Coursework Submission

University ID: 10667974

April 14, 2023

Report for MATH38032 Time Series Analysis

The aim of this report is to observe and model the time series relating to the sales of Australian wine and also try to predict sales relating to the year 1995.

Part 1) Data Import in R

(i) Explanation: In **part 1**, we are asked to import the data in R till the end of 1994. The data itself consists of the monthly sales (in Litres) of fortified wine in Australia, from January 1980 till July of 1995 and ranges from $[1154, 5618]$ and the median value is 2894 litres sold. To import the data, we use $scan()$, which will create a vector of data, with length of 187. We then proceed to cut out the last few data points by creating a new vector adding data points up till index 180.
(ii) R Code:

```
sales <- scan("fortif.txt")  #Read in the data
cut.sales <- sales[1:180] #Use only data points till end of 1994
summary(cut.sales) #Summary of data
```

Part 2) Plotting Data

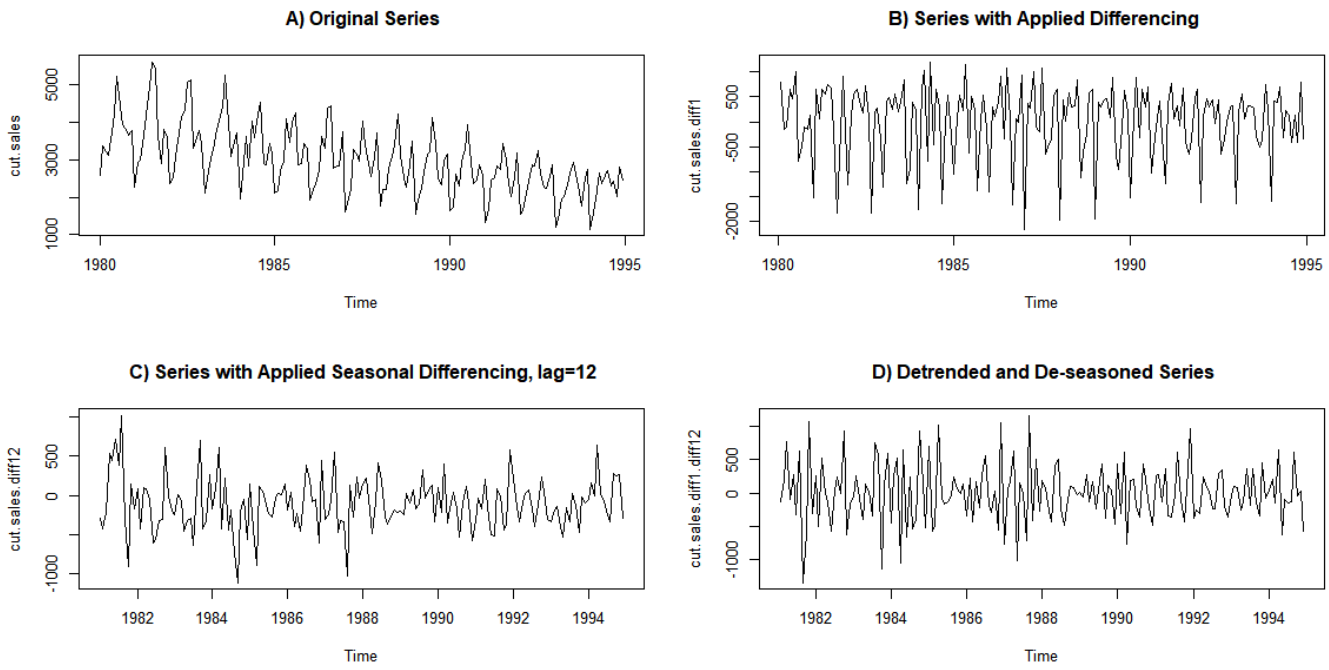

Figure 1: Plots A), B), C) and D) of the Time Series with and without Differencing

