MAS8382 Time Series Data – Project

Luke Kaye

1.1 Applied Part

There is an appendix included at the end of this document, after section 1.2, that includes graphical output relating to this section, as well as the source code used to perform the analysis. The structure of the source code is linear, with functionality written in the same order as the reporting of section 1.1. The contents of the appendix will be referred to throughout section 1.1.

We have a dataset representing monthly sales of a product, from January 2011 to December 2020, totalling 120 observations. We load the data and store it as a time series object, with appropriate parameterisation, naming the time series *y*.

Exploratory Data Analysis

We begin by taking an initial look at the data, to see if we can identify general aspects of its behaviour which may assist with the modelling of the time series. We plot the time series, which can be viewed in **Figure 1** within the appendix (all other Figures referred to can also be found in the appendix).

Each of the data points are marked in red, joined together by the black lines. There is clear, additive seasonality in the data, with each period taking a minimum in February and a maximum in July, and generally higher sales in the spring and summer months; while we could check this formally using the Kruskal Wallis test, this would require fitting a trend to the data which we will avoid doing since this initial analysis is only exploratory. The data appears to take a period of 12 months, thus resulting in 10 total yearly periods. We also note the trend in the time series; briefly, it appears to be linear, but inspecting it closely it appears to have minor quadratic behaviour. By overlaying two lines, one connecting the maxima of each season, and one connecting the minima of each season, we can see this more explicitly, shown within **Figure 2**. While this minor quadratic effect could be ignorable, we will analyse its negligibility later when we fit models to the data. The data in January through April of 2011 seems to be unusually low, but without having access to data before these four months it's hard to tell if they are true outliers. There doesn't appear to be any heteroscedasticity in the data, indicating that ARIMA, time-series regression and dynamic linear models should all be appropriate as opposed to, say, a GARCH model.

The above intuition indicates clear non-stationarity in the data in the form of trend and seasonality, of which there is certainly a time varying local level, which we will consider when fitting models. As the seasonal variation across the time series appears to be constant, there should be no need to transform it, for example, using the Box-Cox family of transformations.

ARIMA Modelling

The first class of model we will fit to the data is an ARIMA model. We will allow for the seasonal generalisation of the ARIMA framework, later deciding if seasonal parameters are required for the data; our analysis so far indicates they will be needed in some capacity. Specifically, we wish to fit a model of the form

$$\mathrm{ARIMA}\{(p,d,q)\times(p^*,d^*,q^*)\} := \Phi(\mathrm{B})\Phi^*(\mathrm{B}^s)\bigtriangledown^d\bigtriangledown_s^{d^*}X_t = \Theta(\mathrm{B})\Theta^*(\mathrm{B}^s)\varepsilon_t,$$

for parameter orders p (autoregression), d (differencing), q (moving average), p^* (seasonal autoregression), d^* (seasonal differencing), q^* (seasonal moving average) and s (seasonal period). For the data, it is extremely likely that our seasonal period will take the value 12. We apply the Box-Jenkins method to attempt to fit the best model of this class to the data.

We have already identified that the data is non-stationary with visible seasonality, through plotting the data in **Figure 1**, however we further analyse a correlogram of the autocorrelation function (ACF) to verify this observation, shown in **Figure 3**.

The lag in the correlogram corresponds to yearly periods, for example, lag 1.0 refers to a lag of one year. We also see the blue dashed lines, indicating 95% confidence bands; an autocorrelation that sits purely within these bands can be deemed non-significant. The ACF for the data decays slowly, with visible spikes at lags 1, 2 and 3. This confirms what we saw earlier, that the data is non-stationary and seasonal, indicating a need for differencing. We additionally formally select $s = 12$ for our seasonal period.

We compute first seasonal differences for the data, then plot the first seasonally differenced data and its ACF, shown in **Figure 4**.

We see that the periodic spikes in the ACF have disappeared, indicating that we don't need to seasonally difference further. The plot for the first seasonally differenced data still shows a time varying local level, and the ACF still decays slowly. We therefore difference the data again, this time non-seasonally, plotting this twice differenced data alongside its ACF, shown in **Figure 5**.

The plot of the time series now looks stationary, and the ACF no longer decays slowly. This is interesting since we noticed a minor quadratic trend in the original data, the act of differencing once both seasonally and non-seasonally seems to have offset this; perhaps this means that the quadratic component of the trend can be ignored. Due to the stationarity of the plot, we select parameters $d = d^* = 1$, for this model.

Having identified the order of differencing, we now select initial values for p, q, p^*, q^*. To do this, we will make use of the ACF plot in **Figure 5**, and additionally plot and use the partial autocorrelation function (PACF), shown in **Figure 6**.

Both the seasonal ACF and PACF cuts off abruptly after lag 1.0 (12 months), suggesting that the paired choice of either $p^* = 1$ and $q^* = 0$, or $p^* = 0$ and $q^* = 1$ are both plausible. Similarly for the non-seasonal orders, the ACF and PACF cuts off abruptly after the first lag, once again indicating that the paired choice of $p = 1$ and $q = 0$, or $p = 0$ and $q = 1$ are both plausible. We note that there are a few scattered non-negligible ACF and PACF values at high lags, but our confidence bands sit at the 95% level so we would expect roughly 5% of our values to output as marginally significant purely by chance, so we ignore them.

As we will make use of significance tests and possibly model selection criteria to further decide on our final model parameters, we arbitrarily choose the combination of $p = 1, q = 0, p^* = 0$ and $q^* = 1$, from the above justification, as our starting point. That is, we currently have an $\mathrm{ARIMA}\{(1,1,0)\times(0,1,1)_{12}\}$ model for our time series. We fit this model to our original data, using the code (which can be found in **Figure 18** within the appendix) on **line 42**, then plot the residuals (i.e., the one-step forecast errors), their ACF and their PACF, shown in **Figure 7**.

The plot of the time series appears to be stationary, and each of the ACF and PACF values are non-significant. We additionally plot the p-values for the Ljung-Box test performed on the residuals for this model, shown in **Figure 8**, to further check their independence and perform the Shapiro-Wilk test on the residuals to assess their normality.

```
        Shapiro-Wilk normality test

data:  resids
W = 0.98878, p-value = 0.4317
```

The p-values returned from the Ljung-Box and Shapiro-Wilk tests all appear to be non-significant, implying a good fit.

We first fit models with additional autoregressive and moving average parameters, to see if we can find a better model than what we currently have. Each candidate model fits a single additional parameter, each of different type, to allow easier analysis of model improvement. Each model is summarised alongside its log-likelihood and likelihood ratio test p-value. These figures were derived using the code on **lines 50 – 71**.

Model	Log-likelihood	p-value
$\mathrm{ARIMA}\{(1,1,0)\times(0,1,1)_{12}\}$	-368.7905	-
$\mathrm{ARIMA}\{(2,1,0)\times(0,1,1)_{12}\}$	-368.7024	0.674676
$\mathrm{ARIMA}\{(1,1,1)\times(0,1,1)_{12}\}$	-368.6996	0.6698855
$\mathrm{ARIMA}\{(1,1,0)\times(1,1,1)_{12}\}$	-368.3623	0.3547365
$\mathrm{ARIMA}\{(1,1,0)\times(0,1,2)_{12}\}$	-368.6346	0.5766173

All these candidate models naturally take higher log-likelihoods than the $\mathrm{ARIMA}\{(1,1,0)\times(0,1,1)_{12}\}$ model since they fit additional parameters, however we make use of the likelihood ratio test to test if this improved fit is worth the additional model complexity, which is suitable since the $\mathrm{ARIMA}\{(1,1,0)\times(0,1,1)_{12}\}$ model is nested within each of the candidate models. None of the candidate models indicate significance in their p-values and we can therefore say that the $\mathrm{ARIMA}\{(1,1,0)\times(0,1,1)_{12}\}$ model is superior to them.

We now fit models with autoregressive and moving average parameters removed. Like before, each candidate model removes a single parameter, each of different type. Each model is summarised alongside its log-likelihood and likelihood ratio test p-value. These figures were derived using the code on **lines 73 – 85**.

Model	Log-likelihood	p-value
$\mathrm{ARIMA}\{(1,1,0)\times(0,1,1)_{12}\}$	-368.7905	-
$\mathrm{ARIMA}\{(0,1,0)\times(0,1,1)_{12}\}$	-375.4661	0.0002582518
$\mathrm{ARIMA}\{(1,1,0)\times(0,1,0)_{12}\}$	-382.4553	1.715732×10^{-7}

Both tests indicate significance in their p-values, however in this case this means that the $\mathrm{ARIMA}\{(1,1,0)\times(0,1,1)_{12}\}$ model, as it is the more complex model in each of the tests, is in fact the better choice.

Ultimately, the $\mathrm{ARIMA}\{(1,1,0)\times(0,1,1)_{12}\}$ model has been selected as the best seasonal ARIMA model to represent the time series, providing a far greater fit than the other models we considered. Had we seen multiple overfit or underfit models that indicated a superior fit to the original model, we could have compared them using model selection criteria such as AIC or

BIC, as the likelihood ratio test would be inappropriate as these models would not have been nested within each other, but in this case it wasn't necessary.

We finish by summarising the $\mathrm{ARIMA}\{(1,1,0)\times(0,1,1)_{12}\}$ model.

```
arima(x = y, order = c(1, 1, 0), seasonal = c(0, 1, 1))

Coefficients:
          ar1     sma1
      -0.3574  -0.6241
s.e.   0.0946   0.1204

sigma^2 estimated as 54.53:  log likelihood = -368.79,  aic = 743.58
```

The fitted model uses the maximum likelihood estimates for the parameters. We see that this model takes autoregression coefficient $\hat{\phi}=-0.3574$ and seasonal moving average coefficient $\hat{\theta}^{*}=-0.6241$ for a period $s=12$; the inclusion of the seasonal coefficient reinforces the belief that there is notable seasonality in the data. Their standard errors are relatively low, further indicating the importance of these parameters. We also have white noise variance $\hat{\sigma}^2=54.53$. This model required only first order non-seasonal differencing, showing that it hasn't found the trend in the time series to be quadratic, only linear. As we have already performed residual analysis for this model, we don't repeat it here.

Time-Series Regression Modelling

We now fit a regression-based model to the data, that is, we wish to fit a model of the form

$$X_t=\beta_0+\beta_1 t+\cdots+\beta_c t^c+s_t+R_t$$

for time t, regression coefficients $\beta_i, i=0,1,2,\cdots,c$, seasonal effects s_t such that $s_i=s_{i+p}=s_{i+2p}=\cdots, i=1,2,\cdots,p$ for period p, and non-independent errors R_t such that $\mathrm{E}(R_t)=0$. We will additionally assume that the errors R_t can be modelled by a stationary, zero-mean ARMA process $\Theta(\mathrm{B})R_t=\Phi(\mathrm{B})\varepsilon_t$, and that $p=12$ as evidenced in the ACF plot of the original time series within the ARIMA modelling stage.

As we noted in the exploratory analysis, the trend within the data is certainly at least linear, but there may also be a minor quadratic component to the trend as well. Even though the ARIMA model found the trend to be linear, we will separately assess in the regression case whether to model the trend as linear or quadratic; we will fit two regression models, one without a squared time term (i.e. $\beta_{i\geq 2}=0$) and one including a squared time term (i.e. $\beta_{i\geq 3}=0$).

We proceed by fitting these two regression models to the time series, using the code on **lines 94 – 99 and lines 106 – 111**, modelling each of the seasonal effects as a dummy indicator variable. While we could use a set of continuous variables, such as Fourier-type variables, to model the seasonal effects, it wouldn't be particularly useful over a simpler set of indicator variables since $p=12$, and thus we only need to fit 11 indicator variables which should be possible with a dataset of this size without introducing a large degree of variance. Contrasting this with a simple set of Fourier-type variables which might fit, say, three sine and three cosine terms, the indicator variables would only utilise an additional 5 degrees of freedom over this, which is not very significant for our 120-length dataset. Additionally, individual indicator variables corresponding to each month are quite easy to interpret, whereas a set of Fourier-type terms would be harder to understand in turn.

We then plot the fitted values for both regression models fitted to the time series, shown in **Figure 9**.

We see that the fitted values of the regression model utilising a squared time term within the trend capture the behaviour of the dataset more closely. We also use the AIC scores of the two models to compare them. We note that $\mathrm{AIC}=2k-2\hat{l}$ for a model, where k is the number of estimated parameters, and $\hat{l}$ is the estimated log-likelihood. Noting this, the penalty for estimating a single parameter is $+2$ to the AIC score. Thus, if our quadratic trend model takes AIC score less than 2 in magnitude than the AIC score for the linear trend model, we have a fit that is better in significance. We find the AIC scores for the models, and their difference.

```
> c(AIC(lin_mean_model), AIC(qua_mean_model),
+   AIC(lin_mean_model)-AIC(qua_mean_model))
[1] 1083.3251  940.1340  143.1911
```

The figures, from left to right, are the linear trend model AIC score, the quadratic trend model AIC score, and their difference. We see that the difference is 143.1911, which is highly significant, providing very strong evidence that our quadratic trend

model is superior. This appears to contrast with the ARIMA model, which only required one non-seasonal differencing; this dissimilarity serves as an example of how different modelling classes can account for behaviour differently in a time series.

While the linear trend isn't a poor fit by any means, we will be conducting forecasting for this time series later; thus, to ensure our forecasting is as accurate as possible, we will proceed with the model featuring the squared time term.

We plot the residuals for this quadratic trend regression model, shown in **Figure 10**.

As evidenced by the residual vs time plot, the residuals are autocorrelated, that is, they are not independent. To account for this, we proceed by modelling the residuals as a stationary, zero-mean ARMA process, once again using the Box-Jenkins method. We plot the ACF and PACF of the residuals, shown in **Figure 11**.

The ACF tapers off slowly while the PACF cuts off after the second lag, suggesting an $\mathrm{AR}(2)$ model would be a good starting point for modelling the residuals of the time series. We therefore fit an $\mathrm{AR}(2)$ model to the residuals, using the code on **line 114 and line 146**, fixing the mean at zero, and plot the residuals, their ACF, their PACF, their Ljung-Box statistic p-values, all shown in **Figure 12**, and perform the Shapiro-Wilk test.

```
        Shapiro-Wilk normality test

data:  model_1$residuals
W = 0.99397, p-value = 0.8884
```

We see that the residuals of this model appear to be stationary, and each of the ACF and PACF values are non-significant (excluding one value on each of the ACF and PACF plots, but as mentioned previously, we expect around 5% of the autocorrelations to output as marginally significant by chance). Additionally, the p-values returned from the Ljung-Box and Shapiro-Wilk tests all appear to be non-significant, implying a good fit.

