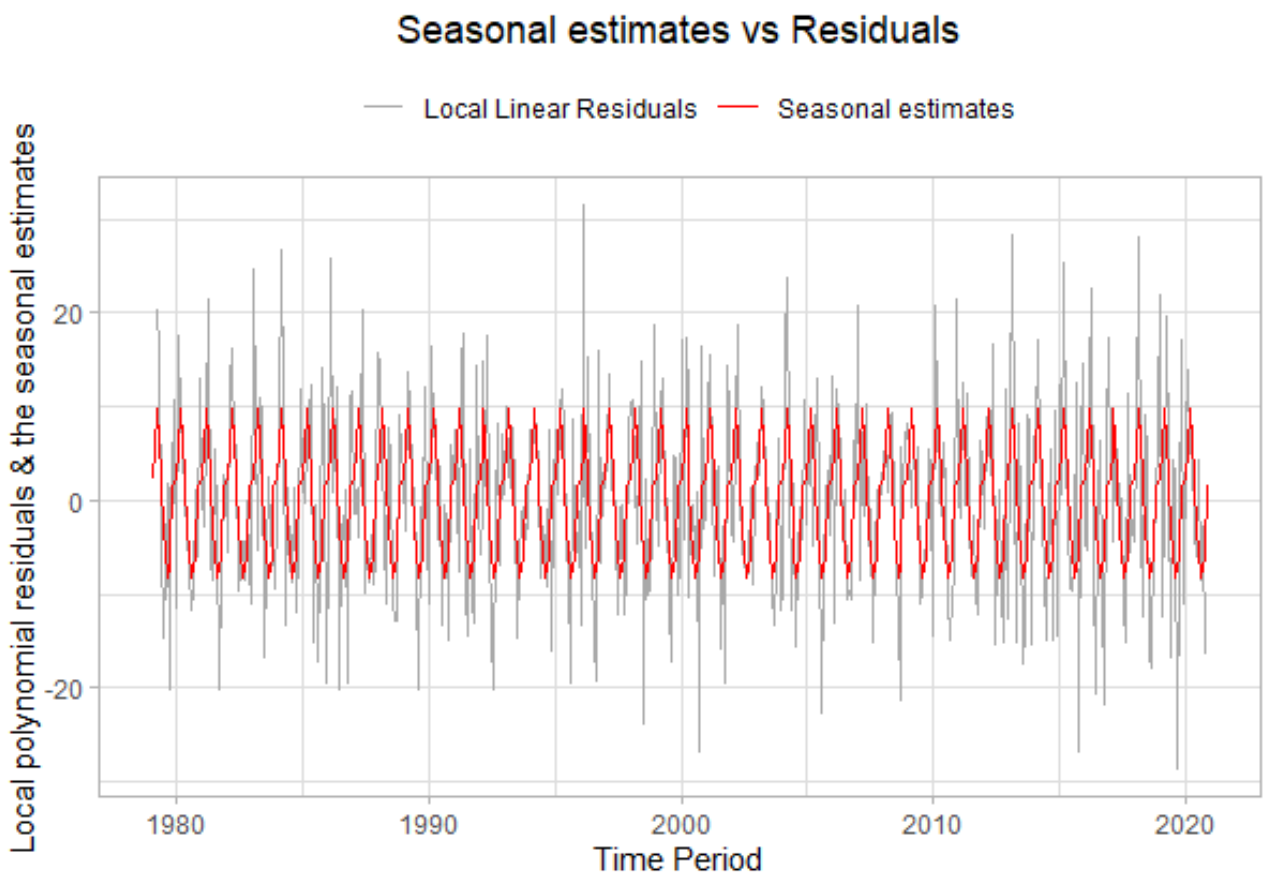


Figure 4.1: Time series Y_t against the seasonal estimates S_t

not all values are close to the seasonal estimates, there are still some much larger peaks. We can further analyze the 'leftover' residuals after we have fitted the seasonality estimates. The plots show the remaining residuals that come from the taking the trend and seasonality estimates from the data points. So, we are looking at the