(i) Explanation: In **part 2**, we are instructed to plot the data. After we convert it to a time series, we use the **plot()** command in **R** to visualise it. Plot A) (Fig. 1 top-left), is the original time series. We can outright spot an yearly pattern, indicated by the yearly "peeks" in the data. A second thing to spot is that variance might not be constant. We see the data have "wild" jumps at the start, while "calming" down in the end. Lastly, adding to the previous point, the mean as well is most certainly not constant. If we only difference once, plot B), we largely remove the trend, but we are left with the seasonal pattern. Contrast to that, if we only difference, at $lag = 12$, we have a mostly "settled" time series, but there does seem to be an evident trend of decreasing variance with time. Finally, we apply both a first difference and a seasonal one, at $lag = 12$. The resulting plot is D), which seems to have both a constant mean and variance over time. Hence, in D), we removed both the trend and seasonality.
(ii) R Code:

```
par(mfrow=c(2,2))# 2 by 2 setup for plots
cut.sales <- ts(cut.sales, start=c(1980, 1), end = c(1994,12), frequency=12) #Convert into Tim
plot(cut.sales, main=" A) Original Series") #Plot data
cut.sales.diff1 <- diff(cut.sales)#First difference to remove trend
plot(cut.sales.diff1, main=" B) Series with Applied Differencing") #Plot data
cut.sales.diff12 <- diff(cut.sales, lag = 12)#Difference at lag 12 to remove seasonality
plot(cut.sales.diff12, main=" C) Series with Applied Seasonal Differencing, lag=12") #Plot dat
cut.sales.diff1.diff12 <- diff(diff(cut.sales), lag=12) #Detrended and deseasoned data
plot(cut.sales.diff1.diff12, main=" D) Detrended and De-seasoned Series") #Plot data
```

Part 3) Plotting the ACF and PACF

Once we are happy with our detrended and de-seasoned time series, we can start inspecing it more closely in order to try and model it. Using **acf()** and **pacf()** in **R**, we can produce the following plots:

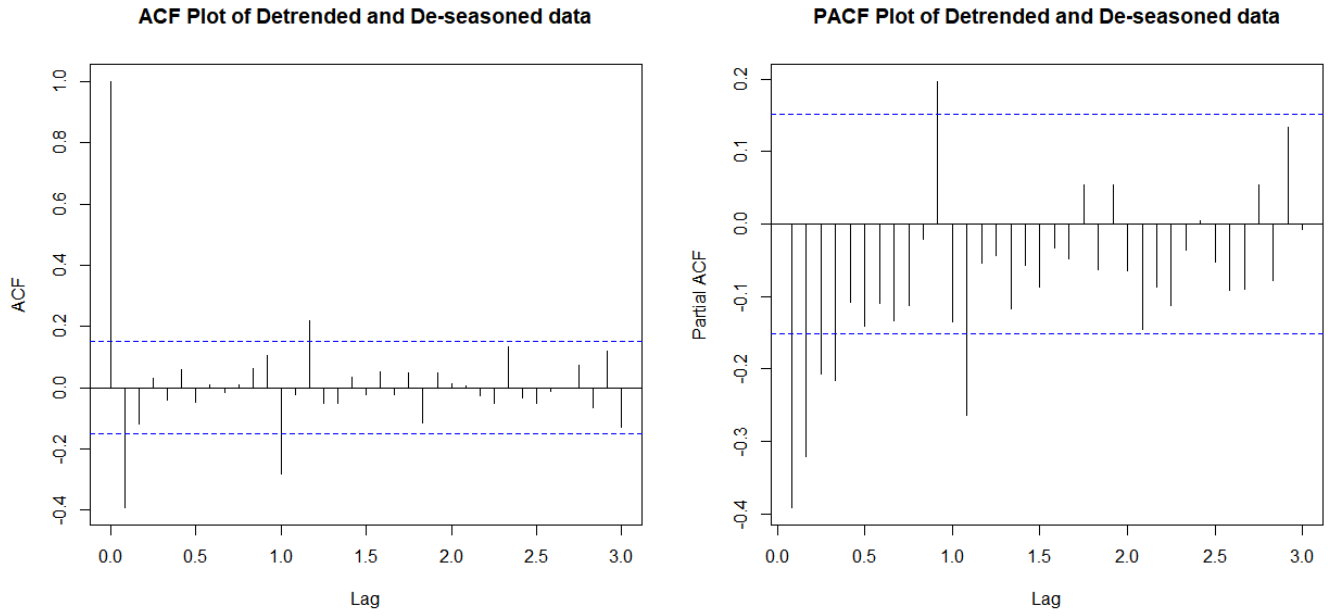

Figure 2: Plots of the ACF and PACF of the Detrended and De-seasoned Data

(i) Explanation: The right plot, of the PACF, does not have a cutoff. We observe a decay, as opposed to the cutoff we spot in the ACF plot at the point 1. This leads us to believe that an adequate model for the data will include an MA(1) component.