Like with the ARIMA model, we will fit additional candidate models, each with a parameter added or removed, and compare them with our current model using the likelihood ratio test to see if we can find a better fit for the residuals. These figures were derived using the code on **lines 153 – 170**.

Model	Log-likelihood	p-value
$\mathrm{AR}(2)$	-393.8892	-
$\mathrm{AR}(3)$	-393.7998	0.6724861
$\mathrm{ARMA}(2,1)$	-393.7748	0.632412
$\mathrm{AR}(1)$	-401.3867	0.0001077966

As we have non-significance for the overfit models, and significance in the underfit model, we therefore choose to remain with the $\mathrm{AR}(2)$ model to fit the residuals.

We finish by summarising our final regression model.

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.803e+02  4.630e+00  38.946  < 2e-16 ***
t            7.643e-01  1.214e-01   6.297 7.02e-09 ***
t_sq         1.534e-02  9.715e-04  15.794  < 2e-16 ***
monthFeb    -9.652e+01  5.107e+00 -18.899  < 2e-16 ***
monthMar    -4.929e+00  5.107e+00  -0.965    0.337
monthApr     4.123e+01  5.108e+00   8.073 1.15e-12 ***
monthMay     1.003e+02  5.108e+00  19.643  < 2e-16 ***
monthJun     9.763e+01  5.109e+00  19.109  < 2e-16 ***
monthJul     1.270e+02  5.110e+00  24.856  < 2e-16 ***
monthAug     8.600e+01  5.111e+00  16.825  < 2e-16 ***
monthSep     2.841e+01  5.113e+00   5.557 2.06e-07 ***
monthOct     8.023e+01  5.114e+00  15.688  < 2e-16 ***
monthNov    -1.582e+00  5.116e+00  -0.309    0.758
monthDec    -1.945e+00  5.118e+00  -0.380    0.705
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11.42 on 106 degrees of freedom
```

```
arima(x = qua_resids, order = c(2, 0, 0), include.mean = FALSE)

Coefficients:
         ar1     ar2
      0.5225  0.3554
s.e.  0.0859  0.0885

sigma^2 estimated as 41.09:  log likelihood = -393.89,  aic = 793.78
```

The model uses the maximum likelihood estimates for the parameters. On the left, we have the summary for the regression component and on the right, we have the summary for the $\mathrm{AR}(2)$ component. Note that the model for the regression component built within R does not include an effect for January, since R defines January to be the baseline for the other months, that is, the other monthly effects are relative to the effect for January, which is defined to be zero. The regression part of the model indicates strong seasonality in the time series, with every seasonal effect excluding March, November and December being significant. Even though some of the seasonal effects are described as non-significant, it would be inappropriate to drop them from the model as they are all essentially factors of the same categorical variable; dropping them from the model would

lead to biased estimates for the other coefficients. It also shows a positive trend overall, with the t and t^2 coefficient estimates taking a statistically significant positive value, noting that we have modelled the trend as quadratic. While the estimated t coefficient is an order of magnitude greater than the estimated t^2 coefficient, our time series takes 120 observations, thus, for precise forecasting, the inclusion of a t^2 term is still important. The $\mathrm{AR}(2)$ part of the model, modelling the residuals, has first and second order autoregression coefficients $\phi_1 = 0.5225$ and $\phi_2 = 0.3554$ respectively, both with relatively low standard errors, indicating their importance in the ARMA component of the model. We additionally have white noise variance $\sigma^2 = 41.09$ for the residuals. As we have already performed residual analysis for this model, we don't repeat it here.

Dynamic Linear Modelling

The final class of model we will fit to the time series is a dynamic linear model (DLM), a special case of a state space model. Specifically, DLMs are state space models that are linear in their parameters in the system and observation equations, and Gaussian in their errors. Specifically, we seek a model of the form

$$Y_t = \mathbf{F}\boldsymbol{\vartheta}_t + v_t, v_t \sim \mathrm{N}(0, \sigma_v^2),$$
$$\boldsymbol{\vartheta}_t = \mathbf{G}\boldsymbol{\vartheta}_{t-1} + \mathbf{w}_t, \mathbf{w}_t \sim \mathrm{N}_q(\mathbf{0}, \mathbf{W}),$$

where we have taken Y_t to be a scalar, as in the data, and hence $\mathbf{F}$ is a $1 \times q$ matrix and the v_t are univariate normally distributed errors. Additionally, $\mathbf{G}$ is a $q \times q$ matrix and the $\mathbf{w}t$ are q-variate normally distributed errors. We assume that $\mathbf{F}$, $\mathbf{G}$, σ_v^2 and $\mathbf{W}$ are independent of time.

We have already seen in **Figure 1** that the time series has an additive seasonal effect and a positive trend, noting that the ARIMA model assumes a linear trend, whereas the regression model assumes a quadratic trend. As a digression, it can be shown that all DLMs and ARIMA models can have equivalent representations derived of them within the other class of model. Therefore, due to this similarity between ARIMA models and DLMs, we will follow the intuition for the ARIMA model, that is, we will attempt to fit a DLM that utilises a linear growth component and a trigonometric seasonal component, where we again assume a seasonal period of 12 (due to the first ACF plot within the ARIMA modelling stage), therefore taking $q = 13$. We make use of the *dlm* package in R for fitting the model of this class.

Using the R code on **lines 180 - 191**, a model featuring a superposition of a linear growth model and a trigonometric seasonal model was fitted. It uses maximum likelihood estimation to estimate the parameters within the model.

We plot the data alongside the smoothed trend values as a brief assessment of fit, shown in **Figure 13**.

The smoothed values seem to represent the time series quite well. We will also perform residual analysis to further assess the fit of the model. To extract the residuals for the model, we use the filtered values. We then plot the residuals, their ACF, their PACF, their Ljung-Box statistic p-values, all shown in **Figure 14**, and perform the Shapiro-Wilk test.

```
        Shapiro-Wilk normality test

data:  resids
W = 0.99088, p-value = 0.615
```

We see that the residuals of this model appear to be stationary, and each of the ACF and PACF values are non-significant. Additionally, the p-values returned from the Ljung-Box and Shapiro-Wilk tests all appear to be non-significant, implying a good fit.

We finish by summarising the selected DLM for the time series. The summary produced within R isn't easily interpretable and is also truncated so we write the information contained within it using the usual notation. Our final selected model takes the form

$$Y_t = \mathbf{F}\boldsymbol{\vartheta}_t + v_t, v_t \sim \mathrm{N}(0, \sigma_v^2),$$
$$\boldsymbol{\vartheta}_t = \mathbf{G}\boldsymbol{\vartheta}_{t-1} + \mathbf{w}_t, \mathbf{w}_t \sim \mathrm{N}_{13}(\mathbf{0}, \mathbf{W}),$$

where

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix},$$
$$\mathbf{G} = \mathrm{blockdiag}\left\{ \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0.8660254 & 0.5 \\ -0.5 & 0.8660254 \end{pmatrix}, \begin{pmatrix} 0.5 & 0.8660254 \\ -0.8660254 & 0.5 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix},$$

$$\left\{ \begin{pmatrix} -0.5 & 0.8660254 \\ -0.8660254 & -0.5 \end{pmatrix}, \begin{pmatrix} -0.8660254 & 0.5 \\ -0.5 & -0.8660254 \end{pmatrix}, -1 \right\},$$

$$\sigma_v^2 = 8.585122,$$

$$\mathbf{W} = \text{diag}(17.59214, 0.03945069, \overbrace{0.06007132, \cdots, 0.06007132}^{11 \text{ elements}}),$$

$$\vartheta_0 \sim \text{N}_{13}(\mathbf{0}, \text{diag}(\overbrace{10^7, \cdots, 10^7}^{13 \text{ elements}})).$$

This model assumes a linear trend in the data, as well as a set of 12 monthly seasonal effects.

Using the smoothed state vectors, we plot the trend and seasonal components of the time series, shown in **Figure 15**.

We see that the model draws quite a detailed representation of the trend and seasonal effects. As we have already performed residual analysis for this model, we don't repeat it here.

Model Selection

We have now fitted three models to the time series, each of a different class: an ARIMA model, a regression-based model and a dynamic linear model.

While each of these models fits the data to a satisfactory degree, and in essence we could use any of them as a suitable representation of the time series, they each have their respective advantages.

- The ARIMA model is parametrically simple, with clear explanation of how the seasonal and non-seasonal effects individually behave. Specifically, non-seasonally it behaves with a first order autoregression, and seasonally it behaves with a first order moving average, with one non-seasonal differencing and one seasonal differencing having taken place.
- The regression-based model uses seasonal effects in the form of dummy indicator variables, and a trend described solely by a quadratic function, which are easy to interpret even for someone not statistically knowledgeable. Regression-based approaches are also well suited to short-term forecasting as they assume the model residuals follow a stationary ARMA process, resulting in a forecast variance that does not increase with the forecast horizon, which was an assumption applicable to this dataset.
- The dynamic linear model, due to how it utilises the Kalman filter, can be readily updated to include new observations, that is, it doesn't require an entirely new cycle of identification and modelling when new data becomes available.

All our models suggest that there are notable additive seasonal effects for a period of 12 seasons and a positive trend, within the data. The ARIMA model and DLM both assume this trend is linear, whereas the regression model assumes it is quadratic.

We note that the residual analysis for the ARIMA model and the DLM are very similar. Noting this in conjunction with the earlier stated equivalence of ARIMA models and DLMs, it is quite likely that our DLM and ARIMA model explain the behaviour of the time series in a similar way.

The main advantage to using a DLM is its ability to update and forecast with the model, both sequentially, as new data are observed. This would be applicable to a situation where, say, we were analysing the monthly sales of a product presently and constantly. For this dataset, we are only retrospectively looking back on a set of observations between 2011 and 2020 inclusive and will be forecasting a further 6 observations, and thus we won't need to make use of this property of DLMs. Therefore, comparing our DLM with our ARIMA model, and noting that the representations they provide of the time series are likely to be quite similar, as indicated by their residuals, use of the ARIMA model would be preferable due to its simple parametric representation, aiding its interpretation.

Contrasting the ARIMA model with the regression model, we note that the forecasting we wish to conduct is for a further 6 observations. Compared to the full dataset of 120 observations, this forecasting period is 5% the size of the actual time series and it is quite unlikely that there would be any external factors that might affect the underlying distribution of the monthly sales in only a 6-month period; we can therefore classify this as a short-term forecast. In this regard, the regression model would be preferable due to its stricter forecast variance. The easily interpretable seasonal effects and trend are an additional bonus, as they would be relatively easy to explain to a less statistically knowledgeable stakeholder interested in these monthly sales. It would also be smart to use a regression model since the trend within the time series is simple enough to be represented by a polynomial.

By the justification above, we will proceed with forecasting using our regression model.

Forecasting Using the Regression Model

Before we forecast the monthly sales from January to June 2021 inclusive, we will first divide the data into a training set and a test set. We will take the first nine periods, that is, the data from January 2011 to December 2019 inclusive, as our training set, and take the final period, the data from January 2020 to December 2020 inclusive, as our test set. With this split of data, we will fit the regression model to the training set and see how it forecasts the period covered by the test set, then comparing this forecast with the actual observations in the test set. This will give us a good indication of the reliability of the forecasting for the model fitted over the entire dataset.

The data is divided into a training set and a test set, using the R code on **lines 221 - 223**.

Proceeding as we did for the regression model fitted over the full data, we fit a regression model of the same form to the training set, that is, with a quadratic trend, dummy indicator seasonal effects with seasonal period equal to 12 and residuals explained by a zero-mean $\mathrm{AR}(2)$ process. As a final comparison between this regression model and the one featuring the linear trend, we will also fit a regression model with a linear trend to the training set and compare its forecasts with those of the quadratic trend regression model. These fits were done using the R code on **lines 225 – 236 and lines 282 – 294**. Again, we use maximum likelihood estimation for this purpose.

As these models serve only for diagnostic purposes, we will plot only their point forecasts and forecast variances for 12 months, overlaid by the actual observations in the test set, shown in **Figure 16**. These forecasts additionally correct for the autocorrelation between the residuals. This forecasting was done using the code on **lines 239 – 259 and lines 297 - 317**.

As expected, the forecasting performed by the regression model featuring the quadratic trend is more accurate, with the test data sitting within the forecast 95% prediction intervals in all the forecasting range, except for marginal deviation outside of this 95% interval within a small segment centred around July. For comparison, the forecasting performed by the model featuring the linear trend has poorer predictive performance, with the test data deviating outside of the 95% interval for a considerable part of the forecasting range, even deviating outside of the 99% interval in places. These results further highlight the importance of modelling the trend for this time series as quadratic, for the regression model class.

Now, exclusively returning to the quadratic trend regression model, we quantify the prediction error between the training set forecasts and the test set by finding the mean absolute percentage error between the forecasts and their corresponding observations.

