MATHEMATICAL STATISTICS

Time Series Analysis Report

Contents

LIST OF FIGURES

LIST OF TABLES

Maths Assignment Question 1

Question 1.1

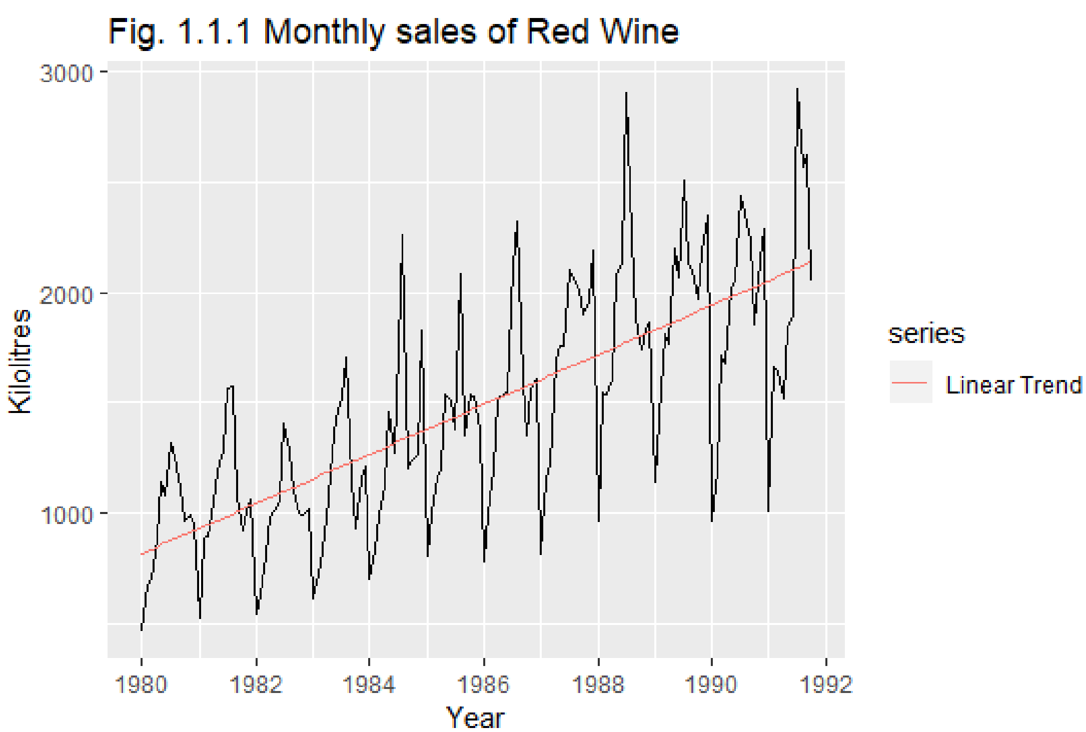
Loading the data from Excel into R:

```
wine.data <- readxl::read_excel("C:/Users/Imy Hull/Desktop/Wine_Data.xlsx")

wine_ts <- ts(wine.data['Sales'],frequency=12,start=c(1980,1))
wine<-wine_ts
```

Basic plots of the data are contained in Figures 1.1.1 to 1.1.5.

Note: It is assumed the monthly sales represent domestic sales of wine rather than exports as it is not clear from the dataset.

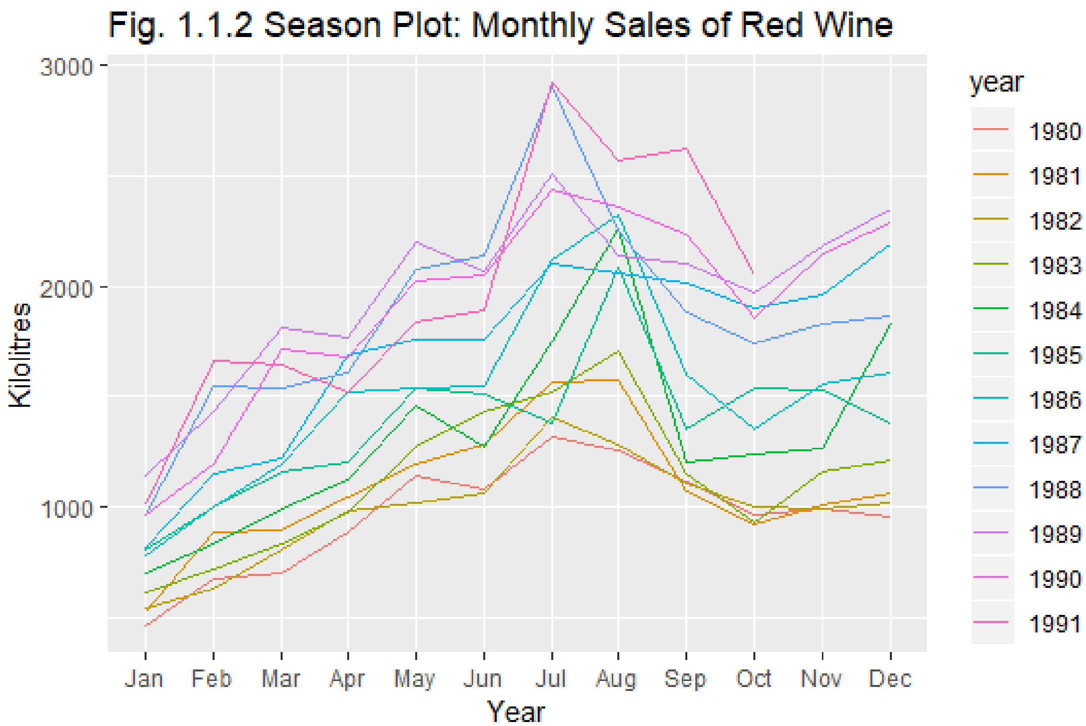
```
wine.line<-tslm(wine~trend)
autoplot(wine) +  autolayer(wine.line$fitted.values, series = "Linear Trend")
+ ggtitle("Fig. 1.1.1 Monthly sales of Red Wine ") +
  xlab("Year") +  ylab("Kilolitres")
```



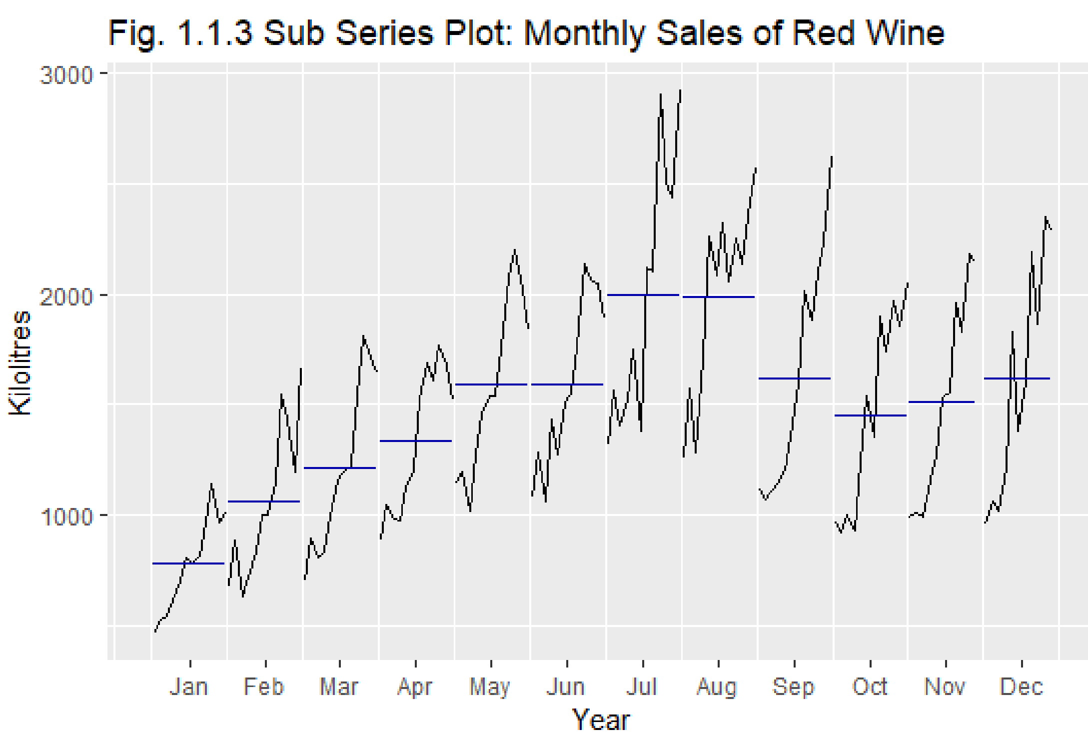

```
wine.line$coefficients

## (Intercept)       trend
##  809.363500    9.348309
```

The time series grows by 112.1796 each year.

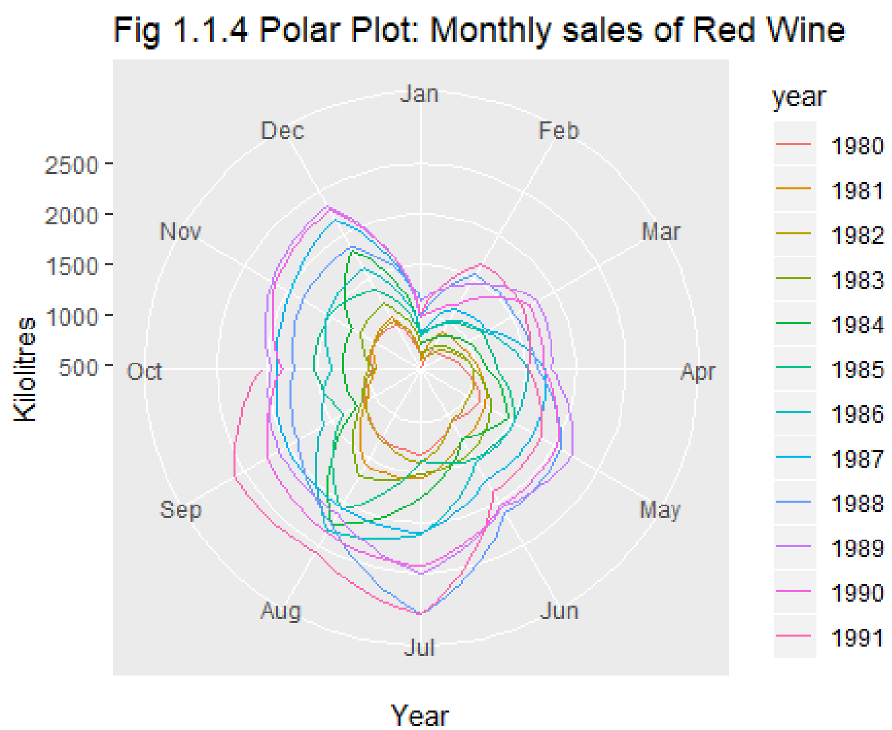
```r
ggseasonplot(wine) +
  ggtitle("Fig. 1.1.2 Season Plot: Monthly Sales of Red Wine ") +
  xlab("Year") +  ylab("Kilolitres")
```




```r
ggsubseriesplot(wine) +
  ggtitle("Fig. 1.1.3 Sub Series Plot: Monthly Sales of Red Wine") +
  xlab("Year") +  ylab("Kilolitres")
```



```r
ggseasonplot(wine,polar = TRUE) +
  ggtitle("Fig 1.1.4 Polar Plot: Monthly sales of Red Wine ") +   xlab("Year"
) +  ylab("Kilolitres")
```




```r
ggAcf(wine)
```

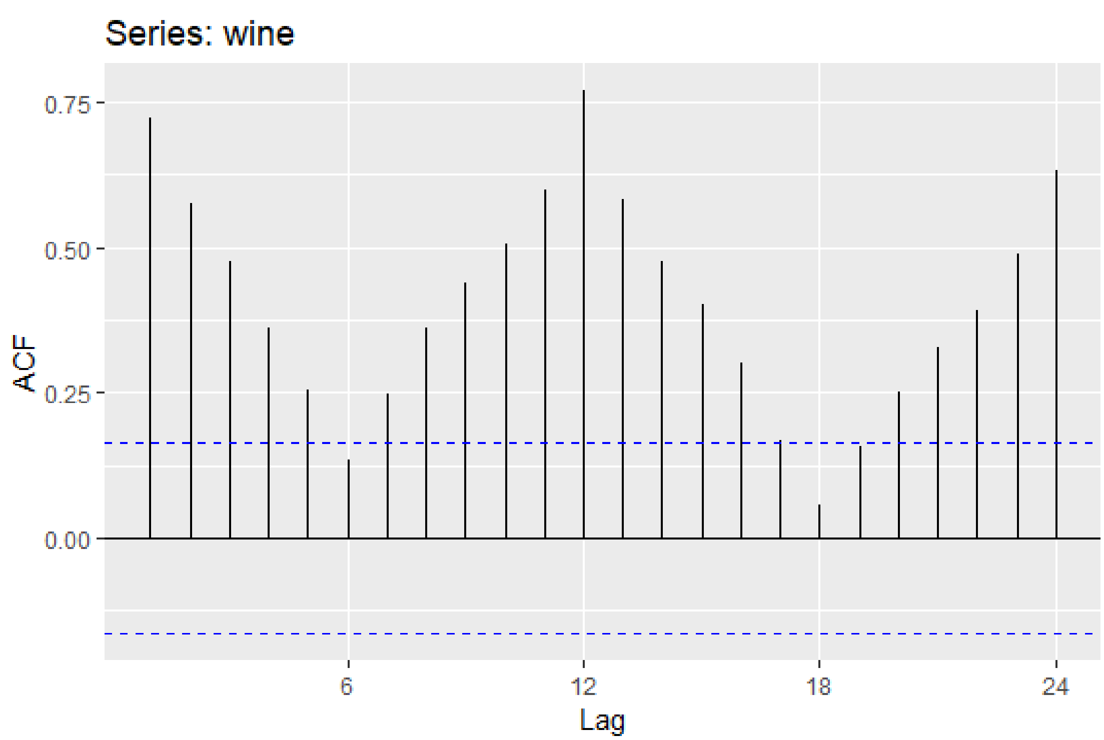


Fig. 1.1.5 ACF Plot for the Wine Data

Observations

The series trends upwards in a linear fashion, and there is monthly seasonality with peaks in July (winter) and lows in January (summer), with some apparent increased variability over the second half of the data so a BoxCox transformation might be needed when forecasting. The lagplots show the strong positive linear relationship at lag 12 and the ACF plot clearly shows a seasonal pattern with peaks at lags 12,24 and 36 and dips at lags 6 and 18. There seem to be several outlier values and the series appears multiplicative but this may be due to the presence of these values.

Running the tsoutlier() function identified four values. Table 1.1 compares these to the monthly averages and suggested replacement values from tsoutlier().

```
tsoutliers(wine)

## $index
## [1]  56  60  67 103
##
## $replacements
## [1] 1633.508 1329.113 2017.038 2436.352

outliers <- read.table("C:/Users/Imy Hull/Desktop/outliers.csv", sep = ",", h
eader = TRUE)
outliers

##     Date Kilolitres MonthAv Diff Replace
## 1 Dec-84       1828    1613  215    1329
## 2 Aug-84       2258    1990  268    1633
## 3 Jul-85       1378    1994 -616    2017
## 4 Jul-88       2907    1994  913    2436
```

Table 1.1 Outliers and Suggested Replacements

Outliers can affect forecast accuracy and should be investigated. Reseach showed that European producers were significantly affected by adverse weather reducing world supply in this period, (see Appendix 1), so Australian wine sales are likely to have increased in response to meet both domestic and export demand. It is interesting that from the start of the series, sales peak in the period June to August, perhaps as this is also the Australian winter, so more red wine is drunk domestically, and it is typical to celebrate with "Christmas in July". As the series seems to become increasingly volatile from 1984, this may indicate a longer-term change in the data. Therefore, outliers were left unadjusted for the purposes of this analysis.

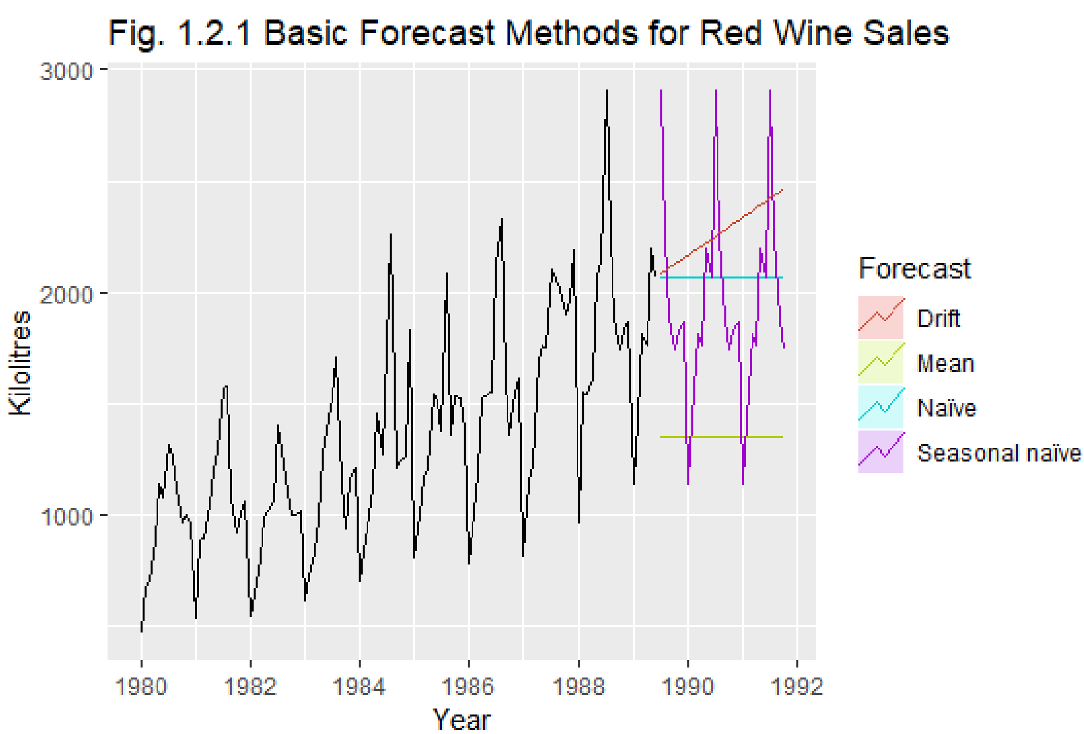
Question 1.2

The dataset was split into a training and test set.

```
wine.train<- window(wine,start=c(1980,1),end=c(1989,6))
wine.test<-window(wine,start = c(1989,7), end = c(1991,10))
h<-length(wine.test)
```

Basic forecasts using the mean, naïve, seasonal naïve and drift methods were calculated and plotted in Fig. 1.2.1.

```
wine.mean <- meanf(wine.train,h=h)
wine.naive <- naive(wine.train,h=h)
wine.snaive <- snaive(wine.train,h=h)
wine.drift<-rwf(wine.train, h =h, drift = TRUE)

autoplot(wine.train) +
  autolayer(wine.mean, series="Mean", PI=FALSE) +
  autolayer(wine.naive, series="Naïve", PI=FALSE) +
  autolayer(wine.snaive, series="Seasonal naïve", PI=FALSE) +
  autolayer(wine.drift, series = "Drift ", PI= FALSE) +
  xlab("Year") + ylab("Kilolitres") +
  ggtitle("Fig. 1.2.1 Basic Forecast Methods for Red Wine Sales") +
  guides (colour=guide_legend(title="Forecast"))
```



Accuracy measures are in Table 1.2

```
wine.accmean<-round(accuracy(wine.mean,wine.test),2)
wine.accnaive<-round(accuracy(wine.naive, wine.test),2)
wine.accsnaive<-round(accuracy(wine.snaive, wine.test),2)
wine.accdrift<-round(accuracy(wine.drift, wine.test),2)
d.table <- rbind(wine.accmean,wine.accnaive,wine.accsnaive,wine.accdrift)
row.names(d.table)<-c("Mean Train", "Mean Test",
                      "Naive Train", "Naive Test",
                      "Seas. Naive Train", "Seas. Naive Test",
                      "Drift Train", "Drift Test")
d.table<-as.data.frame(d.table)
```

```r
kable(d.table, caption = "Table 1.2 Accuracy Measures for the Basic Forecasti
ng Methods", digits = 4)
```

Table 1.2 Accuracy Measures for the Basic Forecasting Methods

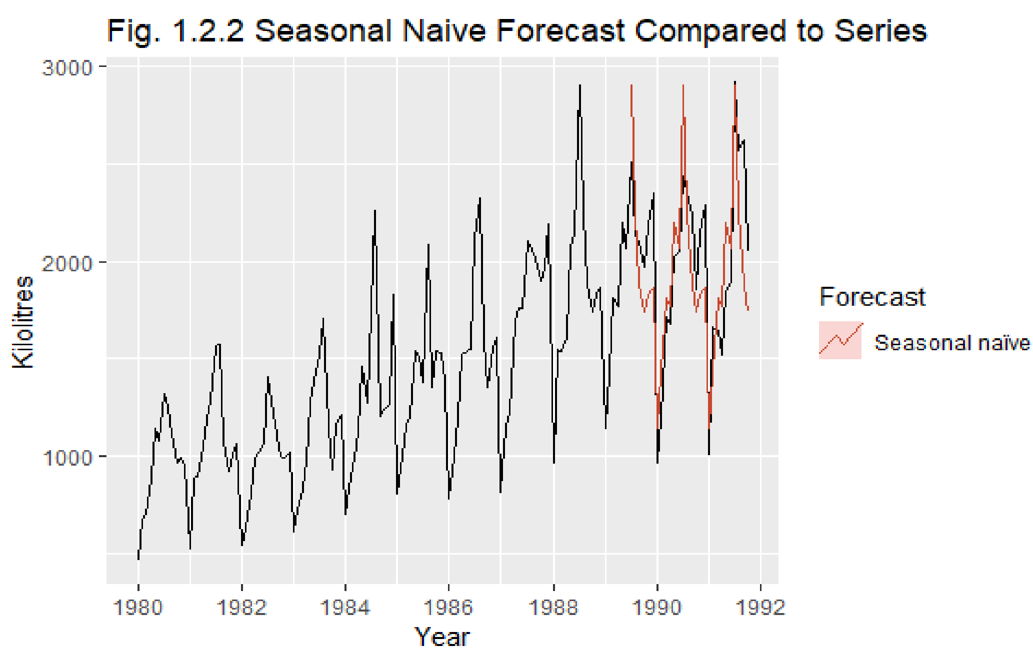
	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Mean Train	0.00	466.13	383.41	-13.63	33.27	1.99	0.69	NA
Mean Test	647.03	792.49	709.50	27.59	33.72	3.69	0.41	1.39
Naive Train	14.19	350.70	250.26	-2.64	21.07	1.30	-0.23	NA
Naive Test	-69.79	462.88	353.00	-10.85	22.10	1.84	0.41	1.05
Seas. Naive Train	110.97	249.16	192.23	6.84	13.14	1.00	0.29	NA
Seas. Naive Test	49.36	295.93	250.36	0.86	12.58	1.30	0.41	0.56
Drift Train	0.00	350.41	246.59	-3.82	20.83	1.28	-0.23	NA
Drift Test	-275.48	535.01	402.54	-21.72	26.66	2.09	0.39	1.29

Table 1.2 Accuracy for the Basic Forecasting Measures

Observations

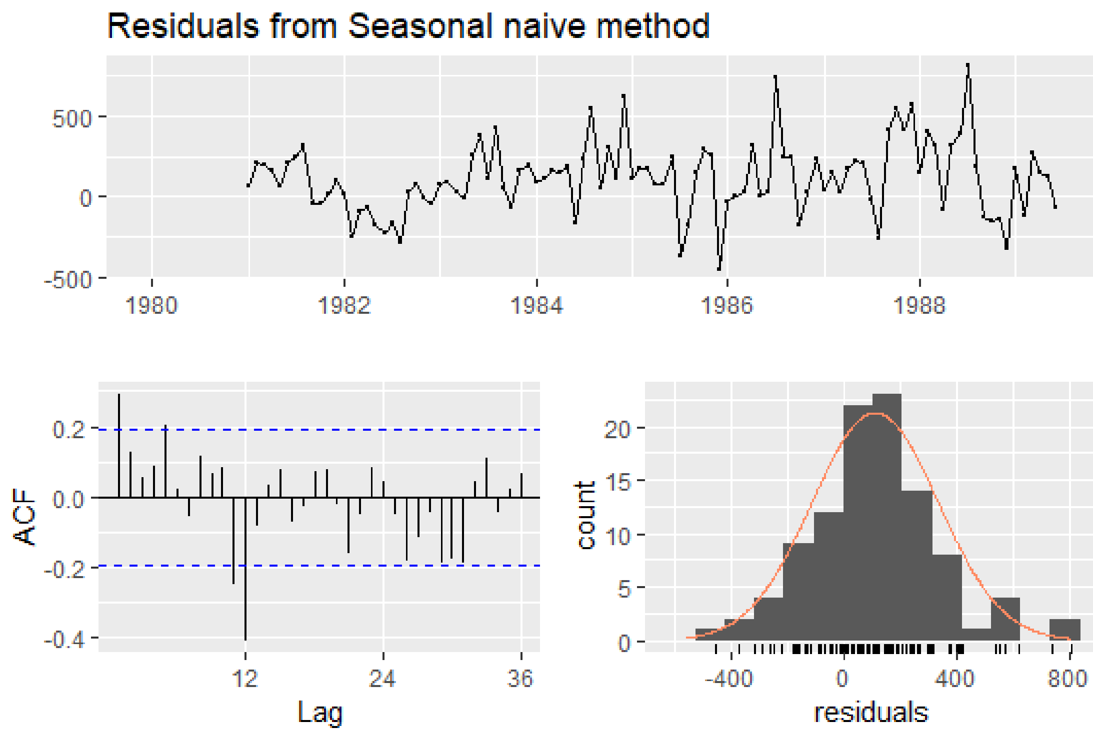
The Seasonal Naive method has the lowest forecast error. Fig. 1.2.2 compares the forecast against the test data which shows that although it picks up the large spike in the previous season, it overforecasts by repeating it over the test period. Also, this method captures seasonality but doesn't account for any trend in the data.

```r
autoplot(wine) +
  autolayer(wine.snaive, series="Seasonal naïve", PI=FALSE) +
  xlab("Year") + ylab("Kilolitres") +
  ggtitle("Fig. 1.2.2 Seasonal Naive Forecast Compared to Series") +
  guides (colour=guide_legend(title="Forecast"))
```



Residuals

```
checkresiduals(wine.snaive)
```

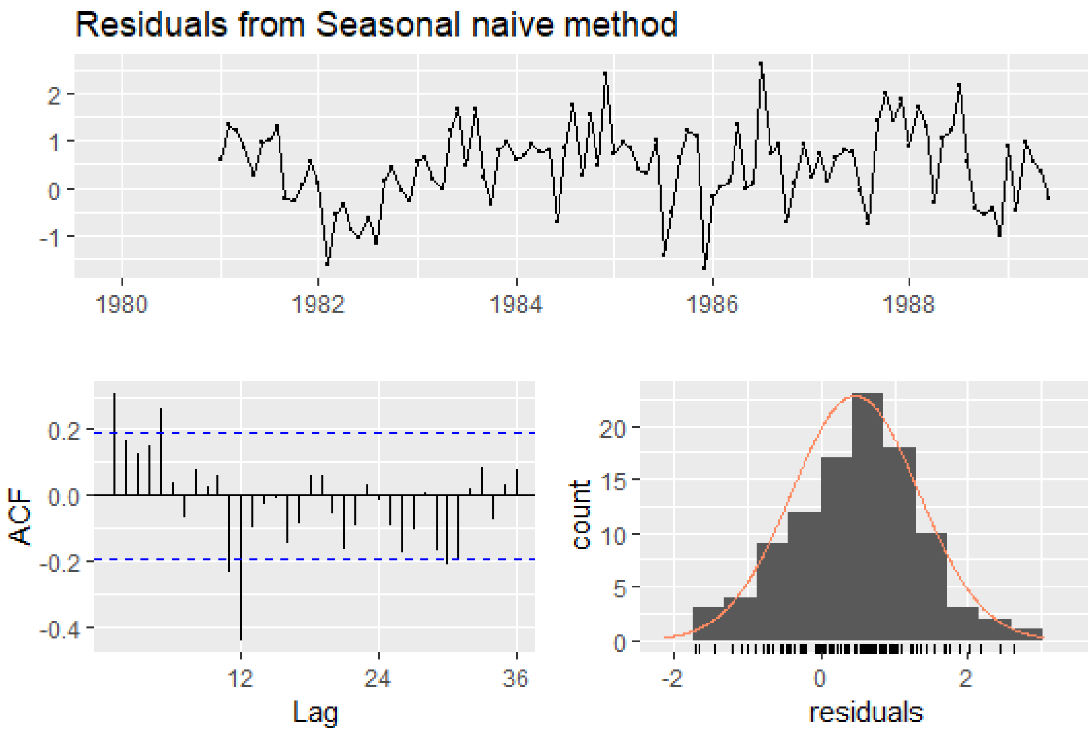



```
##
##  Ljung-Box test
##
## data:  Residuals from Seasonal naive method
## Q* = 55.025, df = 23, p-value = 0.0001934
##
## Model df: 0.   Total lags used: 23
```

There are four points outside the 95% significance bounds on the ACF, with one large spike and a p-value less than 0.05 on the Ljung-Box test. This indicates autocorrelation so information remains in the data which could be used in forecasting. The histogram has a positive skew due to the outliers so is not normally distributed. The residuals plot oscillates around a constant near-zero mean with some apparent increased variance over the latter half of the series. In conclusion, this data does not seem to resemble white noise and it is likely that the model can be improved.

Note: A Box Cox transformation was used to stabilise the variance which increased normality of the data but did not improve the other residual (code used below).

```
wine.log<-BoxCox(wine.train, lambda = "auto")
wine.fc1<-snaive(wine.log, h = h, biasadj = TRUE)
checkresiduals(wine.fc1)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Seasonal naive method
## Q* = 66.068, df = 23, p-value = 4.845e-06
##
## Model df: 0.   Total lags used: 23
```

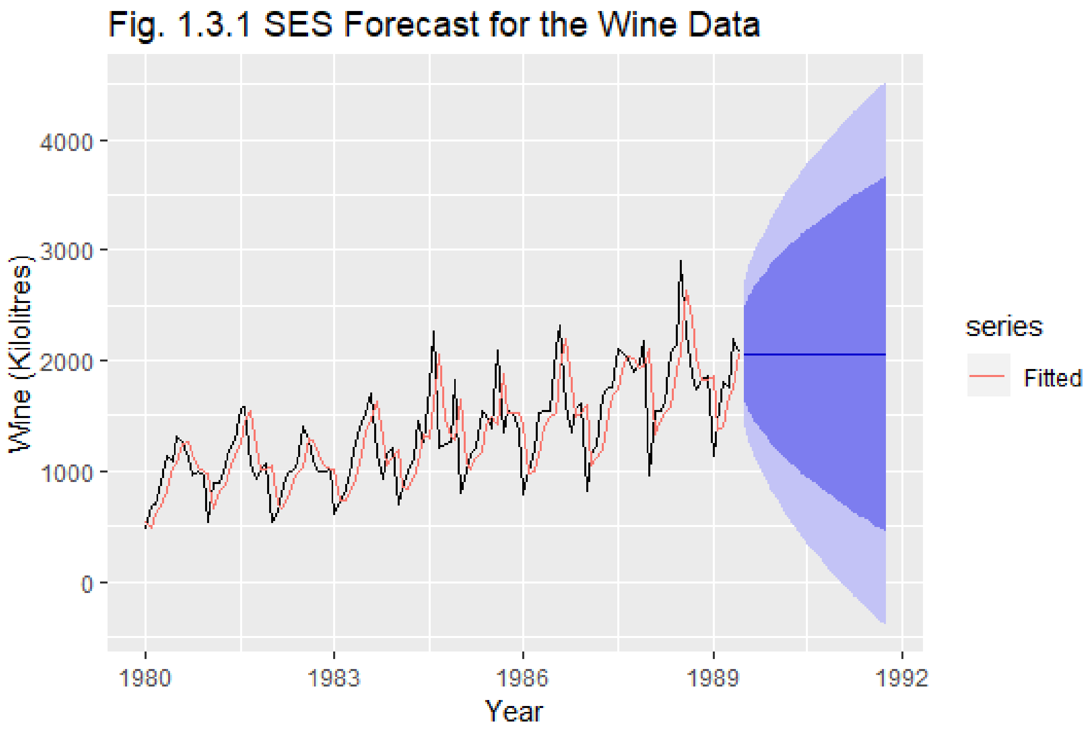
Question 1.3

Exponential Smoothing forecast methods use weighted averages of historical observations with weights reducing exponentially for older observations.