(ii) R Code:

```
acf(cut.sales.diff1.diff12, lag.max=36, main="ACF Plot of Detrended and De-seasoned data")
pacf(cut.sales.diff1.diff12, lag.max=36, main="PACF Plot of Detrended and De-seasoned data")
```

Part 4) Specifying a Tentative Model for the Data

(i) Explanation: From the differencing we did in **part 2)**, we know that our $ARIMA\,(p,d,q) \times (P,D,Q)_s$ model will have $d = D = 1$ and $s = 12$. Further, from **part 3)**, we guessed that the model

will have $q = Q = 1$. Hence a viable guess for our tentative model will be with $p = P = 0$, hence $ARIMA\,(0,1,1) \times (0,1,1)_{12}$, which has the following explicit formula:

$$(1-B)\left(1-B^{12}\right)x_t = (1+\theta_1 B)\left(1+\Theta_1 B^{12}\right)\epsilon_t$$

The notes refer to this model on $p.43$ as the "airline model". To fit this model with **R**, we use **arima()**, with respective arguments. It's important to note we use as an argument our original data *cut.sales*, in stead of the already differenced one.
(ii) R Code:

```
fit1 <- arima(cut.sales, order=c(0,1,1), seasonal=list(order=c(0,1,1), period=12))
```

Part 5) Parameter Estimation

(i) Explanation: Once we have proposed our model, the next step is to estimate its coefficients. We already specified the command we use in **R** in **part 4)**, which was **arima()**. We saved the results of the model to a variable called $fit1$, hence if we run **fit1$coef**, we will be presented with the following result:

ma $= \theta_1$	sma1 $= \Theta_1$
-0.9999991	-0.5422741

Table 1: Tabulated values for θ_1 and Θ_1

(ii) R Code:

```
fit1$coef
```

Part 6) A Closer Look at the Residuals

(i) Explanation: When inspecting residuals, we would like for them to be uncorrelated. We can check this hypothesis with the Ljung-Box test, also referred to as the "portmanteau" test. It is called such, because it is one of the first things we do, when we analyse errors, similar to how the portmanteau is the first room we enter in a house. The hypothesis of test is:

$$H_0 : r_\varepsilon(1) = \cdots = r_\varepsilon(k) = 0 \text{ vs } H_A : r_\varepsilon(i) \neq 0, \text{ for any } i = 1, \ldots, k$$

What the test does is calculate the following statistic, which under H_0, is distributed as $\sim \chi^2_{k-p-q-P-Q}$:

$$Q = n(n+2)\sum_{i=1}^{k} \hat{r}_\varepsilon(i)^2/(n-i)$$

Let's illustrate some plots from R to better observe the residuals:

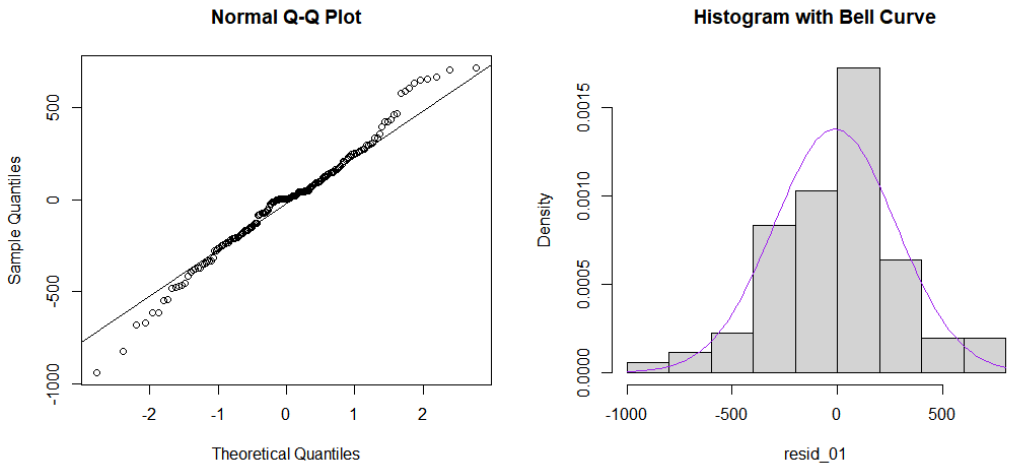

Figure 3: Q-Q plot (left) and Histogram (right) of Residuals From our Model

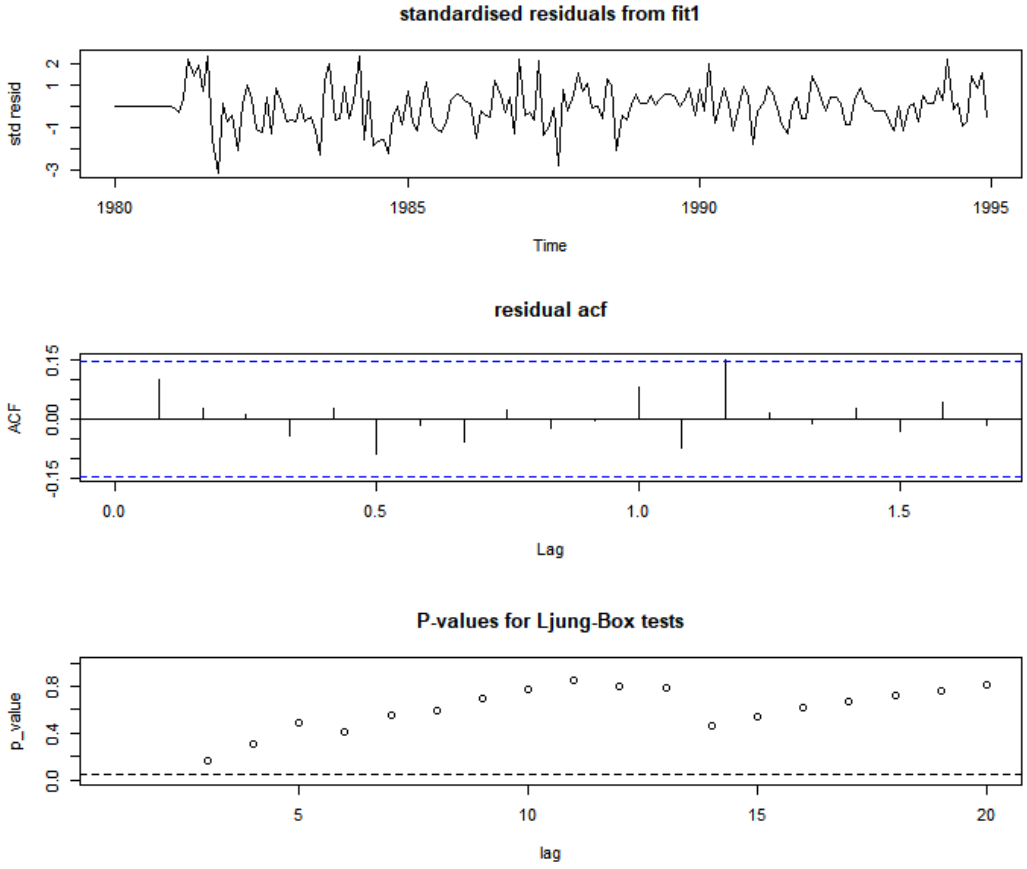

Figure 4: Residual Diagnostic Plots of our Tentative Model

From Fig. 4, the most-bottom plot illustrates the Ljung-Box test. As we can see, the data points are all above the threshold, therefore we cannot reject H_0, i.e we cannot reject the possibility of the correlations all taking values of 0. From the top-most plot, in the same figure, are illustrated the residuals over the time period. The errors are around 0 and we cannot observe any trend in the plot, henceforth this is another indication that our tentative model is a good fit to the wine sales data. The middle plot, checks if the estimations of the acf are within reasonable bounds, i.e $\in \left(\frac{-2}{\sqrt{n}}, \frac{\pm 2}{\sqrt{n}}\right)$, with $n = 180$.

Two other tools we have to visually inspect residuals are illustrated on Fig. 3. To begin with, what we are looking for, is a bell shape in the histogram (right) plot. A bell shape of the residuals will indicate that they follow a normal distribution, which has symmetrical shape around the mean. If true, then most of the errors will be centered around the mean. Using the **curve()** command in **R**, we can superimpose the bell shape we compare with. Although there is a slight deviation, from some residual values, it mostly resembles a bell curve. As for the left plot- that is a QQ-plot. In it, we essentially compare the quantiles of our residuals with the quantiles of a normal distribution. If the residuals are normally distributed, then the points on the x/y plane, will approximately lie along the superimposed straight line.