```
> (mape = mean(100*abs(as.numeric(test_y) - preds)/as.numeric(test_y)))
[1] 2.4467
```

Thus, we have that the point predictions only deviate, on average, by 2.4% from the actual observations. This shows that the point forecasts provide a good estimation of the monthly sales.

We will now proceed by using the regression model for the full time series to produce forecasts for January through June 2021 inclusive.

We plot the time series, alongside its point forecasts and forecast variances for January through June 2021 inclusive, corrected for the autocorrelation between the residuals, shown in **Figure 17**. This forecasting was done using the code on **lines 343 – 370**.

We additionally show the specific forecasted points with red dots on the plot. The forecasting continues the pattern shown in the data, having a positive trend and an additive effect for each season which replicates the shape of each of the previous periods observed in the data. We see that the average sales of the product across a given year will continue to increase, with the monthly pattern in the sales being retained.

Conclusion

We found, from initially exploring the data and fitting various models, that the behaviour of the monthly sales of the product was non-stationary. The monthly sales exhibited a positive trend that could be viewed as linear or quadratic, and clear seasonality with a period of 12 months; both effects were additive. The regression model with quadratic trend, dummy indicator seasonal effects and $\mathrm{AR}(2)$ errors showed that the pattern expressed in the data on the monthly sales would continue, given that the sales of the product in general are forecasted to increase into the future, retaining the characteristic shape in the yearly periods of monthly sales.

1.2 Theoretical Part

Question 1

We have a seasonal ARIMA model

$$X_t = \varepsilon_t + \theta^* \varepsilon_{t-2}, \varepsilon_t \sim \mathrm{N}(0, \sigma^2).$$

Using the usual parametric notation, this describes a seasonal $\mathrm{ARIMA}\{(0,0,0) \times (0,0,1)_2\}$ model.

This model has mean

$$\begin{aligned} \mathrm{E}(X_t) &= \mathrm{E}(\varepsilon_t + \theta^* \varepsilon_{t-2}) \\ &= \mathrm{E}(\varepsilon_t) + \theta^* \mathrm{E}(\varepsilon_{t-2}) \\ &= 0 \end{aligned}$$

and autocovariance

$$\begin{aligned} \gamma_k &\equiv \mathrm{Cov}\left(X_{t-k}, X_t\right) \\ &= \mathrm{E}\left(X_{t-k} X_t\right) \\ &= \mathrm{E}(\left\{\varepsilon_{t-k} + \theta^* \varepsilon_{t-k-2}\right\}\left\{\varepsilon_t + \theta^* \varepsilon_{t-2}\right\}) \\ &= \mathrm{E}(\varepsilon_t \varepsilon_{t-k} + \theta^* \varepsilon_t \varepsilon_{t-k-2} + \theta^* \varepsilon_{t-2} \varepsilon_{t-k} + \theta^{*2} \varepsilon_{t-2} \varepsilon_{t-k-2}) \\ &= \mathrm{E}(\varepsilon_t \varepsilon_{t-k}) + \theta^* \mathrm{E}(\varepsilon_t \varepsilon_{t-k-2}) + \theta^* \mathrm{E}(\varepsilon_{t-2} \varepsilon_{t-k}) + \theta^{*2} \mathrm{E}(\varepsilon_{t-2} \varepsilon_{t-k-2}) \end{aligned}$$

which reduces to

$$\begin{aligned} \gamma_0 &= (1 + \theta^{*2}) \sigma^2, \\ \gamma_2 &= \gamma_{-2} = \theta^* \sigma^2, \\ \gamma_{k \notin \{-2,0,2\}} &= 0. \end{aligned}$$

We therefore have that this seasonal ARIMA model has a mean constant in time and an autocovariance function dependent only on the lag; it must therefore be stationary, regardless of θ^*. This is, in fact, true for any pure moving average model, seasonal or otherwise.

To check this models invertibility, we look at its seasonal moving average characteristic equation as it only consists of a seasonal moving average component, that is, there is no non-seasonal moving average characteristic equation to check. We therefore have