Simple Exponential Smoothing (SES)

SES is the estimated mean of all the future possible outcomes. Figure 1.3.1 shows the one-step forecast on the training data against the original series and the smoothing parameters are below the plot.

```
wine.ses <- ses(wine.train, h=h)
autoplot(wine.ses) +
  autolayer(fitted(wine.ses), series="Fitted") +
  ggtitle("Fig. 1.3.1 SES Forecast for the Wine Data") +
  ylab("Wine (Kilolitres)") + xlab("Year")
```




```
wine.ses$model$fit$par

## [1]   0.6854373 541.3848169
```

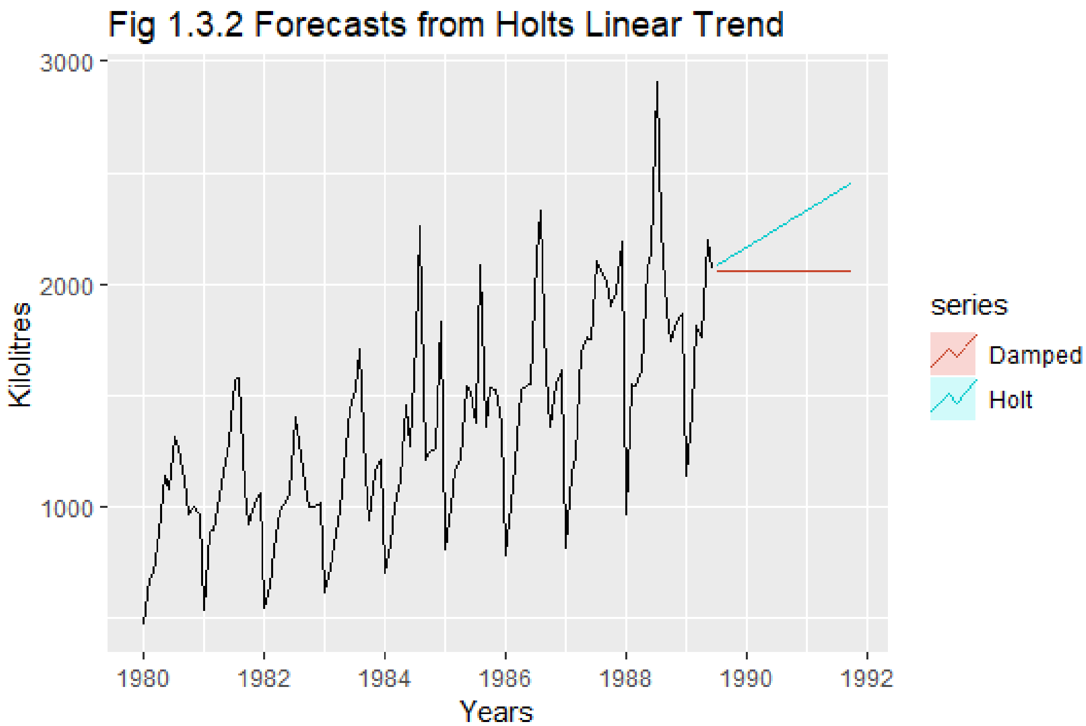
The relatively large smoothing parameter alpha of 0.69 allows greater weight to be attached to the more recent observations, but this method is not ideal for forecasting since it does not account for any trend or seasonality in the data.

Holt Linear Trend and Damped Trend

Holt Linear Methods allow for trend in the data with an additional smoothing parameter and state parameter. The smoothing parameter controls how quickly the slope changes; a small value indicates the trend is close to linear.

This method assumes the trend continues at the same slope for all future forecasts, which can lead to overforecasting. An alternative 'damped' method gives short term trended forecasts and constant longer-term forecasts by adding a damping parameter Phi. Fig. 1.3.2 compares the forecasts of these two methods.

```
wine.holt<-holt(wine.train, h = h)
wine.holtd<-holt(wine.train, h = h, damped = TRUE)
autoplot(wine.train) +
  autolayer(wine.holt, PI = FALSE, series = "Holt") +
  autolayer(wine.holtd, PI = FALSE, series = "Damped") +
  xlab("Years") + ylab("Kilolitres") +
  ggtitle("Fig 1.3.2 Forecasts from Holts Linear Trend")
```



The parameters for the method are as follows:

```
wine.holt$model$fit$par

## [1] 6.811734e-01 1.000030e-04 5.072706e+02 1.383967e+01

wine.holtd$model$fit$par

## [1] 6.754338e-01 1.000003e-04 8.774787e-01 6.301334e+02 7.039480e+01
```

The Holt Linear Trend has captured the upward trend but would not be suitable for longer forecast periods, since the series is likely to level off. The dampening adjustment addresses this and appears to return a similar result to SES, but the method does not take account of the clear seasonality in the data.

Holt Winters Method

This method allows for both seasonality and trend. The additive method is appropriate if seasonal fluctuations about the trend do not vary with the level. A multiplicative method is preferable where seasonal fluctuation about the trend changes proportionately to the level. Both methods can be damped, and each of these was used with the series.

```
wine.hwa<-hw(wine.train, h=h,seasonal = "additive")
wine.hwda<-hw(wine.train, h=h, seasonal= "additive", damped = TRUE)
wine.hwm<-hw(wine.train, h = h, seasonal = "multiplicative")
wine.hwdm<-hw(wine.train, h = h, seasonal = "multiplicative", damped = TRUE)
```

Table 1.3 shows the smoothing parameters for each of the models with and extra parameter phi(0.98) for the damped methods.

```
xa<-round(wine.hwa$model$par[1:3],3)
xb<-round(wine.hwda$model$par[1:4],3)
xc<-round(wine.hwm$model$par[1:3],3)
xd<-round(wine.hwdm$model$par[1:4],3)
e.table <- rbind(xa,xb,xc,xd)

## Warning in rbind(xa, xb, xc, xd): number of columns of result is not a
## multiple of vector length (arg 1)

row.names(e.table)<-c("HWA","HW(D)A","HWM", "HW(D)M")
e.table<-as.data.frame(e.table)
kable(e.table, caption = "Table 1.3 Parameters for HW Models")
```

Table 1.3 Parameters for HW Models

	alpha	beta	gamma	phi
HWA	0.166	0.000	0.001	0.166
HW(D)A	0.185	0.007	0.000	0.980
HWM	0.113	0.000	0.000	0.113
HW(D)M	0.290	0.006	0.000	0.980

A comparison of the accuracy measures from all exponential smoothing methods on the test data is shown in Table 1.4.

```
b1<-accuracy(wine.ses,wine.test)
b2<-accuracy(wine.holt,wine.test)
b3<-accuracy(wine.holtd, wine.test)
b4<-accuracy(wine.hwa,wine.test)
b5<-accuracy(wine.hwda,wine.test)
b6<-accuracy(wine.hwm, wine.test)
b7<-accuracy(wine.hwdm,wine.test)
library("knitr")
a.table <- rbind(b1,b2,b3,b4,b5,b6,b7)
row.names(a.table)<-c("SES Train", "SES Test",
                      "Holt Lin.Train", "Holt Lin.Test",
                      "Holt Lin.(D)Train", "Holt Lin.(D) Test",
                      "HWA Train", "HWA Test",
                      "HW(D)A Train", "HW(D)A Test",
                      "HWM Train", "HWM Test",
                      "HW(D)M Train", "HW(D)M Test")
a.table<-as.data.frame(a.table)
kable(a.table, caption= "Table 1.4 Forecast accuracy of Exponential Smoothing
Methods",digits = 1 )
```

Table 1.4 Forecast accuracy of Exponential Smoothing Methods

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
SES Train	19.5	336.2	247.6	-3.0	20.6	1.3	0.0	NA
SES Test	-65.9	462.3	352.7	-10.6	22.0	1.8	0.4	1.0
Holt Lin.Train	-0.2	335.7	243.1	-4.6	20.4	1.3	0.0	NA

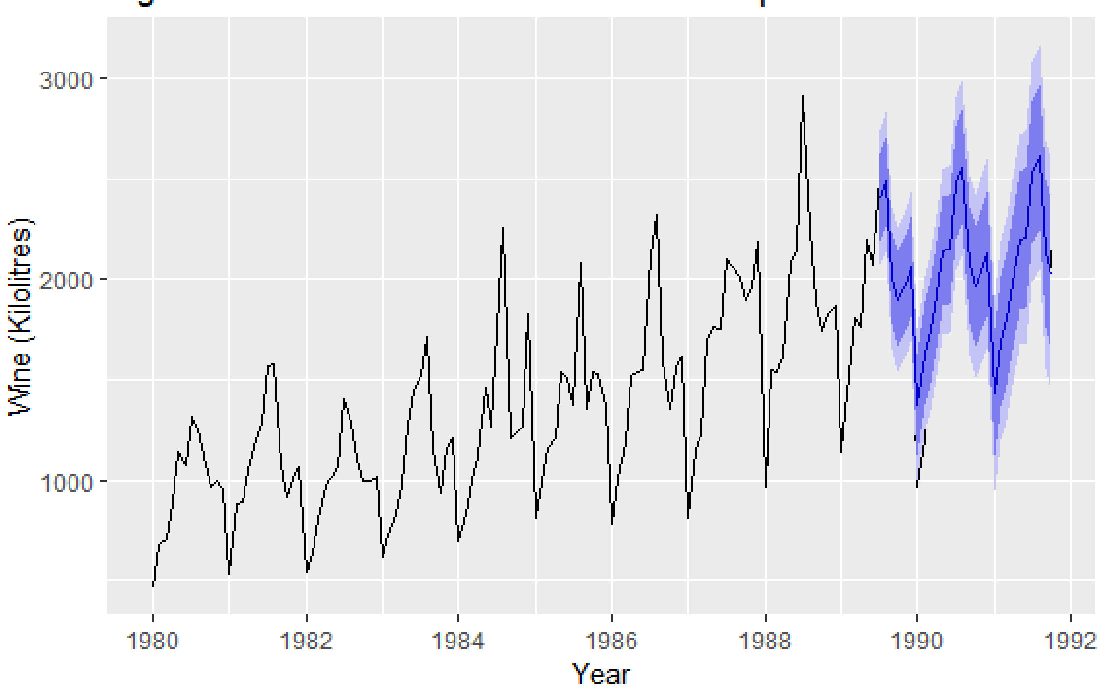
Holt Lin.Test	-272.2	533.0	401.1	-21.5	26.6	2.1	0.4	1.3
Holt Lin.(D)Train	12.0	335.7	246.3	-4.0	20.6	1.3	0.0	NA
Holt Lin.(D) Test	-64.3	462.1	352.6	-10.6	22.0	1.8	0.4	1.0
HWA Train	2.4	157.0	114.9	-0.5	8.7	0.6	0.0	NA
HWA Test	-104.3	269.5	225.2	-8.4	13.4	1.2	0.1	0.6
HW(D)A Train	18.7	159.0	116.3	1.0	8.9	0.6	0.0	NA
HW(D)A Test	-60.1	256.7	209.2	-6.1	12.2	1.1	0.1	0.6
HWM Train	5.5	150.3	102.7	-0.6	7.4	0.5	0.1	NA
HWM Test	-150.9	308.2	238.1	-8.3	12.2	1.2	0.2	0.6
HW(D)M Train	13.9	149.7	102.1	0.1	7.2	0.5	-0.1	NA
HW(D)M Test	-83.8	258.5	215.0	-5.3	11.1	1.1	0.1	0.5

Observations

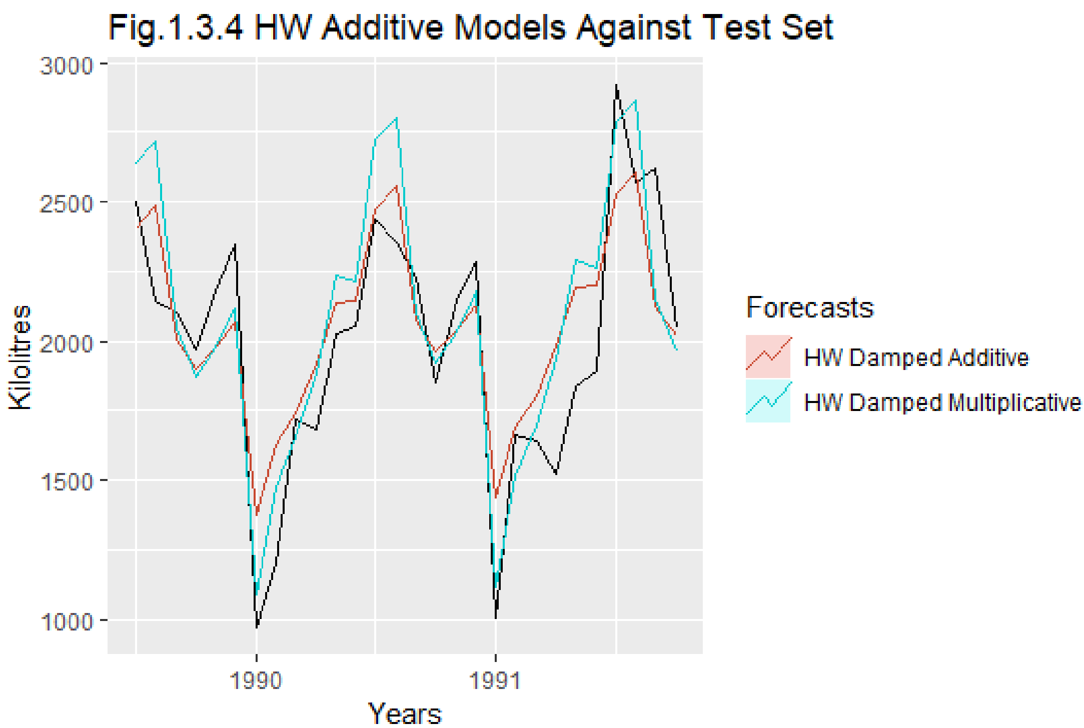
The RMSE of the HW Damped Additive and the Damped Multiplicative methods is quite close with the additive method performing marginally better on the test set. The HW Damped Additive forecast is in Fig. 1.3.3 and Fig. 1.3.4 shows the forecast against the test set for both the preferred and the damped multiplicative methods.

```
autoplot(wine) +
  autolayer(wine.hwda) +
  ggtitle("Fig. 1.3.3 Forecast from Holt Winters Damped Additive Model") +  y
lab("Wine (Kilolitres)") + xlab("Year")
```

Fig. 1.3.3 Forecast from Holt Winters Damped Additive Model

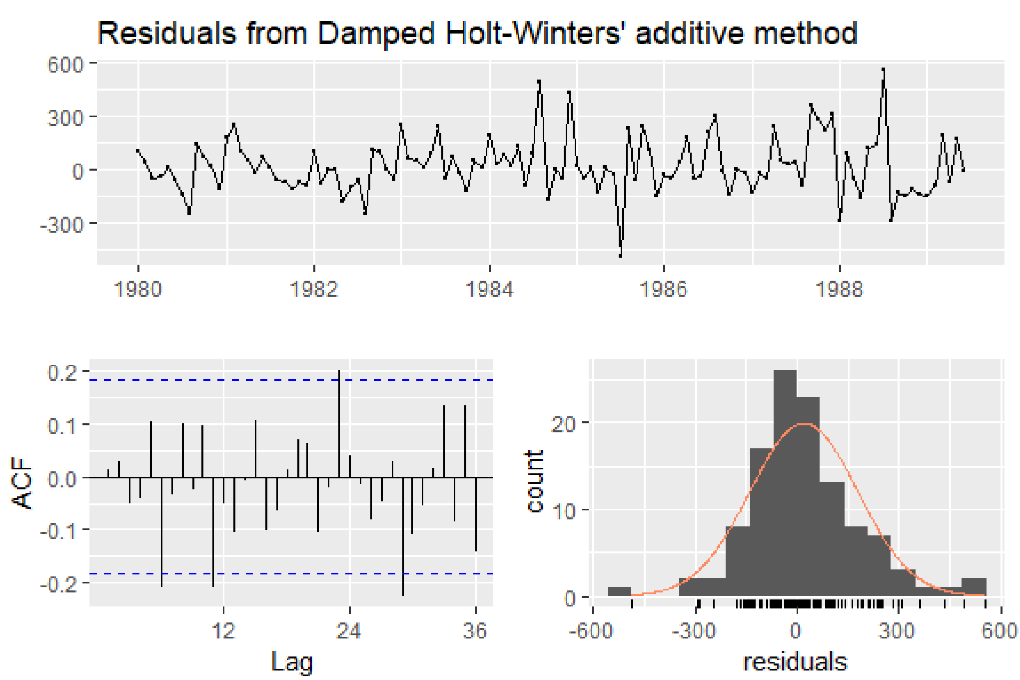


```
autoplot(wine.test) +
  autolayer(wine.hwda, series = "HW Damped Additive", PI = FALSE)+
  autolayer(wine.hwdm, series = "HW Damped Multiplicative", PI = FALSE) +
  xlab("Years") + ylab("Kilolitres") +
  ggtitle("Fig.1.3.4 HW Additive Models Against Test Set") +
  guides(colour = guide_legend(title = "Forecasts"))
```



Residuals

```
checkresiduals(wine.hwda)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from Damped Holt-Winters' additive method
## Q* = 29.489, df = 6, p-value = 4.914e-05
##
## Model df: 17.   Total lags used: 23
```

```
res.hwda<-residuals(wine.hwda)
shapiro.test(res.hwda)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.hwda
## W = 0.96358, p-value = 0.003411
```

There are several lags outside the threshold on the ACF plot and the Ljung-Box test is well below the significance level of 0.05, indicating there is still autocorrelation left in the residuals. The mean appears near-zero and constant but there is still some variability in the data. The series does not appear to be white noise. The histogram appears only roughly normally distributed and the Shapiro-Wilk test indicates the null hypothesis of normality should be rejected. The residuals indicate that the model could still be improved.

In summary, the HW(D) Additive method has better accuracy measures on the test set than the Seasonal Naive method forecasting method and is the preferred method at this point.

Question 1.4

Decompostion forecasting requires seasonality to be additive so a BoxCox transformation was used before using the decomposition.

```
lambda<-BoxCox.lambda(wine.train)
lambda
```

```
## [1] 0.2427797
```

```
wine.log<-BoxCox(wine.train, lambda = "auto")
```

Fig. 1.4.1 shows the components of the decomposed series. The remainder term has some spikes, although generally reasonable, as the scale is small. Altering the s.window did not produce any significant change, so the default was used.

```
wine.trans<-BoxCox(wine.train, lambda = "auto")
wine.decomp<-stl(wine.trans[,1], s.window = 13)
autoplot(wine.decomp)
```

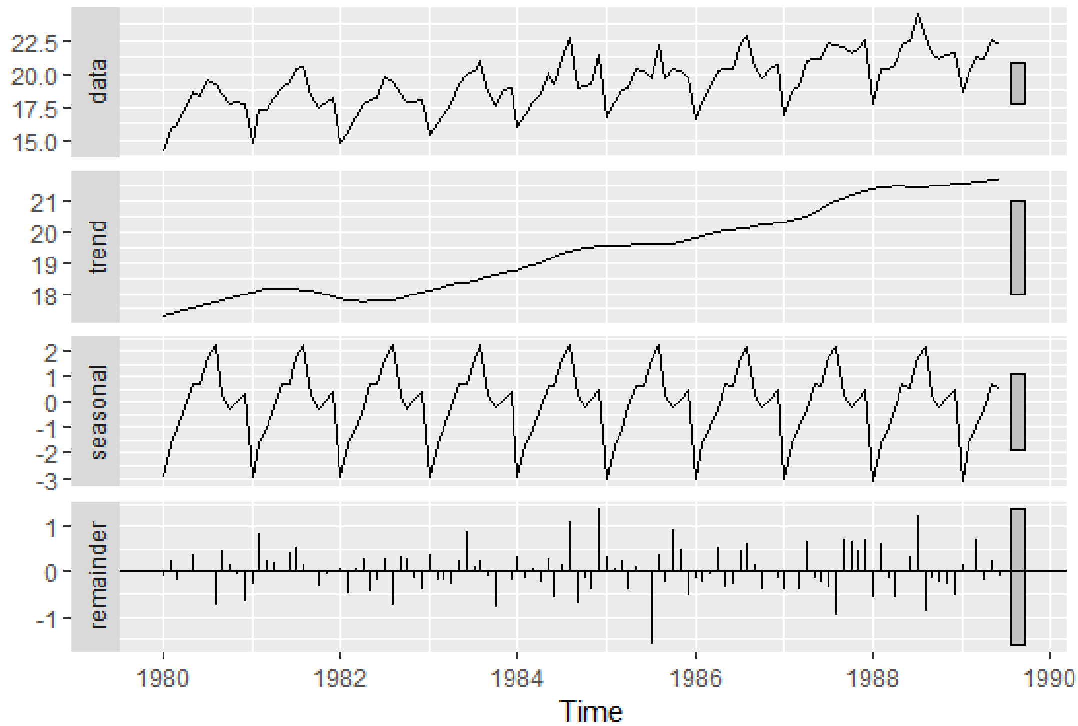


Fig. 1.4.1 Seasonal Decompostion of Wine Data

Forecasting returned an ETS(MAN) model with multiplicative error and additive trend.

```
wine.stlf<-stlf(wine.train, lambda = "auto", s.window = 13, method = "ets", h
= h)
wine.stlf$model

## ETS(M,A,N)
##
## Call:
##  ets(y = x, model = etsmodel, allow.multiplicative.trend = allow.multiplic
ative.trend)
##
##   Smoothing parameters:
##     alpha = 0.2049
##     beta  = 1e-04
##
##   Initial states:
##     l = 17.3001
##     b = 0.0387
##
##   sigma:  0.0282
##
##      AIC     AICc      BIC
## 407.1617 407.7173 420.8427
```

Plotting the forecast:

```
autoplot(wine.stlf) + xlab("Year")+ ylab("Kilolitres")
```

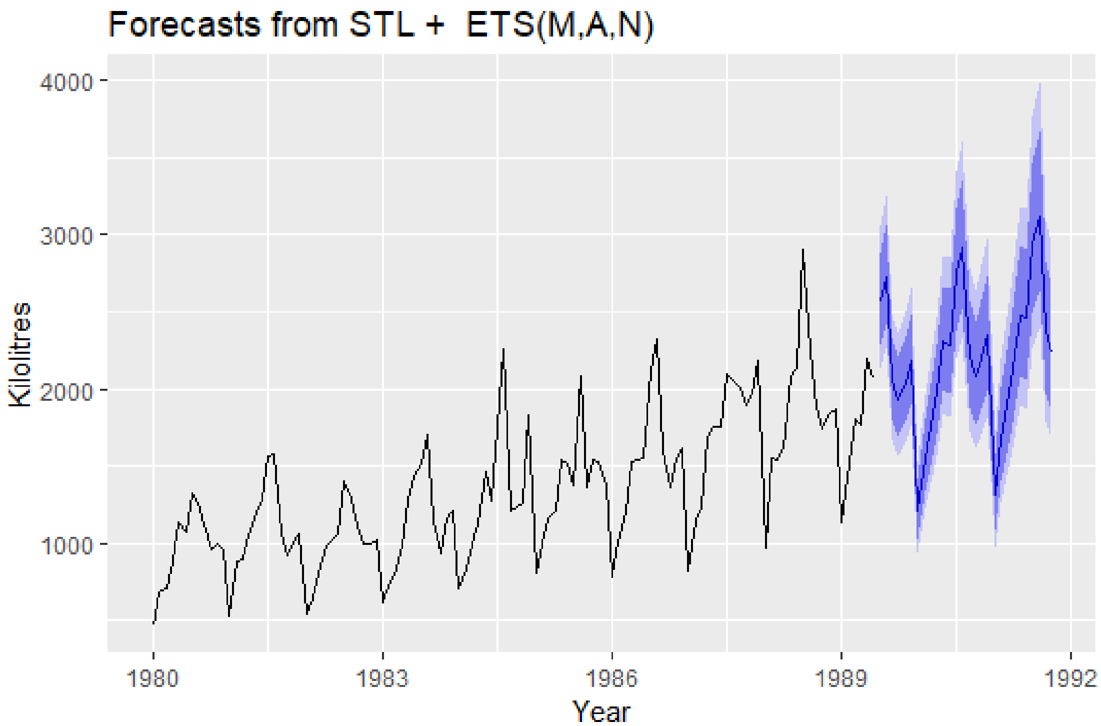


Fig. 1.4.2 STL + MAN Wine Forecast

The HW(D) still seems to perform better over the test period which is supported by the accuracy measures in Table 4.1.

```
acc.hwda<-round(accuracy(wine.hwda,wine.test),1)
acc.stlf<-round(accuracy(wine.stlf,wine.test),1)
library(knitr)
c.table<-rbind(acc.hwda,acc.stlf)
row.names(c.table)<-c("HW(D)A Train",
                      "HW(D)A Test",
                      "STL + ETS Train",
                      "STL + ETS Test")
c.table<-as.data.frame(c.table, caption = "Table 1.5 Accuracy Comparisons")
kable(c.table)
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
HW(D)A Train	18.7	159.0	116.3	1.0	8.9	0.6	0.0	NA
HW(D)A Test	-60.1	256.7	209.2	-6.1	12.2	1.1	0.1	0.6
STL + ETS Train	5.2	144.1	99.7	-0.5	7.0	0.5	-0.1	NA
STL + ETS Test	-207.7	323.6	254.1	-12.1	14.0	1.3	0.0	0.7

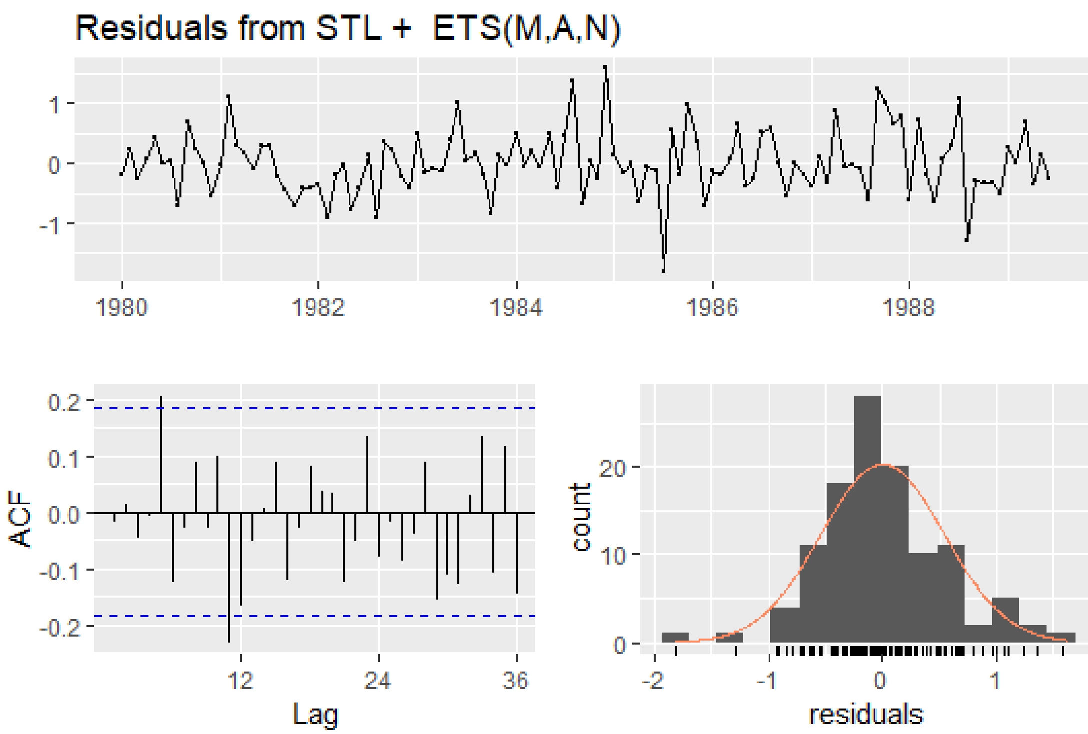
Table 1.5 Accuracy of STL + ETS(MAN)

Observations

Based upon the RMSE accuracy measure, the STL + ETS(MAN) model performed better on the training data than the previously preferred method but less well on the test set. The Holt Winters Damped Additive method is still the best according to the accuracy measures.

Residuals

```
checkresiduals(wine.stlf)

## Warning in checkresiduals(wine.stlf): The fitted degrees of freedom is
## based on the model used for the seasonally adjusted data.
```

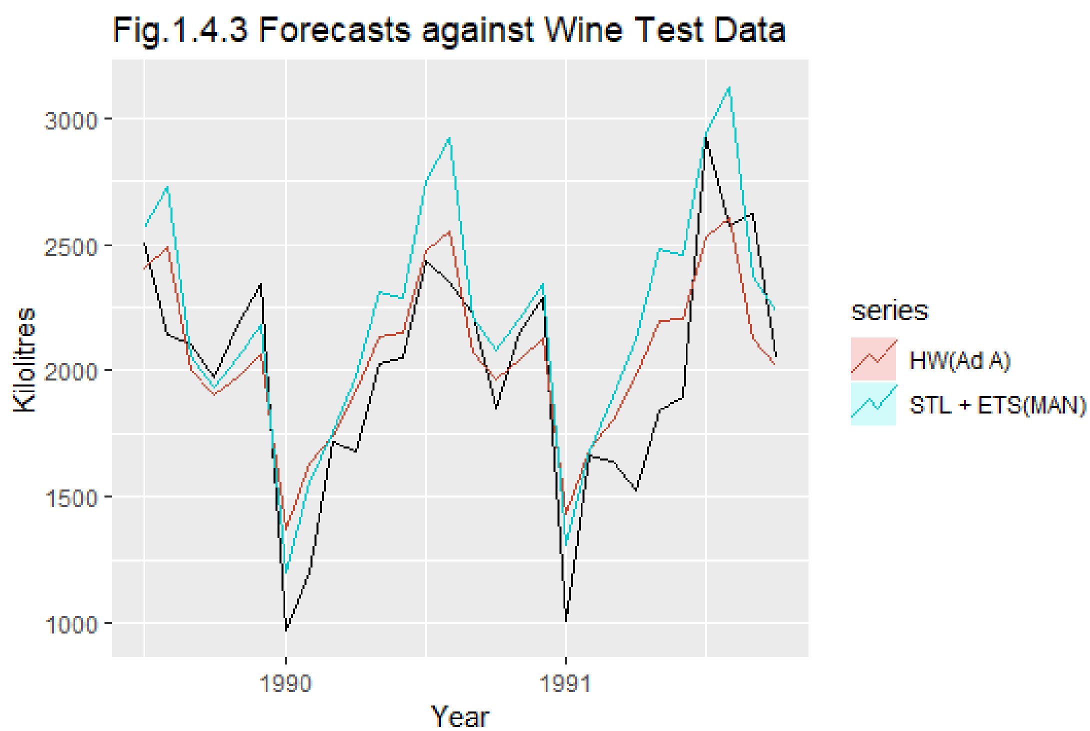



```
##
##  Ljung-Box test
##
## data:  Residuals from STL +  ETS(M,A,N)
## Q* = 29.911, df = 19, p-value = 0.05294
##
## Model df: 4.   Total lags used: 23

res.stlf<-residuals(wine.stlf)
shapiro.test(res.stlf)

##
##  Shapiro-Wilk normality test
##
## data:  res.stlf
## W = 0.97991, p-value = 0.08432
```

There are two lags out of 36 outside the confidence bounds in the ACF plot and the Ljung-Box test p-value is above the 5% level so there is no significant autocorrelation left in the residuals. The residuals plot seems stable with a mean around zero and the histogram appears reasonably normally distributed, which is supported by the Shapiro test statistic being above 0.0.5 so we can accept the null hypothesis of normality.

The previous best method and the STL forecasts against the test set are compared in Fig. 1.4.3.