We can finalise our test, with the **shapiro.test()** command in **R**. It returns a p-value of 0.04849, which is significant at 5%. The null hypothesis of the Shapiro test is that the residuals are from a normal distribution. The significant value we have, means that there is sufficient evidence to reject the null, concluding that the errors are not from a normal distribution which contradicts our findings, from visually inspecting the plots.

(ii) R Code:

```
source("tsdiags.R")
tsdiags(fit1) #Produces the diagnostic plots we inspect
resid_01 <- resid(fit1)
hist(resid_01, freq=FALSE, main="Histogram with Bell Curve") #creates histogram of the residua
curve(dnorm(x, mean=mean(resid_01), sd=sd(resid_01)), add=TRUE, col="purple") #superimpose bel
qqnorm(resid_01); qqline(resid_01) #Produces QQ-plot and a qqline
shapiro.test(resid_01)#p-value = 0.04849
```

Part 7) Log Transforming our Tentative Model

(i) Explanation: We are suggested to log transform the data, and recreate the same model as we had before. To do just that, we can apply the function **arima()** in **R**, with the first argument being **log(cut.sales)**, keeping our previous arguments as before. It is important to note that logarithm is a linear function and hence any operations we do to the logged data, we can recreate to the original data as well. There are many reasons why it is a good idea to log the data, with the most important one being that it may remove some trends in the data. By plotting it, Fig. 5, we can draw some subjective conclusions in that respect, mainly that the variance seems to be constant with time, but there still is a downward sloping mean, which means the variance might be constant, but the mean is not. Therefore it will be required to again apply differencing to remove this trend in the mean. An objective observation we can make- is that the Shapiro test now returns an insignificant value of $p = 0.3338$. Hence, there is insufficient evidence to reject H_0, which states the errors are normally distributed, at the 5% or even 10% SL.

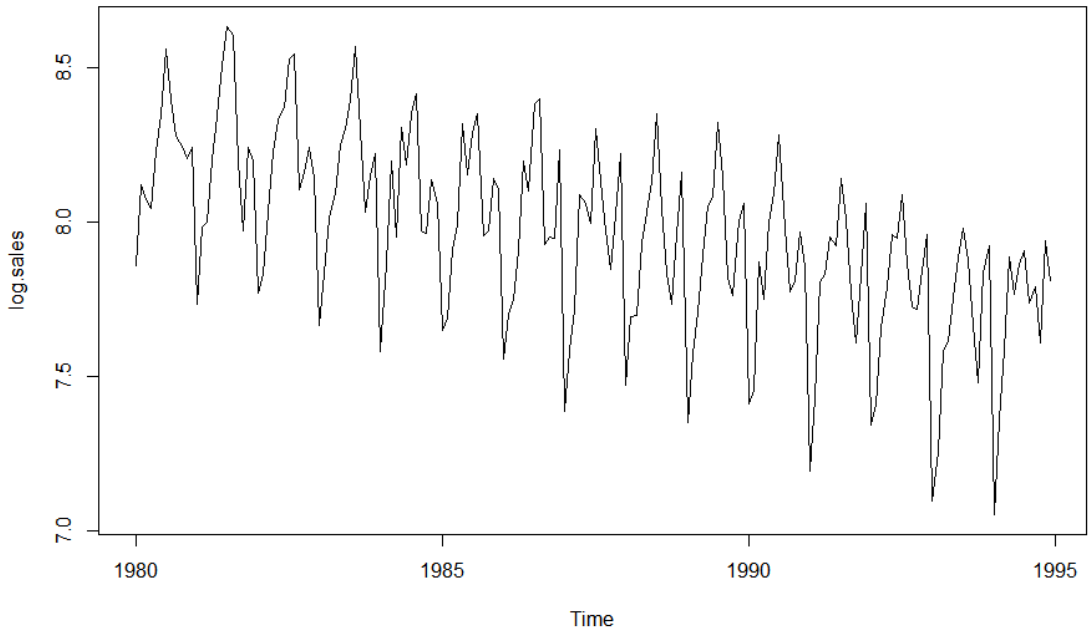

Figure 5: Logged time series

(ii) R Code:

```
log.sales <- ts(log(cut.sales), start=c(1980, 1), end = c(1994,12), frequency=12) #Convert int
plot(log.sales)
fit1.logged <- arima(log(cut.sales), order=c(0,1,1), seasonal=list(order=c(0,1,1), period=12))
resid_02 <- resid(fit1.logged)
shapiro.test(resid_02)#p-value = 0.3338
```