$$\begin{aligned} \Theta^*(x^2) &= 0 \\ \Rightarrow 1 + \theta^* x^2 &= 0 \\ \Rightarrow x &= \pm \frac{i}{\sqrt{\theta^*}}. \end{aligned}$$

For this model to be invertible, we must have $|x| > 1$. We note that

$$\begin{aligned} |x| &= \left| \pm \frac{i}{\sqrt{\theta^*}} \right| \\ &= \frac{1}{\sqrt{|\theta^*|}}. \end{aligned}$$

Imposing the constraint $|x| > 1$, we have

$$1 < \frac{1}{\sqrt{|\theta^*|}}$$
$$\Rightarrow \sqrt{|\theta^*|} < 1$$
$$\Rightarrow |\theta^*| < 1,$$

that is, this series is invertible for $|\theta^*| < 1$.

We can express this model as $\varepsilon_t = X_t - \theta^* \varepsilon_{t-2}$, therefore

$$\begin{aligned} X_t &= \varepsilon_t + \theta^* \varepsilon_{t-2} \\ &= \varepsilon_t + \theta^*(X_{t-2} - \theta^* \varepsilon_{t-4}) \\ &= \varepsilon_t + \theta^* X_{t-2} - \theta^{*2} \varepsilon_{t-4} \\ &= \varepsilon_t + \theta^* X_{t-2} - \theta^{*2}(X_{t-4} - \theta^* \varepsilon_{t-6}) \\ &= \varepsilon_t + \theta^* X_{t-2} - \theta^{*2} X_{t-4} + \theta^{*3} \varepsilon_{t-6} \\ &= \cdots \\ &= \varepsilon_t + \sum_{i=1}^{\infty} (-1)^{i+1} \theta^{*i} X_{t-2i}, \end{aligned}$$

where we constrain $|\theta^*| < 1$ to allow convergence.

We can alternatively parameterise this equation as

$$X_t = \varepsilon_t + \sum_{k=1}^{\infty} \frac{(-1)^{k+1} - 1}{2} (-1)^{\frac{k(k+1)}{2}} \theta^{*\frac{k}{2}} X_{t-k},$$

such that the series now takes terms in $X_{t-k}, k \in \{1, 2, \cdots\}$. This is the $\mathrm{AR}(\infty)$ representation of the model.

Given data $x_1, \cdots, x_n$, assuming that $n \geq 2$, we have

$$\begin{aligned} X_{n+1} &= \varepsilon_{n+1} + \theta^* \varepsilon_{n-1} \\ &= \varepsilon_{n+1} + \theta^* \{x_{n-1} - \hat{x}_{n-2}(1)\}. \end{aligned}$$

This has mean

$$\hat{x}_n(1) = \theta^* \{x_{n-1} - \hat{x}_{n-2}(1)\}$$

and variance

$$\mathrm{Var}(X_{n+1} | \mathbf{X}_{1:n}) = \sigma^2.$$

We additionally have

$$\begin{aligned} X_{n+2} &= \varepsilon_{n+2} + \theta^* \varepsilon_n \\ &= \varepsilon_{n+2} + \theta^* \{x_n - \hat{x}_{n-1}(1)\}. \end{aligned}$$

This has mean

$$\hat{x}_n(2) = \theta^* \{x_n - \hat{x}_{n-1}(1)\}$$

and variance

$$\mathrm{Var}(X_{n+2} | \mathbf{X}_{1:n}) = \sigma^2.$$

Further, we have

$$X_{n+3} = \varepsilon_{n+3} + \theta^* \varepsilon_{n+1}.$$

This has mean

$$\hat{x}_n(3) = 0$$

and variance

$$\mathrm{Var}(X_{n+3}|\mathbf{X}_{1:n}) = (\theta^{*2} + 1)\sigma^2.$$

Using these results, we see that, for $k \geq 3$,

$$\hat{x}_n(k) = 0$$

and

$$\mathrm{Var}(X_{n+k}|\mathbf{X}_{1:n}) = (\theta^{*2} + 1)\sigma^2.$$

Question 2

Using the well-known results

$$e^{ix} = \cos x + i \sin x,$$

$$e^{-ix} = \cos(-x) + i \sin(-x) = \cos x - i \sin x,$$

we can write the cosine and sine functions in terms of complex exponentials. Specifically, we have

$$\cos x = \frac{e^{ix} + e^{-ix}}{2}$$

and

$$\sin x = \frac{e^{ix} - e^{-ix}}{2i}.$$

We can therefore write the seasonal effects as

$$s_t = \sum_{k=1}^{K} \left\{ a_k \left[\frac{\exp\left(\frac{i2\pi kt}{p}\right) + \exp\left(-\frac{i2\pi kt}{p}\right)}{2} \right] + b_k \left[\frac{\exp\left(\frac{i2\pi kt}{p}\right) - \exp\left(-\frac{i2\pi kt}{p}\right)}{2i} \right] \right\}.$$

Summing over the effects in a given period of seasons, we have

$$\sum_{j=1}^{p} s_j = \sum_{j=1}^{p} \sum_{k=1}^{K} \left\{ a_k \left[\frac{\exp\left(\frac{i2\pi kj}{p}\right) + \exp\left(-\frac{i2\pi kj}{p}\right)}{2} \right] + b_k \left[\frac{\exp\left(\frac{i2\pi kj}{p}\right) - \exp\left(-\frac{i2\pi kj}{p}\right)}{2i} \right] \right\}$$

$$= \sum_{k=1}^{K} \left\{ \frac{1}{2} a_k \left[\sum_{j=1}^{p} \exp\left(\frac{i2\pi kj}{p}\right) + \sum_{j=1}^{p} \exp\left(-\frac{i2\pi kj}{p}\right) \right] + \frac{1}{2i} b_k \left[\sum_{j=1}^{p} \exp\left(\frac{i2\pi kj}{p}\right) - \sum_{j=1}^{p} \exp\left(-\frac{i2\pi kj}{p}\right) \right] \right\}.$$