```
autoplot(wine.test) +
  autolayer(wine.hwda, series = "HW(Ad A)", PI = FALSE)+
  autolayer(wine.stlf,series = "STL + ETS(MAN)", PI = FALSE) +
  ggtitle("Fig.1.4.3 Forecasts against Wine Test Data")+
  xlab("Year") + ylab("Kilolitres")
```



The HW Additive method appears to forecast the test data better.

Question 1.5

ARIMA models require stationary data. This series has trend and seasonality, so transformations and differencing are likely to be needed.

Using the kpss unit root test the null hypothesis of stationarity is rejected.

```
library(urca)
summary(ur.kpss(wine.train))

##
## #######################
```

```
## # KPSS Unit Root Test #
## ######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 1.6777
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

As there is variance in the data, a Box Cox transformation was used.

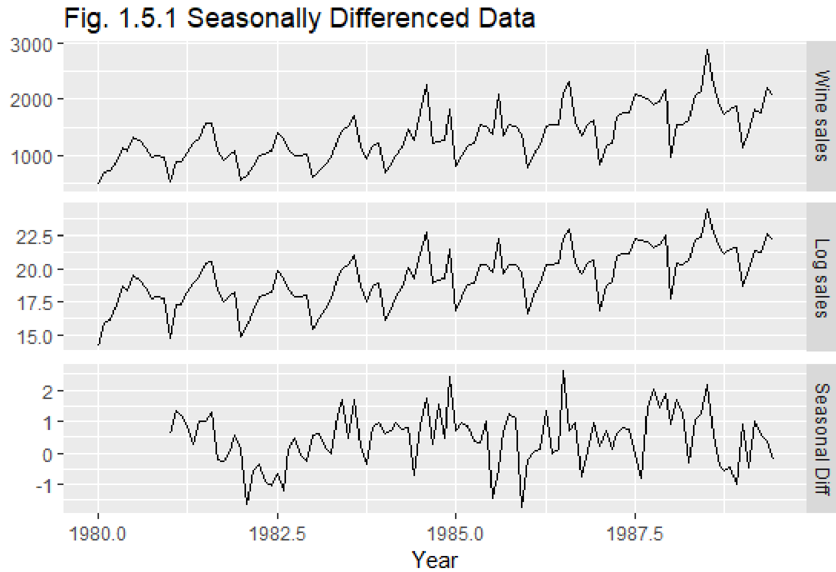
```
wine.trans<-BoxCox(wine.train, lambda = "auto")
```

The nsdiffs() function identified one seasonal and no first order difference is required on the transformed series.

```
wine.trans %>% nsdiffs()

## [1] 1

wine.trans %>% diff(lag=12) %>% ndiffs()

## [1] 0
```

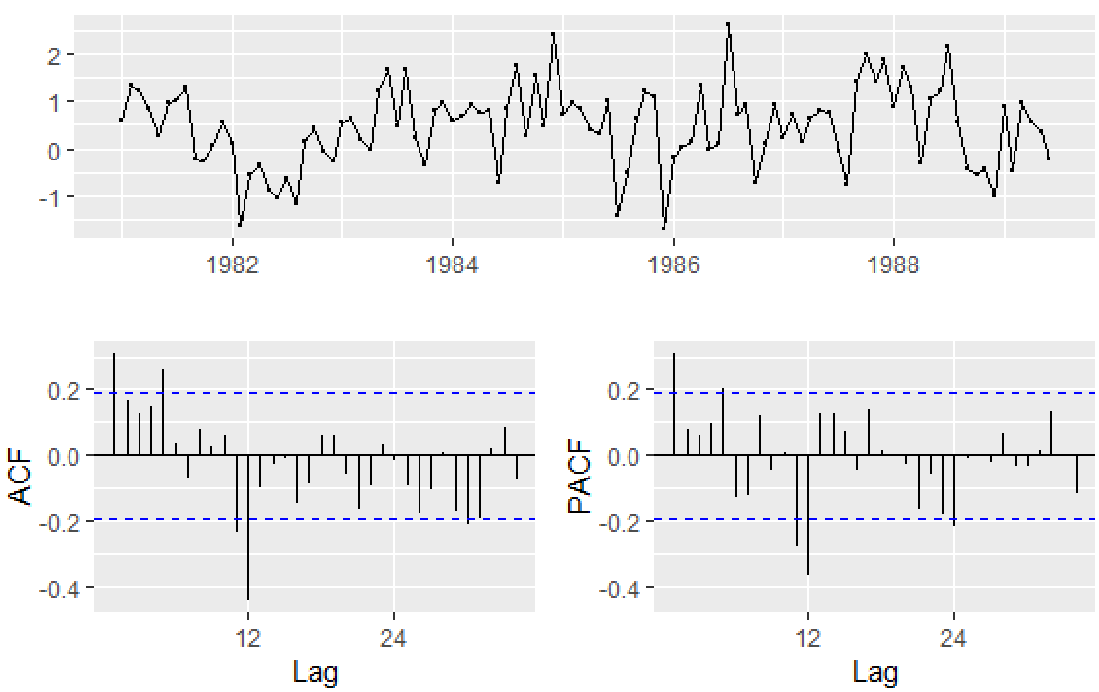
The original, transformed and differenced data are shown in Fig.1.5.1. Seasonal differencing has removed seasonality and appears to have also removed the trend, so first order differencing is not recommended by R. The data appears relatively stationary, although there is still some variance remaining in the data.

```
cbind("Wine sales" = wine.train,
      "Log sales" = wine.trans,
      "Seasonal Diff" = diff(wine.trans,lag = 12))%>%
autoplot(facets=TRUE) +
xlab("Year") + ylab("") +
ggtitle("Fig. 1.5.1 Seasonally Differenced Data")
```



The ACF and PACF plots for the differenced series:

```
Wine.diff<-(wine.trans%>% diff(lag = 12))
ggtsdisplay(Wine.diff)
```



There are significant spikes in both the ACF and PACF at lags 1, 5 and 11 suggesting there may be AR and MA terms present. The PACF and ACF both appear to tail off below the threshold after lag 1 and the ACF seems to cut off after lag 5 suggesting a (1,0,1) or a (1,0,2) model could be appropriate.

There are seasonal spikes at lag 12 on the ACF and two seasonal lags at 12 and 24 on the PACF. The PACF seems to cut off after these lags suggesting an AR(2) model. The ACF tails off after the seasonal lags which would also suggest an AR model. Various Arima models were tried and compared using the corrected AIC statistic.

*Note: lambda = "auto" returned an error on the first model so the value had to be manually entered.

```
ar1 <- Arima(wine.train, order=c(1,0,1), seasonal=c(2,1,0),
             lambda=0.2427797)
ar2<-Arima(wine.train, order = c(2,0,0), seasonal = c(1,1,0),
           lambda= "auto")
ar3<-Arima(wine.train, order = c(1,0,0), seasonal = c(2,1,0),
           lambda= "auto")
ar4<-Arima(wine.train, order = c(1,0,1), seasonal = c(1,1,0),
           lambda= "auto")
ar5<-Arima(wine.train, order = c(2,0,0), seasonal = c(2,1,0),
           lambda= "auto")
ar6<-Arima(wine.train, order = c(1,0,1), seasonal = c(0,1,1),
           lambda= "auto")

ar1$aicc

## [1] 228.6095

ar2$aicc

## [1] 242.8782

ar3$aicc

## [1] 256.0814

ar4$aicc

## [1] 231.2316

ar5$aicc

## [1] 243.5517

ar6$aicc

## [1] 215.3616
```

The best manually fit models based on the AICc are:

ar6: (1,0,1)(0,1,1)

Running auto.arima returned the following model:

```
ar7<-auto.arima(wine.train,stepwise = F, approximation = F, lambda = "auto")
ar7

## Series: wine.train
## ARIMA(1,0,1)(2,1,0)[12] with drift
## Box Cox transformation: lambda= 0.2427797
##
```

```
## Coefficients:
##          ar1      ma1     sar1     sar2   drift
##       0.9015  -0.7289  -0.6336  -0.2392  0.0394
## s.e.  0.0737   0.1062   0.1017   0.1075  0.0082
##
## sigma^2 estimated as 0.4753:  log likelihood=-106.73
## AIC=225.46   AICc=226.34   BIC=241.21
```

Observations

The best manually selected model according to the AICc is (1,0,1)(0,1,1) with an AICc of 215.36.

Setting Trace = True in the auto.arima() function prints out all the models and this model was not listed by auto.arima, which returned (1,0,1)(2,1,0) with an AICc of 226.34 as the best model.

The auto.arima function looks for a model with the lowest AICc but also looks for a model that is numerically sound. A model with roots inside or close to the unit circle will not be selected (Robjhyndman.com, 2019). Plotting them for the manually selected model:

```
Arima(wine.train, order=c(1,0,1), seasonal=c(0,1,1), xreg=seq_along(wine.trai
n)) %>%
  autoplot()
```

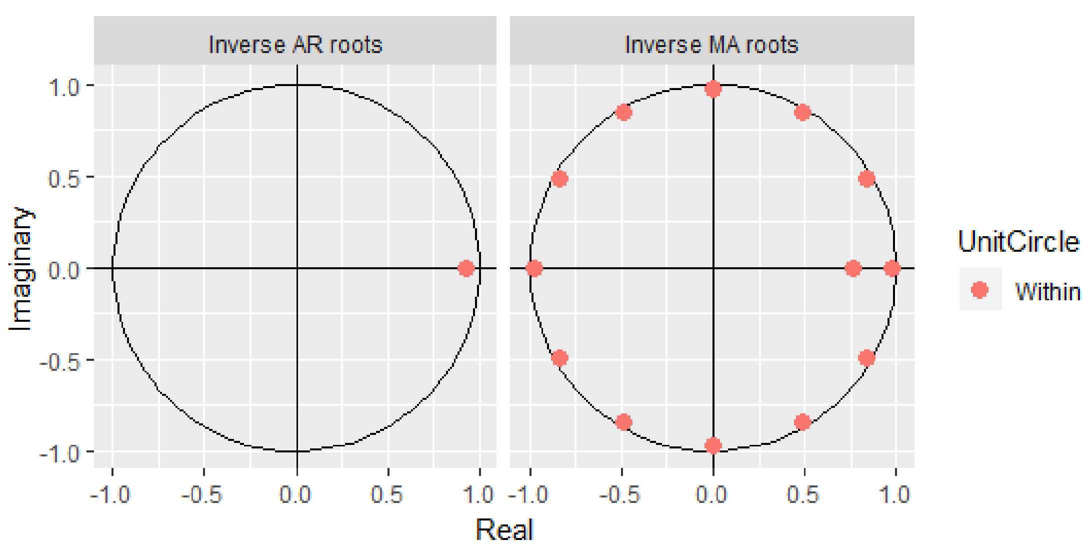
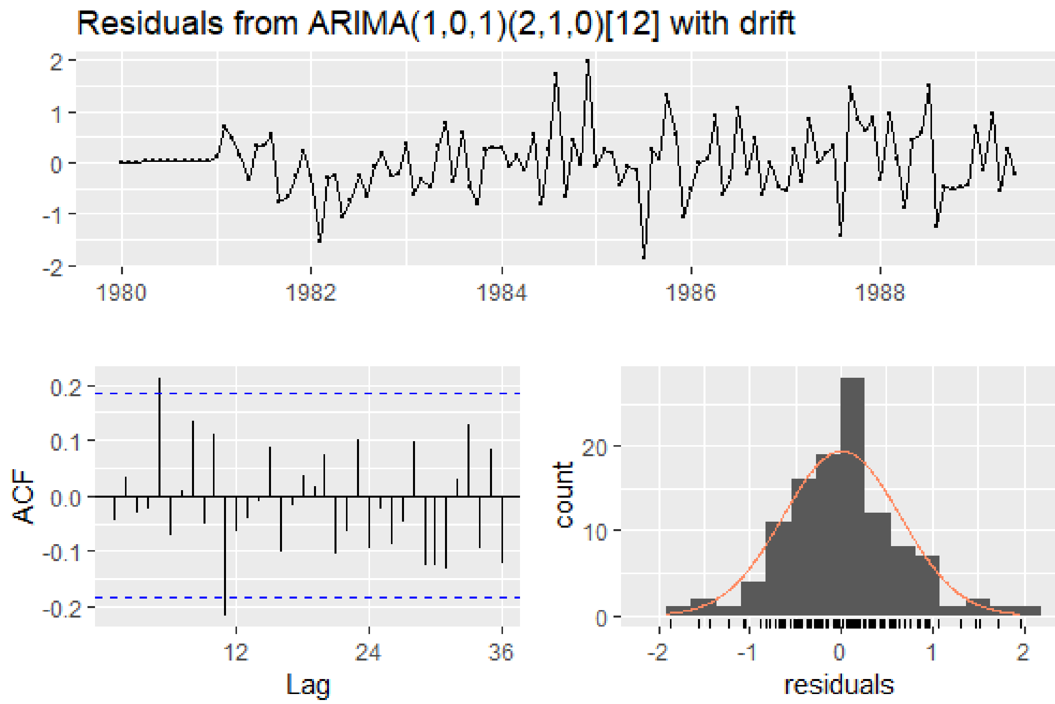


Fig. 1.5.2 Inverse AR Roots

This model is considered numerically unstable so the auto.arima model is selected instead.

Residuals

```
checkresiduals(ar7)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,1)(2,1,0)[12] with drift
## Q* = 24.656, df = 18, p-value = 0.1347
##
## Model df: 5.   Total lags used: 23
```

```
res.ar7<-residuals(ar7)
shapiro.test(res.ar7)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.ar7
## W = 0.98247, p-value = 0.1419
```

Question 1.6

A forecast for the Arima model is shown in Fig. 1.6.1.

```
wine.ar7<-forecast(ar7, h = h, biasadj = TRUE, bootstrap = T)
autoplot(wine.ar7) +xlab("Year") + ylab("Wine sales")
```

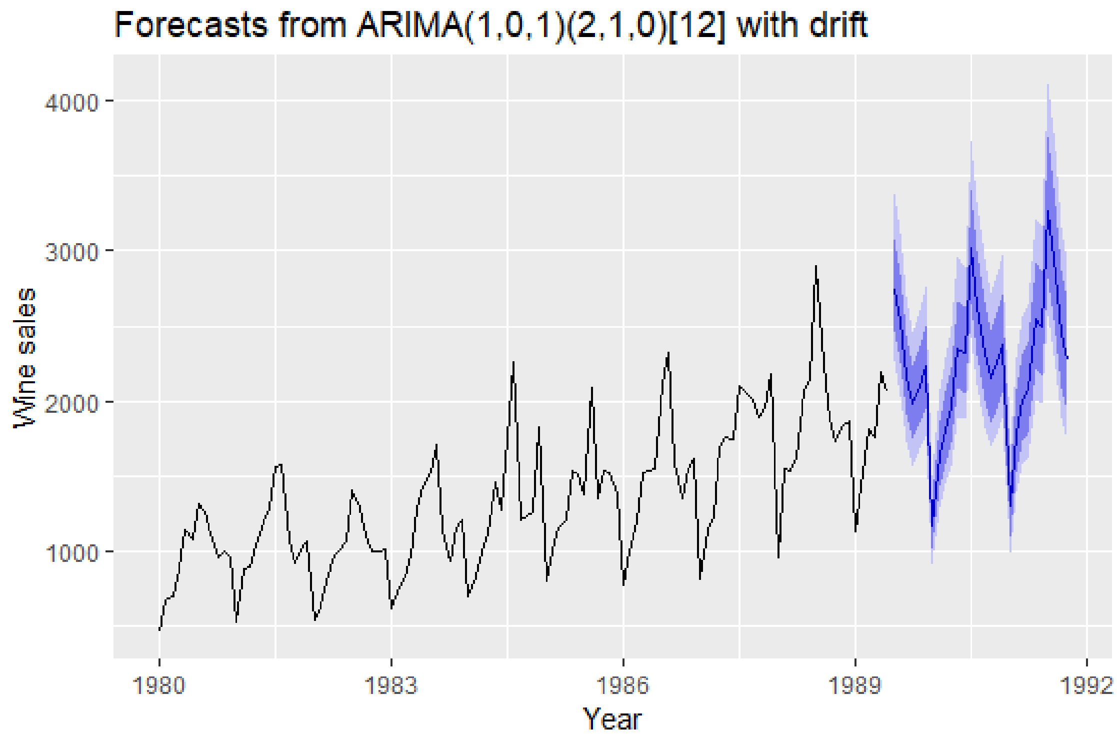
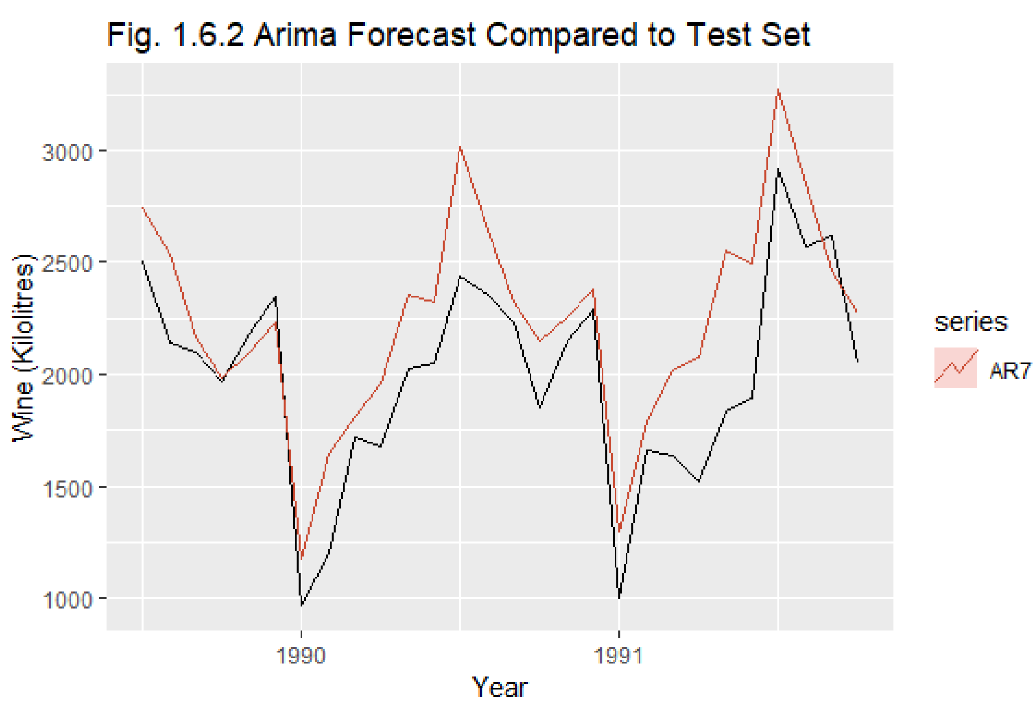


Fig. 1.6.1 Forecast from Arima Model for the Wine Data

A comparison of the Arima forecast against the test set is plotted in Fig. 1.6.2.

```
autoplot(wine.test) + autolayer(wine.ar7, PI = F, series = "AR7") +
  ggtitle("Fig. 1.6.2 Arima Forecast Compared to Test Set")+
  ylab("Wine (Kilolitres)") + xlab("Year")
```



The accuracy of each of the methods and models is shown in Table 1.6.

```
c1<-round(accuracy(wine.snaive, wine.test),1)
c2<-round(accuracy(wine.hwda,wine.test),1)
c3<-round(accuracy(wine.stlf,wine.test),1)
c4<-round(accuracy(wine.ar7,wine.test),1)

a.table <- rbind(c1,c2,c3,c4)
row.names(a.table)<-c("SNaive Train", "SNaive Test", "HW AddD Train", "HW Add
D Test", "STL + ETS Train", "STL + ETS Test","Arima Train", "Arima Test")

a.table<-as.data.frame(a.table, caption = "Table 1.6 Comparison of Accuracy M
easures")
kable(a.table)
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
SNaive Train	111.0	249.2	192.2	6.8	13.1	1.0	0.3	NA
SNaive Test	49.4	295.9	250.4	0.9	12.6	1.3	0.4	0.6
HW AddD Train	18.7	159.0	116.3	1.0	8.9	0.6	0.0	NA
HW AddD Test	-60.1	256.7	209.2	-6.1	12.2	1.1	0.1	0.6
STL + ETS Train	5.2	144.1	99.7	-0.5	7.0	0.5	-0.1	NA
STL + ETS Test	-207.7	323.6	254.1	-12.1	14.0	1.3	0.0	0.7
Arima Train	6.6	172.8	119.0	-0.6	8.2	0.6	-0.1	NA
Arima Test	-248.1	325.8	273.0	-14.0	15.0	1.4	0.4	0.7

Table 1.6 Comparison of Preferred Methods

Observations

Comparing the models:

1) The STL + ETS model performed the best on the training set but considerably worse on the test set.
2) The best performing method against the test set was the HW Damped Additive.
3) Seasonal naive, which is a basic method, only picking up seasonal patterns, performed better than either the STL or the Arima models.

It should be noted that the Holt Winters Additive Method is underpinned by two innovations state space models, one with multipliative and one with additive errors. Both models will return the same point forecasts but with different prediction intervals if the same smoothing parameters are used.

Question 1.7

Conclusions

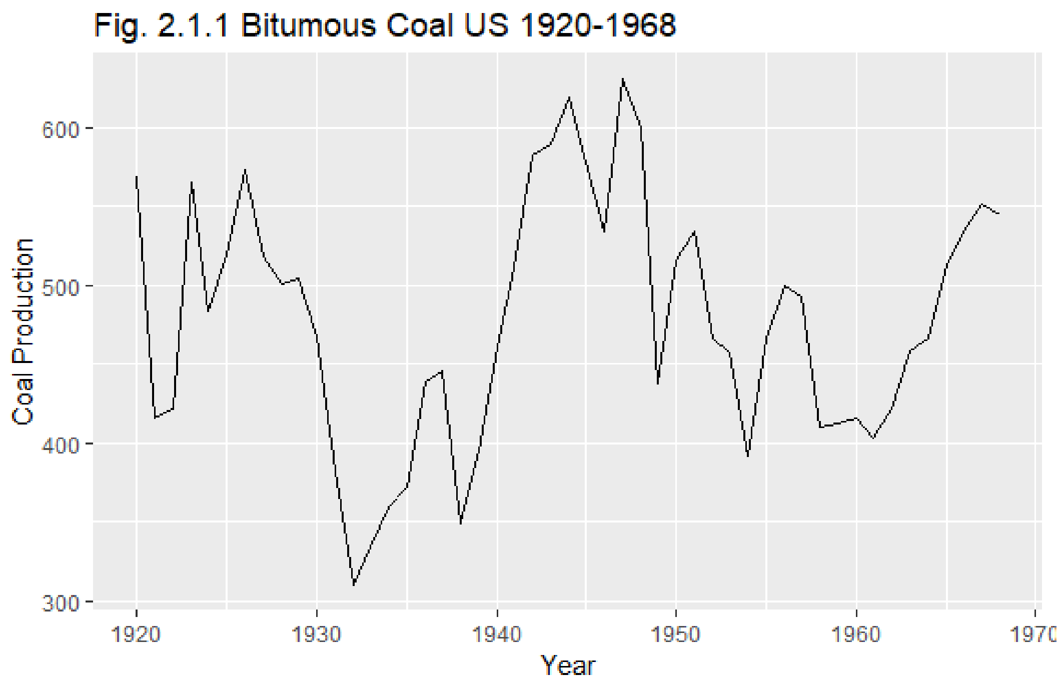
This review has identified that:

1) More complex models will not necessarily provide the best forecasts and in fact basic methods can often perform better.
2) Where a model fits the training set well (overfitting) it will not necessarily produce good forecasts against the test set.
3) Related to this point, residuals are helpful in checking if the model has captured the information in the data and determing how reliable prediction intervals for a forecast may be but should not be used as the basis to choose forecasting models; it is better to test the model on new data.
4) The best forecast accuracy using RMSE was produced by the Holt Winters Additive Damped method which captured the seasonality in the data comparatively well.
5) Methods produce point forecasts, but Innovation State Space models which underpin these methods provide forecasts <u>and</u> can generate prediction intervals - the HW Damped Additive method is underpinned by two state space models, ETS(MadA) and ETS(AAdA) which are generally preferable to the HW method.
6) Outliers in the data can affect the performance of models that weight recent and past values differently but deciding whether to remove them is best left to domain experts, since adjusting the values can result in poor forecasts and performance intervals.

Question 2.1

The bicoal data plots are shown in Figs. 2.1.1. to 2.1.4 below:

```
autoplot(bicoal) + ggtitle("Fig. 2.1.1 Bitumous Coal US 1920-1968") + xlab("Y
ear") + ylab("Coal Production")
```




```
gglagplot(bicoal)
```

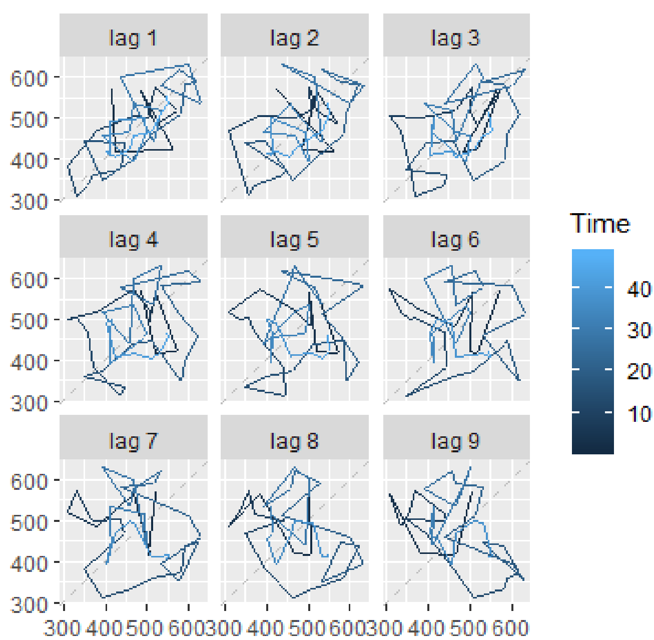


Fig. 2.1.2 Lag Plot - Bicoal

```
ggAcf(bicoal)
```

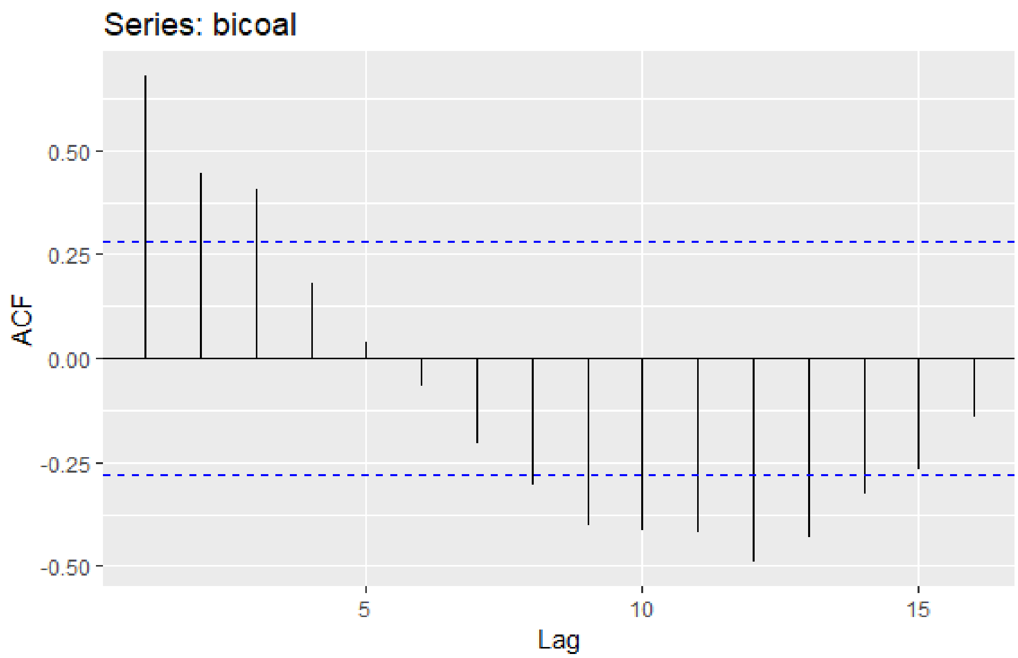


Fig. 2.1.3 ACF Plot - Bicoal

Observations

From the plots, the series displays no significant trend and no seasonality. There is a large dip in the early 1930's around the time of the depression and peaks over the 1940's where production increased substantially during the second world war but there seems to be constant variance over the series. The lagplot shows a roughly linear pattern suggesting there is some autocorrelation in the data, although the points are not tightly clustered. The ACF plot shows some cyclical patterns in the data over a 5 to 10-year period but these are not predictable over the longer-term. The data appears stationary. This was confirmed by running the following tests:

1)KPSS - null hypothesis is accepted, the data is stationary, as the test statistic p is below the 1% critical value.

```
library("urca")
summary(ur.kpss(bicoal))

##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 3 lags.
##
## Value of test-statistic is: 0.0905
##
## Critical value for a significance level of:
```

```
##                   10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

2)Alternatively the ndiffs() function = 0, no differencing required.

```
ndiffs(bicoal)

## [1] 0
```

Question 2.2

In an ARIMA (p, d, q) model the values of p, d and q are given as:

p - the order of the autoregressive (AR) part d - the degree of differencing needed q - the order of the moving average (MA) part

The equation:

$$x_t == c + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \phi_3 x_{t-3} + \phi_4 x_{t-4} + \varepsilon_t$$

is an autoregressive model in the form of AR(p) with forecasts made on the past values of x for number of lags plus a constant and an error term. This model seems appropriate as the series appears stationary with no trend or seasonality so does not need differencing and with no moving average component.