Part 8) Choosing a Model

(i) Explanation: Till now we have been using a "tentative" model, but now we're asked to proceed with a choice of the actual mode. For the purpose of this data, we will stick with our above model, i.e $ARIMA\,(0,1,1) \times (0,1,1)_{12}$, but with logged data. Further, we are to check the significance of our parameters. For a parameter to be significant, it must be greater in magnitude than 1.96. To check this we will create a vector of the coefficient values divided by their respective standard error. The resulting vector is: $(-38.722062, -8.687704)$. Both values are greater in magnitude than 1.96, therefore both are significant and cannot be excluded from the model, at the 5% SL. Consequently, the model cannot be reduced any further.
(ii) R Code:

```
fit1.logged <- arima(log(cut.sales), order=c(0,1,1), seasonal=list(order=c(0,1,1), period=12))
significance.vect <- c(fit1.logged$coef[1]/sqrt(fit1.logged$var.coef[1,1]),
                       fit1.logged$coef[2]/sqrt(fit1.logged$var.coef[2,2])) #coeff divided by
significance.vect#       ma1       sma1
                 #-38.722062  -8.687704
```

Part 9) Forecasting the First Half of 1995

(i) Explanation: The final task is to predict, i.e forecast, the monthly Australian sales of fortified wine, for the first seven months, for the year of 1995. We can do this with the **predict()** command in **R**, instructing it to use our chosen model, along with the number of data points we want to predict. We can further construct 95% prediction intervals for the predicted values, by using the standard error multiplied by 1.96, which is the value of the inverse normal CDF at that point. Hence, we can construct the following table and plot[1]:

Entry	Time (month)	Predicted Value	Prediction Interval, 95%	Actual Value	Act. Val. In Pred. Int	Difference	DEPAV
#181	1995.01	1179.377	(983.0017, 1414.982)	1153	TRUE	26.37674	2.287662%
#182	1995.02	1439.391	(1199.7215, 1726.939)	1482	TRUE	-42.60915	2.875112%
#183	1995.03	1877.313	(1564.7266, 2252.345)	1818	TRUE	59.31319	3.262552%
#184	1995.04	2174.411	(1812.3554, 2608.795)	2262	TRUE	-87.58897	3.872191%
#185	1995.05	2314.184	(1928.8554, 2776.491)	2612	TRUE	-297.81561	11.401823%
#186	1995.06	2467.376	(2056.5394, 2960.285)	2967	FALSE	-499.62418	16.839373%
#187	1995.07	2784.097	(2320.5240, 3340.278)	3179	TRUE	-394.90318	12.422245%

Table 2: Tabulated predicted values, along with their respective prediction interval and corresponding actual value.