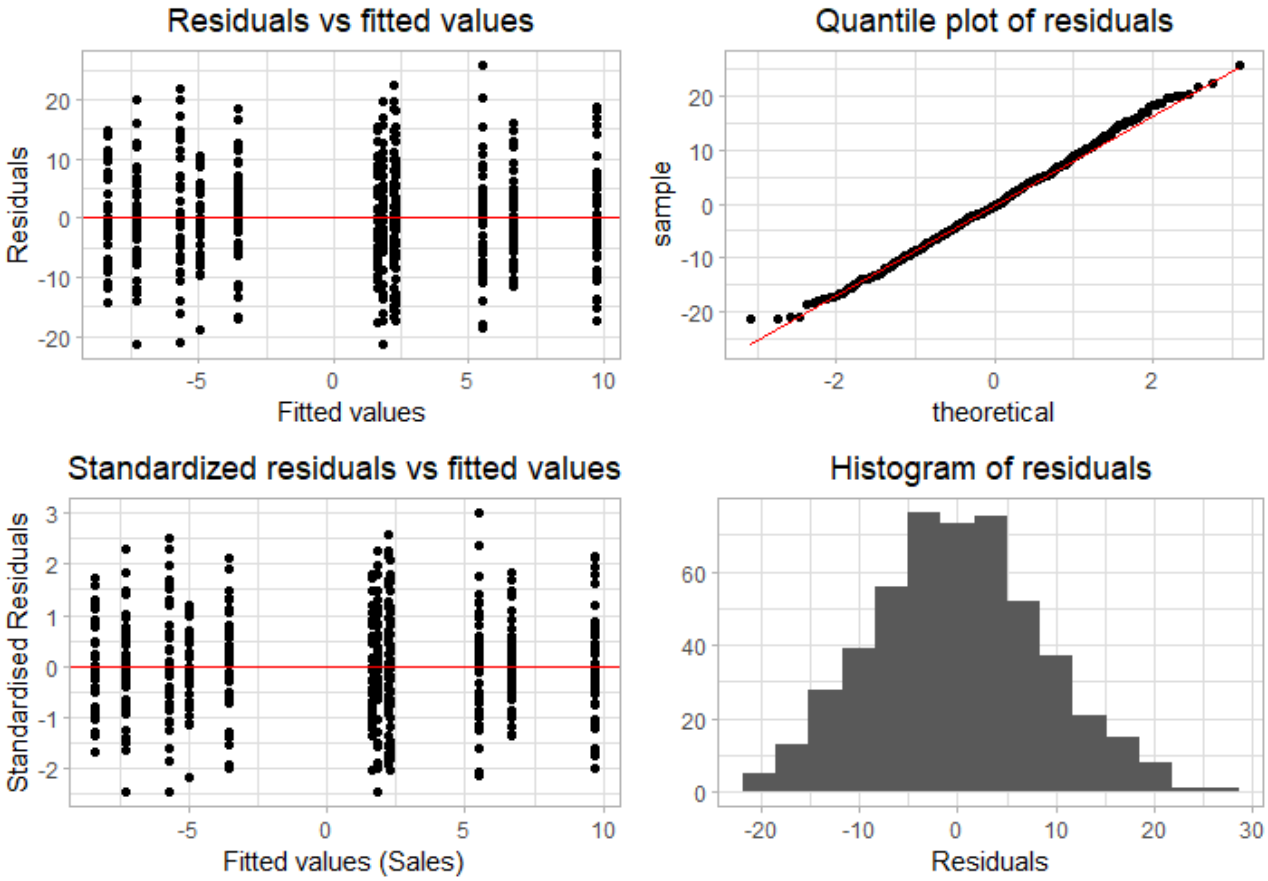


Figure 4.2: Further residual analysis for seasonality estimates

random variables given by

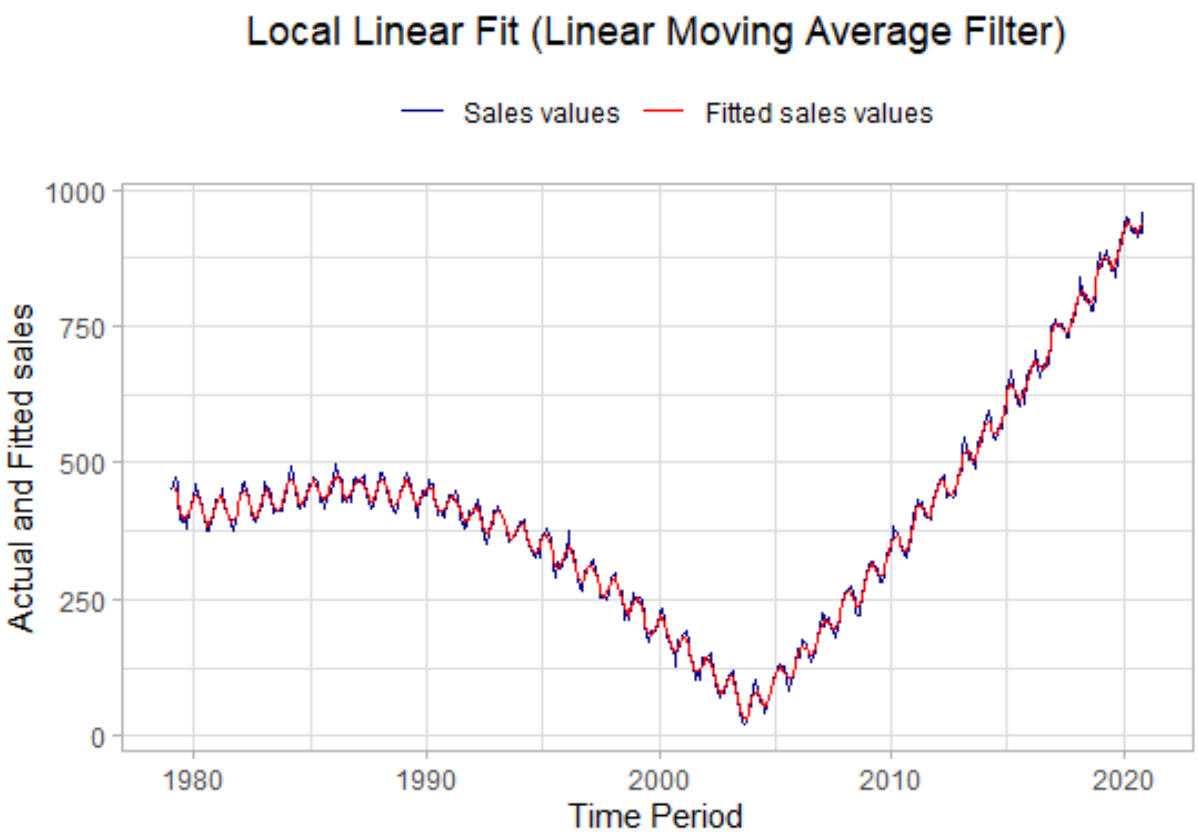
$$Z_t = X_t - \hat{m}_t - \hat{S}_t \tag{4.8}$$

The residuals vs. the fitted values plot as well as the standardized residuals vs fitted values show the seasonality element taken away from the remaining time series data without a trend. As a result of being periodic with **s=12**, the residuals are formed into column-like groups in the graphs. However, the residuals in both plots seem to be nearly evenly spread out around the red horisontal line around zero. There also seems to be a lot of points that go beyond the [-2,2] horisontal lines which indicates that they are outliers. This might suggest that the product has out-performed or under-performed a few times over the years.

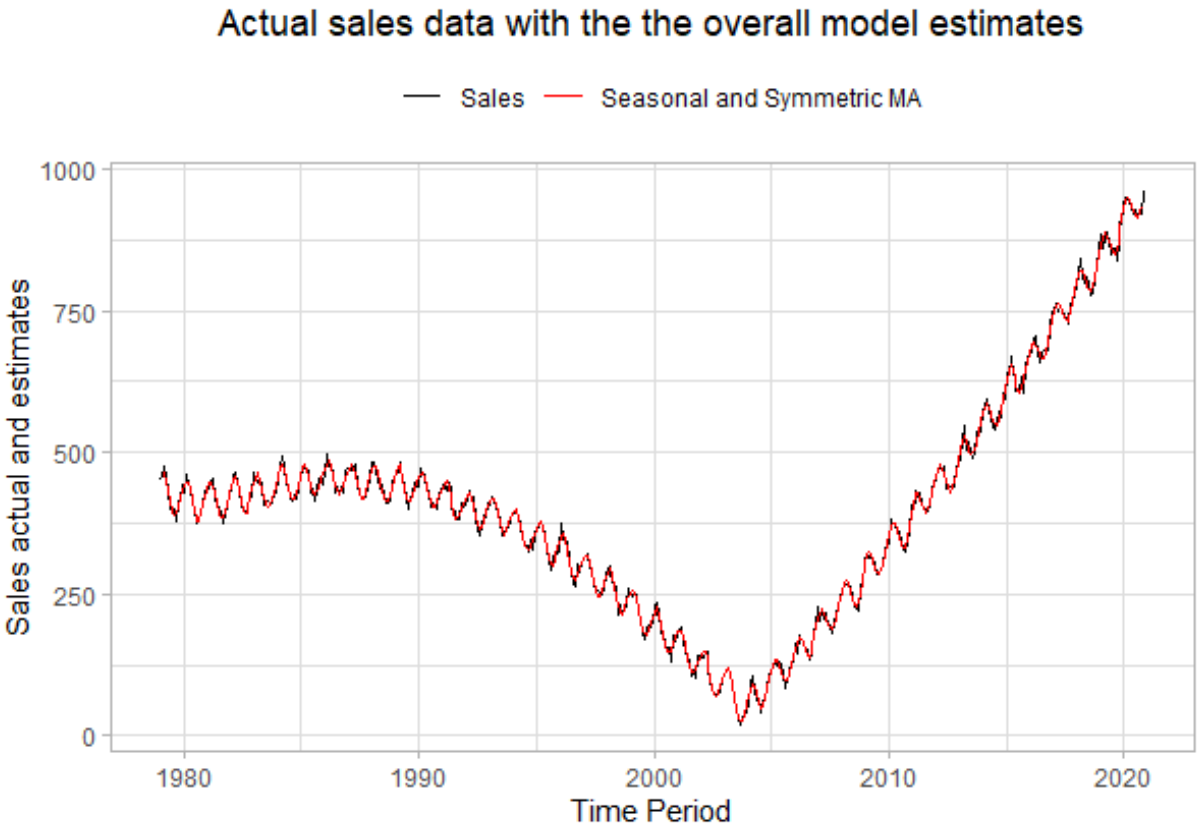
From the qq-plot, the standardized residuals vs fitted values plot and the histogram we can see some potential outlier points. We can see that the sample quantile follows the theoretical quantile line on the qq-plot, apart from a few points in the tails, indicating that overall, the leftover residuals also seem to come from the normal distribution.

The histogram also shows a small tail on the right side where a point could be an outlier. In addition, the peak around zero does not seem to strictly follow a bell-shaped curve with a peak at zero, but the leftover residuals could still be coming from a normal distribution although, further analysis should be done.

In summary, we can plot the local linear fit and seasonal estimation together to see our overall model for the sales data. Fig. 4.3b shows the final fit after using the question 2 local linear fit and seasonality from question 4. Comparing it to the fit from question 2 we can see that the red line follows the extreme values much better within the 12-month cycles.



(a) 2) Actual sales vs. local linear fit



(b) 4) Actual Sales vs. fitted values using the symmetric MA and seasonality

Figure 4.3: Model before seasonality fit vs. after fitting seasonality

Question 5

In this section, we will have a look at the data in more depth and try to fit one more model, evaluate it and suggest whether or not it is a better fit than the ones we have seen so far.

a) Higher order global polynomial fits

Suppose we increase the number of parameters in the global polynomial fit. For example, we can create polynomials of order 3,4,5 and 6 which have the trend equations

$$\begin{aligned} m_t &= x_t + \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3 \\ m_t &= x_t + \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3 + \beta 4 t^4 \\ m_t &= x_t + \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3 + \beta 4 t^4 + \beta 5 t^5 \\ m_t &= x_t + \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3 + \beta 4 t^4 + \beta 5 t^5 + \beta 6 t^6 \end{aligned} \tag{5.1}$$

To find the optimal parameter estimation, we use the least squares method where we construct the sum of squares for each higher order polynomial

$$\begin{aligned} S(\beta_0, \beta_1, \beta_2, \beta_3) &= \sum_{t=1}^{504} \left(x_t - \beta_0 - \beta_1 t - \beta_2 t^2 - \beta_3 t^3 - \beta 4 t^4\right)^2 \\ S(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4) &= \sum_{t=1}^{504} \left(x_t - \beta_0 - \beta_1 t - \beta_2 t^2 - \beta_3 t^3 - \beta 4 t^4\right)^2 \\ S(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5) &= \sum_{t=1}^{504} \left(x_t - \beta_0 - \beta_1 t - \beta_2 t^2 - \beta_3 t^3 - \beta 4 t^4 - \beta 5 t^5\right)^2 \\ S(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6) &= \sum_{t=1}^{504} \left(x_t - \beta_0 - \beta_1 t - \beta_2 t^2 - \beta_3 t^3 - \beta 4 t^4 - \beta 5 t^5 - \beta 6 t^6\right)^2, \end{aligned} \tag{5.2}$$

where we differentiate each model with respect to each parameter value in the model, set to zero and solve the normal equations in order to find the best parameter estimates. I will omit the working out of the estimates for each parameter and will use R's in-built function '*lm()*' instead, to quickly compute and compare models of higher order polynomials. I will instead plot the predicted values in order to briefly assess the fits.

Fig. 5.1 shows the actual sales points as well as the predicted/fitted values of the higher order polynomial fits. The higher order fits look much better than the global quadratic fit in Question 1, represented by the red line. However, we can also see that adding more parameters to the global cubic fit in the green line doesn't change the predicted values too much. The orders 3,4,5 and 6 look similar but the one that fits the data the most seems to be the purple line which is the sextic polynomial.