Question 2.3

```
ggtsdisplay(bicoal)
```

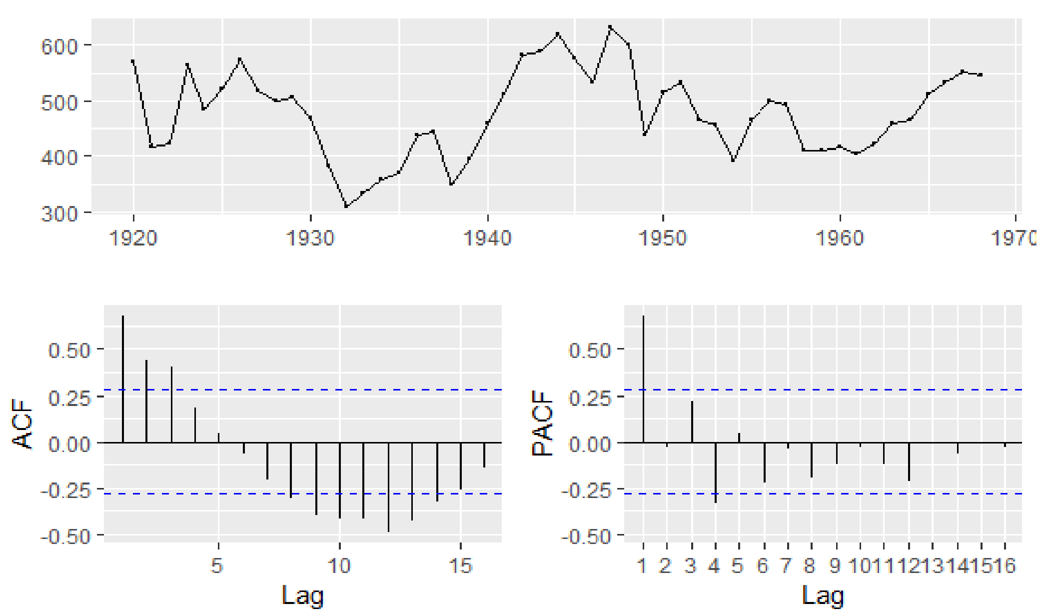


Fig. 2.3.1 Tsdisplay Plots for Bicoal Series

Looking at the PACF, which is associated with the AR or p component, there are 2 significant spikes at lags 1 and 4 after which it appears to taper away. This suggests that there is autocorrelation up to lag 4 and an order of AR(4) may be required. The ACF plot tapers in the early and cyclical lags with a sinusoidal pattern which is also indicative of an AR component.

Question 2.4

The forecasts for the next three years (1969-1971) are shown in Fig. 2.4.1. The manually calculated forecasts are derived from multiplying each coefficient in the equation by the figures from the four previous years plus the constant.

Fig. 2.4.1 Comparison of auto.arima and manually calculated model

	Year	Tonnes	auto.arima	Diff	Recalc	Mean Adj
	1964	467				
	1965	512				
	1966	534				
	1967	552				
	1968	545				
forecast	***1969****	525.81	527.63	1.82	528.18	526.71
	1970*	513.80	517.19	3.39	517.66	516.27
	1971*	499.67	503.81	4.13	504.00	502.89

Calculation of forecast based on equation coefficients

		Sum
1969*	162+(0.83 x 545) + (-0.34 x 552) + (0.55 x 534) + (-0.38 x 512)	525.81
1970*	162 +(0.83 x 526) + (-0.34 x 545) + (0.55 x 552) + (-0.38 x 534)	513.80
1971*	162 + (0.83 x 514) + (-0.34 x 526) + (0.55 x 545) + (-0.38 x 552)	499.67

Comparison of manual and auto.arima coefficients and constant

	constant	coeff 1	coeff 2	coeff 3	coeff 4	Sum
equation	162	0.83	-0.34	0.55	-0.38	0.66
auto.arima	162.984	0.8334	-0.3443	0.5525	-0.378	0.6636

Re calculation of forecasts based on auto.arima coefficients

		Sum
1969	161.984 + (0.8334 x 545) + (-0.3443 x 552) + (0.5525 x 534) + (-0.378 x 512)	528.18490
1970	161.984 + (0.8334 x 528.1849) + (-0.3443 x 545) + (0.5525 x 552) + (-0.378 x 534)	517.65780
1971	161.984 + (0.8334 x 517.6578) + (-0.3443 x 528.1849) + (0.5525 x 545) + (-0.378 x 552)	504.00245

Re calculation of the constant from auto.arima

c = (1 - sum of coefficients) x mean	1 - (0.8334 - 0.3443 + 0.5525 - 0.378) x 481.5221	161.984034

Question 2.5

The model was run and returned forecasts:

```
fit.bc<-arima(bicoal, order = c(4,0,0))
fit.bc
```

```
##
## Call:
## arima(x = bicoal, order = c(4, 0, 0))
##
## Coefficients:
##          ar1      ar2     ar3      ar4  intercept
##       0.8334  -0.3443  0.5525  -0.3780   481.5221
## s.e.  0.1366   0.1752  0.1733   0.1414    21.0591
##
## sigma^2 estimated as 2509:  log likelihood = -262.05,  aic = 536.1
```

Then checked using auto.arima

```
fc.bc<-forecast(auto.arima(bicoal, stepwise = F, approximation = F,), h = 3)
fc.bc$model

## Series: bicoal
## ARIMA(4,0,0) with non-zero mean
##
## Coefficients:
##          ar1      ar2     ar3      ar4      mean
##       0.8334  -0.3443  0.5525  -0.3780  481.5221
## s.e.  0.1366   0.1752  0.1733   0.1414   21.0591
##
## sigma^2 estimated as 2795:  log likelihood=-262.05
## AIC=536.1   AICc=538.1   BIC=547.45
```

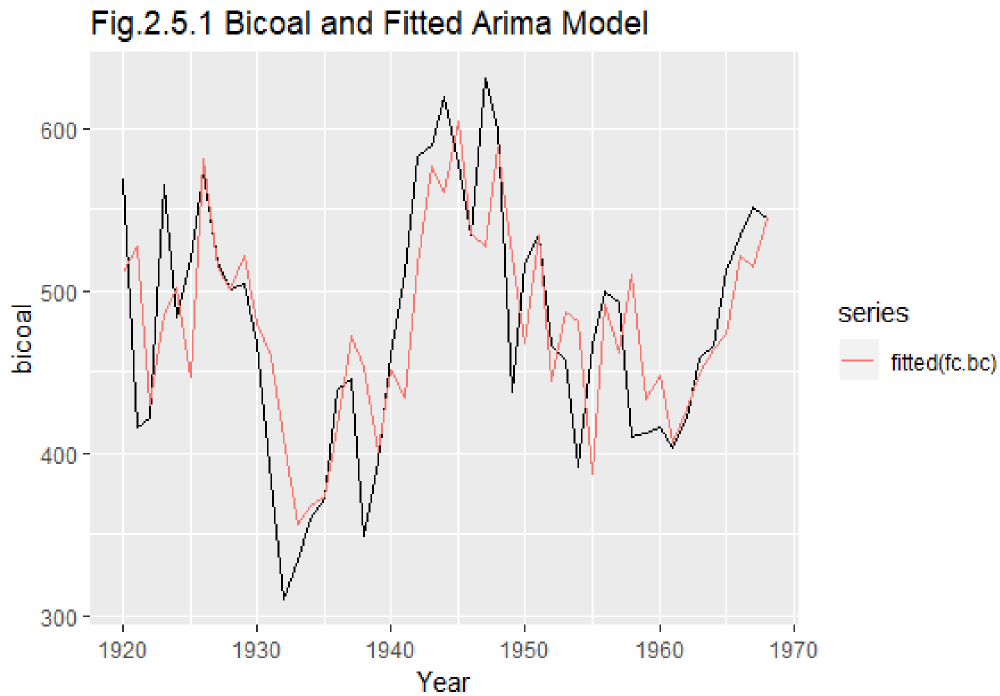
Point forecasts:

```
fc.bc

##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 1969       527.6291 459.8804 595.3779 424.0164 631.2419
## 1970       517.1923 429.0014 605.3832 382.3160 652.0686
## 1971       503.8051 412.4786 595.1315 364.1334 643.4768
```

Table 1. 7 Arima Point Forecasts - Bicoal

The fitted values plotted against the series are shown in Fig. 2.5.1

```
autoplot(bicoal) + autolayer(fitted(fc.bc)) +
  ggtitle("Fig.2.5.1 Bicoal and Fitted Arima Model") + xlab("Year")
```



The forecasts for 1969 to 1971 compared to the Arima model forecasts are shown in Fig. 2.5.2.

```
autoplot(fc.bc)
```

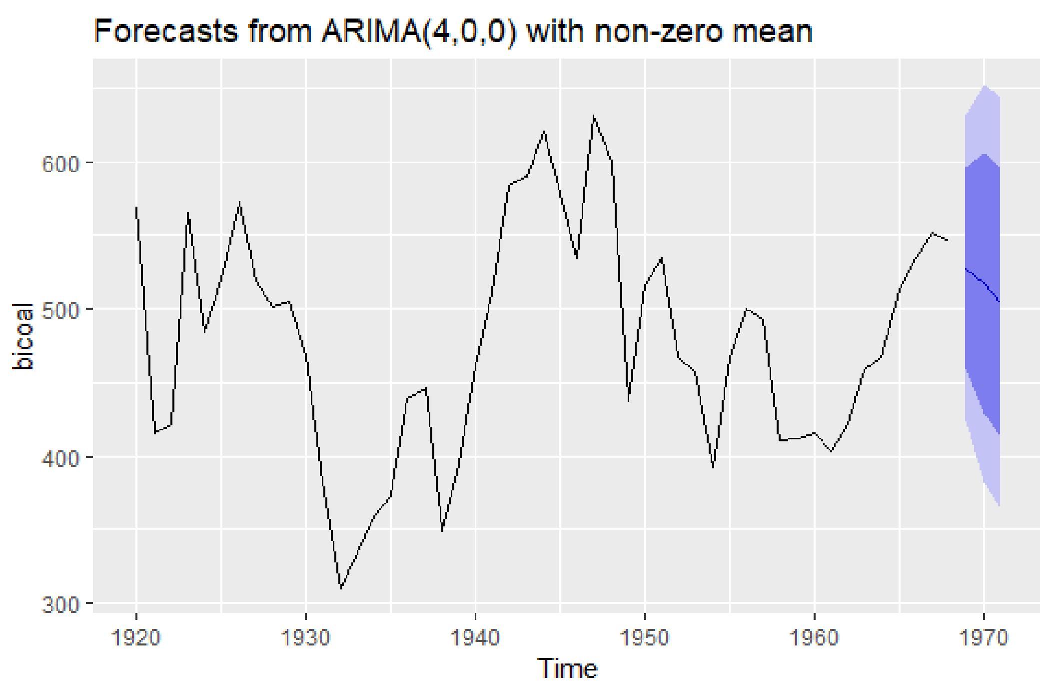
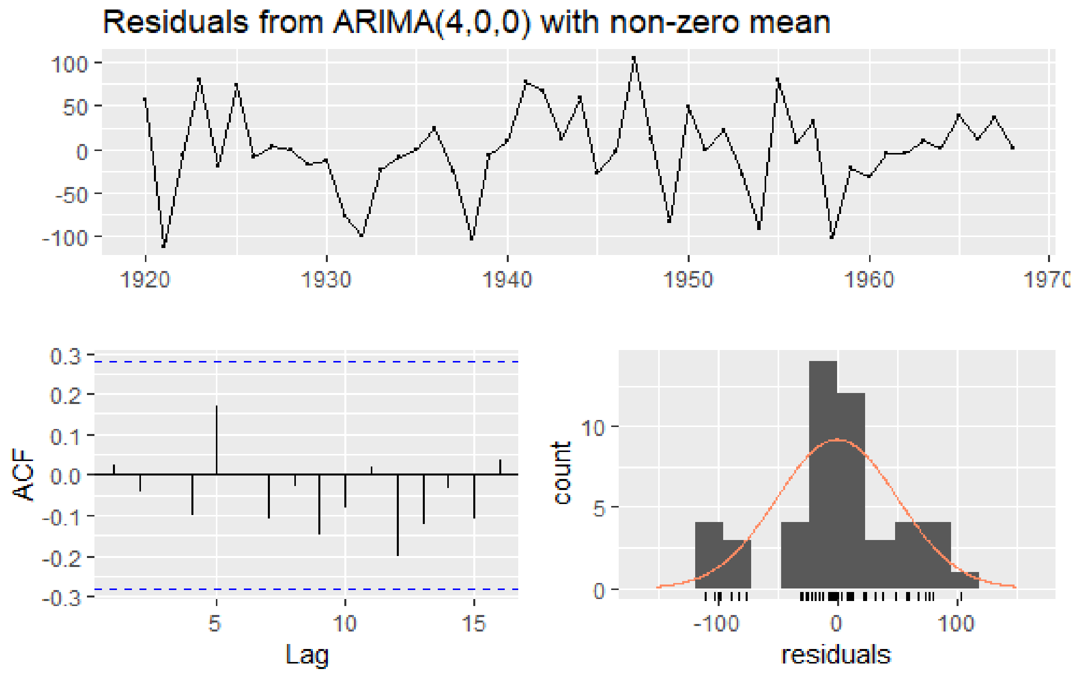


Fig. 2.5.2 Arima Forecast for Bicoal Series

Residuals

```
checkresiduals(fc.bc)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(4,0,0) with non-zero mean
## Q* = 4.852, df = 5, p-value = 0.4342
##
## Model df: 5.   Total lags used: 10
```

```
res.bc<-residuals(fc.bc)
shapiro.test(res.bc)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.bc
## W = 0.94606, p-value = 0.02564
```

Looking at the ACF and Ljung-Box test, the residuals are uncorrelated, so the model has captured the available information in the data. The variance seems stable, but the mean is negative non-zero which causes the forecasts to be slightly biased but can be adjusted by adding back the mean of the residuals to each of the point forecasts. The mean is:

```
res.bc<-fc.bc$residuals
mean(res.bc)
```

```
## [1] -0.9191614
```

The histogram appears roughly normally distributed by eye but the shapiro wilks test rejects the null hypothesis. The prediction intervals are wide anyway reflecting uncertainty in forecasting this data so short forecast periods would be preferable.

Observations

The differences between the forecasts from the auto.arima function and those manually calculated is in part due to roundings on the coefficients and constant term. The auto.arima coefficients are to 4 decimal places compared to 2 decimal places in the manual calculation and the constant term is also rounded in the equation.

The remaining difference between the manual calculation and that produced by auto.arima is likely due to error or randomness term which is not included in the manual calculation.

Question 3.1

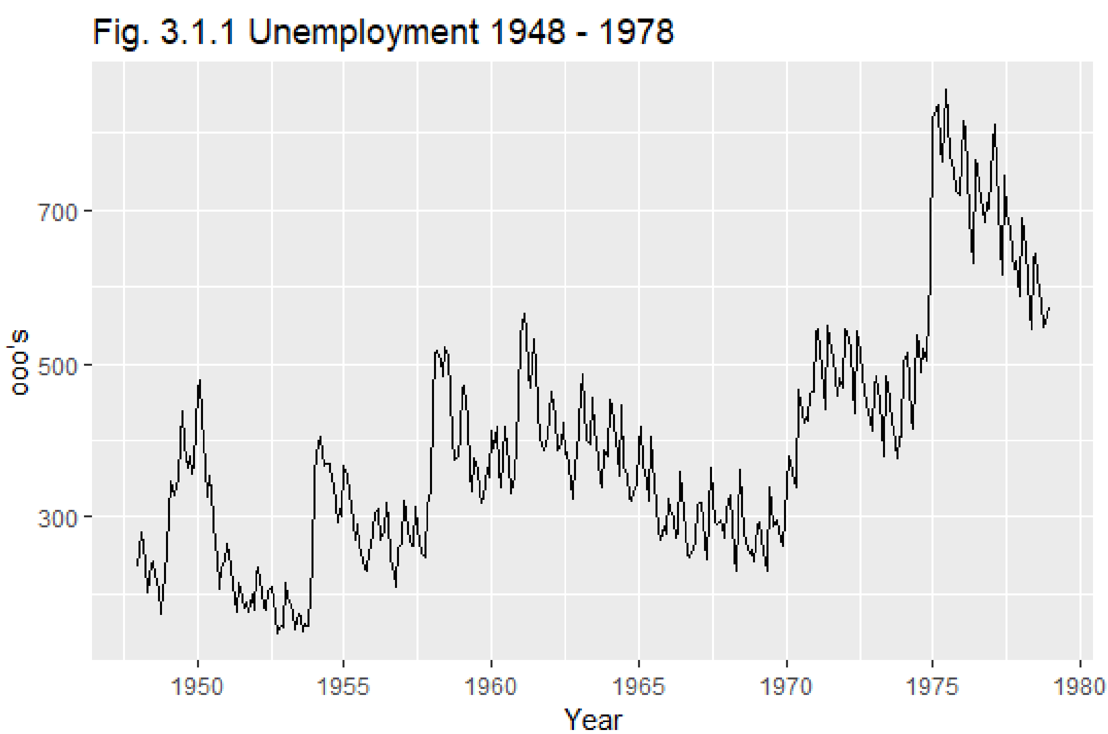
Reading in the unemployment data from Excel to R:

```
unemp.data <- readxl::read_excel("C:/Users/Imy Hull/Desktop/unemp.xlsx")
unemp.ts <- ts(unemp.data['x_t'],frequency=12,start=c(1948,1))
```

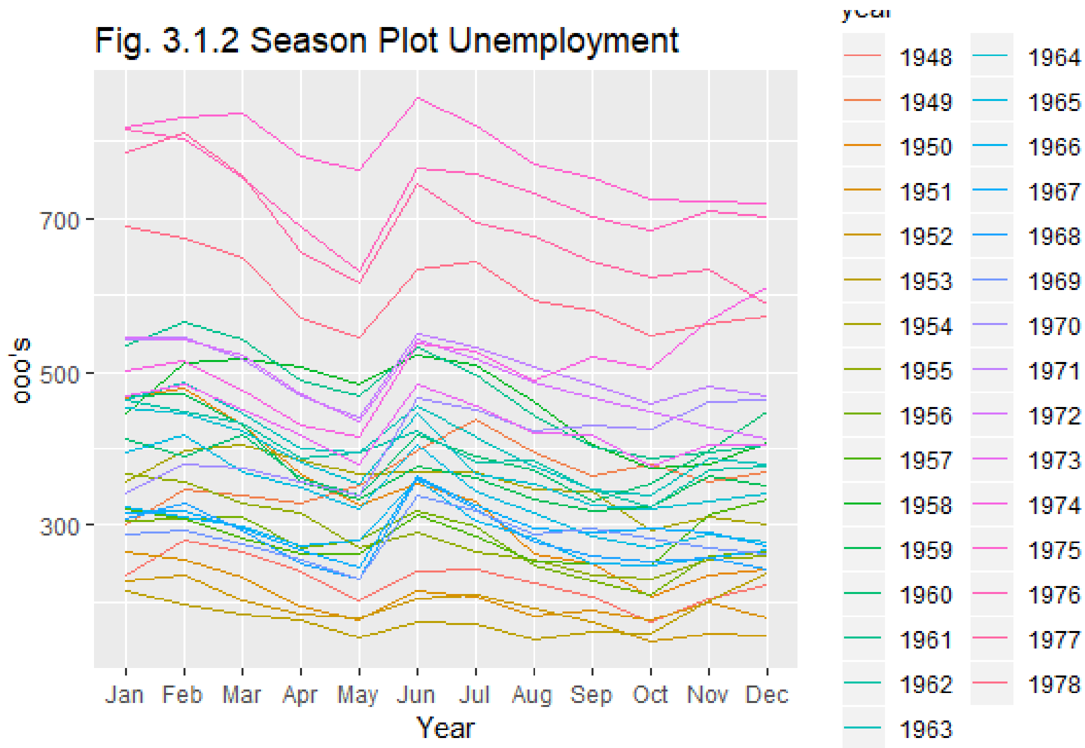
Please note: the starting date of the data as per the spreadsheet is Jan. 1948 and not 1949 as included in the 'loading data sheet' on moodle.

Exporatory plots are contained in Figs. 3.1.1 3.1.6

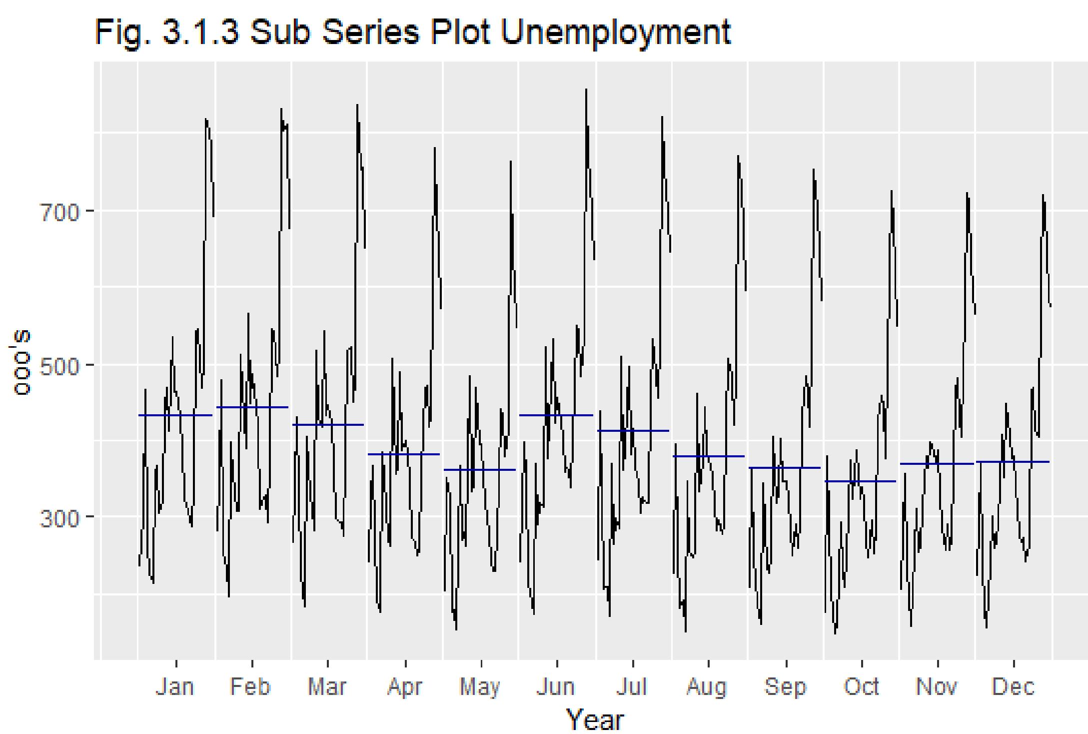
```
autoplot(unemp.ts) + xlab("Year") + ylab("ooo's")+
  ggtitle("Fig. 3.1.1 Unemployment 1948 - 1978")
```



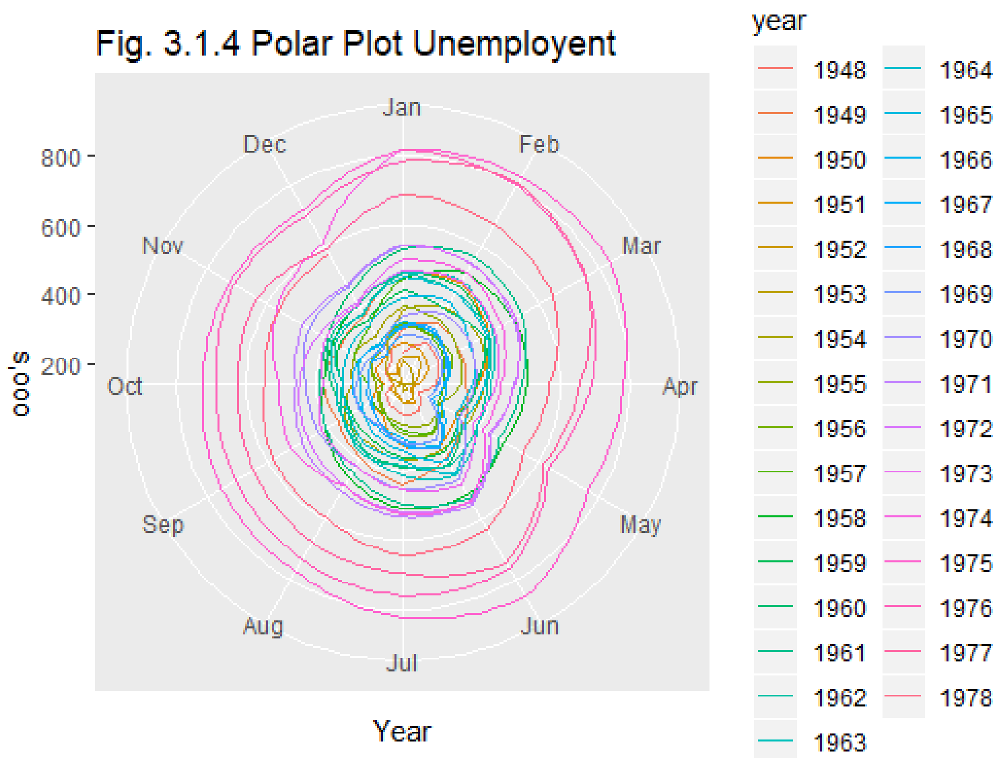

```
ggseasonplot(unemp.ts) + xlab("Year") + ylab("ooo's")+
  ggtitle("Fig. 3.1.2 Season Plot Unemployment")
```



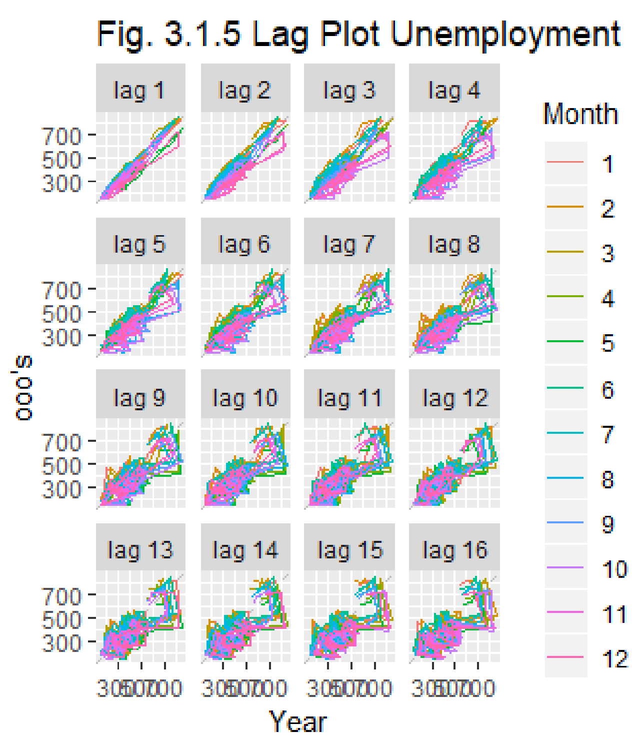

```
ggsubseriesplot(unemp.ts)+ xlab("Year") + ylab("ooo's")+
  ggtitle("Fig. 3.1.3 Sub Series Plot Unemployment")
```




```
ggseasonplot(unemp.ts, polar = TRUE) + xlab("Year") + ylab("ooo's")+
  ggtitle("Fig. 3.1.4 Polar Plot Unemployent")
```




```
gglagplot(unemp.ts)+ xlab("Year") + ylab("ooo's")+
  ggtitle("Fig. 3.1.5 Lag Plot Unemployment")
```




```
ggAcf(unemp.ts)
```

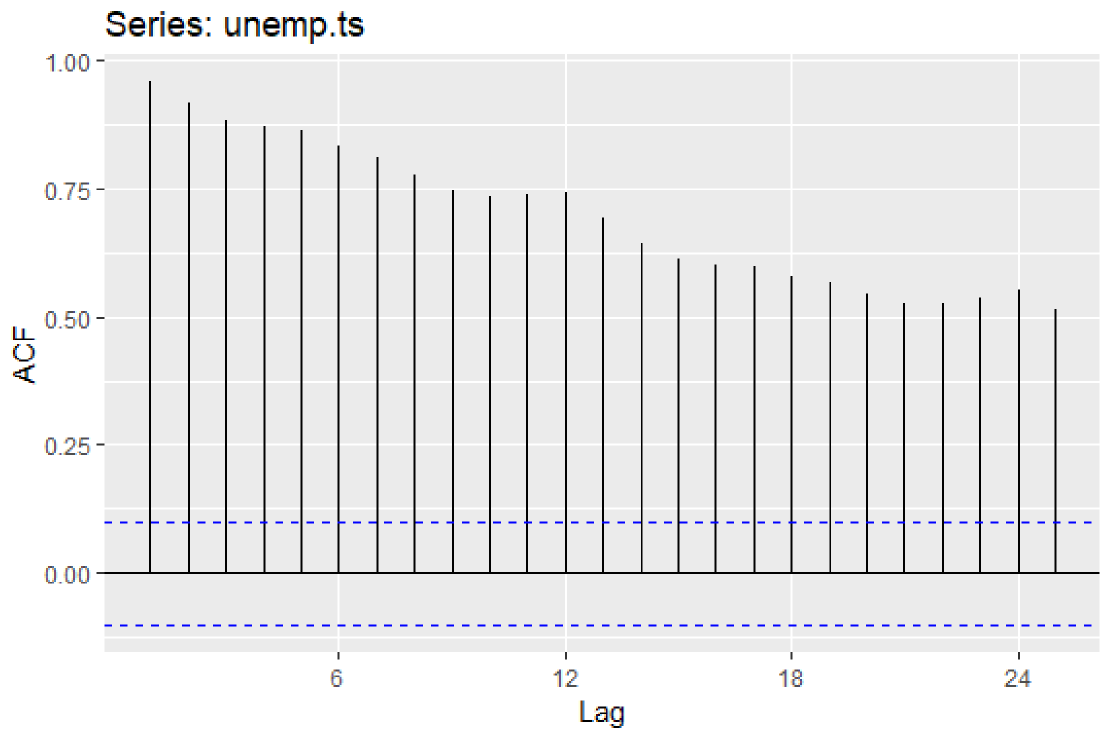


Fig. 3.1.6 ACF Plot - Unemployment

Observations

The series trends upwards from low points in the post war period to a large spike upwards from January 1975 possibly due to a lagged effect from the oil crisis in 1973-4, where it then trends back down again. There are seasonal peaks over the summer period of June to July likely reflecting the effect of school leavers. There is some apparent cyclical pattern over the series and the variance seems to change with larger fluctuations occuring in the first half. The lagplot shows high correlation over the early lags and the ACF plot correspondingly has significant autocorrelation from the early lags, decaying slowly over the series.

A stationary series typically has constant variance, is roughly horizontal and with no long-term predictable patterns.The series does not appear to be stationary as it has both trend and seasonality. From the ACF plot the first lag is large and positive and then decays slowly,also suggesting non-stationarity.

The KPSS test p-value is above the 5% significance level so the null hypothesis of stationary, non-seasonal data is rejected.

```
library("urca")
summary(ur.kpss(unemp.ts))

##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 5 lags.
```

```
##
## Value of test-statistic is: 3.309
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

The data was transformed using BoxCox and one order of seasonal differencing applied.

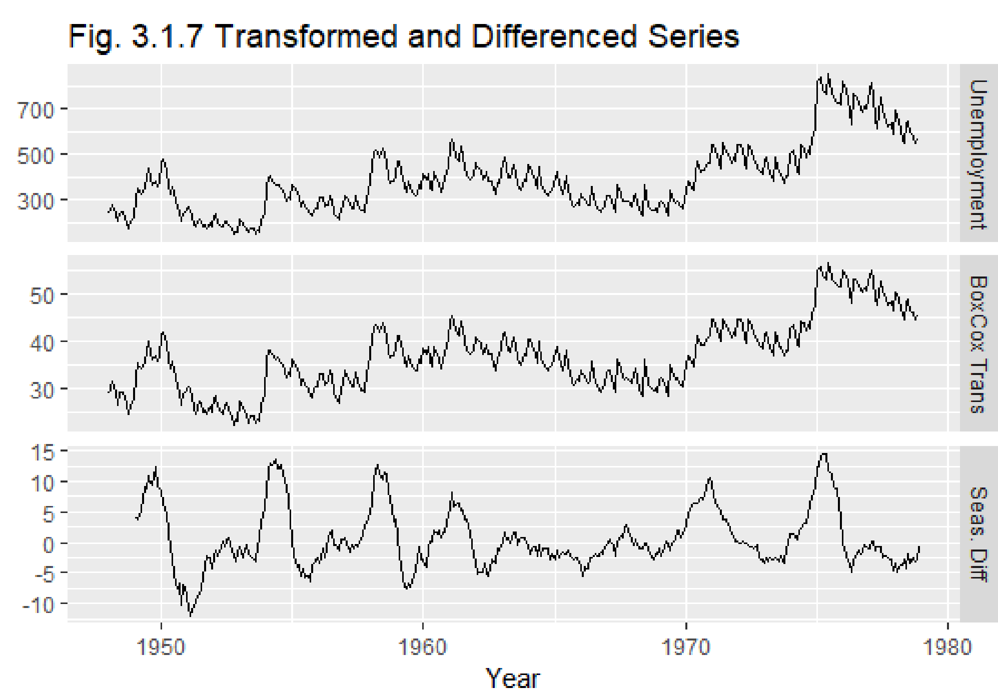
```
un.lambda<-BoxCox.lambda(unemp.ts)
un.lambda

## [1] 0.5096318

un.trans<-BoxCox(unemp.ts, lambda = 1/2)
```

The number of first order differences required on the seasonally differenced series:

```
ndiffs(diff(un.trans,12))

## [1] 0

un.diff<-diff(un.trans,12)
```

The seasonally differenced data is shown in the Fig. 3.1.7. There trend has been removed and much of the variance with just some larger cyclical spikes left in the differenced series.