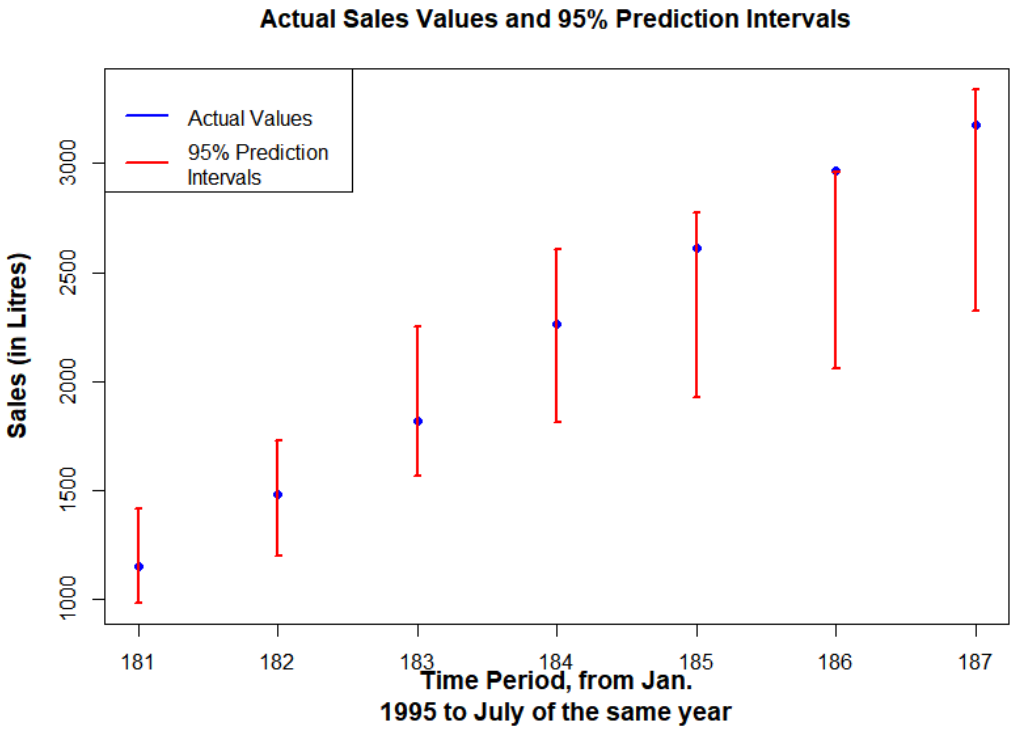

Figure 6: Actual Values and 95% prediction intervals

From the tabulated results, we can observe that six of the seven actual entries fall within our prediction interval, which is a great outcome. Another thing of importance is that the last three values have a higher difference, compared to the initial four predictions. That said, those differences did not exceed 20% of the actual sale's value. Fig. 6 helps us examine how far were the actual values

[1]DEPAV stands for Difference, Expressed as Percentage of Actual Value

from our estimates by visualising it on the x/y plane. Even though the sixth data point, corresponding to June, is outside of our 95% Prediction Interval, it is on the very edge of it.
(ii) R Code:

```
pred <- predict(fit1.logged, n.ahead=7)
predicted.values <- exp(pred$pred)
actual.values <- sales[181:187]# create a vector of actual values
difference <- predicted.values - actual.values
difference.as.ratio.change <- abs((actual.values - predicted.values) / actual.values) * 100
pred.intervals.vector.lower <- c(exp(pred$pred-1.96*pred$se))# create a vector of lower and up
pred.intervals.vector.higher <- c(exp(pred$pred+1.96*pred$se)); par(mfrow=c(1,1))
entry_vector <- c(181:187)# create a vector of indices to use as the x-axis
plot(entry_vector, actual.values, type="n", xlab="Time Period, from Jan.
     1995 to July of the same year", ylab="Sales (in Litres)",
     ylim=c(min(pred.intervals.vector.lower, actual.values),
            max(pred.intervals.vector.higher, actual.values)),
     main="Actual Sales Values and 95% Prediction Intervals",
     cex.lab=1.2, font.lab=2)# Set up the plot
points(entry_vector, actual.values, pch=19, col="blue")# Add points for the actual values
segments(entry_vector, pred.intervals.vector.lower, entry_vector,
         pred.intervals.vector.higher, lwd=2, col="red")# Add vertical lines for the predictio
legend("topleft", legend=c("Actual Values", "95% Prediction
                           Intervals"), col=c("blue", "red"), lty=1, lwd=2)# Add a legend
segments(entry_vector-0.025, pred.intervals.vector.lower, entry_vector+0.025,
         pred.intervals.vector.lower, lwd=2, col="red") # Add line segments at the bottom of t
segments(entry_vector-0.025, pred.intervals.vector.higher, entry_vector+0.025,
         pred.intervals.vector.higher, lwd=2, col="red") # Add line segments at the top of the
```

Conclusion

In this report, we analysed the sales of monthly Australian fortified wine in between Jan. 1980 till Dec. 1994. At first we observed the original data to exhibit trends as well as seasonality. Hence we differenced the data and inspected the ACF and PACF to suggest an adequate ARIMA model for the data. We used a number of tools to analyse if the residuals are independent and if they follow a normal distribution. Subsequently, we compared the log-transformed data to the original and the log data had seemingly a constant variance, as opposed to the original data, which had a time-dependent variance. Using this new model, we predicted seven new entries for the year of 1995, January till July. Our forecasts fell close to the actual values and six out of the seven data points were contained in the prediction interval, with the single value being just outside of the 95% prediction interval. This suggests, that we can conclude that this model does a good job at predicting sales of fortified Australian wine.