Let's have a look at the ANOVA summary of the sextic polynomial fit.

```
> summary(higher_order_fit_4)

Call:
```

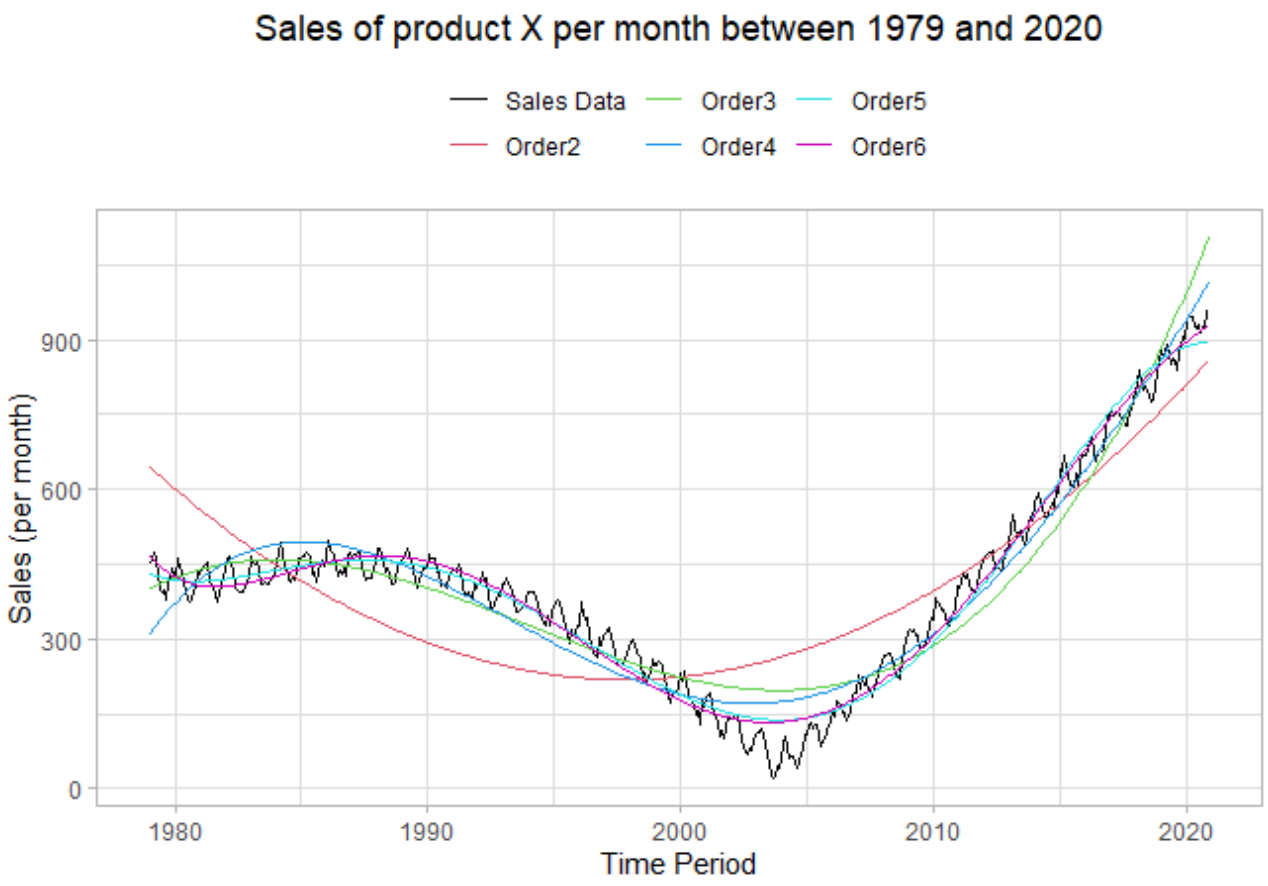


Figure 5.1: Comparing the fit of global polynomials or higher orders

```
lm(formula = X ~ t + I(t^2) + I(t^3) + I(t^4) + I(t^5) + I(t^6))

Residuals:
     Min       1Q   Median       3Q      Max
-112.853  -19.992    1.285   22.865   79.272

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.714e+02  1.055e+01  44.701  < 2e-16 ***
t           -4.859e+00  5.758e-01  -8.439 3.53e-16 ***
I(t^2)       1.124e-01  9.899e-03  11.352  < 2e-16 ***
I(t^3)      -9.005e-04  7.349e-05 -12.253  < 2e-16 ***
I(t^4)       2.948e-06  2.645e-07  11.145  < 2e-16 ***
I(t^5)      -4.152e-09  4.546e-10  -9.133  < 2e-16 ***
I(t^6)       2.104e-12  2.990e-13   7.035 6.62e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 33.02 on 497 degrees of freedom
Multiple R-squared:  0.9739,Adjusted R-squared:  0.9736
F-statistic:  3088 on 6 and 497 DF,  p-value: < 2.2e-16
```

From the ANOVA table we can see that all parameters in the sextic polynomial fit are significant. Therefore, all parameters are needed to fit the data well. We can see that the square root of the mean square residuals or the residual standard error has decreased from 111.2 in the global quadratic fit to 33.02 in the sextic fit as well as an increase in the R^2 or the coefficient of determination statistic from 0.7013 to 0.9739, indicating that the sextic polynomial fit is better. It could be better than the local linear fit, as it doesn't overfit the data, i.e. there is less noise accounted for in the model compared to the local linear fit in Question 2.

In fig. 5.2 we can see that the fitted values and the residuals seem to have some form of correlation. Therefore,

Figure 5.2: Global sextic polynomial fit evaluation

the residuals are not completely random and so the fit is not good even though the R^2 metric is high and the sum of squared errors are quite small compared to the global quadratic fit.

The Q-Q plot and the histogram both suggest that the residuals have some outliers and do not come from the normal distribution as the tails in the qq-plot are below the line and there larger negative residuals in the histogram.

We can conclude that sextic polynomial fit is not a good fit for our data and so we do not fit the seasonal estimates to this model.

b) New model proposal: EWA

In this section, I propose the Exponentially Weighted Average (EWA) model. The model has the least restrictions on the trend estimate but it also introduces lag estimates. This means that f the trend changes quickly the EWA model can lag behind. Suppose you have the stochastic process X_t then the EWA model is

$$\hat{m_t} = \alpha X_t + (a - \alpha)\hat{m_{t-1}} \ for \ 1 \leq 5n, \tag{5.3}$$

where n=504 is the number of sales data points and we introduce a smoothing parameter $\alpha \in (0, 1)$. The reason the model is called an exponentially weighted average can be seen once we expand the model

$$\begin{aligned}
\hat{m_t} &= \alpha X_t + (1-\alpha)\hat{m_{t-1}} \\
&= \alpha X_t + (1-\alpha)(\alpha X_{t-1} + (1-\alpha)\hat{m_{t-2}}) \\
&= \alpha X_t + (1-\alpha)\alpha X_{t-1} + (1-\alpha)^2 \hat{m_{t-2}} \\
&= \alpha X_t + \alpha(1-\alpha) X_{t-1} + \alpha(1-\alpha)^2 X_{t-2} + (1-\alpha)^3 \hat{m_{t-3}} \\
&\cdots \\
&= \alpha X_t + \alpha(1-\alpha) X_{t-1} + \cdots + \alpha(1-\alpha)^{t-1} X_1 + (1-\alpha)^t \hat{m}_0.
\end{aligned} \tag{5.4}$$

We choose a starting point m_0 which is the initial trend value and the weight α. Note the effect of the initial trend value will only be see in the first few estimates but it eventually becomes stable. The α is a form of a weighting factor which decreases older values of X_i exponentially. So for example, the weight of X_1 in the last expansion in the 5.4 is quite small and has a very small affect on the $\hat{m}_t$ estimate. In other words, X_t has weight α, X_{t-1} has weight $\alpha(1-\alpha)$, X_{t-2} has weight $\alpha(1-\alpha)^2, \cdots$. We can use geometric series to show the sum of the weights adds to 1. Let $a = \alpha$, $r = (1-\alpha)$, then the sum of the series becomes

$$S_n = \frac{\alpha - \alpha(1-\alpha)^t}{1-(1-\alpha)} \tag{5.5}$$

Hence the sum is $S_n = 1-(1-\alpha)^t = 1$. The advantages of using this model are that it is less affected by random noise and outlier values for a small α but the opposite is true for a larger α. The trend estimates changes faster with time for a large value of α but it is slow to change with a smaller one. In this case we can choose the value pf $alpha = 0.3$ and the starting value of the trend $\hat{m}_0 = X_0$.

c) Computing and evaluating the model

The code for computing the EWA estimates for different values of α is in the Code Section under Question 5 part c).