```
cbind("Unemployment" = unemp.ts,
      "BoxCox Trans" = un.trans,
      "Seas. Diff" = un.diff) %>%
  autoplot(facets=TRUE) +
    xlab("Year") + ylab("") +
    ggtitle("Fig. 3.1.7 Transformed and Differenced Series")
```



Running the Unit Root test again on the seasonally differenced data we can accept the null hypothesis of stationarity:

```
summary(ur.kpss(un.diff))

##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 5 lags.
##
## Value of test-statistic is: 0.0694
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

Question 3.2

Running ACF and PACF plots allows the most appropriate Arima model to be selected:

```
ggtsdisplay(un.diff)
```

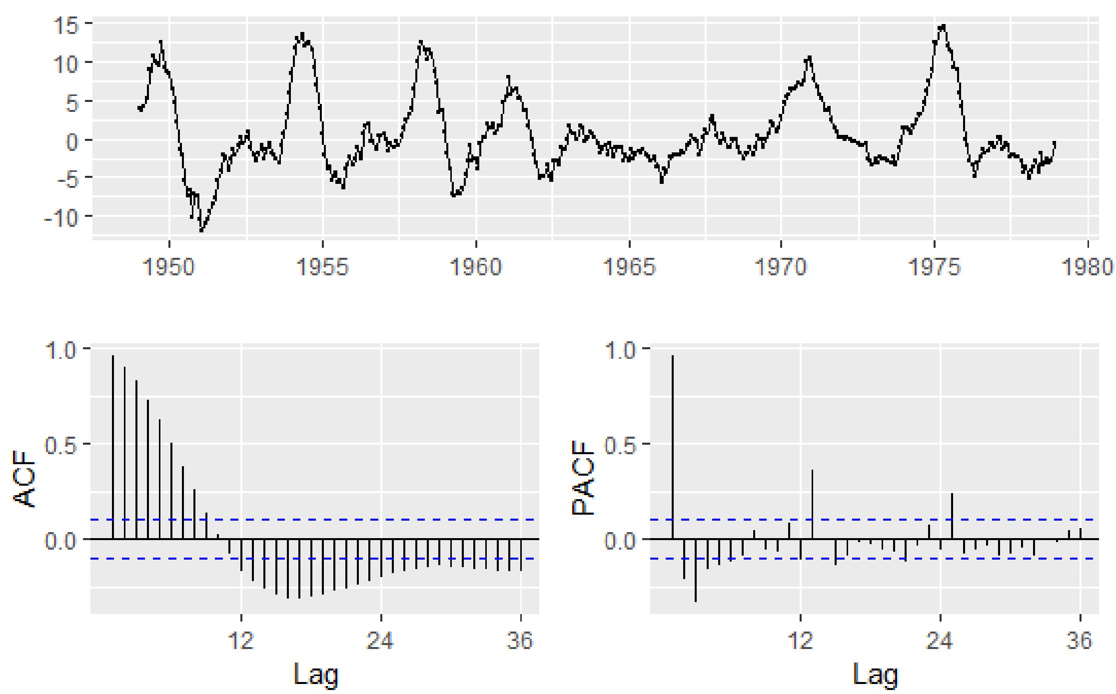


Fig. 3.2.1 ACF and PACF for the Differenced Unemployment Series

Observations

The data has seasonality so a (p,d,q)(P,D,Q)12 model is required.

For the non-seasonal part, there is no differencing term. The ACF has a significant spike at lag 1 and then tapers slowly, suggesting a non-seasonal MA(1) component. The PACF has 4 or 5 significant spikes before decreasing, suggesting an AR()component. An ARMA model of the order (5,0,1) or (4,0,1) could be appropriate.

For the seasonal part of the model there is one order of seasonal difference. There is decay in the seasonal lags of the ACF , with lag 12 negative and positive spikes at lag 12/13 in the PACF. This could suggest a seasonal MA(1) component, (Faculty.fuqua.duke.edu, 2019). The models chosen were:

(5,0,1)(0,1,1) and (4,0,1)(0,1,1)

Running each model and comparing the AICc:

```
un.ar1<-Arima(unemp.ts, order = c(5,0,1), seasonal = c(0,1,1), lambda = "auto
")
un.ar1$aicc

## [1] 1120.259

un.ar2<-Arima(unemp.ts, order = c(4,0,1), seasonal = c(0,1,1), lambda = "auto
")
un.ar2$aicc

## [1] 1119.08
```

Running auto.arima returned a model (3,0,1)(0,1,1).

```
un.ar3<-auto.arima(unemp.ts, stepwise = F, approximation = F, lambda = "auto"
)
summary(un.ar3)

## Series: unemp.ts
## ARIMA(3,0,1)(0,1,1)[12] with drift
## Box Cox transformation: lambda= 0.5096318
##
## Coefficients:
##          ar1      ar2      ar3      ma1     sma1   drift
##       1.6009  -0.4525  -0.1693  -0.5623  -0.7000  0.0607
## s.e.  0.1325   0.1868   0.0669   0.1300   0.0431  0.0322
##
## sigma^2 estimated as 1.233:  log likelihood=-551.08
## AIC=1116.16   AICc=1116.47   BIC=1143.36
##
## Training set error measures:
##                     ME     RMSE      MAE        MPE     MAPE      MASE
## Training set 0.1536276 20.01888 15.25453 -0.2105815 4.241731 0.2098189
##                     ACF1
## Training set -0.01472059
```

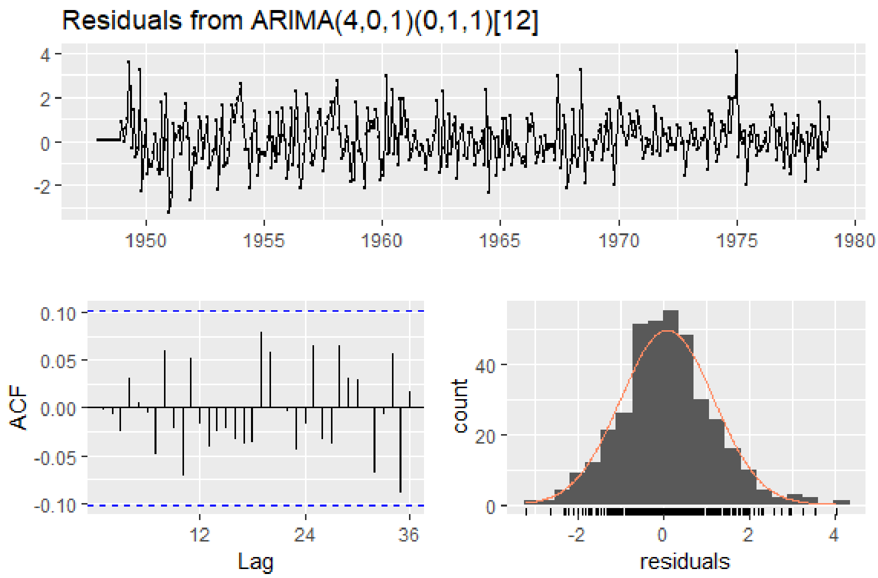
The best manually fitted model is -` ARIMA(4,0,0)(0,1,1) The model returned by auto.arima is - (3,0,1)(0,0,1)

These two models were investigated further.

Question 3.3

Comparing the residuals of both models:

```
checkresiduals(un.ar2)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(4,0,1)(0,1,1)[12]
## Q* = 13.696, df = 18, p-value = 0.7487
##
## Model df: 6.   Total lags used: 24
```

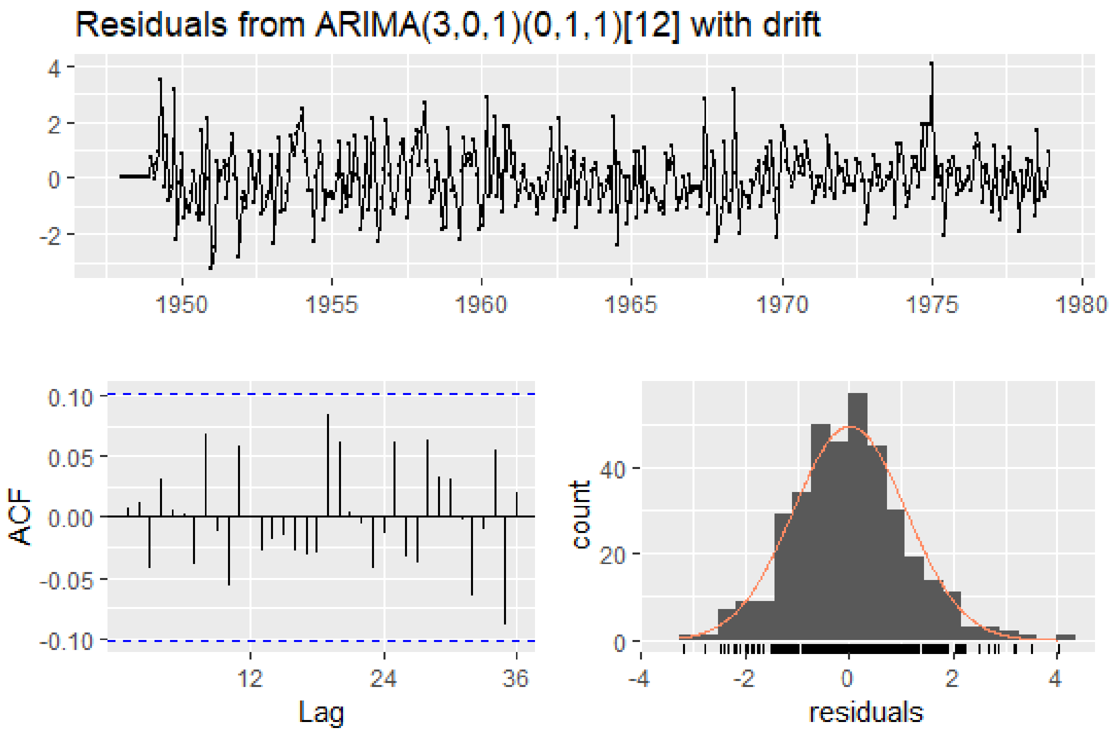
```
res.ar2<-residuals(un.ar2)
shapiro.test(res.ar2)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.ar2
## W = 0.99145, p-value = 0.03032
```

Fig. 3.3.1 Residuals for Manually Chosen Arima - Unemployment

Residuals for the auto.arima model:

```
checkresiduals(un.ar3)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,0,1)(0,1,1)[12] with drift
## Q* = 12.988, df = 18, p-value = 0.7923
##
## Model df: 6.   Total lags used: 24
```

```
res.ar3<-residuals(un.ar3)
shapiro.test(res.ar3)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.ar3
## W = 0.9916, p-value = 0.03332
```

Observations

The residuals for the manual and auto.arima() models are very similar, as there is just one non-seasonal AR order difference between them. For both, the ACF has no values outside the confidence bounds and the Ljung-Box test is well above 0.05, so the models picked up all the available information in the series. The mean and variance are constant, so both model residuals appear to be white noise.

The histogram appears to have a positive skew with the Shapiro-wilk test for normality rejecting the null hypothesis of normally distributed residuals.

As, auto.arima produces the lowest AICc it is therefore the preferred model.

Question 3.4

The forecast for the preferred model is shown in Fig 3.4.1.

```
fc.un1<-forecast(un.ar3, h = 12, biasadj = TRUE)
autoplot(fc.un1) + ggtitle("Fig. 3.4.1 ARIMA Unemployment Forecast")+ xlab("Y
ear") + ylab("000s")
```

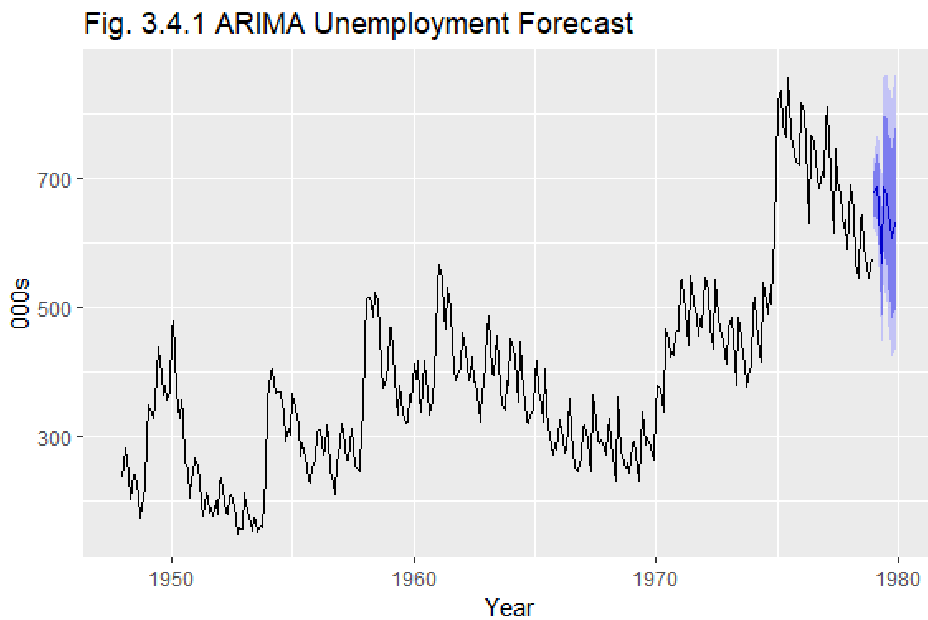


Fig. 3.4.1 ARIMA Model Unemployment Forecast

Question 3.5

Running ets() on the series based on the maximum likelihood returns a model associated with the Holt Winters Damped Additive method:

```
un.ets<-ets(unemp.ts)
summary(un.ets)

## ETS(A,Ad,A)
##
## Call:
##  ets(y = unemp.ts)
##
##   Smoothing parameters:
##     alpha = 0.8481
##     beta  = 0.1214
##     gamma = 0.1519
##     phi   = 0.8001
##
##   Initial states:
##     l = 178.9834
##     b = 4.0007
```

```
##     s = -24.0306 -25.0901 -46.2798 -28.6777 -19.0167 9.0941
##            46.23 -27.8982 -7.9895 28.2684 55.4583 39.9317
## 
##   sigma:  23.9484
## 
##      AIC     AICc      BIC 
## 4583.300 4585.238 4653.840 
## 
## Training set error measures:
##                     ME     RMSE      MAE     MPE    MAPE      MASE
## Training set 0.8454685 23.39484 18.42953 0.10783 5.23666 0.2534894
##                   ACF1
## Training set 0.2007737
```

Plotting this model shows the clear seasonality seeming to increase towards the end of the series:

```
autoplot(un.ets)
```

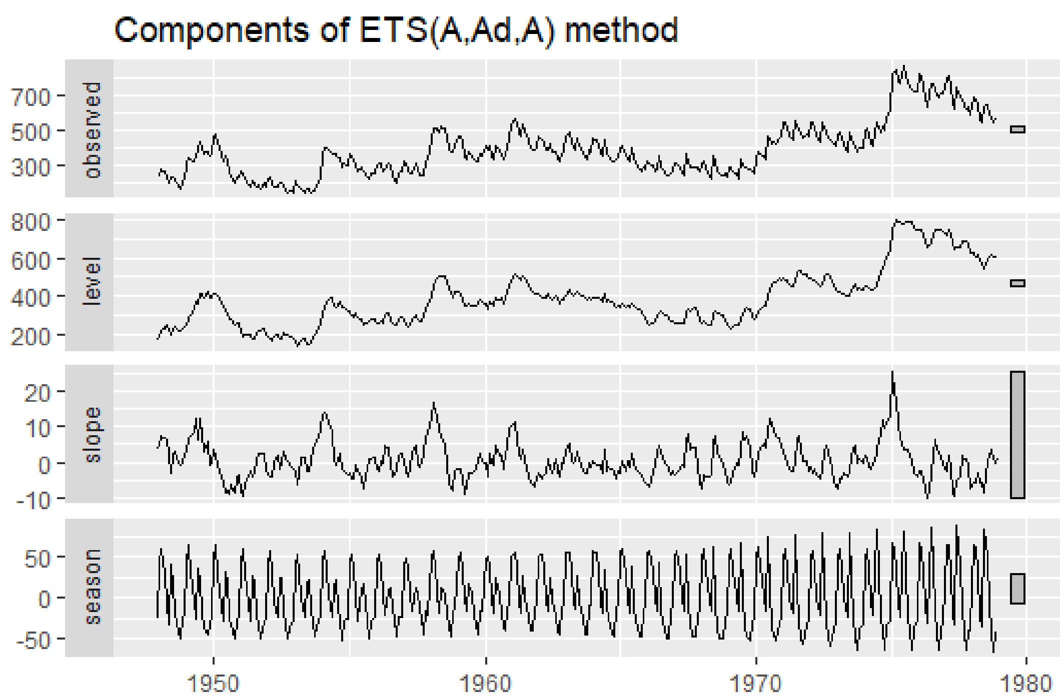
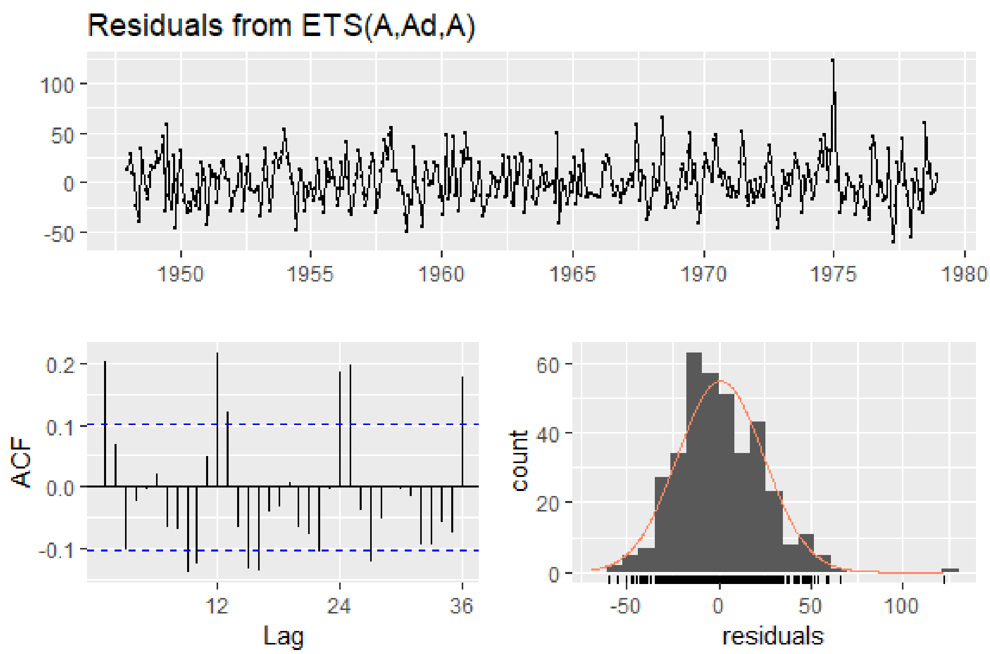


Fig. 3.5.1 ETS(A,Ad,A) for Unemployment

Question 3.6

```
checkresiduals(un.ets)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,Ad,A)
## Q* = 101.86, df = 7, p-value < 2.2e-16
##
## Model df: 17.   Total lags used: 24
```

```
res.un<-residuals(un.ets)
shapiro.test(res.un)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.un
## W = 0.97917, p-value = 3.336e-05
```

Fig. 3.5.2 Residuals ETS - Unemployment

Observations

There are quite a few significant spikes on the seasonal lags and the Ljung Box test is very small, rejecting the null hypothesis of no autocorrelation. The ETS model has not captured all the information in the data. The residuals plot appears to have a constant mean, but the variance may be increasing towards the end of the series. There is also one large spike from the increase in the mid 1970s which is also shown by the significant postive skew in the histogram which is therefore not normally distributed.

Question 3.7

Forecast for the ETS model:

```
fc.un2<-(forecast(un.ets, h = 12))
autoplot(fc.un2) + xlab("Year") + ylab("000's")
```

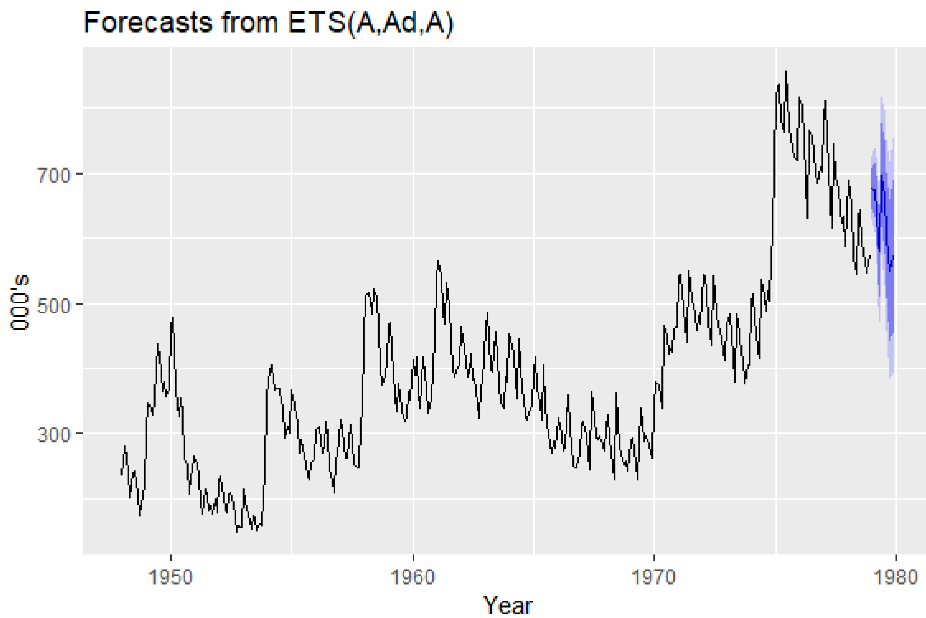
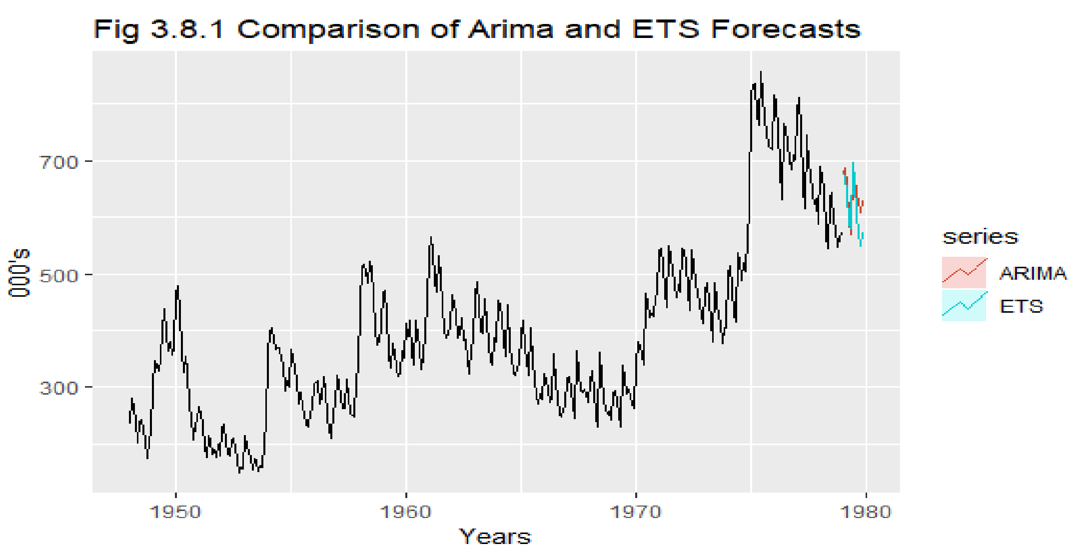


Fig 3.7.1 Forecasts for ETS(A,Ad,A)

Question 3.8

The models cannot be compared using the AICc since they are in completely different classes, so the maximum likelihood is calculated differently. The two forecasts are plotted in Fig. 3.8.1

```
autoplot(unemp.ts)+ autolayer(fc.un1, PI = F, series = "ARIMA") + autolayer(f
c.un2, PI = F, series = "ETS") + ylab("000's") + xlab("Years") + ggtitle("Fig
3.8.1 Comparison of Arima and ETS Forecasts")
```



Comparing the models based on the lowest RMSE against the training set, the Arima model fits the data better:

Arima 20.019 ETS 23.395

```r
summary(un.ar3)
```

```
## Series: unemp.ts
## ARIMA(3,0,1)(0,1,1)[12] with drift
## Box Cox transformation: lambda= 0.5096318
##
## Coefficients:
##          ar1      ar2      ar3      ma1     sma1   drift
##       1.6009  -0.4525  -0.1693  -0.5623  -0.7000  0.0607
## s.e.  0.1325   0.1868   0.0669   0.1300   0.0431  0.0322
##
## sigma^2 estimated as 1.233:  log likelihood=-551.08
## AIC=1116.16   AICc=1116.47   BIC=1143.36
##
## Training set error measures:
##                     ME     RMSE      MAE        MPE     MAPE      MASE
## Training set 0.1536276 20.01888 15.25453 -0.2105815 4.241731 0.2098189
##                     ACF1
## Training set -0.01472059
```

```r
summary(un.ets)
```

```
## ETS(A,Ad,A)
##
## Call:
##  ets(y = unemp.ts)
##
##   Smoothing parameters:
##     alpha = 0.8481
##     beta  = 0.1214
##     gamma = 0.1519
##     phi   = 0.8001
##
##   Initial states:
##     l = 178.9834
##     b = 4.0007
##     s = -24.0306 -25.0901 -46.2798 -28.6777 -19.0167 9.0941
##            46.23 -27.8982 -7.9895 28.2684 55.4583 39.9317
##
##   sigma:  23.9484
##
##      AIC     AICc      BIC
## 4583.300 4585.238 4653.840
##
## Training set error measures:
##                     ME     RMSE      MAE     MPE    MAPE      MASE
## Training set 0.8454685 23.39484 18.42953 0.10783 5.23666 0.2534894
```

```
##                    ACF1
## Training set 0.2007737
```

Comparing against the training set alone is not the best way to choose between the performance of the models. A better method would be to use a training and test set.

```
un.train<-window(unemp.ts, start = c(1949,1), end = c(1972,12))
un.test<-window(unemp.ts, start = c(1973,1), end = c(1978,12))
h<-length(un.test)
un.fit1 <- auto.arima(un.train)
un.fit2<-ets(un.train)
un.fc1<-forecast(un.fit1, h = h)
un.fc2<-forecast(un.fit2, h = h)
un.acc1<-accuracy(un.fc1,un.test)
un.acc2<-accuracy(un.fc2,un.test)
un.acc1
```

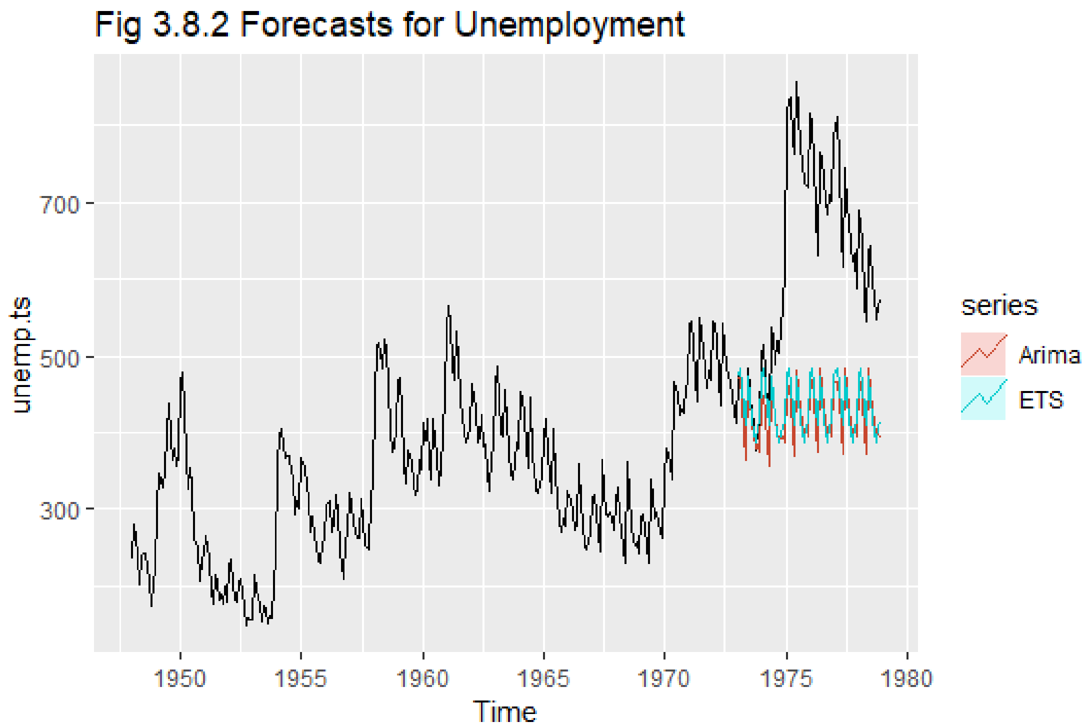
```
##                      ME      RMSE       MAE         MPE      MAPE
## Training set   0.658541  19.46227  15.23974 -0.01033995  4.855106
## Test set     200.124653 235.16569 200.31495 29.11105950 29.156840
##                   MASE        ACF1 Theil's U
## Training set 0.2402385 -0.03420667        NA
## Test set     3.1577552  0.95743465  3.295376
```

```
un.acc2
```

```
##                       ME      RMSE       MAE         MPE      MAPE
## Training set   0.3074896  22.10862  16.91586 -0.07409406  5.239738
## Test set     189.3740649 228.78997 191.59914 27.05759733 27.608904
##                   MASE       ACF1 Theil's U
## Training set 0.2666607 0.02159396        NA
## Test set     3.0203596 0.95348279  3.195929
```

The result of this is that this time the ETS model performs better. However, the large spike from January 1975 onwards included in the test set, causes the models to underforecast significantly as shown in Fig.3.8.2:

```
autoplot(unemp.ts) + autolayer(un.fc1, PI = F, series = "Arima") + autolayer(
un.fc2, PI = F, series = "ETS") + ggtitle("Fig 3.8.2 Forecasts for Unemployme
nt")
```



With such a volatile data series, a shorter time-span may be more useful when making shorter term forecasts, as the series is in effect two separate ones, split by the significant jump in the mid 1970's.

Taking the last 48 data points from January 1975 and running a time series cross validation:

```
example<-window(unemp.ts, start = c(1975,1))

f.arima <- function(x, h) {
  forecast(auto.arima(x), h=h)
}
f.ets <- function(x, h) {
  forecast(ets(x), h = h)
}

tsv.ac1<-tsCV(example,f.arima, h = 1)
tsv.ac2<-tsCV(example,f.ets, h = 1)

mean(tsv.ac1^2, na.rm=TRUE)

## [1] 1846.71

mean(tsv.ac2^2, na.rm=TRUE)

## [1] 3242.278
```

Observations

Neither the Arima nor ETS model performed well on the test set since the large increase from January 1975 which fell within the test set period was not modelled by the training data. It might be better to consider the series as two separate datasets as volatile and unpredictable series can be difficult to forecast, and it may be better to use a shorter time period.