If we let

$$y := x + x^2 + \cdots + x^n$$

then

$$xy = x^2 + x^3 + \cdots + x^{n+1}$$

and therefore

$$xy - y = y(x-1) = x^{n+1} - x.$$

Provided $x \neq 1$, we can compute

$$\frac{y(x-1)}{x-1} = y = \frac{x^{n+1} - x}{x-1} = \frac{x(x^n - 1)}{x-1}.$$

If we additionally let

$$z := x^{-1} + x^{-2} + \cdots + x^{-n}$$

then

$$xz = 1 + x^{-1} + \cdots + x^{1-n}$$

and therefore

$$xz - z = z(x-1) = 1 - x^{-n}.$$

Provided $x \neq 1$, we can compute

$$\frac{z(x-1)}{x-1} = z = \frac{1 - x^{-n}}{x-1} = \frac{x^{-n}(x^n - 1)}{x-1}.$$

Using these results in conjunction with the assumption that $p \geq 2$, we have

$$\sum_{j=1}^{p} s_j = \sum_{k=1}^{K} \left\{ \frac{1}{2} a_k \left[\frac{\exp\left(\frac{i2\pi k}{p}\right)\left[\exp\left(i2\pi k\right) - 1\right]}{\exp\left(\frac{i2\pi k}{p}\right) - 1} + \frac{\exp\left(-i2\pi k\right)\left[\exp\left(i2\pi k\right) - 1\right]}{\exp\left(\frac{i2\pi k}{p}\right) - 1} \right] + \right.$$
$$\left. \frac{1}{2i} b_k \left[\frac{\exp\left(\frac{i2\pi k}{p}\right)\left[\exp\left(i2\pi k\right) - 1\right]}{\exp\left(\frac{i2\pi k}{p}\right) - 1} - \frac{\exp\left(-i2\pi k\right)\left[\exp\left(i2\pi k\right) - 1\right]}{\exp\left(\frac{i2\pi k}{p}\right) - 1} \right] \right\}.$$

Noting that $\exp(i2\pi k) = \cos(i2\pi k) + i\sin(i2\pi k) = 1, k \in K$, each of the $\exp(i2\pi k) - 1$ factors in the terms, and therefore each of the terms in their entirety, in the summation are zero, and therefore we have that

$$\sum_{j=1}^{p} s_j = 0.$$

Writing the jth seasonal effect as

$$s_j = \sum_{k=1}^{K} \left\{ a_k \cos\left(\frac{2\pi k j}{p}\right) + b_k \sin\left(\frac{2\pi k j}{p}\right) \right\},$$

we make the substitution $j = j + p$ and have

$$s_{j+p} = \sum_{k=1}^{K} \left\{ a_k \cos\left(\frac{2\pi k(j+p)}{p}\right) + b_k \sin\left(\frac{2\pi k(j+p)}{p}\right) \right\}$$
$$= \sum_{k=1}^{K} \left\{ a_k \cos\left(\frac{2\pi kj + 2\pi kp}{p}\right) + b_k \sin\left(\frac{2\pi kj + 2\pi kp}{p}\right) \right\}$$

$$= \sum_{k=1}^{K} \left\{ a_k \left[\cos\left(\frac{2\pi kj}{p}\right) \cos\left(\frac{2\pi kp}{p}\right) - \sin\left(\frac{2\pi kj}{p}\right) \sin\left(\frac{2\pi kp}{p}\right) \right] + b_k \left[\sin\left(\frac{2\pi kj}{p}\right) \cos\left(\frac{2\pi kp}{p}\right) + \cos\left(\frac{2\pi kj}{p}\right) \sin\left(\frac{2\pi kp}{p}\right) \right] \right\}$$

$$= \sum_{k=1}^{K} \left\{ a_k \left[\cos\left(\frac{2\pi kj}{p}\right) \cos\left(2\pi k\right) - \sin\left(\frac{2\pi kj}{p}\right) \sin\left(2\pi k\right) \right] + b_k \left[\sin\left(\frac{2\pi kj}{p}\right) \cos\left(2\pi k\right) + \cos\left(\frac{2\pi kj}{p}\right) \sin\left(2\pi k\right) \right] \right\},$$

where we made use of the identities

$$\cos(u+v) \equiv \cos u \cos v - \sin u \sin v,$$

$$\sin(u+v) \equiv \sin u \cos v + \cos u \sin v.$$

Given that $k \in K$,

$$\cos(2\pi k) = 1, \sin(2\pi k) = 0,$$

and thus the summation reduces to

$$s_{j+p} = \sum_{k=1}^{K} \left\{ a_k \cos\left(\frac{2\pi kj}{p}\right) + b_k \sin\left(\frac{2\pi kj}{p}\right) \right\},$$

that is, the jth and $(j+p)$th seasonal effects are precisely the same.

The above procedure can be repeated indefinitely to see that

$$s_j = s_{j+p} = s_{j+2p} = \cdots,$$

that is, the seasonal effects are equal across each period.

Appendix

Included within this appendix is graphical output relating to section 1.1, followed by the source code used to perform the analysis within section 1.1. Each set of graphical output is contained within a labelled figure, describing what it contains, and the source code is listed with line numbers corresponding to each line within the source code.