Fig. 5.3 shows the model estimates for the values $\alpha = 0.3, \alpha = 0.5, \alpha = 0.7, \alpha = 0.9$. From the plot we

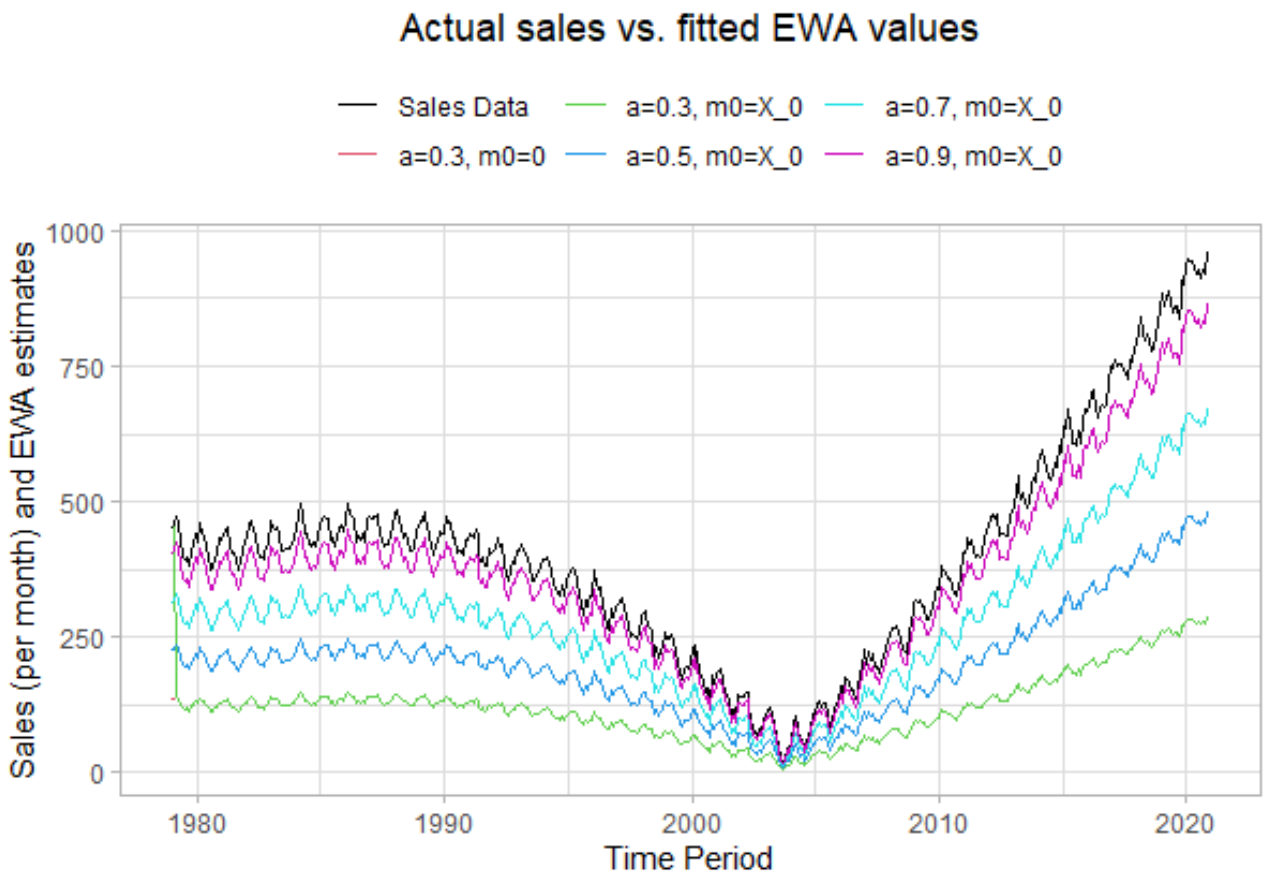


Figure 5.3: Comparing the EWA model for different values of α

can see straight away that no EWA fit is a good fit because they all sit below the raw sales data. This will skew the residuals and where they will sit below the line of $y = 0$, making the residuals skewed. Although, the higher the alpha, the better the model becomes, there is a risk of overfitting the data using a higher value of α.

Therefore, we can conclude that the EWA model is also not a good fit for the Sales data without computing the residuals nor fitting the seasonality estimates.

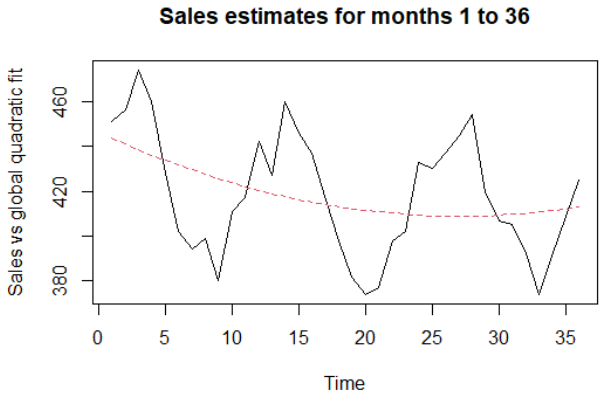
d) Final comments and conclusion

In summary, the local linear trend seems to fit the sales data quite well compared to all the models suggested. The residuals are scattered, and the histogram shows the best spread around the mean zero. So the local linear trend with the seasonal estimates provides a good model of the sales data which we can use to predict future sales.

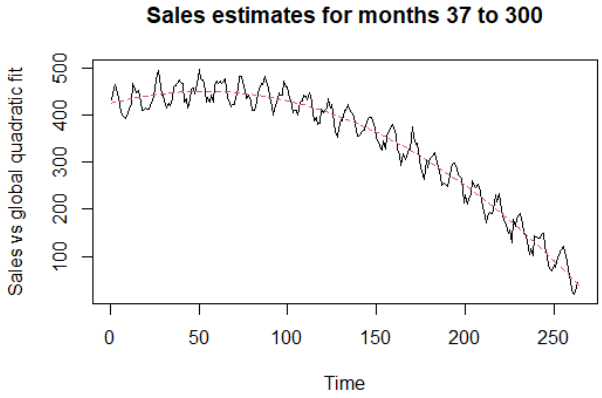
However, we can also see that the trend changes a few times over the time period and fitting one trend might not be good enough if market conditions change again. The current local linear trend can predict an increase in sales but if the consumer interest changes or a competitor enters the market, a new model will be needed.

We can briefly examine the three obvious parts of the sales data. Fig. 5.4a, 5.4b and 5.4c show simple models that fit parts of the sales data that look like they come from the same trend. For example, 5.4a shows a global quadratic fit for the months 1 to 36. All the fits look quite good in terms of following the individual trends of the sales. They also seem to not overfit the data, i.e. they are not modelling the noise, something the local linear fit could be doing. In fig. 5.5 we can see how the individual parts add up to the whole graph and fig.5.6 shows the combined residuals plots for evaluation.
Fig. 5.6 shows evenly spread residuals but the q-q plot and histogram suggest the residuals are nor exactly coming from the normal distribution. Although, evaluating it all together suggests the overall residuals might no be normally distributed, we can see in fig. 5.4a, 5.4b and 5.4c that the trend estimates follow the sales data quite well without overfitting it.

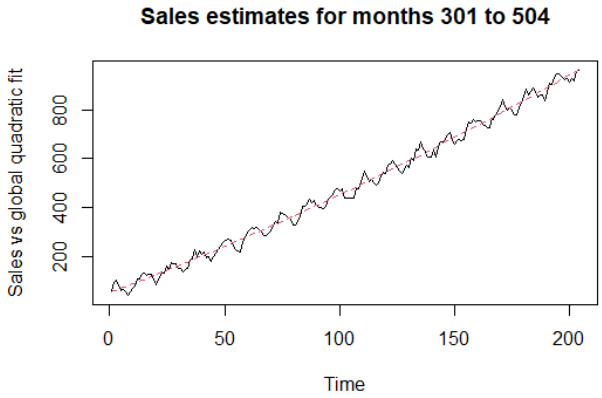
Therefore, as a conclusion I would suggest using the local linear trend with the seasonality estimates if one wants to model and make predictions using all the sales data in the short term. However, splitting the data up where there is a new trend forming and modelling each one might give a better overall prediction in the short term without taking noise into account. This could be used to analyse general sales estimates whereas the local linear model can be used to help the supply chain of the firm.