The forecasts produced for the following 12 years appeared reasonable. A one step time series cross validation performed on a subset of the data from January 1975 indicated that the ARIMA model would be the better forecasting model for this series.

Question 4.1

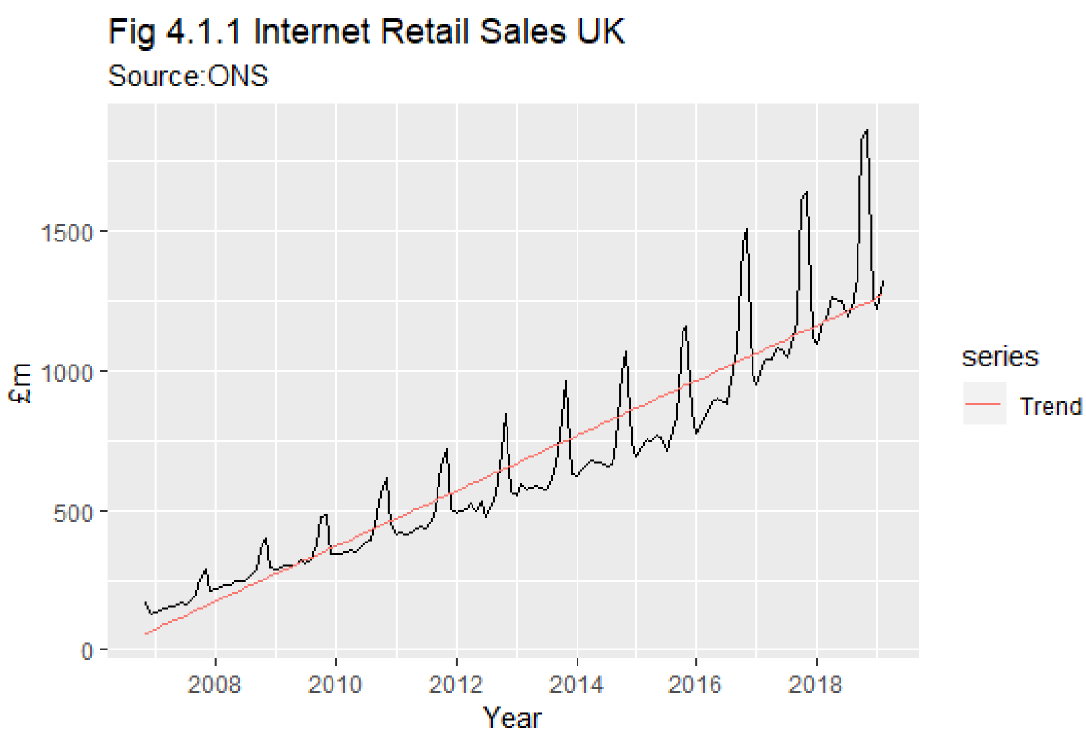
Exploratory Plots

The data is for Internet Retail Sales from the following souce:

https://www.ons.gov.uk/businessindustryandtrade/retailindustry/timeseries/je2j/drsi

(Ons.gov.uk, 2019)

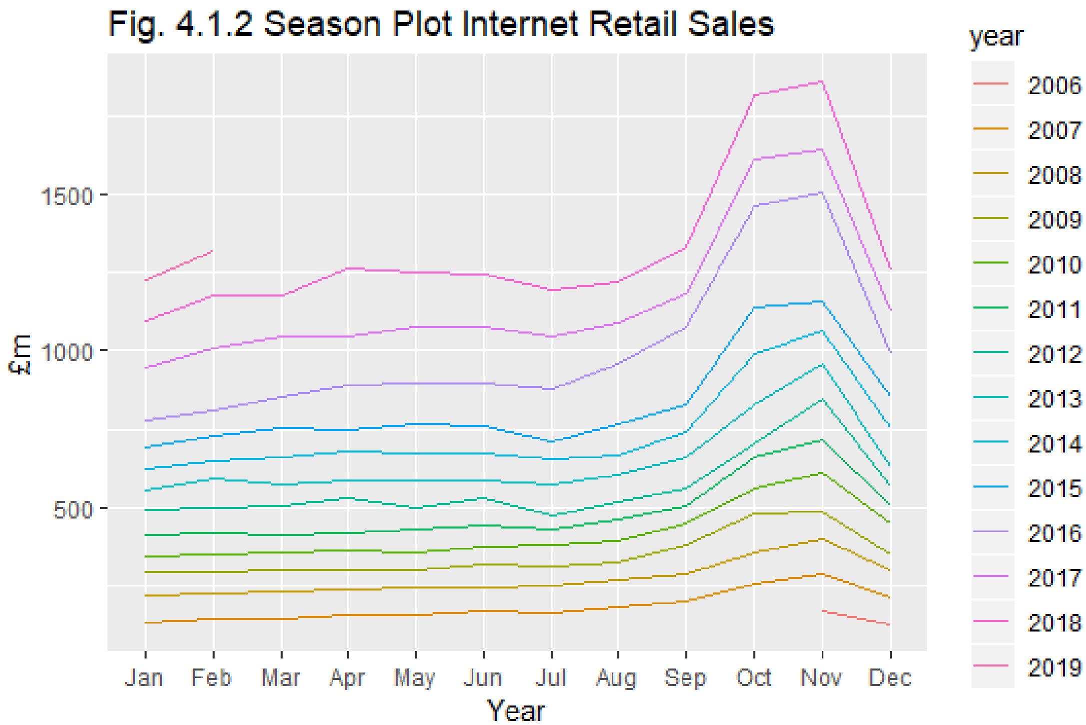
```
internet.data <- readxl::read_excel("C:/Users/Imy Hull/Desktop/internetsales.
xlsx")
internet<- ts(internet.data, start=c(2006,11), end = c(2019,2), frequency = 1
2)
length(internet)

## [1] 148

int.line<-tslm(internet~trend)

autoplot(internet) + autolayer(int.line$fitted.values, series = "Trend")+ ggt
itle("Fig 4.1.1 Internet Retail Sales UK", subtitle = "Source:ONS") + xlab("Y
ear") + ylab("£m")
```



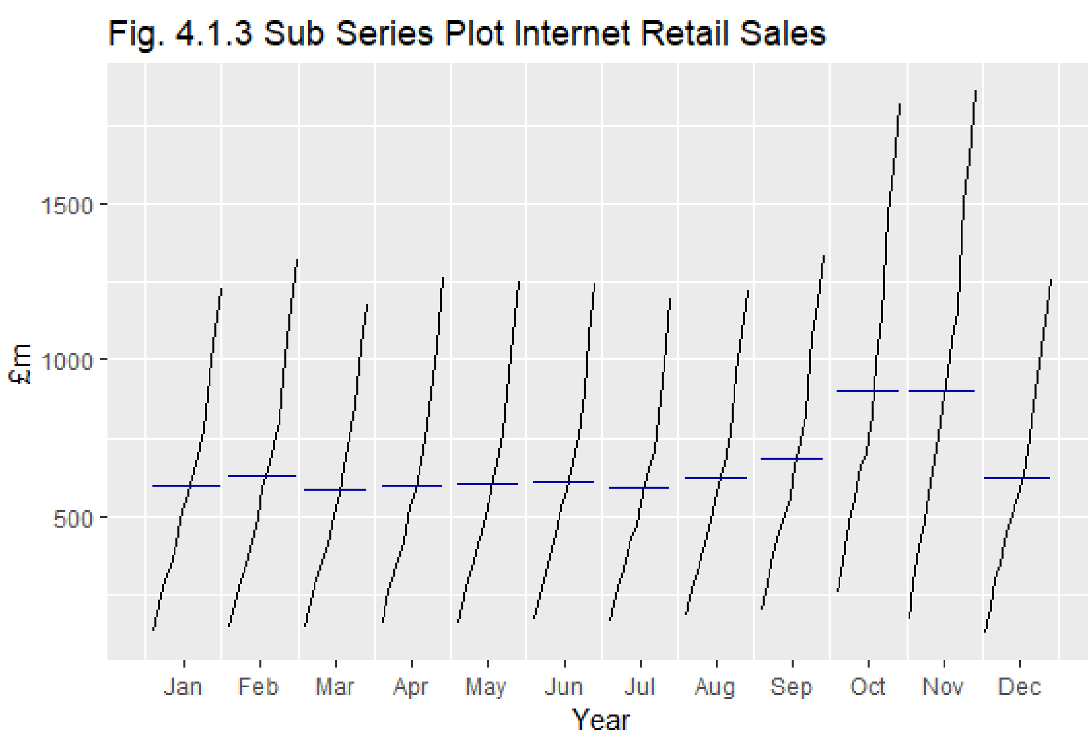

```
int.line$coefficients

## (Intercept)       trend
##   50.263293    8.233107
```

Growing by £98.797284 per year.

```r
ggseasonplot(internet) + xlab("Year") + ylab("£m")+
  ggtitle("Fig. 4.1.2 Season Plot Internet Retail Sales")
```

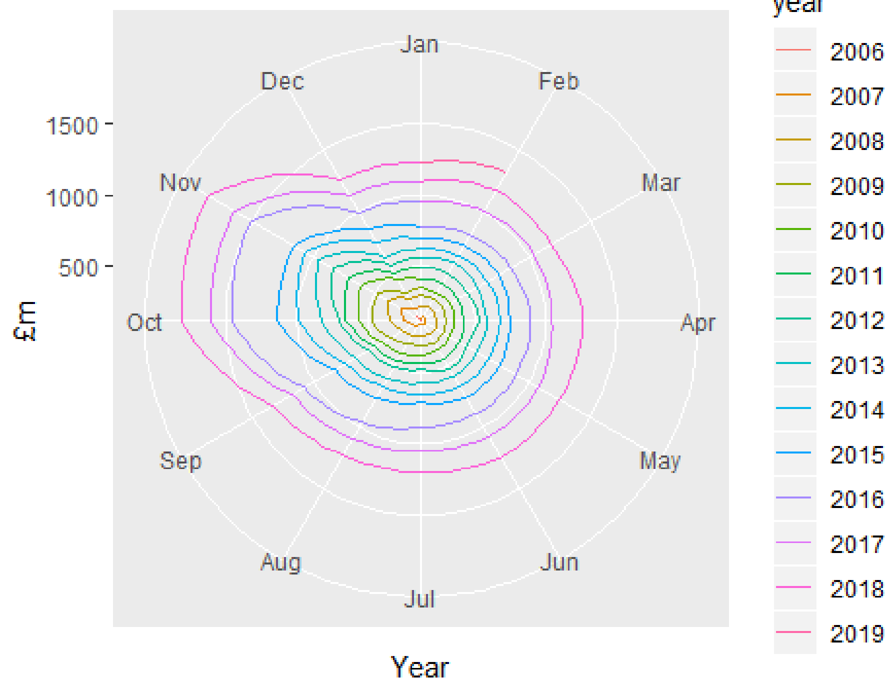



```r
ggsubseriesplot(internet)+ xlab("Year") + ylab("£m")+
  ggtitle("Fig. 4.1.3 Sub Series Plot Internet Retail Sales")
```




```r
ggseasonplot(internet, polar = TRUE) + xlab("Year") + ylab("£m")+
  ggtitle("Fig. 4.1.4 Season Plot Internet")
```

Fig. 4.1.4 Season Plot Internet




```r
gglagplot(internet)+ xlab("Year") + ylab("£m")
```

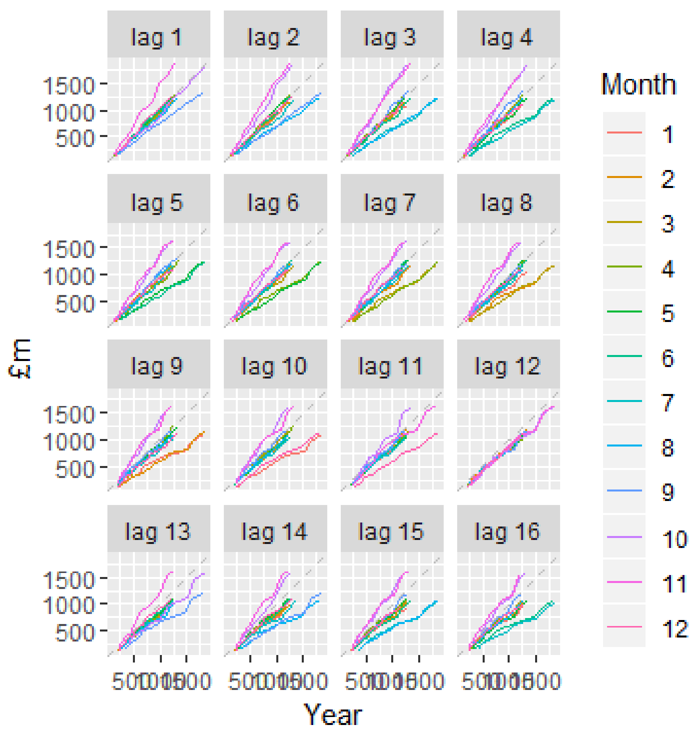


Fig. 4.1.5 Lag Plot - Internet Sales

```r
ggAcf(internet)
```

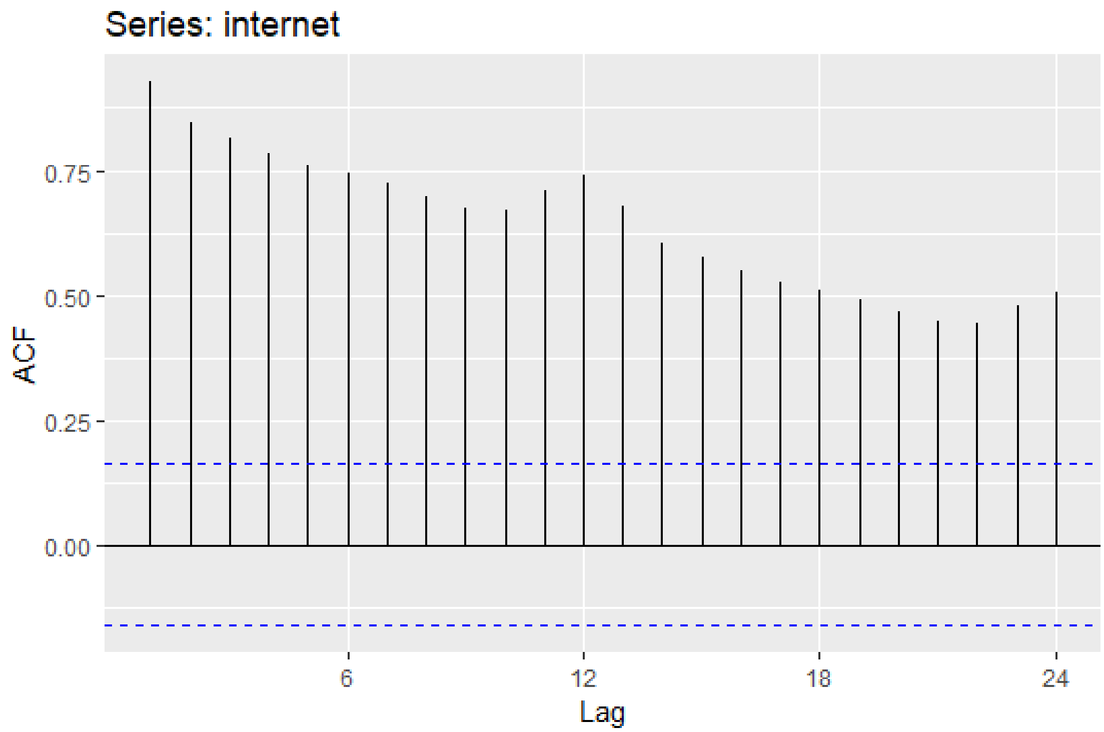


Fig. 4.1.6 ACF - Internet Sales

```
autoplot(stl(internet[,1], s.window = 7))
```

4.1.7 Decompostion - Internet Sales

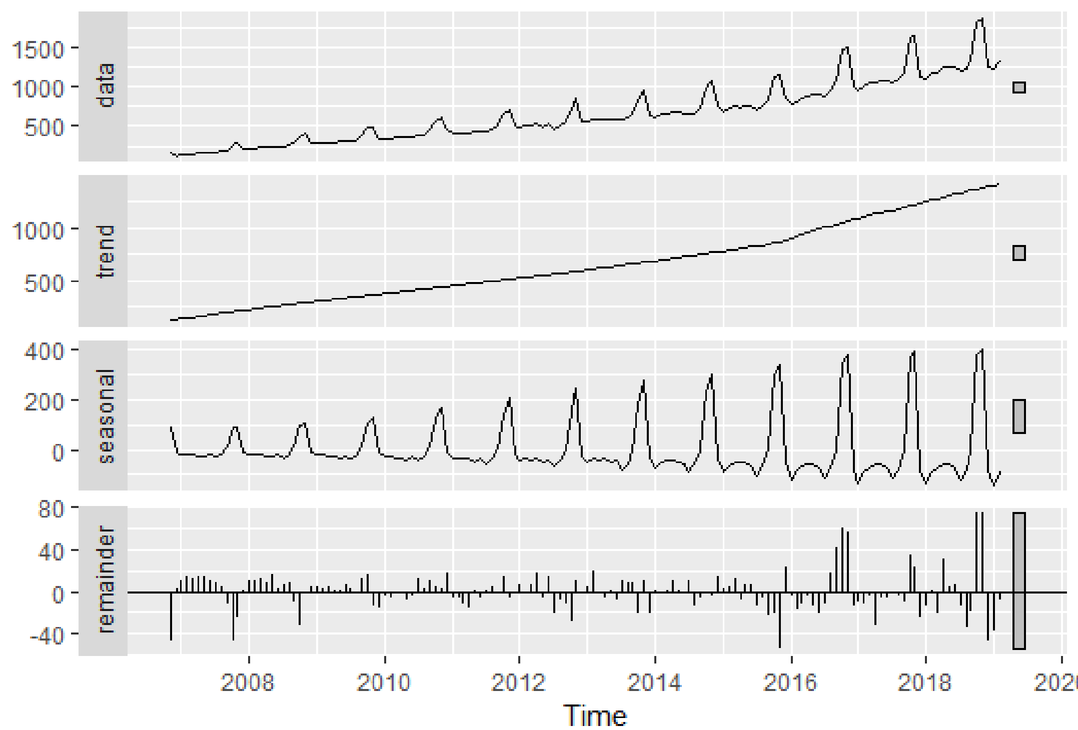


Fig. 4.1.7 Decompostion - Internet Sales

Observations

The series has a linear upward trend with a clear seasonal pattern and variance increasing with the level of the series, indicating a multiplicative seasonality.

Looking at the season plot, sub-series and polar plots, the highest sales occur in November. Since 2015, October sales have also increased, resulting in 'plateau' appearance between October and November in later years where sales in both months are similar. The overall rising level of sales in November reflects the increased popularity of 'Black Friday' and 'Cyber Monday' and Halloween has become an increasingly important sales opportunity for retail in October. (We Make Websites, 2019)

The seasonality is clearly shown on the lagplot at lag 12 and in the ACF plot. The seasonal decompostion also shows the upwards trend and clear seasonal pattern increasing over the series. There isn't much pattern in the remainder apart from some spikes towards the end of the series, but the scale is relatively small.

Question 4.2

Seasonal Naive

Splitting the series into a training and test set:

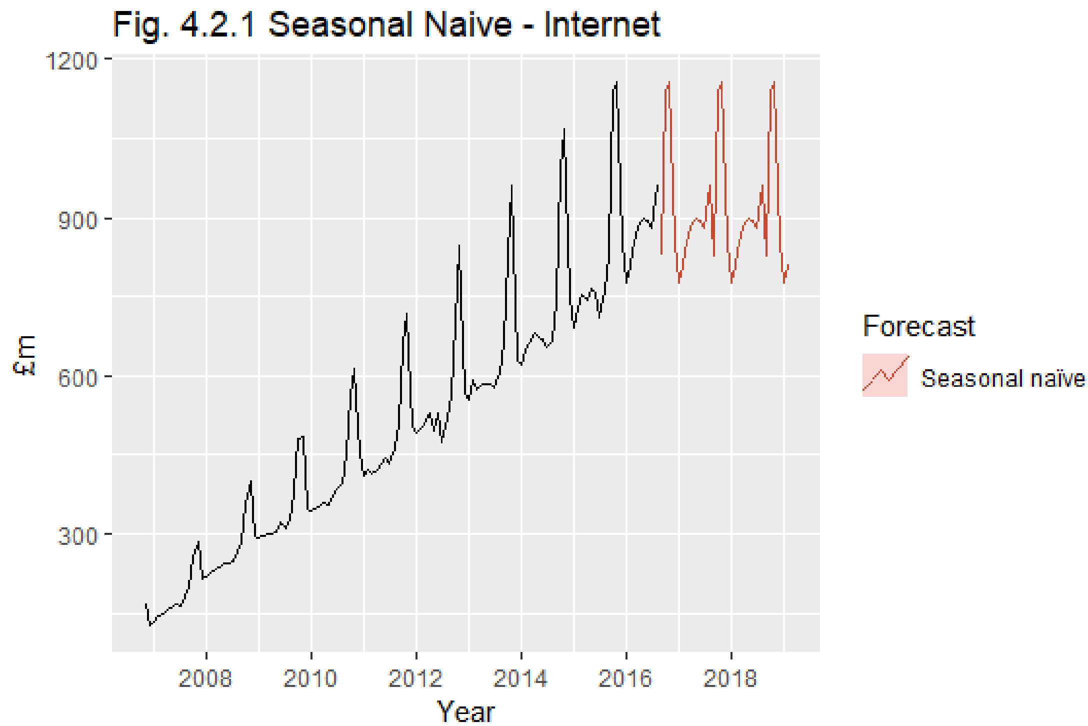
```
int.train<-window(internet, end = c(2016,8))
int.test<-window(internet, start = c(2016,9))
h<-length(int.test)
```

Performing seasonal naive method to forecast:

```
int.snaive <- snaive(int.train, h = h)
```

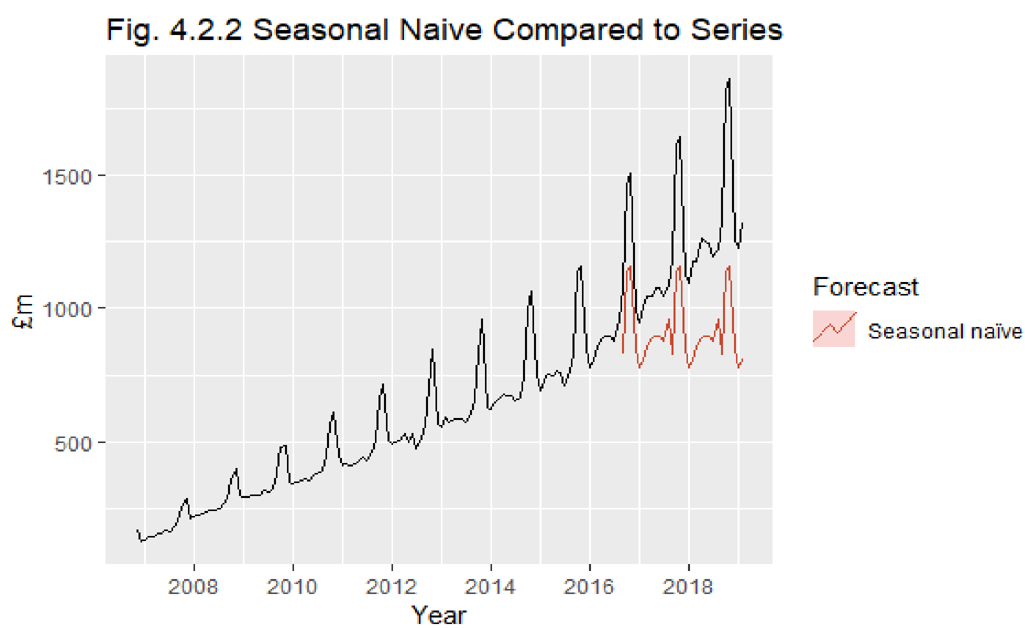
The forecast is plotted in Fig. 4.2.1 which shows that it captures the seasonal pattern:

```
autoplot(int.train) +  autolayer(int.snaive, series="Seasonal naïve", PI=FALS
E) + xlab("Year") + ylab("£m") + ggtitle("Fig. 4.2.1 Seasonal Naive - Interne
t") +
  guides (colour=guide_legend(title="Forecast"))
```



Comparing the forecast to the actual data in the test set, the rising trend and increasing seasonal variance is not captured well by the Seasonal Naive Method.

```
autoplot(internet) +
  autolayer(int.snaive, series="Seasonal naïve", PI=FALSE) +
  xlab("Year") + ylab("£m") +
  ggtitle("Fig. 4.2.2 Seasonal Naive Compared to Series") +
  guides (colour=guide_legend(title="Forecast"))
```



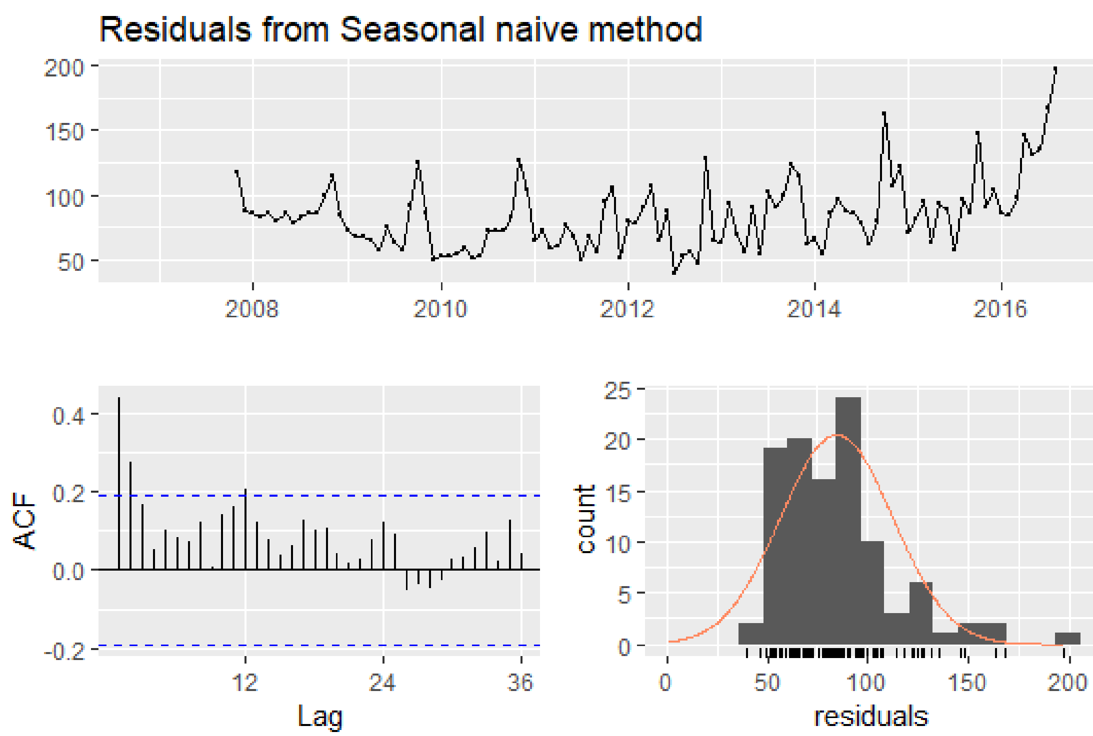
The accuracy of the method is shown by the measures below:

```r
int.acc1<-accuracy(int.snaive,int.test)
int.acc1
```

```
##                    ME      RMSE       MAE      MPE     MAPE     MASE
## Training set  84.37453  88.81308  84.37453 17.18630 17.18630 1.000000
## Test set     330.28667 361.48558 330.28667 25.60166 25.60166 3.914531
##                   ACF1 Theil's U
## Training set 0.4402204        NA
## Test set     0.6902082  1.749058
```

The residuals can be examined:

```r
checkresiduals(int.snaive)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Seasonal naive method
## Q* = 59.505, df = 24, p-value = 7.501e-05
##
## Model df: 0.   Total lags used: 24
```

```r
res.int1<-residuals(int.snaive)
shapiro.test(res.int1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.int1
## W = 0.91087, p-value = 2.68e-06
```

There are just three spikes on the ACF which are outside the confidence bounds, so the method has captured a fair amount of the information in the data but there is still some autocorrelation. The Ljung-Box test statistic is very small, although this represents correlation from the whole series which is quite long at 148 data points. There is non-constant mean and variance and the residuals are not normally distributed with a large positive skew to the histogram.

Question 4.3

Holt Winters Seasonal Methods

Holt Winters Additive and Multiplicative Seasonal Methods were used, and the accuracy is shown in Table 1.8:

```
int.hwa<-hw(int.train, h=h,seasonal = "additive")
int.hwda<-hw(int.train, h=h, seasonal= "additive", damped = TRUE)
int.hwm<-hw(int.train, h = h, seasonal = "multiplicative")
int.hwdm<-hw(int.train, h = h, seasonal = "multiplicative", damped = TRUE)

int.acc2<-round(accuracy(int.hwa,int.test),1)
int.acc3<-round(accuracy(int.hwda,int.test),1)
int.acc4<-round(accuracy(int.hwm, int.test),1)
int.acc5<-round(accuracy(int.hwdm,int.test),1)
library("knitr")
a.table <- rbind(int.acc2,int.acc3,int.acc4,int.acc5)
row.names(a.table)<-c("HWA Train", "HWA Test",
                      "HW(D)A Train", "HW(D)A Test", "HWM Train", "HWM Test",
"HW(D)M Train", "HW(D)M Test")
a.table<-as.data.frame(a.table)
kable(a.table, caption= "Table 1.8 Forecast accuracy of Holt Winters Methods"
,digits = 4 )
```

Table 1.8 Forecast accuracy of Holt Winters Methods

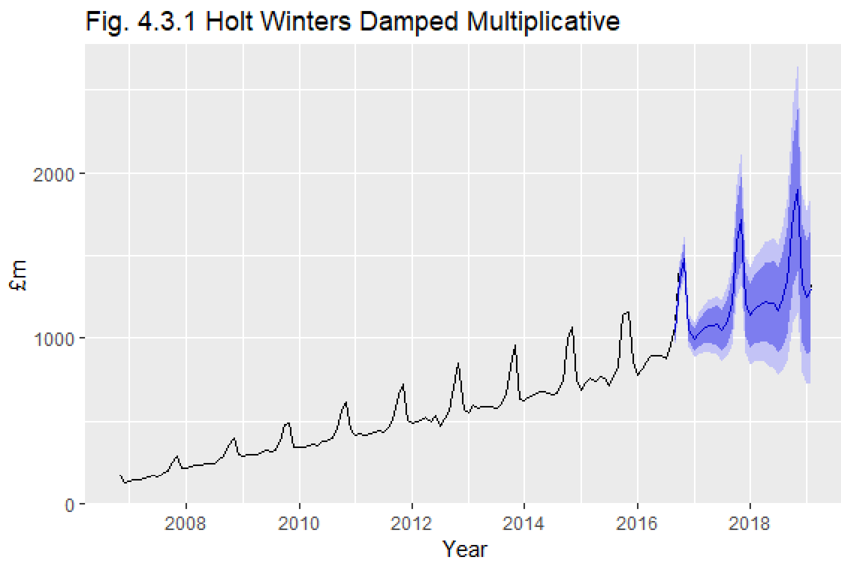
	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
HWA Train	2.4	31.0	21.6	-0.1	5.7	0.3	0.2	NA
HWA Test	101.9	133.7	102.0	7.5	7.5	1.2	0.4	0.6
HW(D)A Train	8.1	32.0	22.5	1.4	6.0	0.3	0.3	NA
HW(D)A Test	108.0	141.2	108.4	7.9	7.9	1.3	0.4	0.7
HWM Train	2.3	18.4	13.6	-0.2	2.8	0.2	0.2	NA
HWM Test	78.1	89.7	78.1	6.1	6.1	0.9	0.3	0.5
HW(D)M Train	2.6	17.7	12.6	0.4	2.5	0.1	-0.2	NA
HW(D)M Test	-5.5	42.1	34.3	-0.6	2.7	0.4	0.3	0.2

Table 1.8 Accuracy Measures for HW Models – Internet Sales

From the table, the Holt Winters Damped Multiplicative has the lowest RMSE and is the preferred model so far.