Figure 1: the dataset, modelled as a time series.

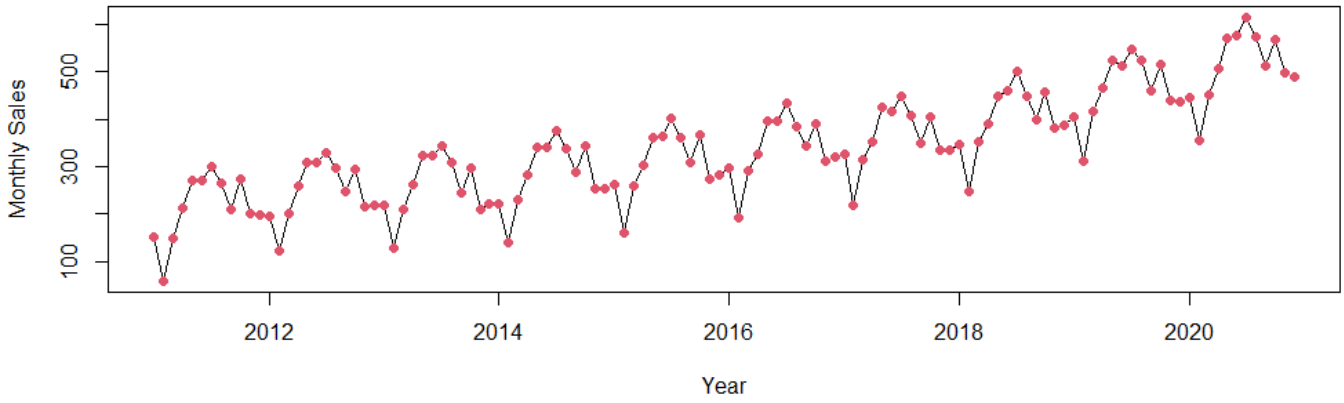


Figure 2: the dataset, modelled as a time series, with lines connecting the minima and maxima of each season.

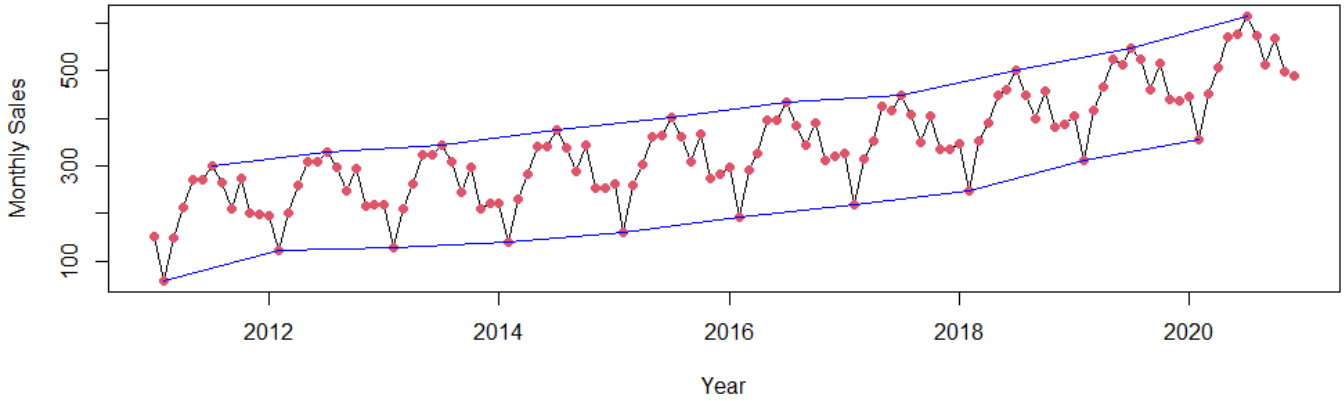


Figure 3: the autocorrelation function of the time series.

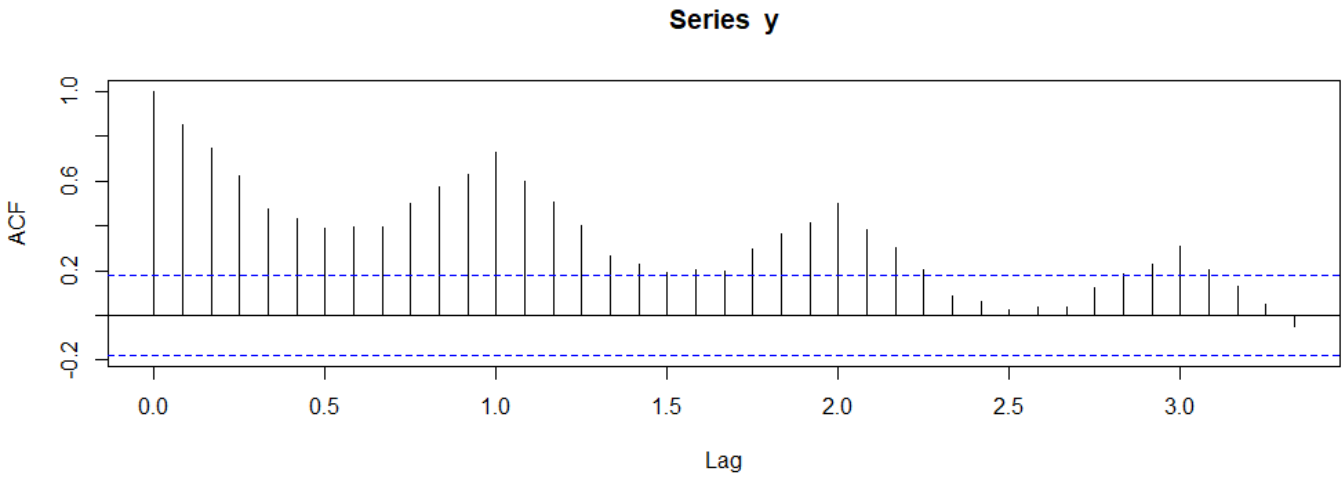


Figure 4: the seasonally differenced time series, then the autocorrelation function of the seasonally differenced time series.

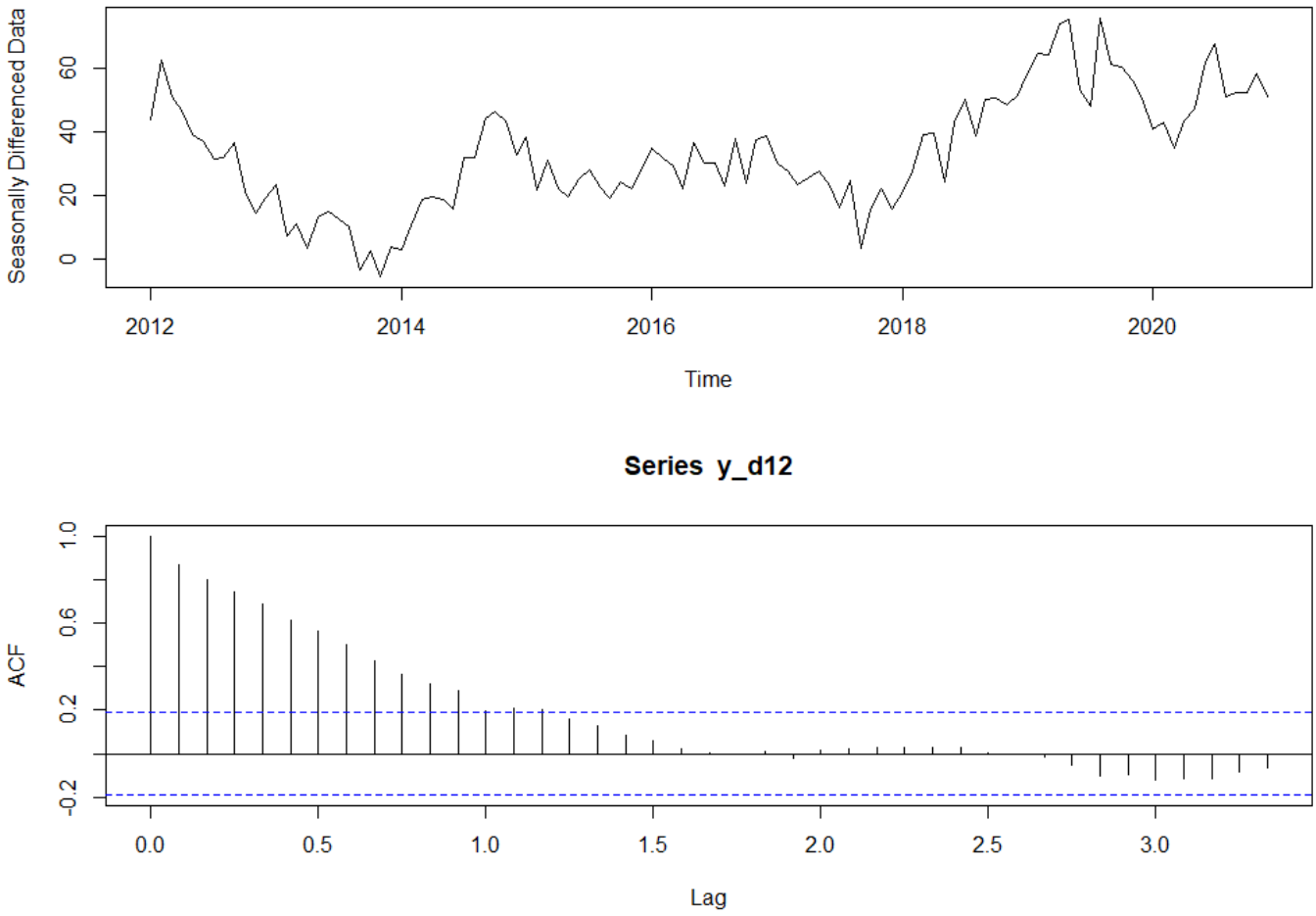


Figure 5: the time series differenced both seasonally and non-seasonally, then its autocorrelation function.

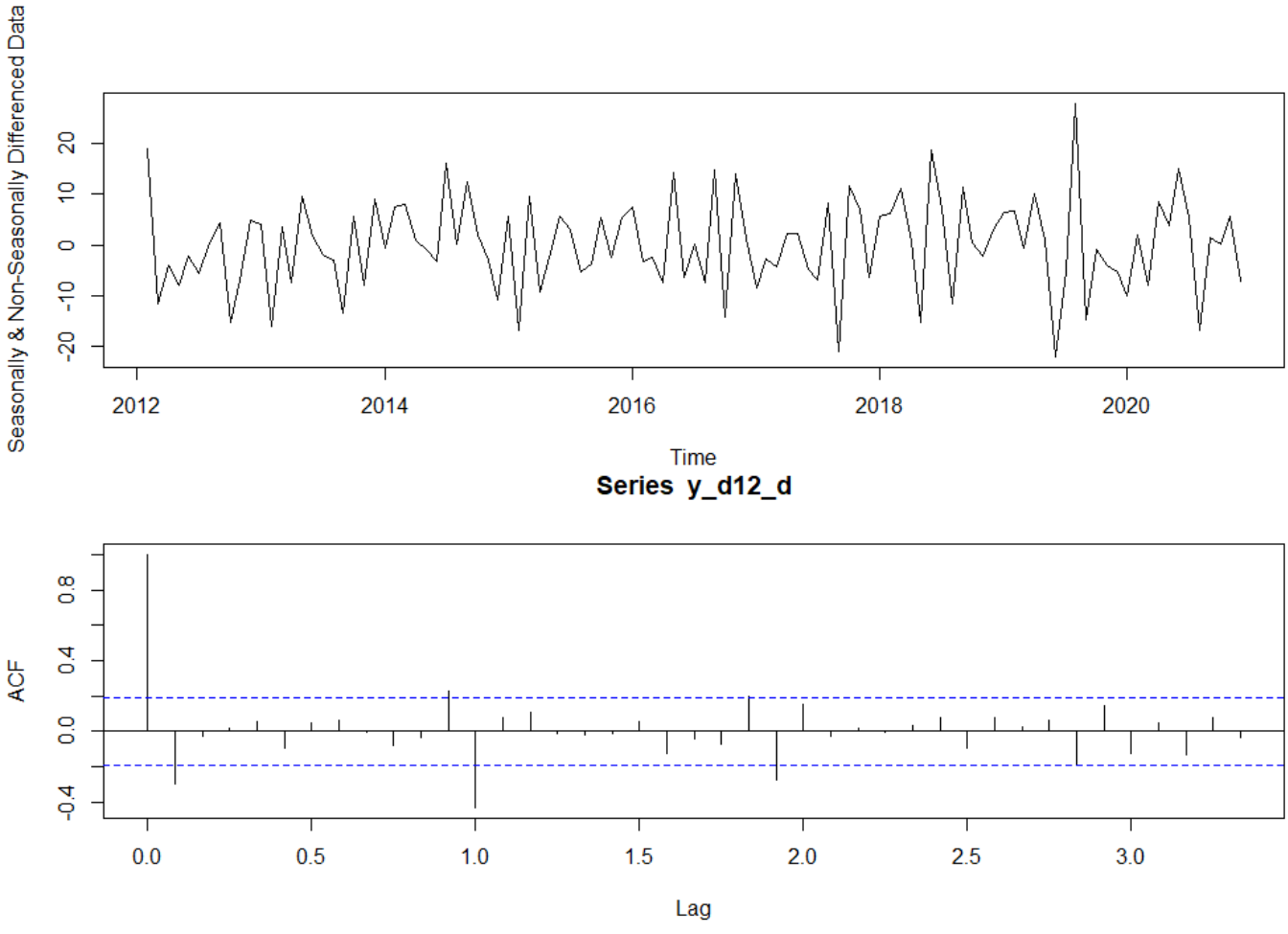


Figure 6: the partial autocorrelation function for the time series differenced both seasonally and non-seasonally.

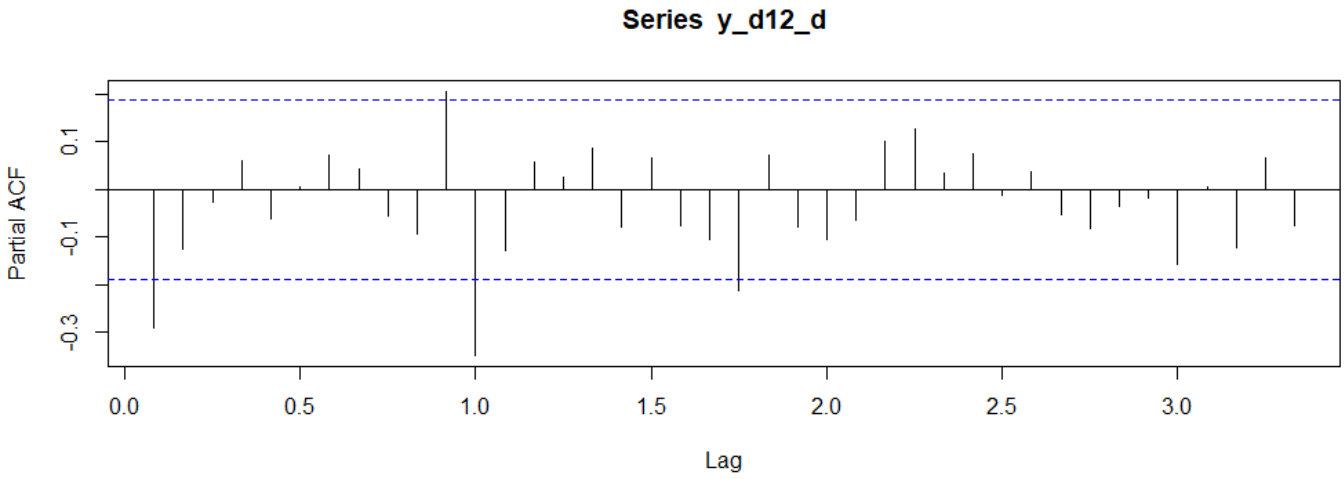
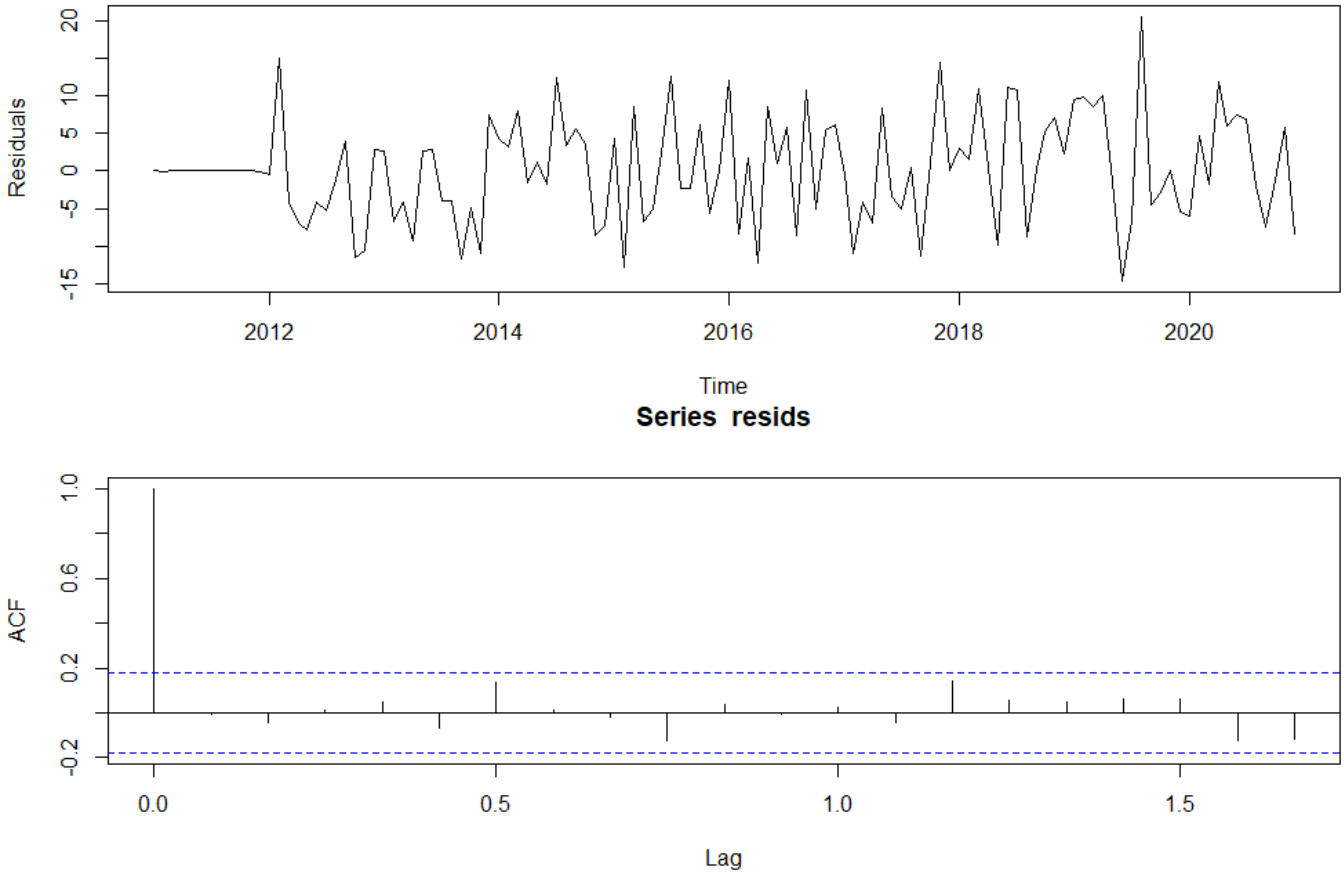


Figure 7: the residuals of the $\mathrm{ARIMA}\{(1,1,0)\times(0,1,1)_{12}\}$ model fitted to the time series, with parameters estimated through maximum likelihood, then the residuals' autocorrelation function, then the residuals' partial autocorrelation function.



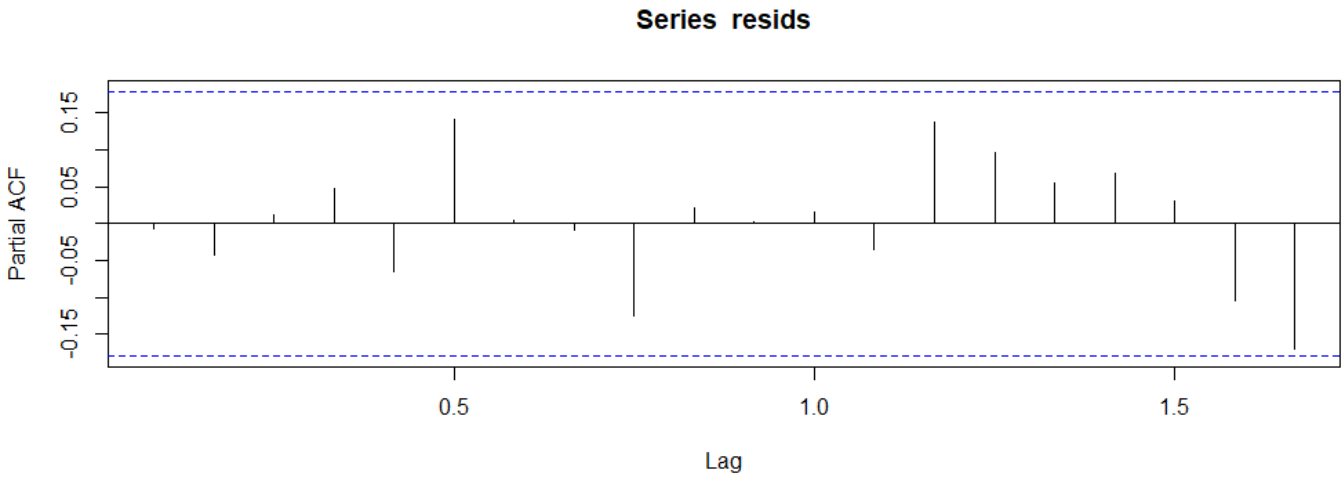


Figure 8: the p-values for the Ljung-Box test performed on the residuals for the $\mathrm{ARIMA}\{(1,1,0)\times(0,1,1)_{12}\}$ model.

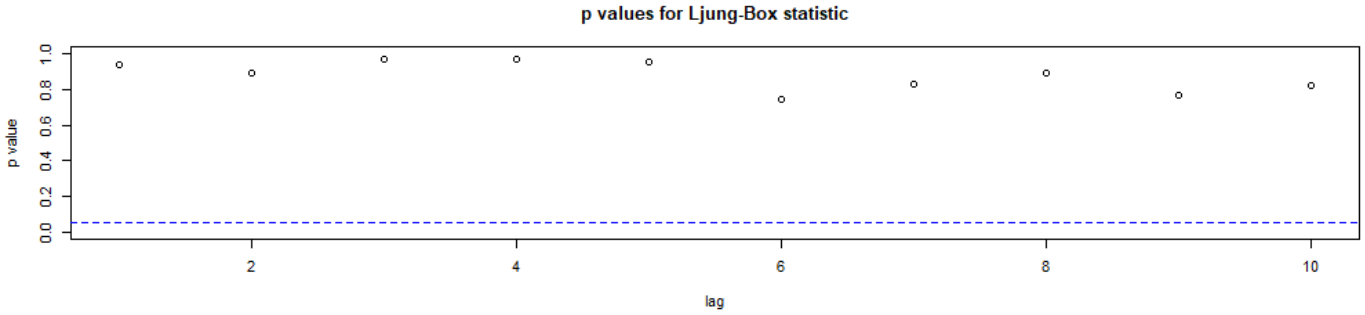


Figure 9: the fitted values for the linear and quadratic trend regression models fitted to the time series.

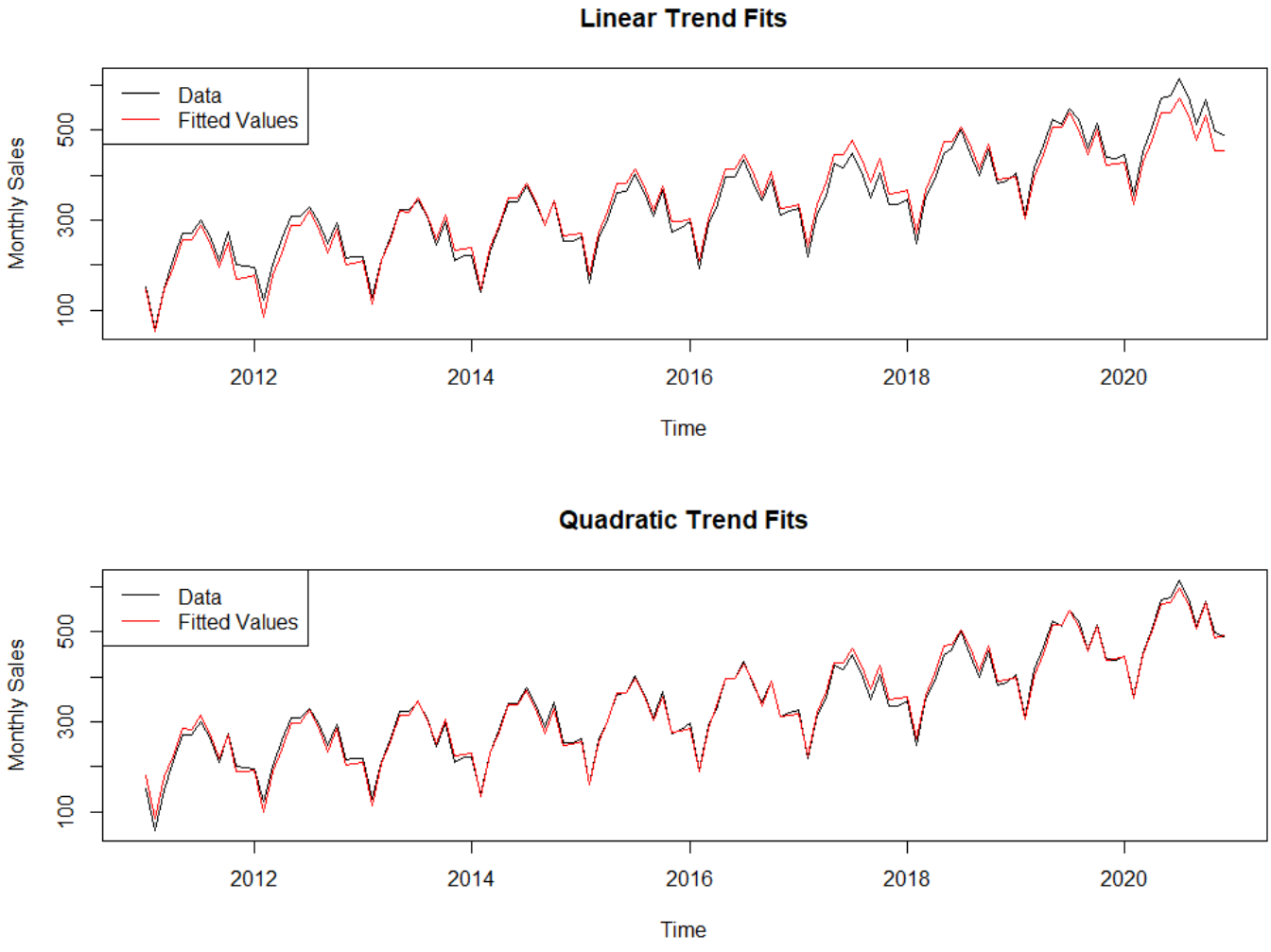


Figure 10: the residuals of the quadratic trend regression model fitted to the time series.

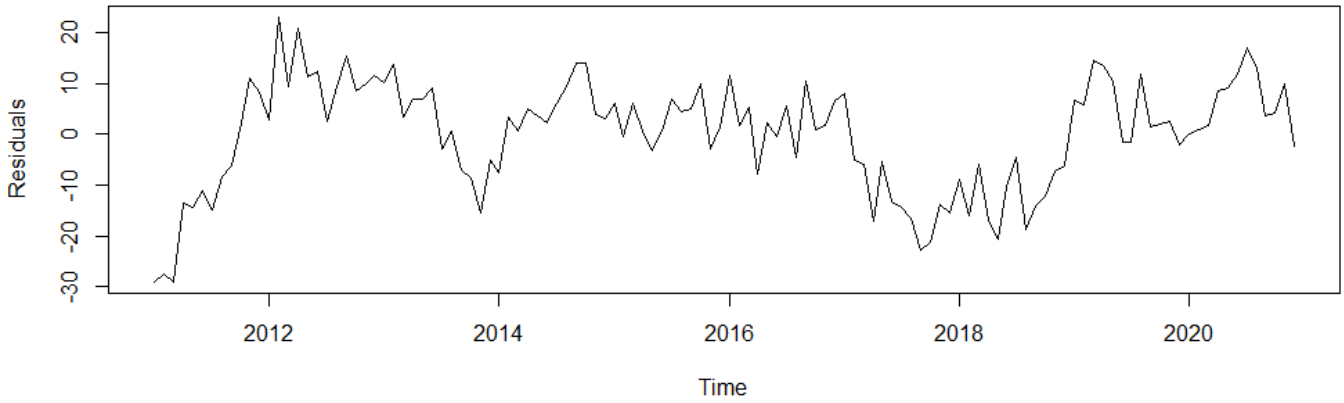


Figure 11: the autocorrelation function for the residuals of the quadratic trend regression model fitted to the time series, then its partial autocorrelation function.

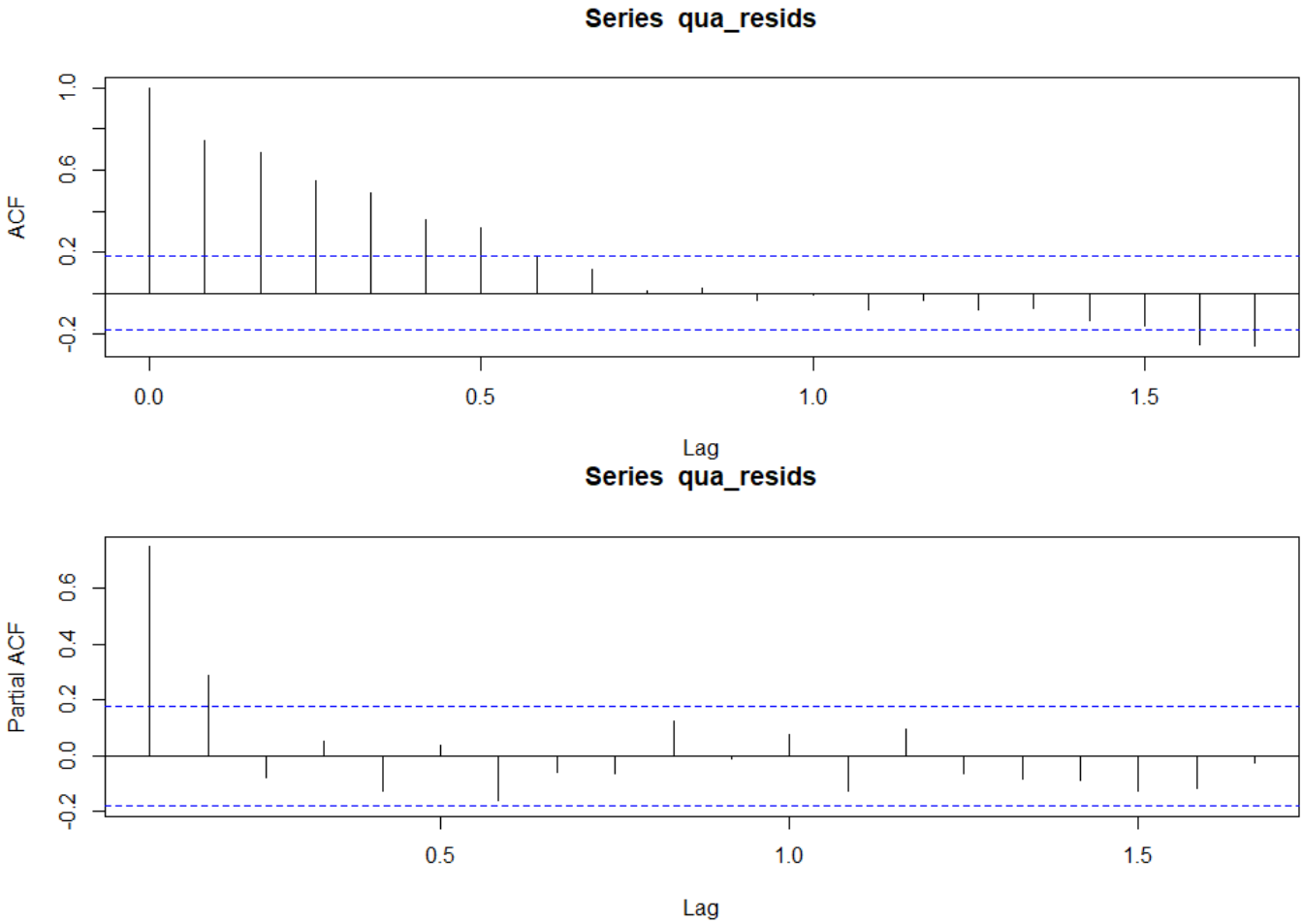


Figure 12: the residuals of the $\mathrm{AR}(2)$ model fitted to the residuals of the quadratic trend regression model, then its autocorrelation *function*, then its partial autocorrelation function, then the p-values for the Ljung-Box test performed on the residuals of the $\mathrm{AR}(2)$ model.

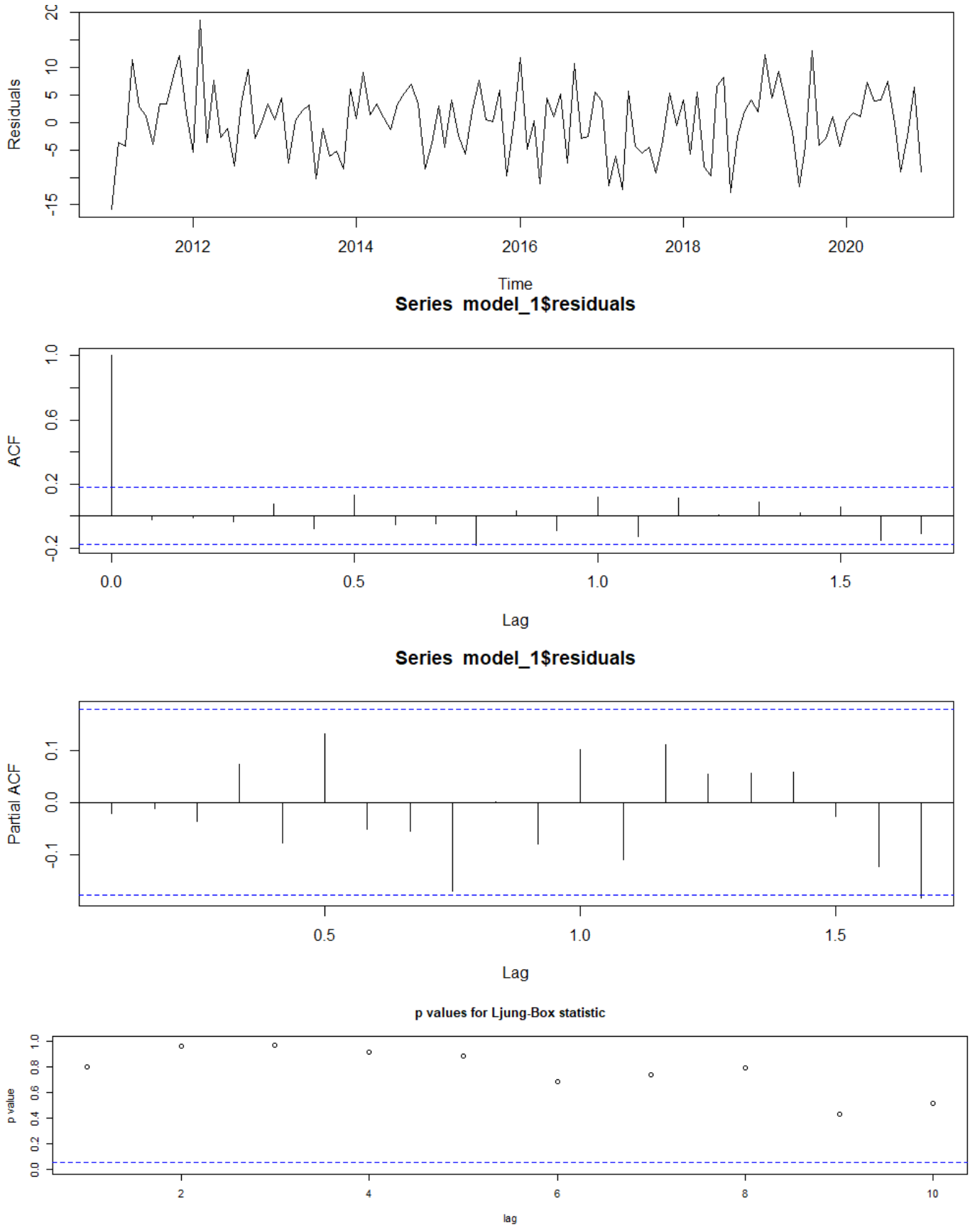


Figure 13: the time series overlaid by the smoothed trend values of the dynamic linear model, with parameters estimated by maximum likelihood.

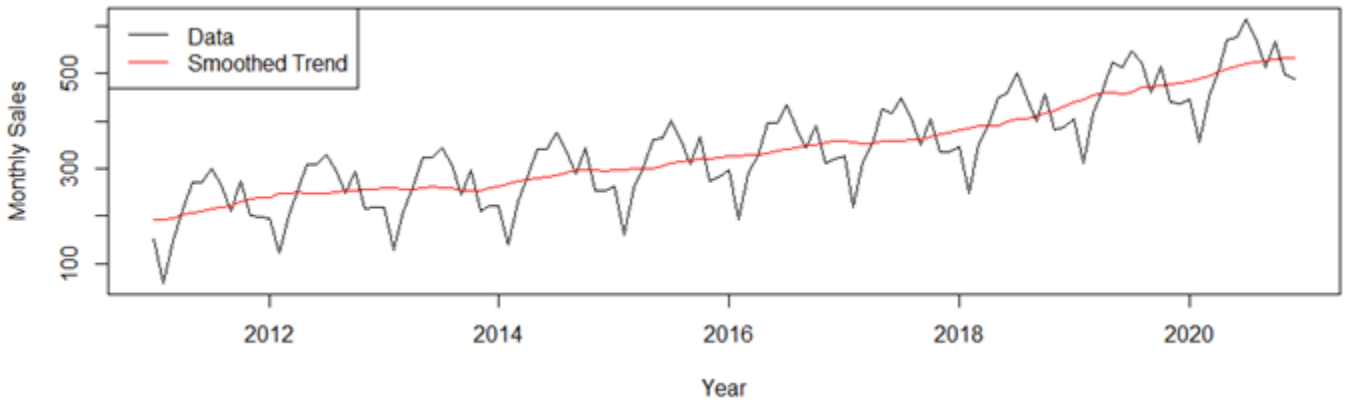
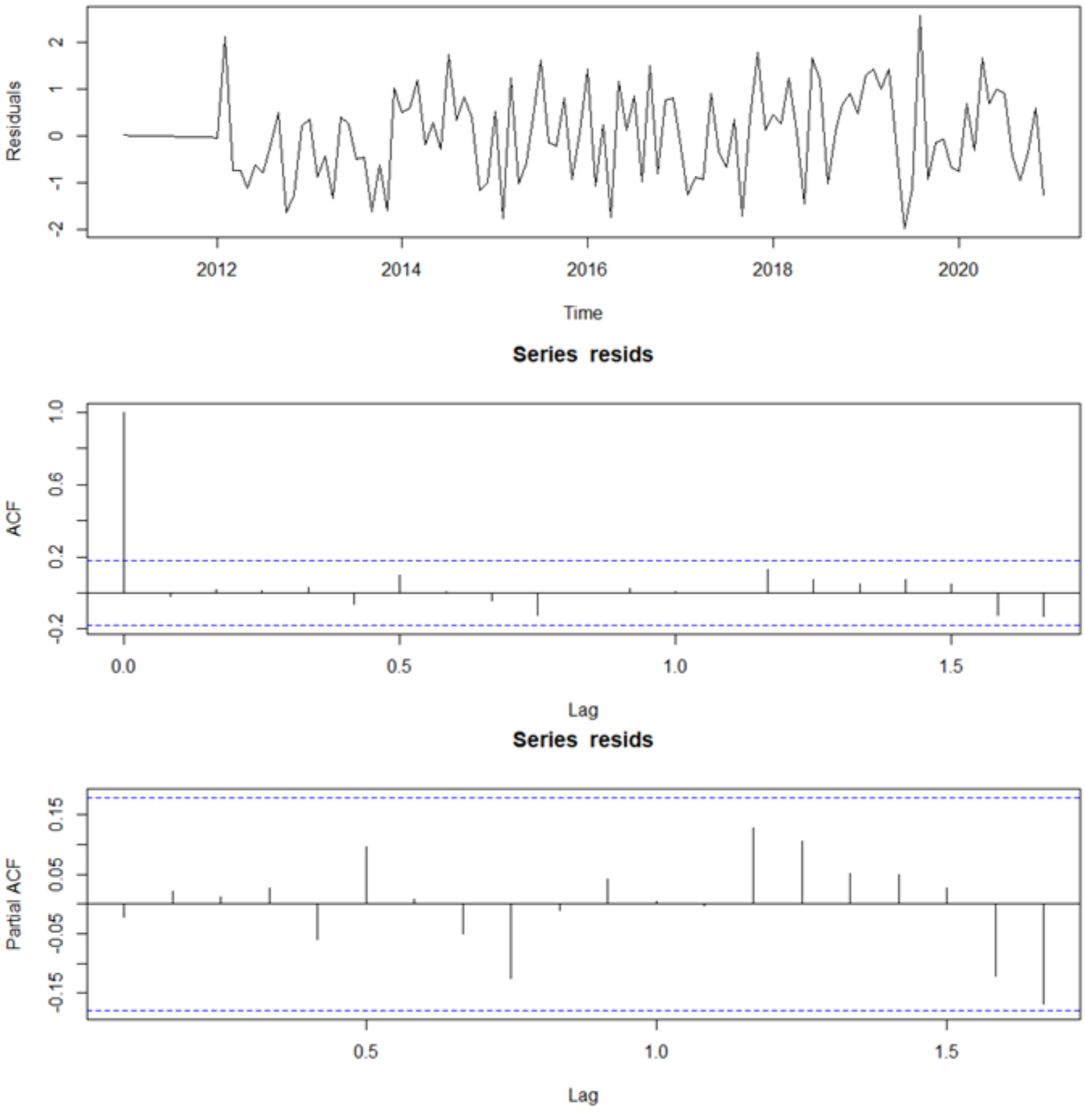


Figure 14: the residuals of the filtered values of the dynamic linear model fitted to the time series, then its autocorrelation function, then its partial autocorrelation function, then the p-values for the Ljung-Box test performed on the residuals of the filtered values of the dynamic linear model.



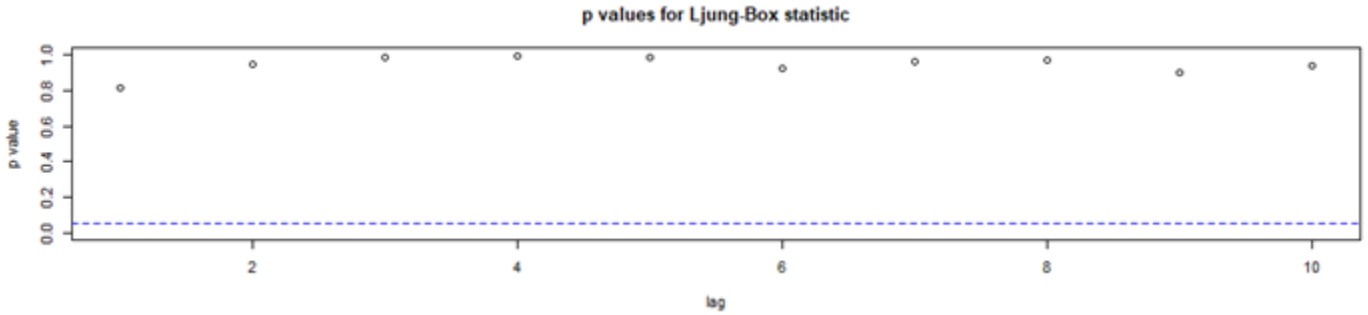


Figure 15: the decomposition of the time series into trend and seasonal effects, using the smoothed state vectors from the fitted dynamic linear model

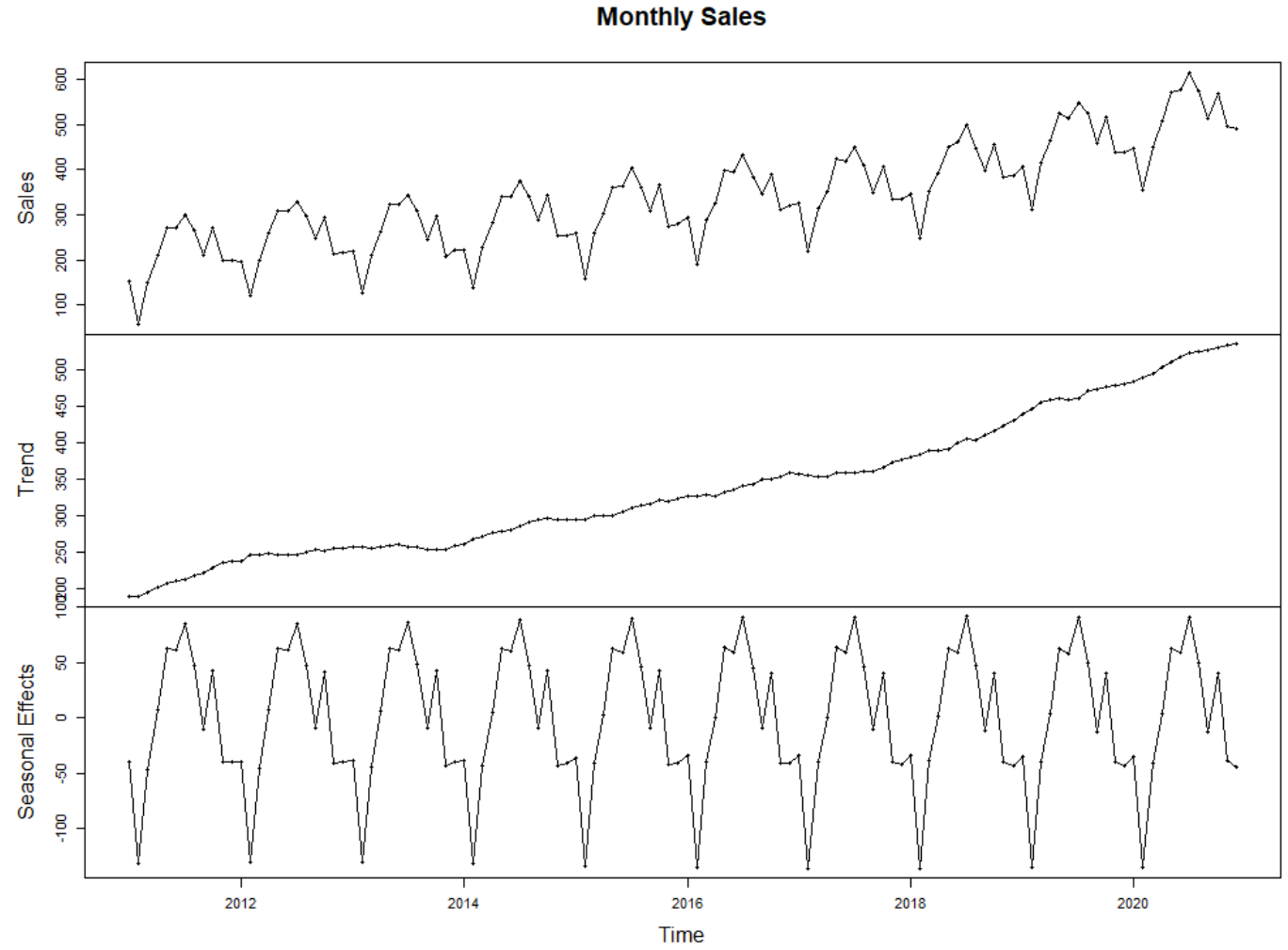


Figure 16: the 12 month point forecasts and associated prediction intervals for the linear trend and quadratic trend regression models fitted over the training data, overlaid by the test set data.

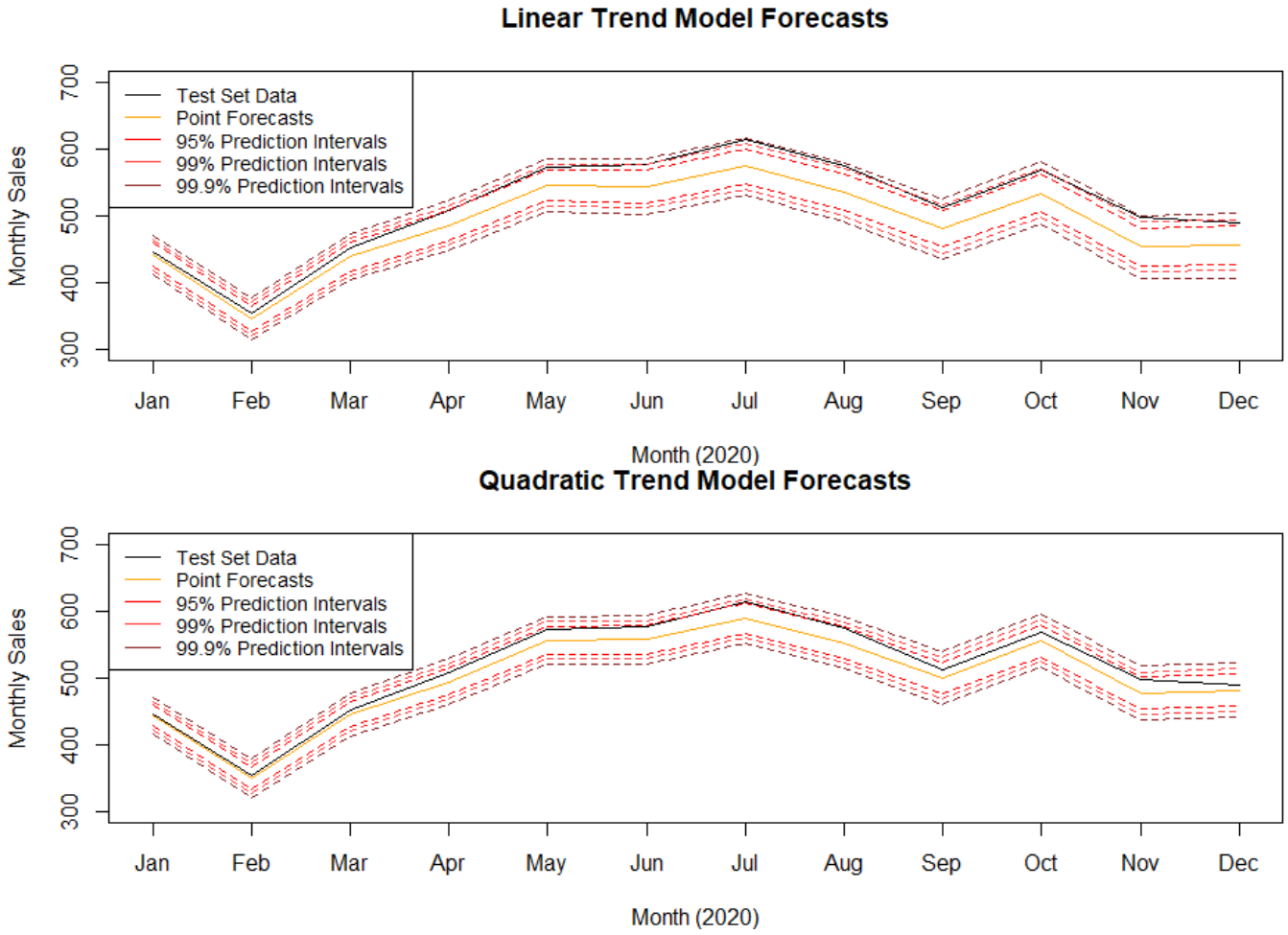


Figure 17: the time series attached to its point forecasts and forecast variances for January through June 2021 inclusive, constructed using the quadratic trend regression model fitted over the full data, then the same plot again, zoomed in on January 2019 through June 2021.

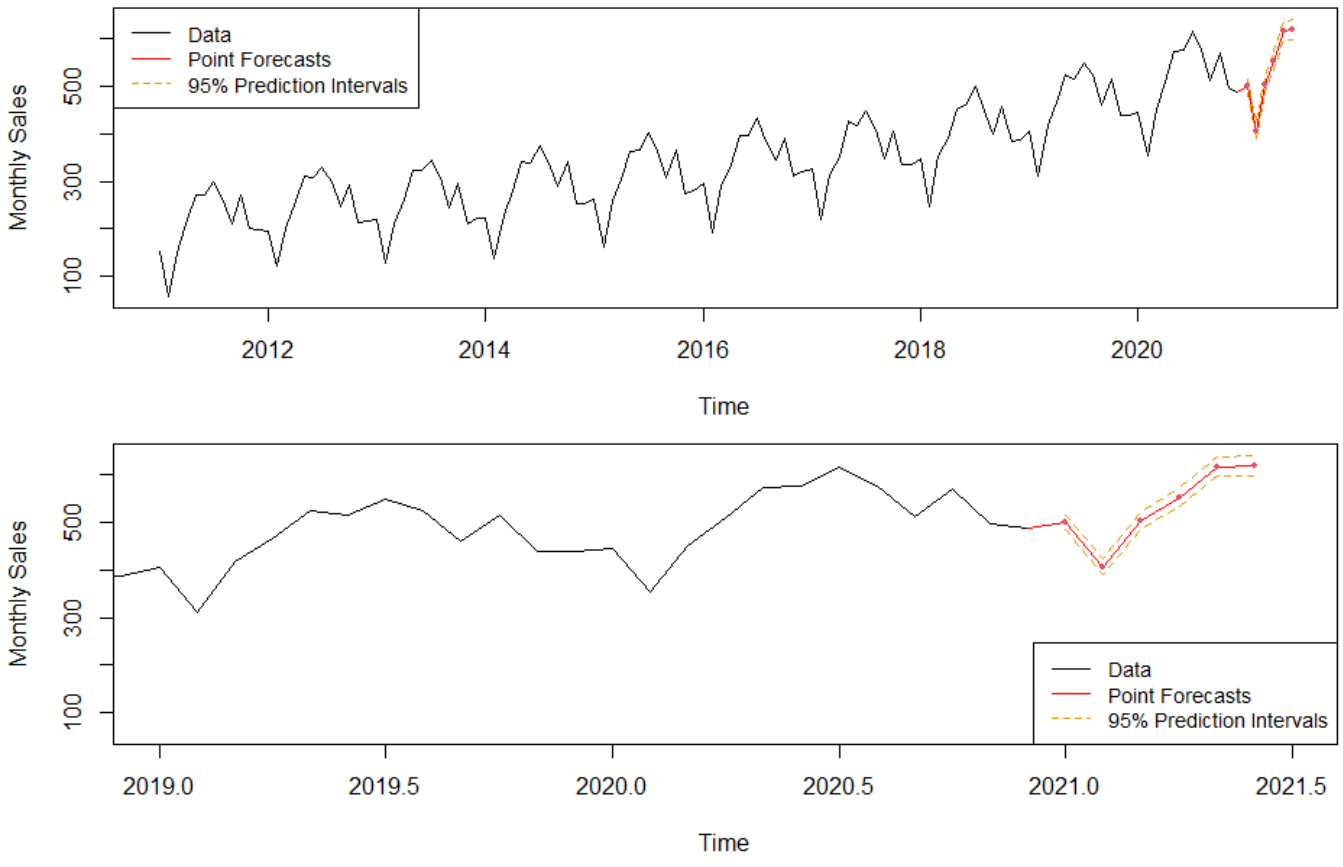


Figure 18: the R source code used to produce the analysis within section 1.1.