(a) Sales data from month 1 to 36, global quadratic fit



(b) Sales data from month 37 to 300, global quadratic fit



(c) Sales data from month 301 to 504, global quadratic fit

Figure 5.4: Fitting global quadratic models to parts of data

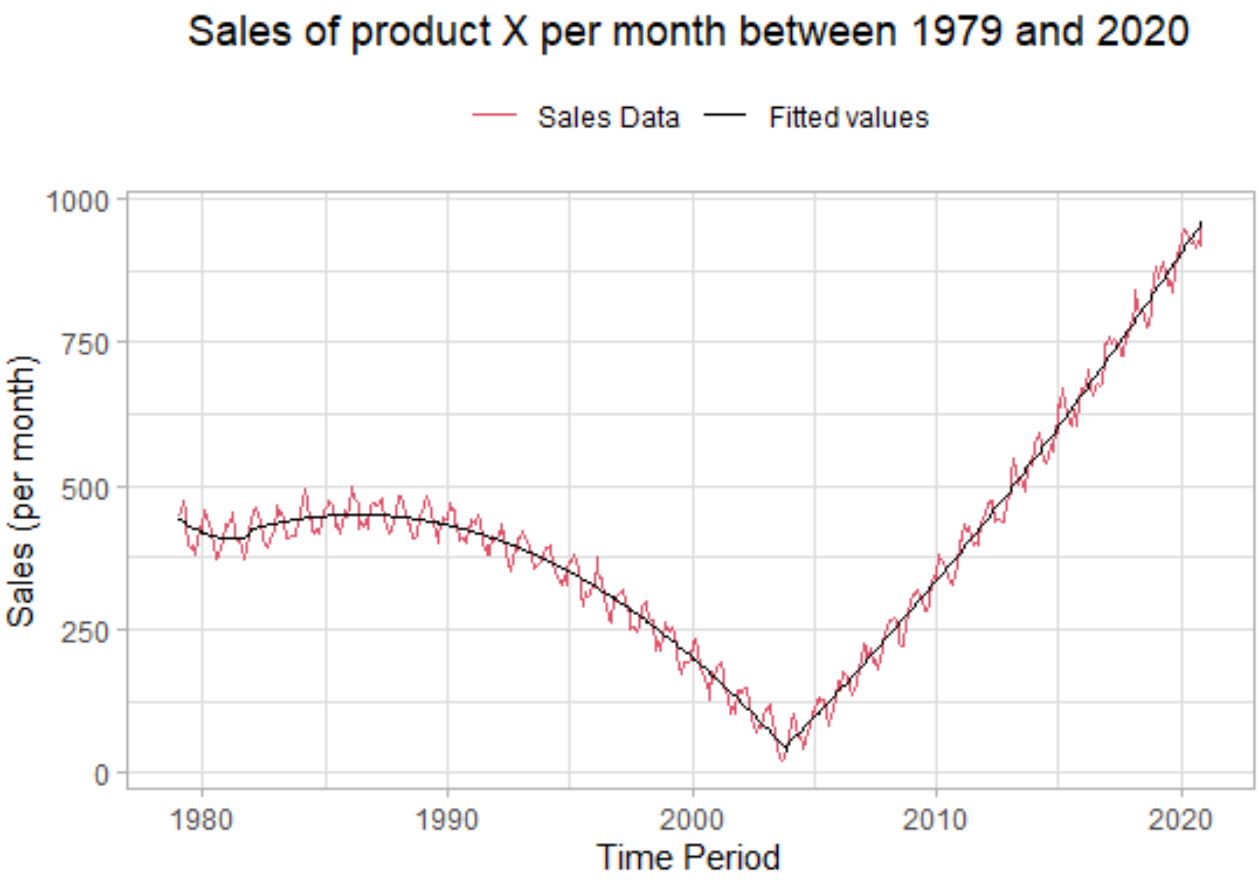


Figure 5.5: Overall model of the trend parts

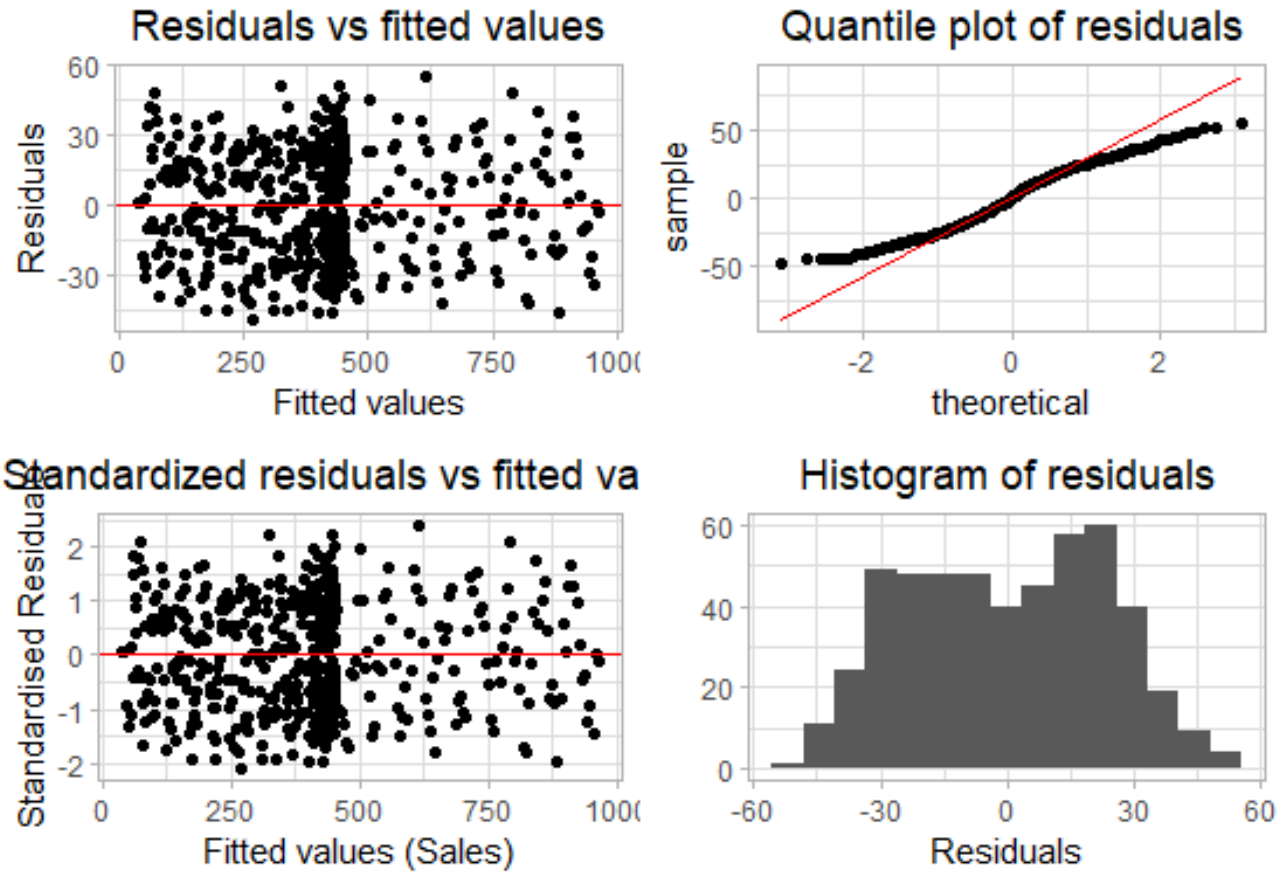


Figure 5.6: Model evaluation of individual residuals

Code

```
# Question 1
## Initial Plot - Plot time series data only
## ----------------------------

## Reading in out data from a text file.
data <- read.table("~/University work/02 Statistical Methods/03.
Coursework/Assessed Data 20-21.txt", col.names = c("sales"))

## Assigning data to variables A and n
X <- data$sales  ## a time series vector from the sales column
n <- length(X) # the length of the vector X

## Importing packages
library(ggplot2)
library(dplyr)
library(ggeasy)  ## package to help me centre the title of the chart

# Creating a dataframe with Sales and date values
ts_data <- data.frame(
  months = seq(as.Date("1979-01-01"), as.Date("2020-12-31"), by="months"), # adding dates
  Sales = X) # the Sales data

# Using the ggplot package to create a plot using the dataframe
p <- ggplot(ts_data, aes(x=months, y=Sales)) + ## plot sales (=y) against time (=x)
  geom_line( color="darkblue") +  ## in the colour blue
  ## geom_point(pch=20) + I don't want to clutter the chart so I won't add the points
  xlab("Time Period") + ## labeling the x axis
  ylab("Sales (per month)") +
  ggtitle("Sales of product X per month between 1979 and 2020") +  ## add title
  theme(plot.title = element_text(size=14, hjust = 'right', face="bold.italic")) + # main title
  theme_bw() +  ## white background
  theme_light() + ## light background lines
  scale_x_date(date_labels = "%Y") +  ## add only the years of the time period
  ggeasy::easy_center_title()  ## centered title

## plotting the data
p

1. b) The Global Polynomial fit
## --------------------------

## we create a range of values from 1 to the length as X as T
t <- 1:n
## here we fit the linear model
global_fit <- lm(X~t + I(t^2))
## output the global linear fit short summary
```

```
global_fit




>>
Call:
lm(formula = X ~ t + I(t^2))

Coefficients:
(Intercept)            t       I(t^2)
 648.997860    -3.793496     0.008355

## 1.b) Visualising the fitted data

## save predictions of the model, actual Sales and time stamp in the new data frame
## we want to plot against the time

df <- data.frame(global_fit = predict(global_fit, ts_data), sales=X,
time=seq(as.Date("1979-01-01"), as.Date("2020-12-31"), by="months"))

## show first 4 rows of our dataframe
## head(df,3)

# Create plot p using ggplot2 and ggeasy libraries we imported above

p <-ggplot(data = df, aes(x = time, y = sales)) +
  geom_line(aes( y= sales, color="Sales Data")) +  ## plot Sales against time
  geom_line(data = df, aes(x=time, y=global_fit, color='Fitted Values')) +
  ## plot fitted values against time
  xlab("Time Period") + ## labeling the x axis
  ylab("Sales (per month)") +
  ggtitle("Sales of product X per month between 1979 and 2020") +
  ## add title
  theme(plot.title = element_text(size=14, hjust = 'right', face="bold.italic")) + # main title
  theme_bw() +  ## white background
  theme_light() +  ## light background lines
  theme(legend.position="top") +
  ggeasy::easy_center_title()  + ## centered title
  scale_colour_manual("",
                      breaks = c("Sales Data", "Fitted Values"), ## legend names
                      values = c("darkblue", "red")) ## define legend colors

p

######################################################################
# Question 2
## 2.b) Local linear fit code
###----------------------------------------

## Code for the iterative local linear fit

X <- df$sales  ## a time series vector from the sales column
n <- length(X) # the length of the vector X

## vector to store the trend estimates
vect <- (1:n)

## Setting the first two and last two to 'NA' because they would not be calculated
vect[1] <- NA; vect[2] <- NA; vect[n-1] <- NA; vect[n] <- NA
```

```
## Set starting point of iteration
t <- 3

## iteration to update our empty vector with the trend estimates

## the while loop does calculation of trend estimate at each point in the time series data
## it starts at t=3 and loops/increments until t reaches n-1=503th point
while(t < n-1)
{
  ## Step 1
  vect[t] <- (X[t-2]+X[t-1]+X[t]+X[t+1]+X[t+2] ) / 5
  ## calculating the moving average at each fixed point
  ## and updating our empty vector

  ## Step 2
  t <- t+1 ## go to the next point t+1 and repeat step 1
}

####--- 2.b) Local linear fit plot

## Add new vector to our dataframe so we can plot our data
df$linear_polynomial <- vect

# Create plot p using ggplot2 and ggeasy libraries we imported above

p <-ggplot(data = df, aes(x = time, y = sales)) +
  geom_line(aes( y= sales, color="Sales values")) +  ## plot Sales against time
  geom_line(data = df, aes(x=time, y=linear_polynomial, color='Fitted sales values')) +
  ## plot fitted values against time
  xlab("Time Period") + ## labeling the x axis
  ylab("Actual and Fitted sales") +
  ggtitle("Local Linear Fit (Linear Moving Average Filter)") +  ## add title
  theme(plot.title = element_text(size=10, hjust = 'right', face="bold.italic")) + # main title
  theme_bw() +  ## white background
  theme_light() +  ## light background lines
  theme(legend.position="top") +
  ggeasy::easy_center_title()  + ## centered title
  scale_colour_manual("",
                      breaks = c("Sales values", "Fitted sales values"), ## legend names
                      values = c("darkblue", "red")) ## define legend colors
## 'print' the plot
p

#########################################################################