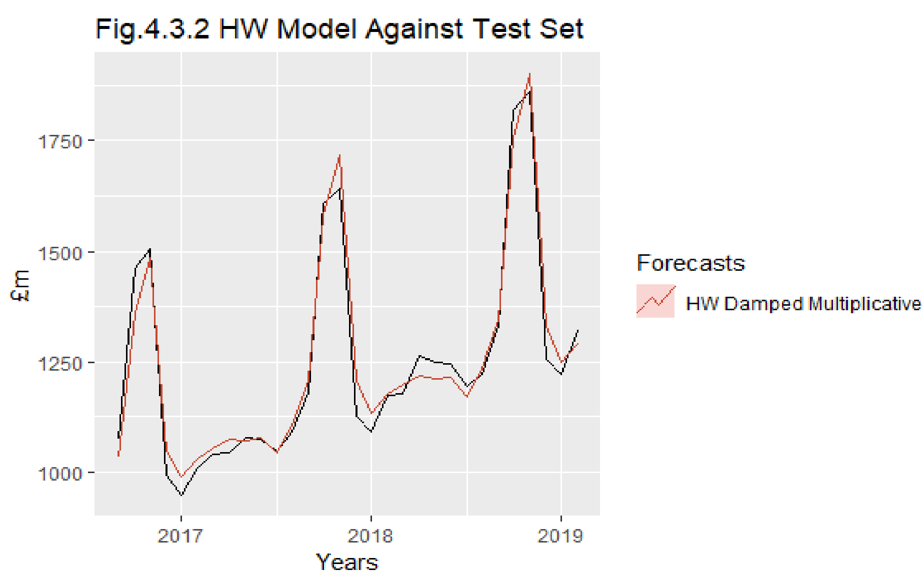
The forecast is plotted in Fig. 4.3.1.

```
autoplot(internet) +
  autolayer(int.hwdm) +
  ggtitle("Fig. 4.3.1 Holt Winters Damped Multiplicative") +  ylab("£m") + xl
ab("Year")
```



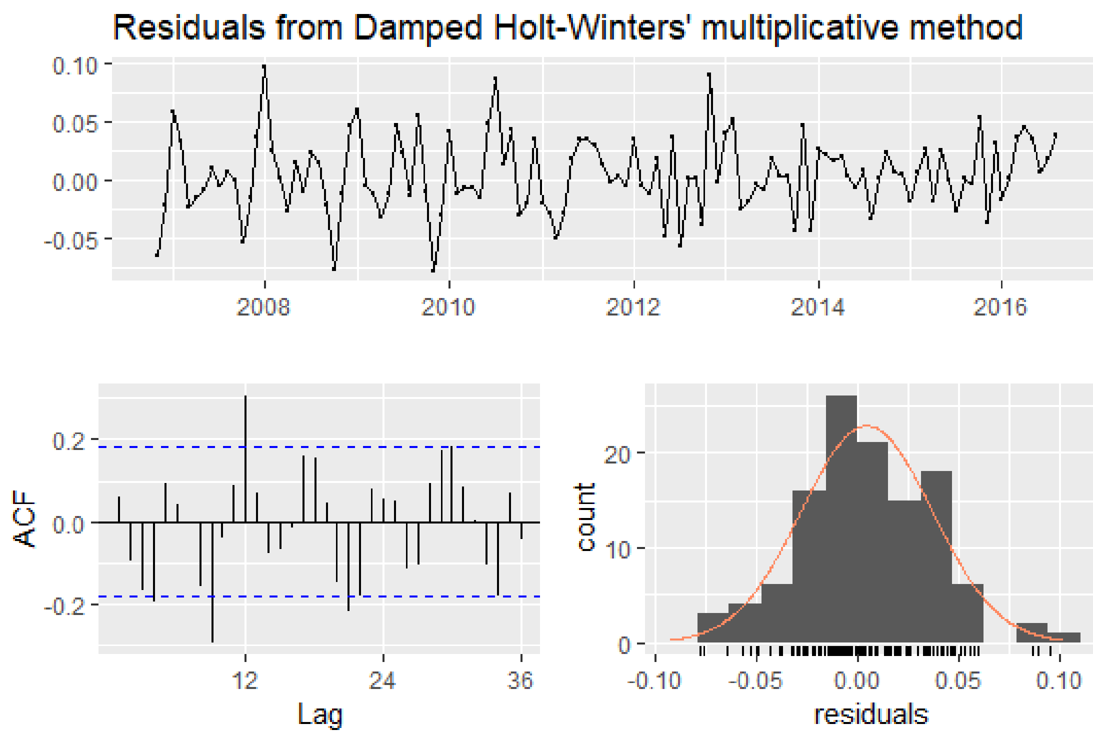
The forecast is shown against the test set in Fig.4.3.2:

```
autoplot(int.test) +
  autolayer(int.hwdm, series = "HW Damped Multiplicative", PI = FALSE)+ xlab(
"Years") + ylab("£m") + ggtitle("Fig.4.3.2 HW Model Against Test Set") + guid
es(colour = guide_legend(title = "Forecasts"))
```



The HW method appears to follow the test well. Examining the residuals:

```
checkresiduals(int.hwdm)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Damped Holt-Winters' multiplicative method
## Q* = 65.019, df = 7, p-value = 1.491e-11
##
## Model df: 17.   Total lags used: 24

res.int2<-residuals(int.hwdm)
shapiro.test(res.int2)

##
##  Shapiro-Wilk normality test
##
## data:  res.int2
## W = 0.99043, p-value = 0.5855
```

There are two significant lags outside the confidence bounds and a several others just outside, so the method didn't capture all the information in the data. The Ljung-Box test statistic is also very small supporting autocorrelation over the series, although this is a long series. The mean of the residuals is near zero and constant and the histogram appears normally distributed.

The Holt Winters Damped Multiplicative is the preferred method at this stage.

Question 4.4

STL and ETS()

STL was used to forecast the series which returned an ETS model of additive errors, additive damped trend and no seasonality.

```
int.stlf<-stlf(int.train, h = h, s.window = 7, lambda = "auto", biasadj = T)
autoplot(int.stlf)
```

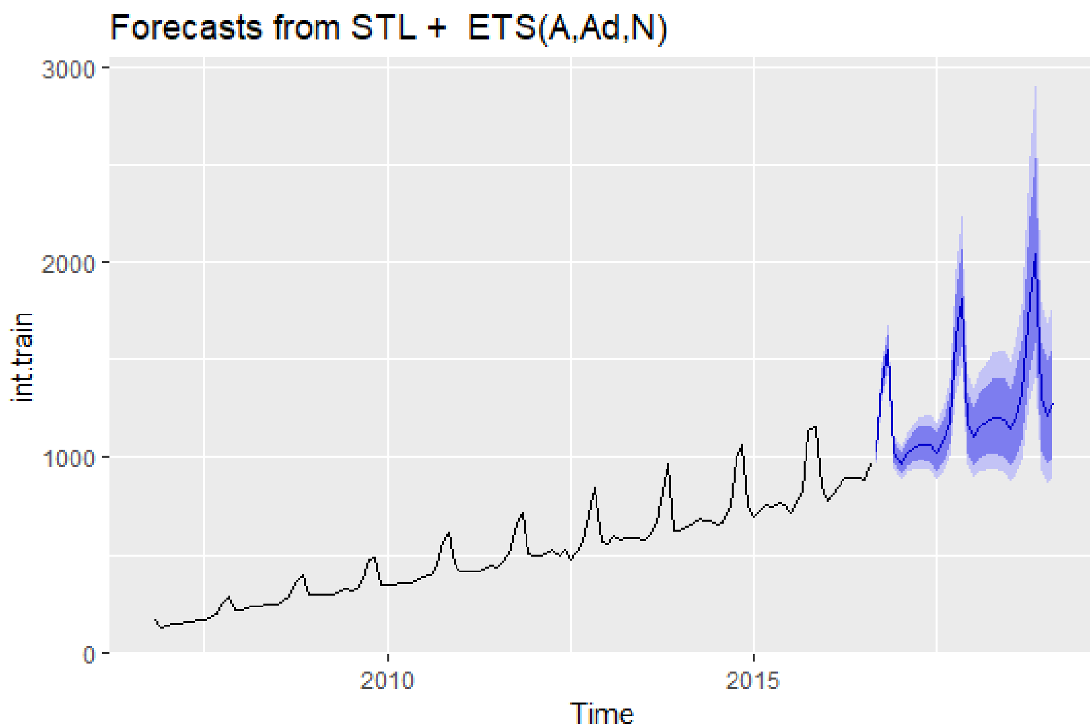
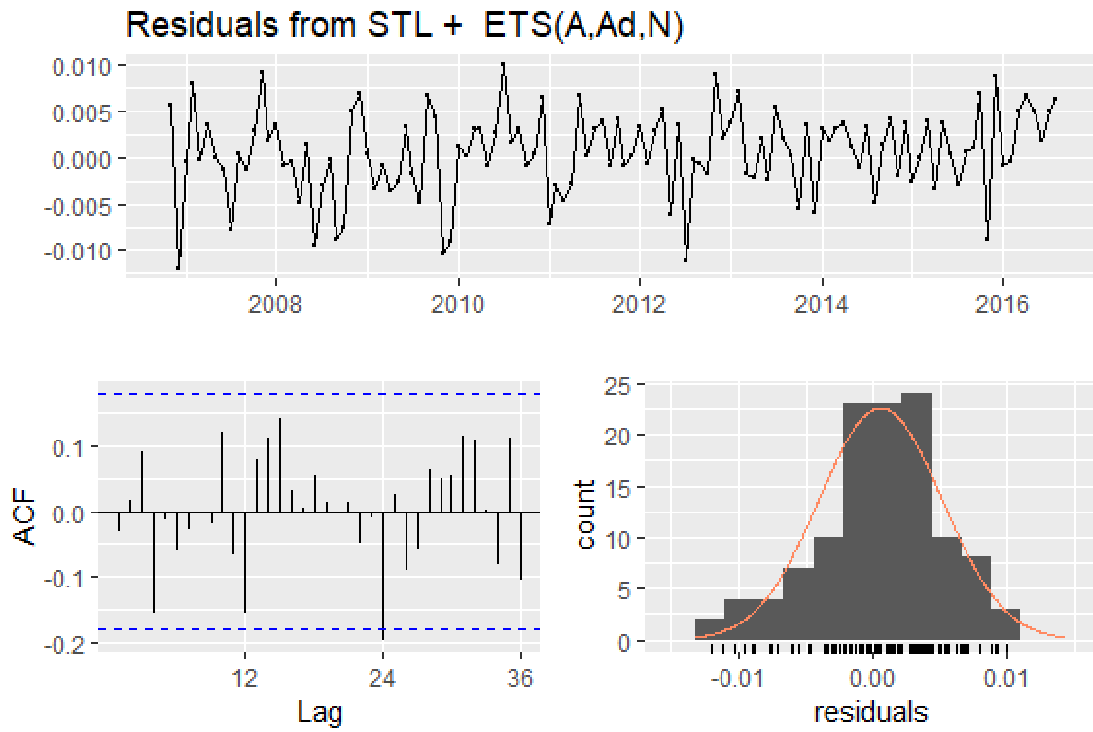


Fig. 4.4.1 STL Forecast

The residuals showed that there is one lag outside the confidence bounds and the Ljung-Box test p-value suggests that there is no autocorrelation left in the data. The residuals are not normally distributed with a negative skew and the mean oscillates around zero.

```
checkresiduals(int.stlf)

## Warning in checkresiduals(int.stlf): The fitted degrees of freedom is base
d
## on the model used for the seasonally adjusted data.
```




```
##
##  Ljung-Box test
##
## data:  Residuals from STL +  ETS(A,Ad,N)
## Q* = 22.683, df = 19, p-value = 0.2516
##
## Model df: 5.   Total lags used: 24
```

```
res.int3<-residuals(int.stlf)
shapiro.test(res.int3)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.int3
## W = 0.97611, p-value = 0.03349
```

The accuracy measures do not improve on the previous method.

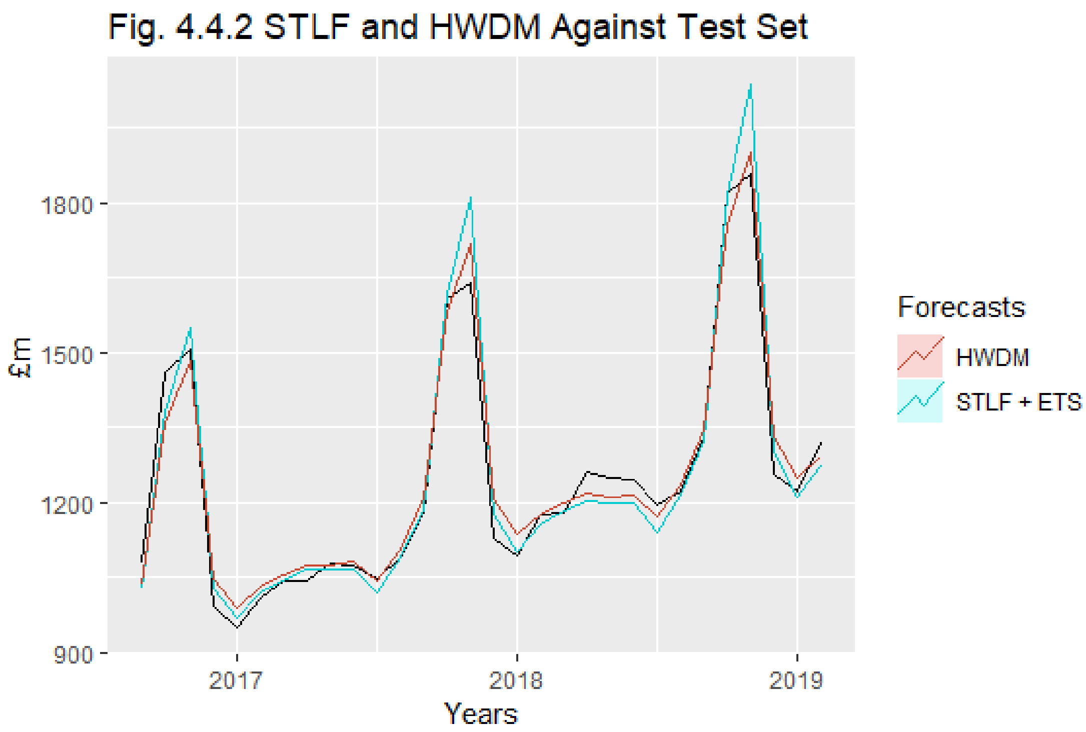
```
int.acc6<-round(accuracy(int.stlf,int.test),1)
int.acc6
```

```
##                ME RMSE  MAE  MPE MAPE MASE ACF1 Theil's U
## Training set  2.8 15.8 10.6  0.3  1.9  0.1 -0.2        NA
## Test set     -4.1 55.7 36.0 -0.1  2.7  0.4  0.3       0.2
```

Plotting the forecasts of both the STLF and HW Damped Multiplicative against the test set:

```
autoplot(int.test) +
  autolayer(int.stlf, series = "STLF + ETS", PI = FALSE)+ autolayer(int.hwdm,
PI = F, series = "HWDM") + xlab("Years") + ylab("£m") + ggtitle("Fig. 4.4.2 S
```

```
TLF and HWDM Against Test Set") + guides(colour = guide_legend(title = "Forec
asts"))
```



Both these models seem to capture the data in the series quite well although the STLF appears to overforecast on the peaks.

Question 4.5

ETS Model

Using ETS() on the training set, returns a model with multiplicative error, additive trend and multiplicative seasonality.

```
int.ets<-ets(int.train)
summary(int.ets)
```

```
## ETS(M,A,M)
##
## Call:
##  ets(y = int.train)
##
##   Smoothing parameters:
##     alpha = 0.3795
##     beta  = 0.0425
##     gamma = 2e-04
##
##   Initial states:
##     l = 122.4654
##     b = 8.0495
```

```
##     s = 1.2692 1.0043 0.9297 0.8936 0.9281 0.9227
##            0.935 0.931 0.9329 0.9191 0.9634 1.371
##
##   sigma:  0.0321
##
##      AIC     AICc      BIC
## 1212.255 1218.375 1259.357
##
## Training set error measures:
##                    ME     RMSE      MAE         MPE     MAPE      MASE
## Training set 1.580125 18.57932 12.35272 -0.04171495 2.297639 0.1464035
##                     ACF1
## Training set -0.03271944
```

Plotting the ETS components, it shows the seasonal pattern, slope and level produced up by the model.

```
autoplot(int.ets)
```

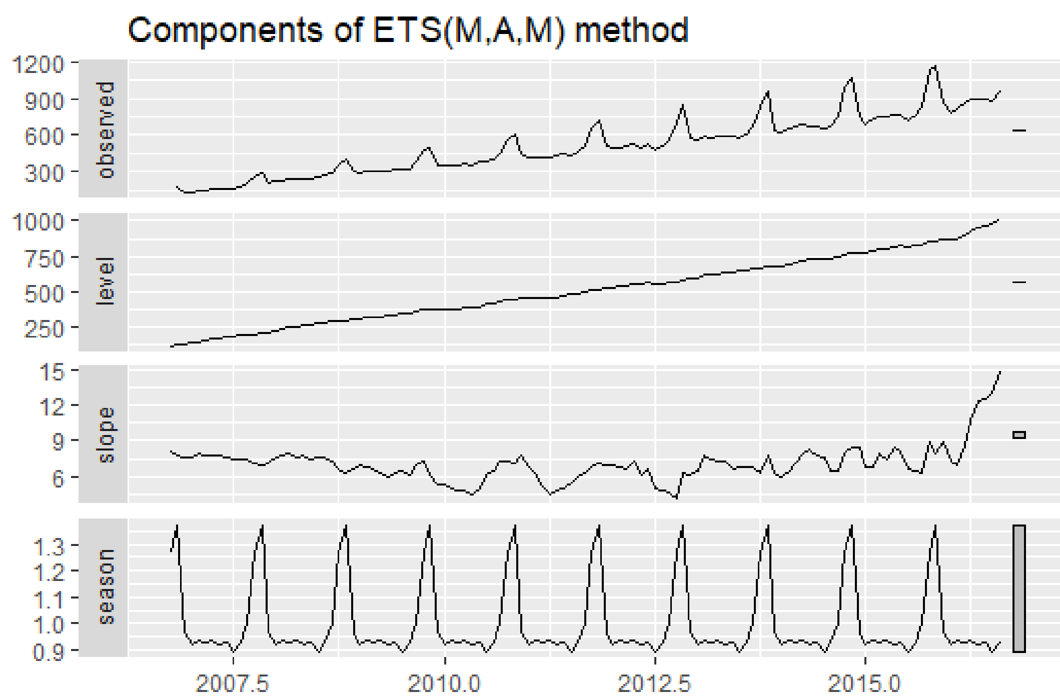
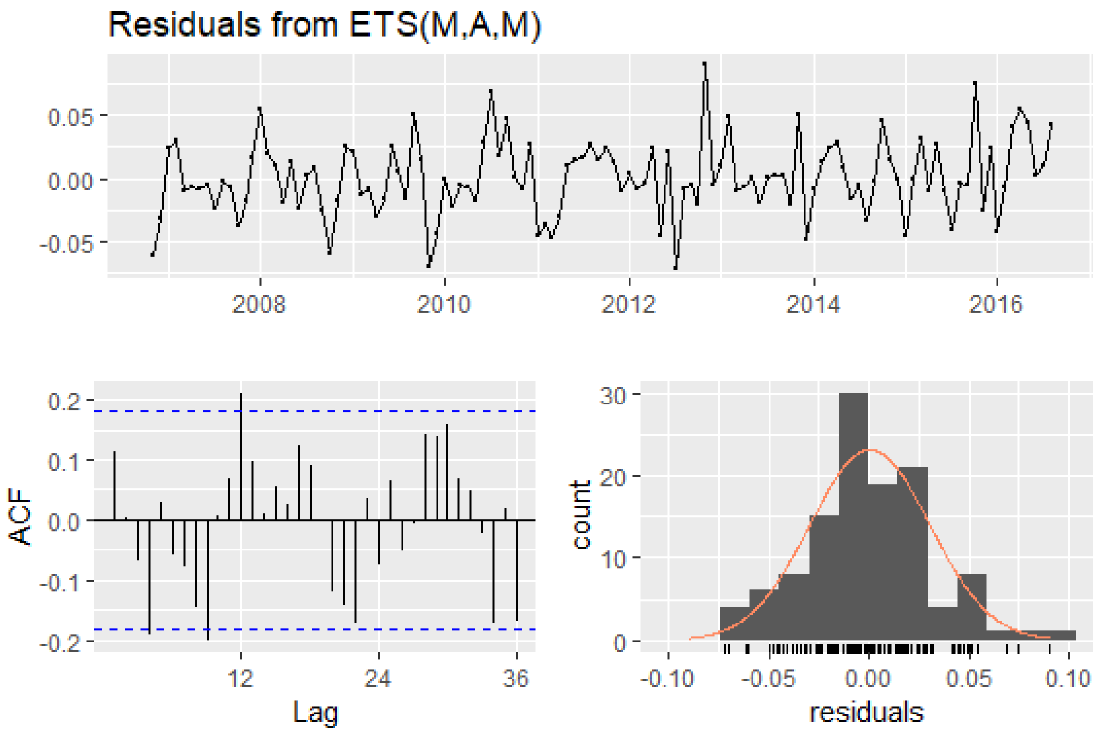


Fig.
4.5.1 ETS Model Components

Checking the residuals:

```
checkresiduals(int.ets)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(M,A,M)
## Q* = 37.688, df = 8, p-value = 8.595e-06
##
## Model df: 16.   Total lags used: 24

res.int4<-residuals(int.ets)
shapiro.test(res.int4)

##
##  Shapiro-Wilk normality test
##
## data:  res.int4
## W = 0.98979, p-value = 0.5285
```

The model hasn't picked up all the information in the data as there are a few lags outside the confidence bounds and there is autocorrelation in the data over the whole series shown by the Ljung-Box test statistic. Residuals are almost normally distributed, and the mean is oscillating around zero. There is still some apparent variance in the data.

Plotting the forecast:

```
fc.int1<-(forecast(int.ets, h = h))
autoplot(fc.int1) + xlab("Year") + ylab("£m")
```

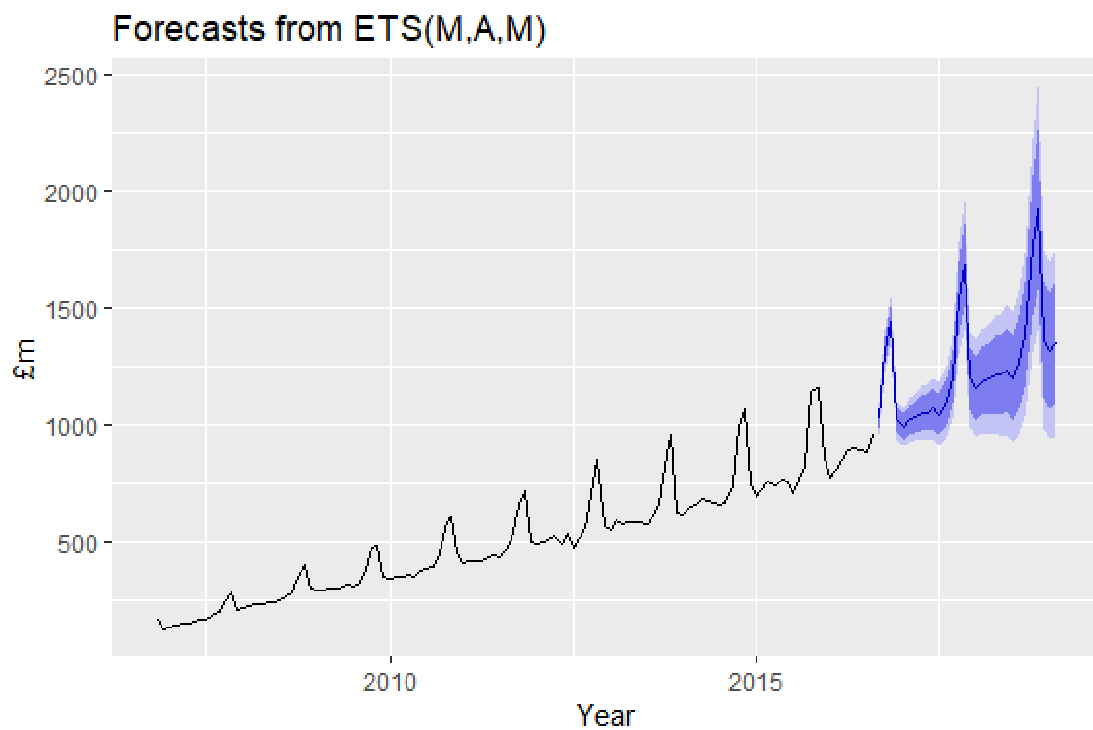


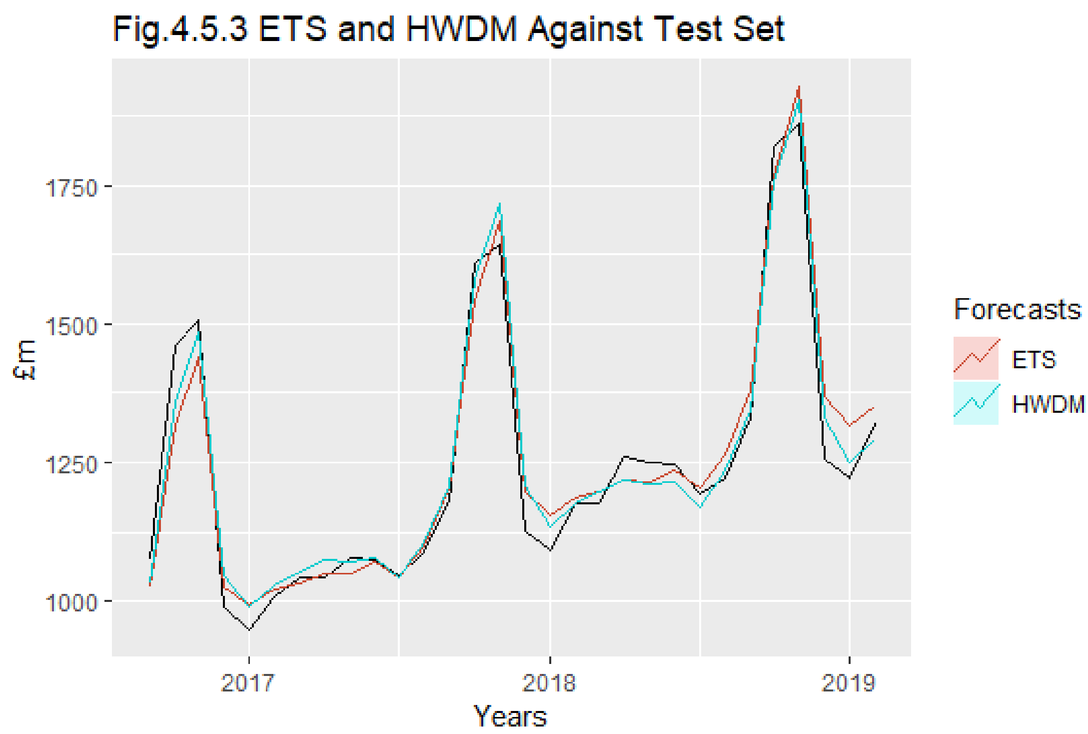
Fig. 4.5.2 Forecasts from ETS Model

The accuracy measures show that this model does not improve on the Holt Winters method but the RMSE is quite close.

```
int.acc7<-accuracy(fc.int1,int.test)
int.acc7

##                      ME     RMSE      MAE         MPE     MAPE      MASE
## Training set  1.580125 18.57932 12.35272 -0.04171495 2.297639 0.1464035
## Test set     -7.948355 53.56864 42.06193 -0.80628531 3.275215 0.4985145
##                     ACF1 Theil's U
## Training set -0.03271944        NA
## Test set      0.43360046 0.2611793
```

Plotting the ETS model with the HW method against the test set:

```
autoplot(int.test) +
  autolayer(fc.int1, series = "ETS", PI = FALSE)+ autolayer(int.hwdm, PI = F,
series = "HWDM") + xlab("Years") + ylab("£m") + ggtitle("Fig.4.5.3 ETS and HW
DM Against Test Set") + guides(colour = guide_legend(title = "Forecasts"))
```



ETS models are underpinned by the HW methods and are generally preferable as they can generate performance intervals. The corresponding models are:

1)ETS(MAdM) - multiplicative error, additive damped trend and multiplicative seasonality.

2)ETS(AAdM) - additive error, additive damped trend and multiplicative seasonality.

AAdM is a forbidden model combination, so the accuracy of the MAdM forecast was investigated to see if a better ETS model could be produced.

```
int.ets2<-ets(int.train, model = "MAM", damped = T)

fc.int3<-(forecast(int.ets2, h = h))
accuracy(fc.int3,int.test)

##                      ME     RMSE      MAE      MPE     MAPE      MASE
## Training set  3.402446 19.61033 12.52191 0.386545 2.269242 0.1484087
## Test set     56.500772 80.31913 62.25765 4.075891 4.658184 0.7378726
##                    ACF1 Theil's U
## Training set -0.1801344        NA
## Test set      0.4028048 0.4020314
```

The accuracy of this model was worse than the ETS(MAM)

Question 4.6

ARIMA

Arima models require stationary data. There appears to be some variance, so a Box Cox transformation was used:

```r
lambda<-BoxCox.lambda(int.train)
lambda

## [1] -0.2709233

int.trans<-BoxCox(int.train, lambda = "auto")
```

Running a KPSS test on the transformed data, we can reject the null hypothesis of stationarity:

```r
library("urca")
summary(ur.kpss(int.trans))

##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 2.2223
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

To identify the order of seasonal differencing:

```r
nsdiffs(int.trans)

## [1] 1
```

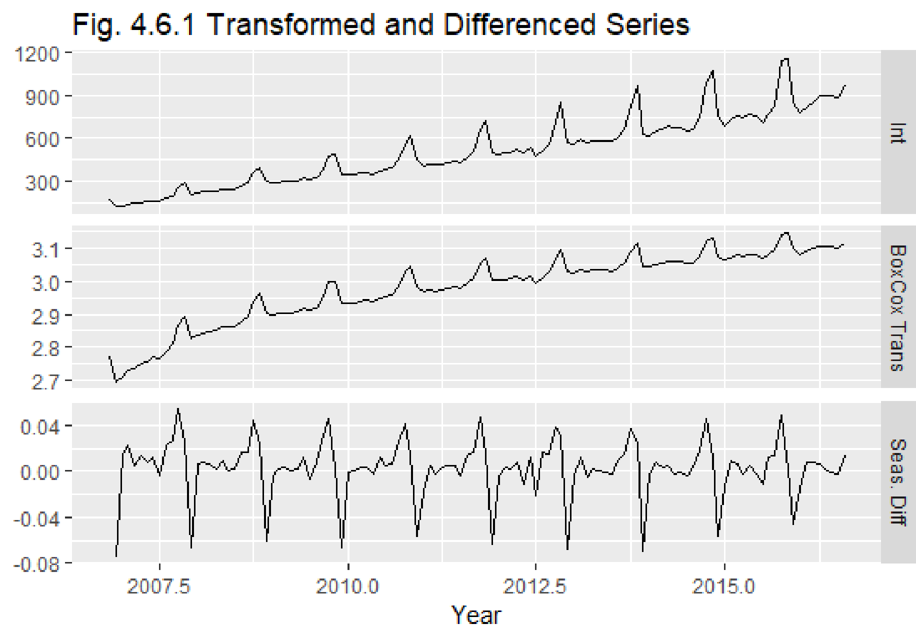
Then the order of first differencing:

```r
int.diff<-diff(int.trans, frequency = 12)
ndiffs(int.diff)

## [1] 0

summary(ur.kpss(int.diff))

##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 0.0707
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

With one order of seasonal differencing, the data is stationary as test statistic is below the 1% significance level.