```r
# dlm package for dynamic linear modelling
library(dlm)

# read in project data
filename = 'projectdata.txt'
data = read.table(filename, header = FALSE)
y = data[, 7]

# convert data to time series
y = ts(y, start = c(2011,1), end = c(2020, 12), frequency = 12)

# exploratory plot
par(mfrow = c(2,1))
plot(y, xlab = 'Year', ylab = 'Monthly Sales')
points(y, pch = 21, col = 2, bg = 2)
# overlay lines connecting the minima and maxima points of each season
minima_times = seq(2011.083, 2020.083, 1)
lines(minima_times, y[seq(2, 110, 12)],  col = 'blue')
maxima_times = seq(2011.5, 2020.5, 1)
lines(maxima_times, y[seq(7, 115, 12)],  col = 'blue')

# arima modelling


# acf correlogram
acf(y, lag.max = 40)

# first seasonal differences and plots, d* = 1
y_d12 = diff(y, lag = 12)
plot(y_d12, ylab = 'Seasonally Differenced Data')
acf(y_d12, lag.max = 40)

# first non-seasonal differences and plots, d* = 1, d = 1
y_d12_d = diff(y_d12)
plot(y_d12_d, ylab = 'Seasonally & Non-Seasonally Differenced Data')
acf(y_d12_d, lag.max = 40)

# pacf plot, d* = 1, d = 1
pacf(y_d12_d, lag.max = 40)

# ARIMA(1,1,0)x(0,1,1)_12
model_1 = arima(y, order = c(1,1,0), seasonal = c(0,1,1))
resids = model_1$residuals
plot(resids, ylab = 'Residuals')
acf(resids, lag.max = 40)
pacf(resids, lag.max = 40)
tsdiag(model_1) # ljung-box, use the third plot
shapiro.test(resids)

# overfit models
model_1$loglik
# model fits
model_1a = arima(y, order = c(2,1,0), seasonal = c(0,1,1))
model_1b = arima(y, order = c(1,1,1), seasonal = c(0,1,1))
model_1c = arima(y, order = c(1,1,0), seasonal = c(1,1,1))
model_1d = arima(y, order = c(1,1,0), seasonal = c(0,1,2))
# log likelihoods
model_1a$loglik
model_1b$loglik
model_1c$loglik
model_1d$loglik
# likelihood ratio test statistics
test_stata = 2 * (model_1a$loglik - model_1$loglik)
test_statb = 2 * (model_1b$loglik - model_1$loglik)
test_statc = 2 * (model_1c$loglik - model_1$loglik)
test_statd = 2 * (model_1d$loglik - model_1$loglik)
# p-values for the tests assuming chi-square distribution
```

```r
1 - pchisq(test_stata, df=1)
1 - pchisq(test_statb, df=1)
1 - pchisq(test_statc, df=1)
1 - pchisq(test_statd, df=1)

# underfit models
# model fits
model_1e = arima(y, order = c(0,1,0), seasonal = c(0,1,1))
model_1f = arima(y, order = c(1,1,0), seasonal = c(0,1,0))
# log likelihoods
model_1e$loglik
model_1f$loglik
# likelihood ratio test statistics
test_state = 2 * (model_1$loglik - model_1e$loglik)
test_statf = 2 * (model_1$loglik - model_1f$loglik)
# p-values for the tests assuming chi-square distribution
1 - pchisq(test_state, df=1)
1 - pchisq(test_statf, df=1)