# Question 3

# QS3: Evaluating the global model in QS1
###---------------------------------------

## package that let's us have a grid of plots
library(cowplot)

## Add residuals to dataframe
df$residual <- global_fit$residuals

# Make a scatter plot of residuals against fitted values
p1 <- qplot(global_fit$fitted, global_fit$residuals, geom = "point") +
  geom_abline(intercept = 0,slope = 0,colour = "red") +
  ##
  xlab("Fitted values") + ## labeling the x axis
  ylab("Residuals") +
```

```
  ggtitle("Residuals vs fitted values") +  ## add title
  theme(plot.title = element_text(size=4, hjust = 'right', face="bold.italic")) + # main title
  theme_bw() +  ## white background
  theme_light() +  ## light background lines
  ggeasy::easy_center_title()

# Make a quantile-quantile plot
p2 <-ggplot(data = df,aes(sample = residual)) +
  geom_qq() +
  geom_qq_line(colour = "red") +
  labs(title = "Quantile plot of residuals") +
  theme_bw() +  ## white background
  theme_light() +  ## light background lines
  theme(legend.position="top") +
  ggeasy::easy_center_title()

p3 <- qplot(global_fit$fitted, rstandard(global_fit), geom = "point") +
  geom_abline(intercept = 0,slope = 0,colour = "red") +
  labs(title = "Standardised residuals vs fitted values", x = "Fitted values (Sales)",
  y = "Standardised Residuals") +
  theme_bw() +  ## white background
  theme_light() +  ## light background lines
  ggeasy::easy_center_title()

# Make a histogram of the residuals
p4 <- qplot(global_fit$residuals,geom = "histogram",bins = 15) +
  labs(title = "Histogram of residuals",x = "Residuals") +
  theme_bw() +  ## white background
  theme_light() +  ## light background lines
  theme(legend.position="top") +
  ggeasy::easy_center_title()

# The plots using
plots <- plot_grid(p1, p2,p3,p4, ncol=2)
plots

summary(global_fit)

Call:
lm(formula = X ~ t + I(t^2))

Residuals:
    Min      1Q  Median      3Q     Max
-239.35  -92.28   13.16   93.35  180.30

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.490e+02  1.492e+01   43.50   <2e-16 ***
t           -3.793e+00  1.364e-01  -27.80   <2e-16 ***
I(t^2)       8.355e-03  2.616e-04   31.93   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 111.2 on 501 degrees of freedom
Multiple R-squared:  0.7013,Adjusted R-squared:  0.7001
F-statistic: 588.2 on 2 and 501 DF,  p-value: < 2.2e-16

### QS3: Evaluating the local linear model in qs 2
##### ------------------------------------------------

## Calculating the residuals
```

```
df$local_residual <- df$sales - df$linear_polynomial

## Calculating the standardized residuals

## assign local residual to a vector
local_res <- df$local_residual[3:(n-2)]  ## exclude the first two and last two vector values
## since they are NA

df$local_std_res <- NA  ## create an empty column in the data frame

## standardized residuals are found by dividing each residual in vector 'local_res' by
## the standard deviation of the residuals calculated by 'sd(local_res)'
## we assign the new (calculated) vector to a the new column
## in our data frame called 'local_std_res'
## but we only assign it to the the rows from 3 to n-2=502
## because the first and last two rows are NA

df$local_std_res[3:(n-2)] <- local_res / sd(local_res)

# Make a scatter plot of residuals against fitted values
p1 <- qplot(df$linear_polynomial, df$local_residual, geom = "point") +
  geom_abline(intercept = 0,slope = 0,colour = "red") +
  ##
  xlab("Fitted values") + ## labeling the x axis
  ylab("Residuals") +
  ggtitle("Residuals vs fitted values") +  ## add title
  theme(plot.title = element_text(size=4, hjust = 'right', face="bold.italic")) + # main title
  theme_bw() +  ## white background
  theme_light() +  ## light background lines
  ggeasy::easy_center_title()

# Make a quantile-quantile plot
p2 <-ggplot(data = df,aes(sample = local_residual)) +
  geom_qq() +
  geom_qq_line(colour = "red") +
  labs(title = "Quantile plot of residuals") +
  theme_bw() +  ## white background
  theme_light() +  ## light background lines
  theme(legend.position="top") +
  ggeasy::easy_center_title()

p3 <- qplot(df$linear_polynomial, df$local_std_res, geom = "point") +
  geom_abline(intercept = 0,slope = 0,colour = "red") +
  labs(title = "Standardized residuals vs fitted values", x = "Fitted values (Sales)",
  y = "Standardised Residuals") +
  theme_bw() +  ## white background
  theme_light() +  ## light background lines
  ggeasy::easy_center_title()

# Make a histogram of the residuals
p4 <- qplot(df$local_residual,geom = "histogram",bins = 15) +
  labs(title = "Histogram of residuals",x = "Residuals") +
  theme_bw() +  ## white background
  theme_light() +  ## light background lines
  theme(legend.position="top") +
  ggeasy::easy_center_title()

# Produce plots
plots <- plot_grid(p1, p2,p3,p4, ncol=2)
plots
```

```
#########################################################################
# Question 4

## ---- 4b) Adding seasonality to the local linear fit

## define a vector that contains the local linear fit residuals,
## excluding the first and last two data points as they are NA
X <- df$local_residual[3:502]

## create an indicator value
ind <- rep(1:12,42) ## repeat values from 1 to 12, 42 times
# - create a vector of 504 indicator values

## Exclude the first two and last two data points from our indicator vector
ind <- ind[3:502]

## take the average over all values with the same indicator variable
avr_values <- c( mean(X[ind == 1]), mean(X[ind == 2]), mean(X[ind == 3]), mean(X[ind == 4]),
mean(X[ind ==5]), mean(X[ind == 6]), mean(X[ind == 7]), mean(X[ind == 8]), mean(X[ind == 9]),
mean(X[ind == 10]), mean(X[ind == 11]), mean(X[ind == 12]))

## Create seasonal estimates by repeating the averages k=42 number of times
sfit <- rep(avr_values, 42)

df$seasonal_fit <- sfit

head(df,5)
```

Output

```
  global_fit sales       time  residual linear_polynomial local_residual local_std_res seasonal_fit
1   645.2127   451 1979-01-01 -194.2127               NA             NA            NA     2.239024
2   641.4443   456 1979-02-01 -185.4443               NA             NA            NA     5.502439
3   637.6926   474 1979-03-01 -163.6926            453.8           20.2     1.9586434     9.714286
4   633.9576   460 1979-04-01 -173.9576            444.0           16.0     1.5514007     6.671429
5   630.2393   428 1979-05-01 -202.2393            431.6           -3.6    -0.3490652     2.295238
   seas_res Seas_std_res local_and_seasonal
1        NA           NA                NA
2        NA           NA                NA
3 10.485714    1.2143975          463.5143
4  9.328571    1.0803836          450.6714
5 -5.895238   -0.6827539          433.8952
```

```
## ---- 4c) Plotting the model with the seasonality element

# Create plot p using ggplot2 and ggeasy libraries we imported above

p <-ggplot(data = df, aes(x = time, y = local_residual)) +
 ## plot Sales against time
  geom_line(aes( y= local_residual, color="Local Linear Residuals")) +
  ## plot fitted values against time
  geom_line(data = df, aes(x=time, y=seasonal_fit, color='Seasonal estimates')) +
  ##geom_point(data = df, aes(x=time, y=local_residual, color='Local Linear Residuals'), pch=4) +
  xlab("Time Period") + ## labeling the x axis
  ylab("Local polynomial residuals & the seasonal estimates") +
  ggtitle("Seasonal estimates vs Residuals") +  ## add title
  theme(plot.title = element_text(size=10, hjust = 'right', face="bold.italic")) + # main title
  theme_bw() +  ## white background
  theme_light() +  ## light background lines
  theme(legend.position="top") +
  ggeasy::easy_center_title()  + ## centered title
```

```
  scale_colour_manual("",
                      breaks = c("Local Linear Residuals", "Seasonal estimates"), ## legend names
                      values = c("darkgrey", "red")) ## define legend colors

p

### ------ 4c) Evaluating the model with seasonality

# Analysing leftover residuals from seasonality estimates removal
df$seas_res <- df$local_residual - df$seasonal_fit

## Calculating the standardized residuals

## assign local residual to a vector
seas_res <- df$seas_res[3:(n-2)]  ## exclude the first two and last two vector values

df$Seas_std_res <- NA  ## create an empty column in the data frame

## Standardized residuals are found by dividing each residual in the vector 'seas_res' by
## the standard deviation of the residuals calculated by 'sd(seas_res)'.
## We assign the new (calculated) vector to a the new column in our data frame
## called 'Seas_std_res'
## but we only assign it to the the rows from 3 to n-2=502 because
## the first and last two rows are NA

df$Seas_std_res[3:(n-2)] <- seas_res / sd(seas_res)

# Make a scatter plot of residuals against fitted values
p1 <- qplot(df$seasonal_fit, df$seas_res, geom = "point") +
  geom_abline(intercept = 0,slope = 0,colour = "red") +
  ##
  xlab("Fitted values") + ## labeling the x axis
  ylab("Residuals") +
  ggtitle("Residuals vs fitted values") +  ## add title
  theme(plot.title = element_text(size=4, hjust = 'right', face="bold.italic")) + # main title
  theme_bw() +  ## white background
  theme_light() +  ## light background lines
  ggeasy::easy_center_title()

# Make a quantile-quantile plot
p2 <-ggplot(data = df,aes(sample = seas_res)) +
  geom_qq() +
  geom_qq_line(colour = "red") +
  labs(title = "Quantile plot of residuals") +
  theme_bw() +  ## white background
  theme_light() +  ## light background lines
  theme(legend.position="top") +
  ggeasy::easy_center_title()

p3 <- qplot(df$seasonal_fit, df$Seas_std_res, geom = "point") +
  geom_abline(intercept = 0,slope = 0,colour = "red") +
  labs(title = "Standardized residuals vs fitted values", x = "Fitted values (Sales)",
  y = "Standardised Residuals") +
  theme_bw() +  ## white background
  theme_light() +  ## light background lines
  ggeasy::easy_center_title()

# Make a histogram of the residuals
p4 <- qplot(df$seas_res,geom = "histogram",bins = 15) +
  labs(title = "Histogram of residuals",x = "Residuals") +
  theme_bw() +  ## white background
```

```
  theme_light() +  ## light background lines
  theme(legend.position="top") +
  ggeasy::easy_center_title()

# Plot the plots
plots <- plot_grid(p1, p2,p3,p4, ncol=2)
plots

#### QS4: ------ Overall model plot


df$local_and_seasonal <- df$linear_polynomial + df$seasonal_fit

p <-ggplot(data = df, aes(x = time, y = sales)) +
  geom_line(aes( y= sales, color="Sales")) +  ## plot Sales against time
  geom_line(data = df, aes(x=time, y=local_and_seasonal, color='Seasonal and Symmetric MA')) +
  ## plot fitted values against time
  xlab("Time Period") + ## labeling the x axis
  ylab("Sales actual and estimates") +
  ggtitle("Actual sales data with the the overall model estimates") +  ## add title
  theme(plot.title = element_text(size=10, hjust = 'right', face="bold.italic")) + # main title
  theme_bw() +  ## white background
  theme_light() +  ## light background lines
  theme(legend.position="top") +
  ggeasy::easy_center_title()  + ## centered title
  scale_colour_manual("",
                      breaks = c("Sales", "Seasonal and Symmetric MA"), ## legend names
                      values = c("Black", "red")) ## define legend colors

p

######################################################################
# Question 5

### a) Further observations code

## Assigning data to variables A, n and t
X <- data$sales  ## a time series vector from the sales column
n <- length(X) # the length of the vector X
t <- 1:n ## we create a range of values from 1 to the length as X as T

## Higher order global polynomials

higher_order_fit_0 <- lm(X~ t + I(t^2))
higher_order_pred_0 <- predict(higher_order_fit_0)

higher_order_fit_1 <- lm(X~ t + I(t^2)+ I(t^3))
higher_order_pred_1 <- predict(higher_order_fit_1)

higher_order_fit_2 <- lm(X~ t + I(t^2)+ I(t^3)+ I(t^4))
higher_order_pred_2 <- predict(higher_order_fit_2)

higher_order_fit_3 <- lm(X~ t + I(t^2)+ I(t^3)+ I(t^4)+ I(t^5))
higher_order_pred_3 <- predict(higher_order_fit_3)

higher_order_fit_4 <- lm(X~ t + I(t^2)+ I(t^3)+ I(t^4)+ I(t^5)+ I(t^6))
higher_order_pred_4 <- predict(higher_order_fit_4)

## Plot for higher order polynomials

## Create a dataframe that stores the actual and predicted
```

```
## sales as well as the time period

df_global_fits <- data.frame(
  months = seq(as.Date("1979-01-01"), as.Date("2020-12-31"), by="months"), # adding dates
  Sales = X,
  Order2_pred = higher_order_pred_0,
  Order3_pred = higher_order_pred_1,
  Order4_pred = higher_order_pred_2,
  Order5_pred = higher_order_pred_3,
  Order6_pred = higher_order_pred_4)

head(df_global_fits,3)
      months Sales Order2_pred Order3_pred Order4_pred Order5_pred Order6_pred
1 1979-01-01   451    645.2127    400.5629    310.0668    430.8917    466.6800
2 1979-02-01   456    641.4443    402.6310    315.7332    429.3518    462.1518
3 1979-03-01   474    637.6926    404.6578    321.2936    427.9070    457.8378

## Plotting the higher order polynomials

p <-ggplot(data = df_global_fits, aes(x = months, y = sales)) +
  geom_line(aes(y= Sales, color="Sales Data")) +
  geom_line(aes(x=months, y=Order2_pred, color='Order2')) +
  geom_line(aes(x=months, y=Order3_pred, color='Order3')) +
  geom_line(aes(x=months, y=Order4_pred, color='Order4')) +
  geom_line(aes(x=months, y=Order5_pred, color='Order5')) +
  geom_line(aes(x=months, y=Order6_pred, color='Order6')) +
  xlab("Time Period") +
  ylab("Sales (per month)") +
  ggtitle("Sales of product X per month between 1979 and 2020") +
  theme(plot.title = element_text(size=14, hjust = 'right', face="bold.italic")) +
  theme_bw() +
  theme_light() +
  theme(legend.position="top") +
  ggeasy::easy_center_title()  +
  scale_colour_manual("",
                      breaks = c("Sales Data", "Order2","Order3","Order4","Order5","Order6"),
                      values = c(1,2,3,4,5,6))

p

#### ---- Evaluating the Global Sextic Polynomial using residual plots


## standardized residuals are found by dividing each residual in vector 'local_res' by
## the standard deviation of the residuals calculated by 'sd(local_res)'

df_global_fits$global_order6_std_res <- df_global_fits$Residuals / sd(df_global_fits$Residuals)

head(df_global_fits,3)

# Make a scatter plot of residuals against fitted values
p1 <- qplot(df_global_fits$Order6_pred, df_global_fits$Residuals, geom = "point") +
  geom_abline(intercept = 0,slope = 0,colour = "red") +
  ##
  xlab("Fitted values") + ## labeling the x axis
  ylab("Residuals") +
  ggtitle("Residuals vs fitted values") +  ## add title