# final model summary
model_1


# regression modelling


# fit regression model with linear trend
# construct data frame
month = rep(factor(month.abb, levels = month.abb), length.out = length(y))
y_lin_df = data.frame(y = y, t = 1:length(y), month = month)
# regression fit
lin_mean_model = lm(y ~., data = y_lin_df)
# fitted values and residuals
lin_fits = lin_mean_model$fitted.values
lin_resids = lin_mean_model$residuals
# model summary
summary(lin_mean_model)

# fit regression model with quadratic trend
# construct data frame
y_qua_df = data.frame(y = y, t = 1:length(y), t_sq = (1:length(y))^2,
                      month = month)
# regression fit
qua_mean_model = lm(y ~., data = y_qua_df)
# fitted values and residuals
qua_fits = qua_mean_model$fitted.values
qua_resids = qua_mean_model$residuals

# fitted values and residuals plots
# convert fits and resids to ts
lin_fits = ts(lin_fits, start = start(y), end = end(y),
              frequency = frequency(y))
lin_resids = ts(lin_resids, start = start(y), end = end(y),
                frequency = frequency(y))
qua_fits = ts(qua_fits, start = start(y), end = end(y),
              frequency = frequency(y))
qua_resids = ts(qua_resids, start = start(y), end = end(y),
                frequency = frequency(y))
# plot the data with overlaid fits
plot(y, ylab = 'Monthly Sales', main = 'Linear Trend Fits')
lines(lin_fits, col = 'red')
legend('topleft', c('Data', 'Fitted Values'), col = c('black', 'red'), lty = 1)
plot(y, ylab = 'Monthly Sales', main = 'Quadratic Trend Fits')
lines(qua_fits, col = 'red')
legend('topleft', c('Data', 'Fitted Values'), col = c('black', 'red'), lty = 1)

#aic scores for quadratic and linear trend models, and their difference
```

```r
c(AIC(lin_mean_model), AIC(qua_mean_model),
  AIC(lin_mean_model)-AIC(qua_mean_model))

# plot quadratic trend model residuals
plot(qua_resids, ylab = 'Residuals')

# acf and pacf of residuals
acf(qua_resids)
pacf(qua_resids)

# fit zero-mean AR(2) model to residuals, then residual plots
model_1 = arima(qua_resids, order = c(2,0,0), include.mean = FALSE)
plot(model_1$residuals, ylab = 'Residuals')
acf(model_1$residuals)
pacf(model_1$residuals)
tsdiag(model_1) # ljung-box, use the third plot
shapiro.test(model_1$residuals)

# candidate models to replace current
model_1$loglik
# model fits
model_1a = arima(qua_resids, order = c(3,0,0), include.mean = FALSE)
model_1b = arima(qua_resids, order = c(2,0,1), include.mean = FALSE)
model_1c = arima(qua_resids, order = c(1,0,0), include.mean = FALSE)
# log likelihoods
model_1a$loglik
model_1b$loglik
model_1c$loglik
# likelihood ratio test statistics
test_stata = 2 * (model_1a$loglik - model_1$loglik)
test_statb = 2 * (model_1b$loglik - model_1$loglik)
test_statc = 2 * (model_1$loglik - model_1c$loglik)
# p-values for the tests assuming chi-square distribution
1 - pchisq(test_stata, df=1)
1 - pchisq(test_statb, df=1)
1 - pchisq(test_statc, df=1)

# final residual arma model and main model summary
summary(qua_mean_model)
model_1


# dynamic linear modelling


# function to build dlm
build_model = function(params) {
  dlmModPoly(order = 2, dV = exp(params[1]), dW = exp(params[2:3])) +
    dlmModTrig(s = 12, dV = 0, dW = exp(rep(params[4], 11)))
}

# find mles for dlm and check convergence
model_fit = dlmMLE(y, parm = c(0, 0, 0, 0), build = build_model)
model_fit$convergence # should equal 0

# build model for data
model = build_model(model_fit$par)

# view smoothed values alongside data
model_smooth = dlmSmooth(y, model)
model_level = dropFirst(model_smooth$s[, 1])
plot(y, xlab = 'Year', ylab = 'Monthly Sales')
lines(model_level, col = 'red')
legend('topleft', c('Data', "Smoothed Trend"), col=c("black", "red"), lty = 1)

# residual analysis for dlm
model_filter = dlmFilter(y, model)
```

```r
resids = residuals(model_filter, sd = FALSE)
plot(resids, ylab = 'Residuals')
acf(resids)
pacf(resids)
tsdiag(model_filter) # ljung-box, use the third plot
shapiro.test(resids)

# final dlm summary
str(model) # derive components from this
# trend and seasonal component decomposition
x = cbind(y, dropFirst(model_smooth$s[,1]),
          dropFirst(model_smooth$s[,-(1:2)]) %*% t(model$FF)[-(1:2)])
colnames(x) = c("Sales", "Trend", "Seasonal Effects")
plot(x, type="o", main="Monthly Sales")


# regression model forecasting


# divide the data into a training and test set
train_y = ts(y[1:108], start = c(2011,1), end = c(2019, 12), frequency = 12)
test_y = ts(y[109:120], start = c(2020,1), end = c(2020, 12), frequency = 12)

# fit linear regression model on training set
# construct data frame
month = rep(factor(month.abb, levels = month.abb), length.out = length(train_y))
train_y_df = data.frame(train_y = train_y, t = 1:length(train_y), month = month)
# regression fit
train_y_mean_model = lm(train_y ~., data = train_y_df)
summary(train_y_mean_model)
# residual ARMA fit
resids = train_y_mean_model$residuals
resids = ts(resids, start = start(train_y), end = end(train_y),
            frequency = frequency(train_y))
train_y_resids_model = arima(resids, order = c(2,0,0), include.mean = FALSE)
train_y_resids_model

# linear forecasts for the test set period
# forecast of the residuals
forecast_resids = predict(train_y_resids_model, n.ahead = 12)
preds_resids = forecast_resids$pred # point forecasts
predvar_resids = forecast_resids$se^2 # forecast variances
# forecasts of the linear and seasonal components
new_month = factor(month.abb, levels = month.abb)
new_y_df = data.frame(t = 109:120, month= new_month)
forecast = predict(train_y_mean_model, new_y_df) # naive point forecasts
preds_naive = ts(forecast, frequency = 12, start = c(2020,1))
# construct overall forecasts
confvar = predict(train_y_mean_model, new_y_df, interval = 'confidence',
                  se.fit = TRUE)$se.fit^2 # mean model confidence int. variances
preds = preds_naive + preds_resids # overall point forecasts
predvar = confvar + predvar_resids # overall forecast variances
lower0.95 = preds - 1.96 * sqrt(predvar) # lower bound at 95%
upper0.95 = preds + 1.96 * sqrt(predvar) # upper bound at 95%
lower0.99 = preds - 2.58 * sqrt(predvar) # lower bound at 99%
upper0.99 = preds + 2.58 * sqrt(predvar) # upper bound at 99%
lower0.999 = preds - 3.29 * sqrt(predvar) # lower bound at 99.9%
upper0.999 = preds + 3.29 * sqrt(predvar) # upper bound at 99.9%

# plot linear point forecasts, forecast prediction intervals, test set
train_y_preds_forecast = ts(preds, start = c(2020,1), frequency = 12)
plot(test_y, xlab = 'Month (2020)', ylab = 'Monthly Sales', xaxt = 'n',
     ylim = c(300,700), main = 'Linear Trend Model Forecasts')
tsp = attributes(test_y)$tsp
axis(1, at = seq(tsp[1], tsp[2], along = test_y), labels=new_month)
lines(train_y_preds_forecast, col = 'orange')
lines(lower0.95, lty=2, col= 'red')
```

```r
lines(upper0.95, lty=2, col= 'red')
lines(lower0.99, lty=2, col= 'brown1')
lines(upper0.99, lty=2, col= 'brown1')
lines(lower0.999, lty=2, col= 'brown4')
lines(upper0.999, lty=2, col= 'brown4')
legend("topleft", c('Test Set Data', 'Point Forecasts',
                    '95% Prediction Intervals',
                    '99% Prediction Intervals',
                    '99.9% Prediction Intervals'),
       col=c('black', "orange", "red", "brown1", "brown4"),
       lty = c(1,1,1,1,1),
       cex = 0.85)

# fit quadratic regression model on training set
# construct data frame
month = rep(factor(month.abb, levels = month.abb), length.out = length(train_y))
qua_train_y_df = data.frame(train_y = train_y, t = 1:length(train_y),
                            t_sq = (1:length(train_y))^2, month = month)
# regression fit
qua_train_y_mean_model = lm(train_y ~., data = qua_train_y_df)
summary(qua_train_y_mean_model)
# residual ARMA fit
resids = qua_train_y_mean_model$residuals
resids = ts(resids, start = start(train_y), end = end(train_y),
            frequency = frequency(train_y))
qua_train_y_resids_model = arima(resids, order = c(2,0,0), include.mean = FALSE)
qua_train_y_resids_model

# quadratic forecasts for the test set period
# forecast of the residuals
forecast_resids = predict(qua_train_y_resids_model, n.ahead = 12)
preds_resids = forecast_resids$pred # point forecasts
predvar_resids = forecast_resids$se^2 # forecast variances
# forecasts of the linear and seasonal components
new_month = factor(month.abb, levels = month.abb)
new_y_df = data.frame(t = 109:120, t_sq = (109:120)^2, month= new_month)
forecast = predict(qua_train_y_mean_model, new_y_df) # naive point forecasts
preds_naive = ts(forecast, frequency = 12, start = c(2020,1))
# construct overall forecasts
confvar = predict(qua_train_y_mean_model, new_y_df, interval = 'confidence',
                  se.fit = TRUE)$se.fit^2 # mean model confidence int. variances
preds = preds_naive + preds_resids # overall point forecasts
predvar = confvar + predvar_resids # overall forecast variances
lower0.95 = preds - 1.96 * sqrt(predvar) # lower bound at 95%
upper0.95 = preds + 1.96 * sqrt(predvar) # upper bound at 95%
lower0.99 = preds - 2.58 * sqrt(predvar) # lower bound at 99%
upper0.99 = preds + 2.58 * sqrt(predvar) # upper bound at 99%
lower0.999 = preds - 3.29 * sqrt(predvar) # lower bound at 99.9%
upper0.999 = preds + 3.29 * sqrt(predvar) # upper bound at 99.9%

# plot quadratic point forecasts, forecast prediction intervals, test set
qua_train_y_preds_forecast = ts(preds, start = c(2020,1), frequency = 12)
plot(test_y, xlab = 'Month (2020)', ylab = 'Monthly Sales', xaxt = 'n',
     ylim = c(300,700), main = 'Quadratic Trend Model Forecasts')
tsp = attributes(test_y)$tsp
axis(1, at = seq(tsp[1], tsp[2], along = test_y), labels=new_month)
lines(qua_train_y_preds_forecast, col = 'orange')
lines(lower0.95, lty=2, col= 'red')
lines(upper0.95, lty=2, col= 'red')
lines(lower0.99, lty=2, col= 'brown1')
lines(upper0.99, lty=2, col= 'brown1')
lines(lower0.999, lty=2, col= 'brown4')
lines(upper0.999, lty=2, col= 'brown4')
legend("topleft", c('Test Set Data', 'Point Forecasts',
                    '95% Prediction Intervals',
                    '99% Prediction Intervals',
                    '99.9% Prediction Intervals'),
```

```
       col=c('black', "orange", "red", "brown1", "brown4"),
       lty = c(1,1,1,1,1),
       cex = 0.85)

# mean absolute percentage error of linear forecasts and observations
(mape = mean(100*abs(as.numeric(test_y) - preds)/as.numeric(test_y)))

# forecasts for the full model
# fit full regression model again
# construct data frame
month = rep(factor(month.abb, levels = month.abb), length.out = length(y))
y_df = data.frame(y = y, t = 1:length(y), t_sq = (1:length(y))^2, month = month)
# regression fit
mean_model = lm(y ~., data = y_df)
# residuals
resids = mean_model$residuals
resids = ts(resids, start = start(y), end = end(y), frequency = frequency(y))
# fit zero-mean AR(2) model
resids_model = arima(resids, order = c(2,0,0), include.mean = FALSE)
# forecast of the residuals
forecast_resids = predict(resids_model, n.ahead = 6)
preds_resids = forecast_resids$pred # point forecasts
predvar_resids = forecast_resids$se^2 # forecast variances
# forecast of the linear and seasonal components
new_month = factor(month.abb[1:6], levels = month.abb[1:6])
new_y_df = data.frame(t = 121:126, t_sq = (121:126)^2, month= new_month)
forecast = predict(mean_model, new_y_df) # naive point forecasts
preds_naive = ts(forecast, frequency = 12, start = c(2021,1))
# construct overall forecasts
confvar = predict(mean_model, new_y_df, interval = 'confidence',
                  se.fit = TRUE)$se.fit^2 # mean model confidence int. variances
preds = preds_naive + preds_resids # overall point forecasts
predvar = confvar + predvar_resids # overall forecast variances
lower = preds - 1.96 * sqrt(predvar) # lower bound at 95%
upper = preds + 1.96 * sqrt(predvar) # upper bound at 95%

# plot point forecasts, forecast prediction intervals for 2021 jan - jun
y_forecast = ts(c(y, upper), start = start(y), frequency = frequency(y))
y_preds = ts(c(y[length(y)], preds), start = end(y), frequency = frequency(y))
plot(y_forecast, ylab = 'Monthly Sales', type = 'n')
lines(y, col = 'black')
lines(y_preds, col = 'red')
lines(lower, lty=2, col= 'orange2')
lines(upper, lty=2, col= 'orange2')
points(ts(y_preds[-1], start = c(2021, 1), frequency = 12),
       pch = 21, col = 2, bg = 2, cex = 0.5)
legend("topleft", c('Data', 'Point Forecasts',
                    '95% Prediction Intervals'),
       col=c('black', "red", "orange2"),
       lty = c(1,1,2),
       cex = 0.9)

# zoomed in version of the above plot
plot(y_forecast, ylab = 'Monthly Sales', type = 'n', xlim = c(2019,2021.5))
lines(y, col = 'black')
lines(y_preds, col = 'red')
lines(lower, lty=2, col= 'orange2')
lines(upper, lty=2, col= 'orange2')
points(ts(y_preds[-1], start = c(2021, 1), frequency = 12),
       pch = 21, col = 2, bg = 2, cex = 0.5)
legend("bottomright", c('Data', 'Point Forecasts',
                        '95% Prediction Intervals'),
       col=c('black', "red", "orange2"),
       lty = c(1,1,2),
       cex = 0.9)
```