  theme(plot.title = element_text(size=4, hjust = 'right', face="bold.italic")) + # main title
  theme_bw() +  ## white background
  theme_light() +  ## light background lines
  ggeasy::easy_center_title()
```

```
# Make a quantile-quantile plot
p2 <-ggplot(data = df_global_fits,aes(sample = Residuals)) +
  geom_qq() +
  geom_qq_line(colour = "red") +
  labs(title = "Quantile plot of residuals") +
  theme_bw() +  ## white background
  theme_light() +  ## light background lines
  theme(legend.position="top") +
  ggeasy::easy_center_title()

p3 <- qplot(df_global_fits$Order6_pred, df_global_fits$global_order6_std_res, geom = "point") +
  geom_abline(intercept = 0,slope = 0,colour = "red") +
  labs(title = "Standardized residuals vs fitted values", x = "Fitted values (Sales)",
  y = "Standardised Residuals") +
  theme_bw() +  ## white background
  theme_light() +  ## light background lines
  ggeasy::easy_center_title()

# Make a histogram of the residuals
p4 <- qplot(df_global_fits$Residuals,geom = "histogram",bins = 15) +
  labs(title = "Histogram of residuals",x = "Residuals") +
  theme_bw() +  ## white background
  theme_light() +  ## light background lines
  theme(legend.position="top") +
  ggeasy::easy_center_title()

# Plot the plots
plots <- plot_grid(p1, p2,p3,p4, ncol=2)
plots

#-----------------------------------------------------------
##----- c) computing and evaluating the model
## EWA fit

# define the weight parameter a and first trend value m0
a <- 0.3
X <- df$sales
m0 <- 0

## Define an empty vector with 504 elements
m <- rep(NA,50)

# define the first trend in m
m[1] <- a*X[1] + (1-a)*m0

## loop to find all EWA estimates
for(i in 2:504)
{
  m[i] <- a*X[i] + (1-a)^m[i-1]
}

ts.plot(X)
lines(m,col=2)

##-------
## a = 0.3, value of m0 = X_0

a <- 0.3
b0 <- X[1]
```

```
## Define an empty vector with 504 elements
b <- rep(NA,50)

# define the first trend in m
b[2] <- a*X[2] + (1-a)*b0

## loop to find all EWA estimates
for(i in 3:504)
{
  b[i] <- a*X[i] + (1-a)^b[i-1]
}

ts.plot(X)
lines(b,col=2)

## -- larger value of a

# define the weight parameter a and first trend value m0
a <- 0.5
c0 <- 0

## Define an empty vector with 504 elements
c <- rep(NA,50)

# define the first trend in m
c[1] <- a*X[1] + (1-a)*c0

## loop to find all EWA estimates
for(i in 2:504)
{
  c[i] <- a*X[i] + (1-a)^c[i-1]
}

ts.plot(X)
lines(c,col=2)

## --- a = 0.7

# define the weight parameter a and first trend value m0
a <- 0.7
d0 <- 0

## Define an empty vector with 504 elements
d <- rep(NA,50)

# define the first trend in m
d[1] <- a*X[1] + (1-a)*d0

## loop to find all EWA estimates
for(i in 2:504)
{
  d[i] <- a*X[i] + (1-a)^d[i-1]
}

ts.plot(X)
lines(d,col=2)

### a = 0.9

# define the weight parameter a and first trend value m0
a <- 0.9
```

```
p0 <- 0

## Define an empty vector with 504 elements
p <- rep(NA,50)

# define the first trend in m
p[1] <- a*X[1] + (1-a)*p0

## loop to find all EWA estimates
for(i in 2:504)
{
  p[i] <- a*X[i] + (1-a)^p[i-1]
}

ts.plot(X)
lines(p,col=2)

### Create a dataframe with all EWA estimates

df_EWA <- data.frame(
  months = seq(as.Date("1979-01-01"), as.Date("2020-12-31"), by="months"), # adding dates
  Sales = X)

df_EWA$alpha_0.3_mo_0 <- m
df_EWA$alpha_0.3_mo_X_0 <-  b
df_EWA$alpha_0.5_mo_X_0 <- c
df_EWA$alpha_0.7_mo_X_0 <-  d
df_EWA$alpha_0.9_mo_X_0 <-  p

head(df_EWA, 3)

## plot them

p <-ggplot(data = df_EWA, aes(x = months, y = Sales)) +
  geom_line(aes(y= Sales, color="Sales Data")) +
  geom_line(aes(x=months, y=alpha_0.3_mo_0, color='a=0.3, m0=0')) +
  geom_line(aes(x=months, y=alpha_0.3_mo_X_0, color='a=0.3, m0=X_0')) +
  geom_line(aes(x=months, y=alpha_0.5_mo_X_0, color='a=0.5, m0=X_0')) +
  geom_line(aes(x=months, y=alpha_0.7_mo_X_0, color='a=0.7, m0=X_0')) +
  geom_line(aes(x=months, y=alpha_0.9_mo_X_0, color='a=0.9, m0=X_0')) +
  xlab("Time Period") +
  ylab("Sales (per month) and EWA estimates") +
  ggtitle("Actual sales vs. fitted EWA values") +
  theme(plot.title = element_text(size=14, hjust = 'right', face="bold.italic")) +
  theme_bw() +
  theme_light() +
  theme(legend.position="top") +
  ggeasy::easy_center_title()  +
  scale_colour_manual("",
                      breaks = c("Sales Data", "a=0.3, m0=0","a=0.3, m0=X_0","a=0.5, m0=X_0",
                      "a=0.7, m0=X_0","a=0.9, m0=X_0"),
                      values = c(1,2,3,4,5,6))

p
##-------------------------

## QS 5 d) Final comments and conclusion


#### split the data into 3 parts
```

```
part1 <- df$sales[1:36]

part2 <- df$sales[37:300]

part3 <- df$sales[301:504]

### makeing time values for each split

t1<- 1:36

t2<- 37:300

t3<- 301:504

### fitting models

part1_fit <- lm(part1~t1 +I(t1^2))

part2_fit <- lm(part2~t2 +I(t2^2))

part3_fit <- lm(part3~t3 +I(t3^2))

####

part1_predict <- predict(part1_fit)

part2_predict <- predict(part2_fit)

part3_predict <- predict(part3_fit)


###-----------------------------------------------------------------

## Make dataframe

df_splitted_data <- data.frame(
  months = seq(as.Date("1979-01-01"), as.Date("2020-12-31"), by="months"), # adding dates
  Sales = X)

df_splitted_data$pred_parts <- NA  ## create an empty column in the data frame

df_splitted_data$pred_parts[1:36] <- part1_predict
df_splitted_data$pred_parts[37:300] <- part2_predict
df_splitted_data$pred_parts[301:504] <- part3_predict

head(df_splitted_data)

>
      months Sales pred_parts
1 1979-01-01   451   443.5665
2 1979-02-01   456   440.9555
3 1979-03-01   474   438.4466
4 1979-04-01   460   436.0400
5 1979-05-01   428   433.7355
6 1979-06-01   402   431.5332

## Whole plot

p <-ggplot(data = df_splitted_data, aes(x = months, y = Sales)) +
  geom_line(aes(y= Sales, color="Sales Data")) +
  geom_line(aes(x=months, y=pred_parts, color='Fitted values')) +
```

```
  xlab("Time Period") +
  ylab("Sales (per month)") +
  ggtitle("Sales of product X per month between 1979 and 2020") +
  theme(plot.title = element_text(size=14, hjust = 'right', face="bold.italic")) +
  theme_bw() +
  theme_light() +
  theme(legend.position="top") +
  ggeasy::easy_center_title()  +
  scale_colour_manual("",
                      breaks = c("Sales Data", "Fitted values"),
                      values = c(1,2,3,4,5,6,7,8))

p

## Individual plots

ts.plot(part1, main="Sales estimates for months 1 to 36", ylab="Sales vs global quadratic fit")
lines(part1_predict,col=2, lty=2)

ts.plot(part2, main="Sales estimates for months 37 to 300", ylab="Sales vs global quadratic fit")
lines(part2_predict,col=2, lty=2)

ts.plot(part3, main="Sales estimates for months 301 to 504", ylab="Sales vs global quadratic fit")
lines(part3_predict, col=2, lty=2)


### Evaluation plots and Residuals

df_splitted_data$Res <- df_splitted_data$Sales - df_splitted_data$pred_parts

df_splitted_data$split_data_std_res <- df_splitted_data$Res / sd(df_splitted_data$Res)

head(df_splitted_data,3)

>
      months Sales pred_parts       Res split_data_std_res
1 1979-01-01   451   443.5665  7.433499          0.3207793
2 1979-02-01   456   440.9555 15.044513          0.6492189
3 1979-03-01   474   438.4466 35.553350          1.5342408

# Make a scatter plot of residuals against fitted values

p1 <- qplot(df_splitted_data$pred_parts, df_splitted_data$Res, geom = "point") +
  geom_abline(intercept = 0,slope = 0,colour = "red") +
  ##
  xlab("Fitted values") + ## labeling the x axis
  ylab("Residuals") +
  ggtitle("Residuals vs fitted values") +  ## add title
  theme(plot.title = element_text(size=4, hjust = 'right', face="bold.italic")) + # main title
  theme_bw() +  ## white background
  theme_light() +  ## light background lines
  ggeasy::easy_center_title()

# Make a quantile-quantile plot
p2 <-ggplot(data = df_splitted_data,aes(sample = Res)) +
  geom_qq() +
  geom_qq_line(colour = "red") +
  labs(title = "Quantile plot of residuals") +
  theme_bw() +  ## white background
  theme_light() +  ## light background lines
  theme(legend.position="top") +
```

```
  ggeasy::easy_center_title()

p3 <- qplot(df_splitted_data$pred_parts, df_splitted_data$split_data_std_res, geom = "point") +
  geom_abline(intercept = 0,slope = 0,colour = "red") +
  labs(title = "Standardized residuals vs fitted values", x = "Fitted values (Sales)",
  y = "Standardised Residuals") +
  theme_bw() +  ## white background
  theme_light() +  ## light background lines
  ggeasy::easy_center_title()

# Make a histogram of the residuals
p4 <- qplot(df_splitted_data$Res,geom = "histogram",bins = 15) +
  labs(title = "Histogram of residuals",x = "Residuals") +
  theme_bw() +  ## white background
  theme_light() +  ## light background lines
  theme(legend.position="top") +
  ggeasy::easy_center_title()

# Plot the plots
plots <- plot_grid(p1, p2,p3,p4, ncol=2)
plots
```