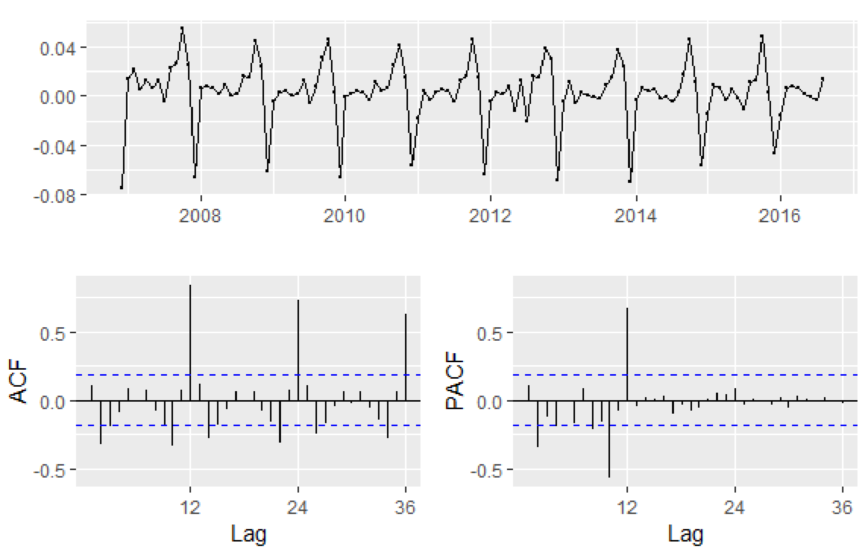
Plotting the transformed and differenced data against the original series, the data appears stationary.

```
cbind("Int" = int.train,
      "BoxCox Trans" = int.trans,
      "Seas. Diff" = int.diff) %>%
  autoplot(facets=TRUE) +
    xlab("Year") + ylab("") +
    ggtitle("Fig. 4.6.1 Transformed and Differenced Series")
```



Plotting the ACF and PACF plots:

```
ggtsdisplay(int.diff)
```



For the non-seasonal part, the PACF and ACF decays after lag 2 or 3 suggesting an AR(2) might be suitable. For the non-seasonal part, the PACF and ACF cut off after seasonal lag 12 so an MA(1) model could be appropriate. The following models were tried with arima:

(2,0,0)(0,1,1) (3,0,0)(0,1,1) (2,0,1)(0,1,1) (1,0,1)(1,1,0) (3,0,0)(1,1,0) (2,0,1)(1,1,0)

```
int.ar1 <- Arima(int.train, order=c(2,0,0), seasonal=c(0,1,1), lambda="auto")
int.ar2 <- Arima(int.train, order=c(3,0,0), seasonal=c(0,1,1), lambda="auto")
int.ar3 <- Arima(int.train, order=c(2,0,1), seasonal=c(0,1,1), lambda="auto")
int.ar4 <- Arima(int.train, order=c(1,0,1), seasonal=c(1,1,0), lambda="auto")
int.ar5 <- Arima(int.train, order=c(3,0,0), seasonal=c(1,1,0), lambda="auto")
int.ar6 <- Arima(int.train, order=c(2,0,1), seasonal=c(1,1,0), lambda="auto")

int.ar1$aicc

## [1] -735.2391

int.ar2$aicc

## [1] -733.0408

int.ar3$aicc

## [1] -733.0369

int.ar4$aicc

## [1] -731.2113

int.ar5$aicc

## [1] -731.564

int.ar6$aicc

## [1] -731.4435
```

Using auto.arima produces a warning message that the model has 3 differencing operations and that this should be reduced.

```
int.ar7<-auto.arima(int.train, lambda = "auto", stepwise = F, approximation =
F)

## Warning in auto.arima(int.train, lambda = "auto", stepwise = F,
## approximation = F): Having 3 or more differencing operations is not
## recommended. Please consider reducing the total number of differences.

int.ar7

## Series: int.train
## ARIMA(2,2,1)(0,1,1)[12]
## Box Cox transformation: lambda= -0.2709233
##
## Coefficients:
##           ar1      ar2      ma1     sma1
##       -0.6005  -0.2423  -0.8535  -0.5794
```

```
## s.e.    0.1096   0.1084   0.0619   0.1368
##
## sigma^2 estimated as 4.37e-05:  log likelihood=369.99
## AIC=-729.99   AICc=-729.37   BIC=-716.76
```

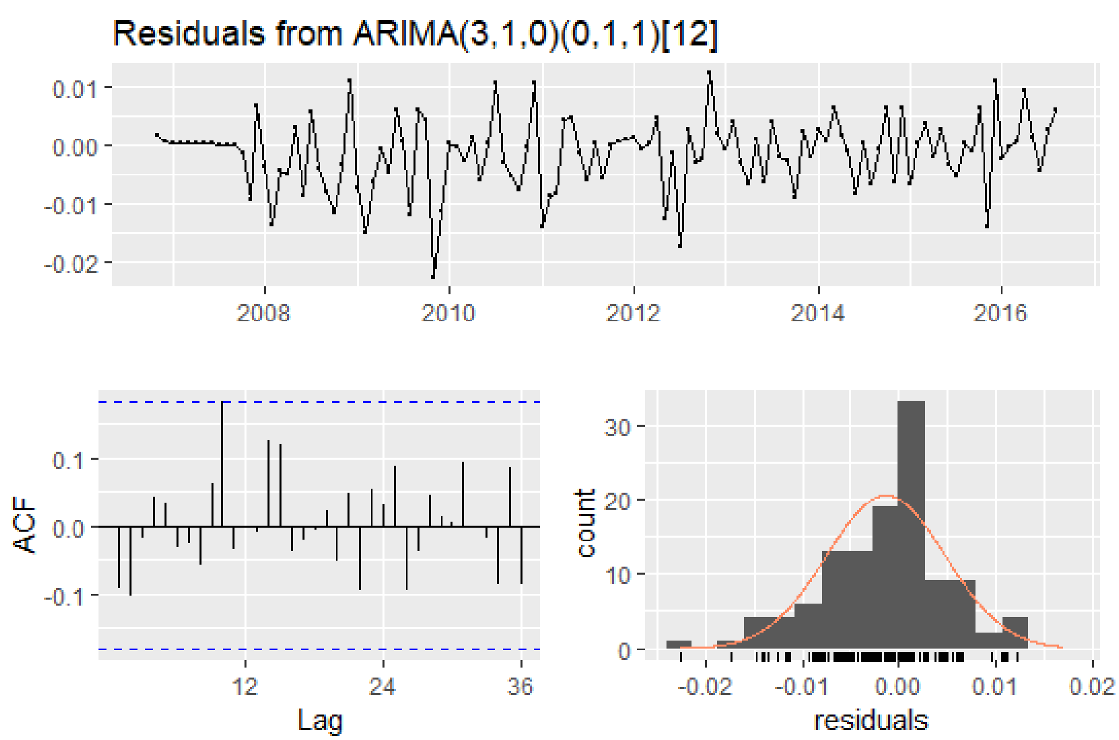
The code was amended to limit the maximum differences to 1. This returns a model (3,1,0)(0,1,1)12 with AICc of -741.2.

```
int.ar8<-auto.arima(int.train, lambda = "auto", stepwise = F, approximation =
F, max.d=1, max.D=1)
int.ar8
```

```
## Series: int.train
## ARIMA(3,1,0)(0,1,1)[12]
## Box Cox transformation: lambda= -0.2709233
##
## Coefficients:
##           ar1     ar2     ar3     sma1
##       -0.3173  0.1686  0.3542  -0.5966
## s.e.   0.0944  0.1099  0.1024   0.1354
##
## sigma^2 estimated as 4.564e-05:   log likelihood=375.9
## AIC=-741.8   AICc=-741.2   BIC=-728.53
```

checking the residuals:

```
checkresiduals(int.ar8)
```




```
##
##  Ljung-Box test
##
```

```
## data:  Residuals from ARIMA(3,1,0)(0,1,1)[12]
## Q* = 15.46, df = 20, p-value = 0.7495
##
## Model df: 4.   Total lags used: 24
```

```
res.int5<-residuals(int.ar8)
shapiro.test(res.int5)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.int5
## W = 0.9724, p-value = 0.0156
```

From the ACF plot, there are no significant lags, and the Ljung-Box test p-value is above 0.05 so the model has picked up all the information in the series. The mean appears near zero and constant, but the residuals are not normally distributed.

Forecasting using this model:

```
fc.int2<-forecast(int.ar8,h=h)
autoplot(fc.int2)
```

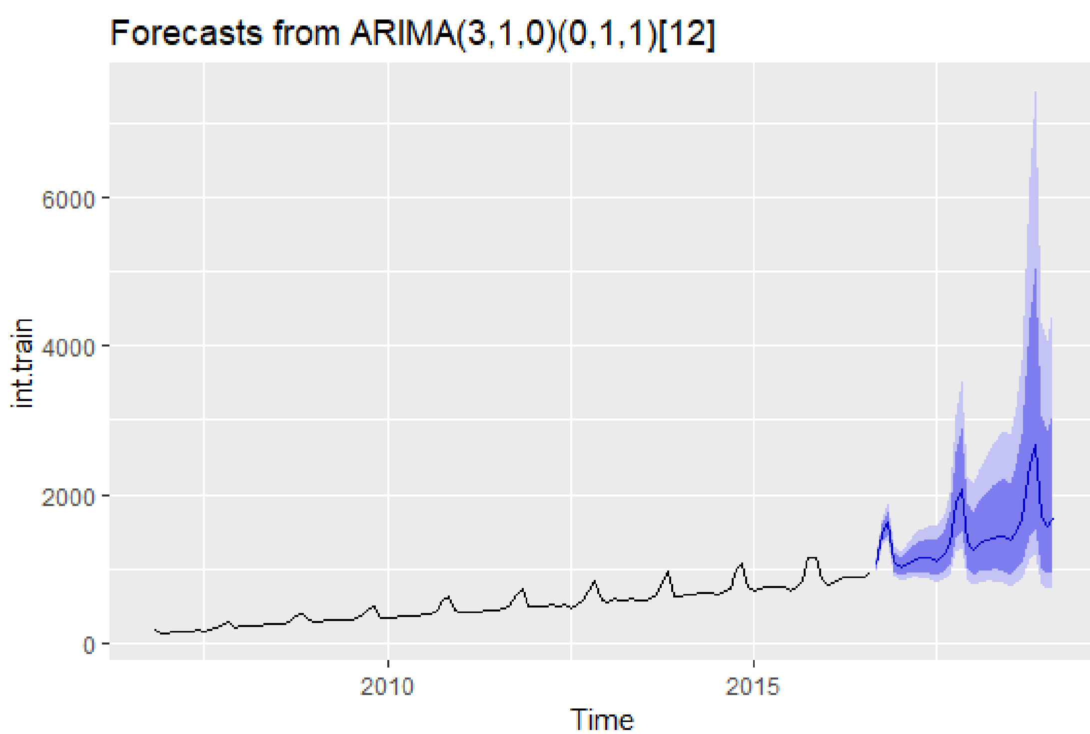


Fig.

4.6.2 ARIMA Forecast

Checking the accuracy of this model against the test set it does not improve on the previous best method as shown in Table 1.9

```
int.acc8<-accuracy(fc.int2,int.test)
int.acc8
```

```
##                        ME       RMSE        MAE         MPE       MAPE
## Training set   -2.900036  21.67726  13.62788  -0.7580537  2.457047
## Test set     -209.164735 274.12768 210.41822 -15.5877585 15.704167
##                   MASE       ACF1 Theil's U
## Training set 0.1615166 -0.3161933        NA
## Test set     2.4938595  0.7723161  1.209463
```

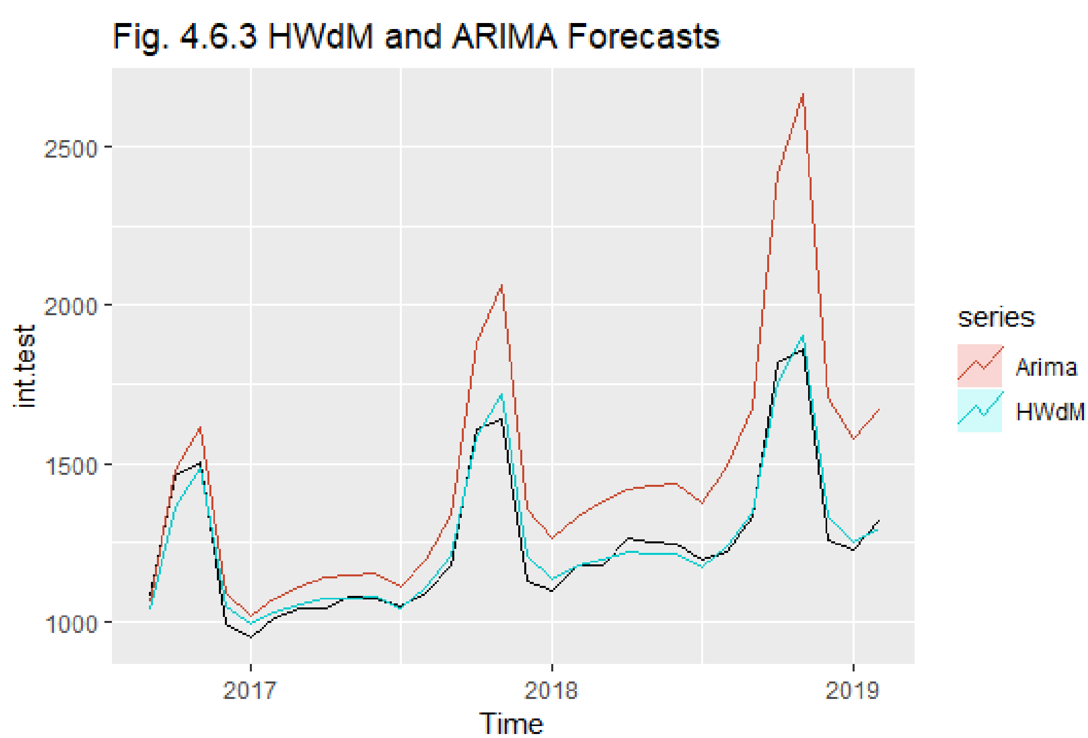
```
xx<-round(int.acc8,2)
xy<-round(int.acc5,2)
a.table <- rbind(xx, xy)
row.names(a.table)<-c("ARIMA Train","ARIMA Test", "HWdM Train", "HwdM Test")
a.table<-as.data.frame(a.table)
kable(a.table, caption= "Table 1.9 Forecast accuracy of Arima compared to HW
",digits = 5)
```

Table 1.9 Forecast accuracy of Arima compared to HW

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
ARIMA Train	-2.90	21.68	13.63	-0.76	2.46	0.16	-0.32	NA
ARIMA Test	-209.16	274.13	210.42	-15.59	15.70	2.49	0.77	1.21
HWdM Train	2.60	17.70	12.60	0.40	2.50	0.10	-0.20	NA
HwdM Test	-5.50	42.10	34.30	-0.60	2.70	0.40	0.30	0.20

Plotting the two the HW method clearly matches the test set much better than the ARIMA model.

```
autoplot(int.test) + autolayer(fc.int2, PI = F, series = "Arima") + autolayer
(int.hwdm, PI = F, series = "HWdM") + ggtitle("Fig. 4.6.3 HWdM and ARIMA Fore
casts")
```



Conclusion

For this series, the best forecasting method on the training set was STL + ETS(MAN) and for the test set, the Holt Winters Damped Multiplicative method performed the best in predicting internet retail sales.

Of the ETS models, ETS(MAM) had an RMSE close to the Holt Winters method, therefore, to generate performance intervals for the point forecasts, this model would be preferable.

Question 5.1

Data Source:https://www.metoffice.gov.uk/hadobs/hadsst3/data/download.html

Northern Hemisphere Median Sea Surface Temperatures Monthly Data (1950 - 2019)
HadSST.3.1.1.0_monthly_nh_ts.txt

Data notes

This data is taken from a larger data set dating back to the 1800's. There is inconsistency in the data over the full period due to changes in data collection techniques , so data from 1950 onwards was used . See Appendix 2.

The series contains sea surface temperature (SST) anomalies relative to the mean SST of 1961 until 1990. It is common practice to use anomalies rather than absolute temperatures as *"This effectively normalizes the data so they can be compared and combined to more accurately represent temperature patterns with respect to what is normal for different places within a region."(*Ncdc.noaa.gov, 2019).

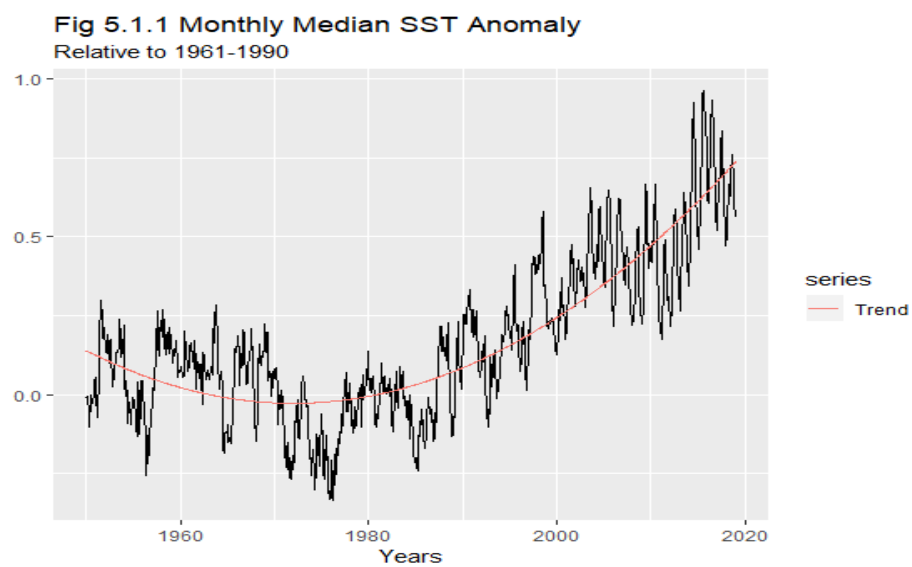
```
sst.data <- readxl::read_excel("C:/Users/Imy Hull/Desktop/SST.xlsx")
sst.ts <- ts(sst.data, start=c(1950,1), end = c(2019,1), frequency = 12)
```

Examining the dataset:

There are 829 observations over a small value range of 1.297 including negative and near zero values.

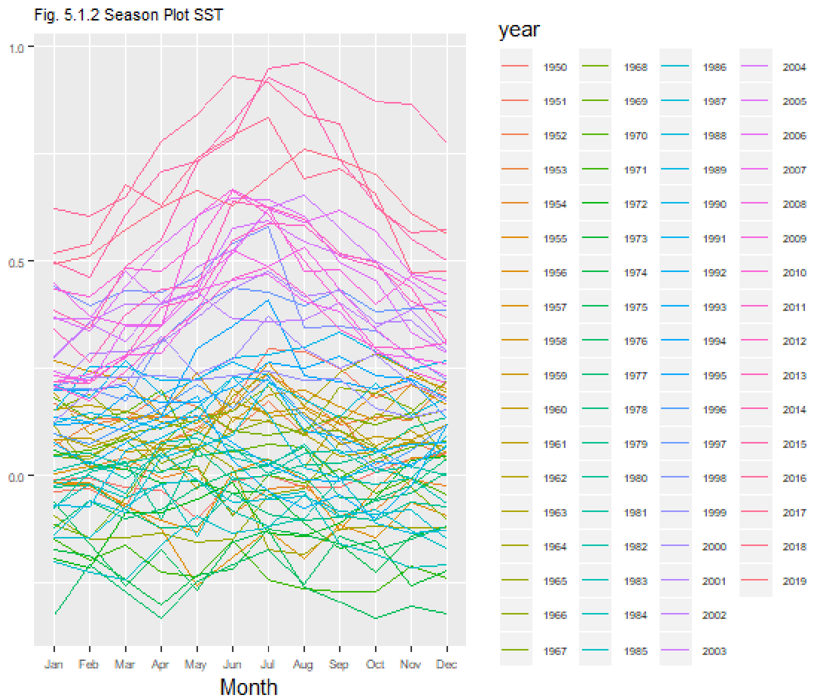
The data was plotted in Figs. 5.1.1 to 5.1.6.

```
sst.line<-tslm(sst.ts~trend + I(trend^2))
autoplot(sst.ts) + autolayer(sst.line$fitted.values, series = "Trend")+ xlab(
"Years") + ylab("") + ggtitle("Fig 5.1.1 Monthly Median SST Anomaly", subtitl
e = "Relative to 1961-1990")
```

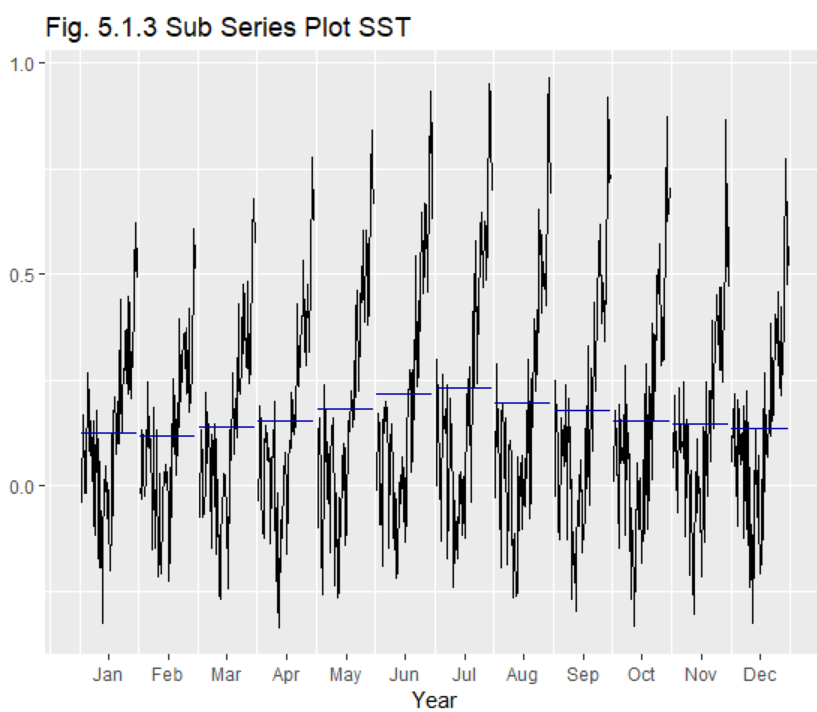


Note:The data has a non-linear trend. Research suggested a polynomial 2nd order trendline should be used, (Shmueli and Lichtendahl, n.d.p.34)

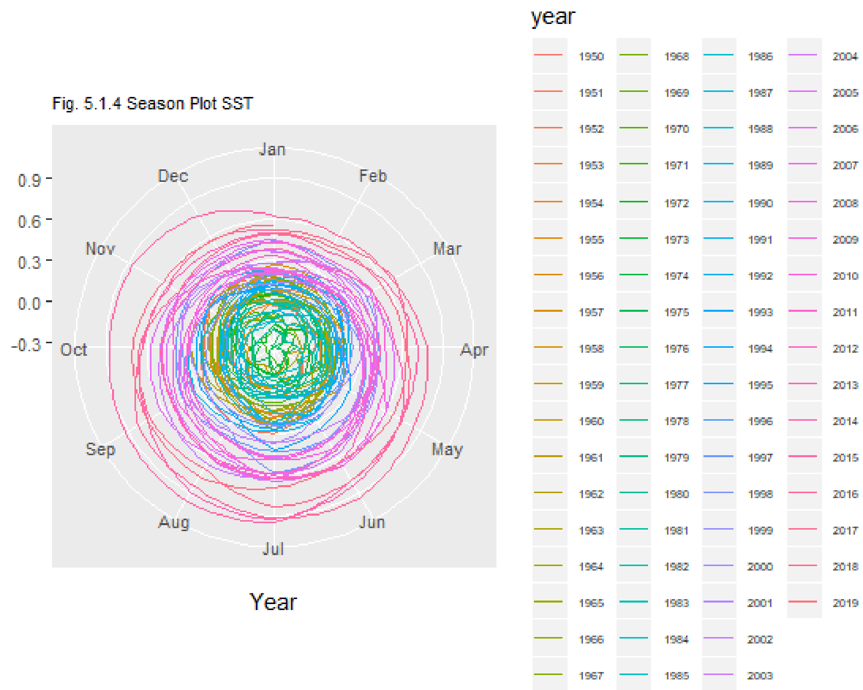
```
ggseasonplot(sst.ts) + theme(legend.text = element_text(size = 5),axis.text =
element_text(size = 6), plot.title = element_text(size = 8))+ ylab("")+   ggt
itle("Fig. 5.1.2 Season Plot SST")
```



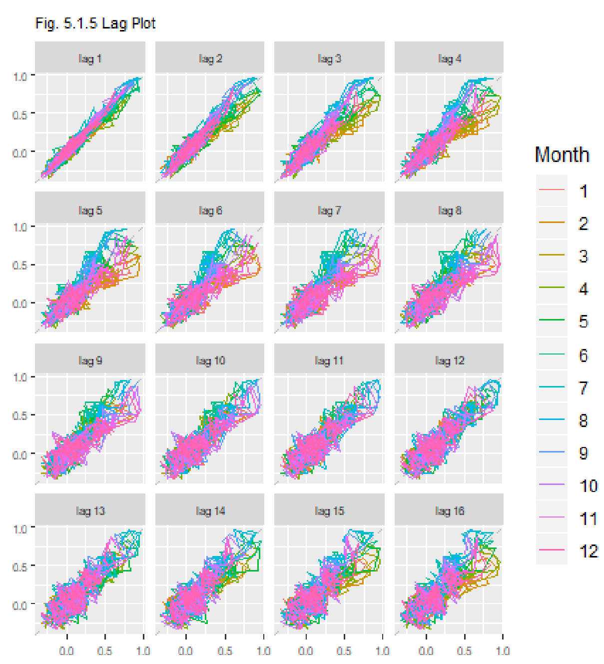

```
ggsubseriesplot(sst.ts)+ xlab("Year") + ylab("")+
  ggtitle("Fig. 5.1.3 Sub Series Plot SST")
```



```r
ggseasonplot(sst.ts, polar = TRUE) + theme(legend.text = element_text(size =
5), axis.text = element_text(size = 8), plot.title = element_text(size = 8))+
xlab("Year") + ylab("")+  ggtitle("Fig. 5.1.4 Polar Plot SST")
```




```r
gglagplot(sst.ts) + theme(axis.text = element_text(size = 6)) + theme(axis.ti
tle = element_text(size = 4),plot.title = element_text(size = 8),strip.text.x
= element_text(size = 6)) + xlab("Year") + ylab("")+
  ggtitle("Fig. 5.1.5 Lag Plot")
```



```r
ggAcf(sst.ts)
```

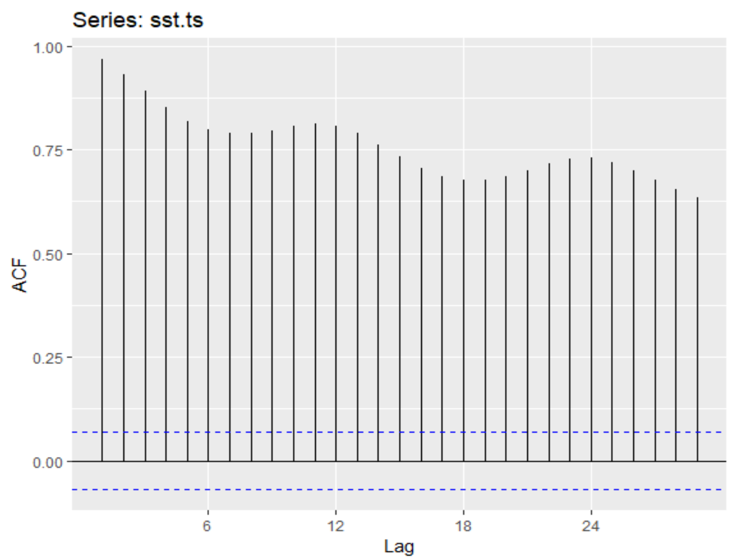


Fig. 5.1.6 ACF SST

Observations

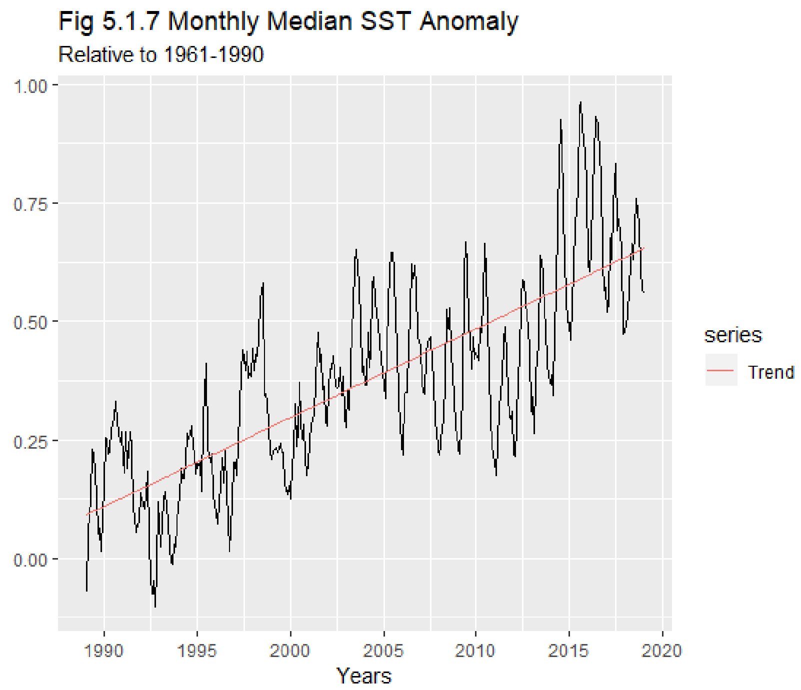
The SST Northern Hemisphere anomaly seems to have trended downwards until about the mid 1970's, where it started to rise. Since the late 1980's there has been an upward trend, peaking in the years 2014 to 2016 where the El Nino effect caused surface sea temperatures to rise to record levels (Climate.gov, 2019).

There are seasonal peaks over the northern summer months of July and August and the Polar plot shows that 2014 and 2015 also experienced particularly warm seas over the winter period of November and December. The Lag plot shows a high level of autocorrelation in the early lags and at the seasonal lag (12) and this seasonal pattern is also reflected in the ACF plot.

Subsetting the Data Series

It was decided that since satellite data was consistently being used to collect data from the later 1980's it would be more consistent and reliable, so the data was subsetted for this period to give 30 years of monthly data.

```r
sst<-window(sst.ts, start = c(1989,2))
sst.line1<-tslm(sst~trend)
autoplot(sst) + autolayer(sst.line1$fitted.values, series = "Trend")+ xlab("Y
ears") + ylab("") + ggtitle("Fig 5.1.7 Monthly Median SST Anomaly", subtitle
= "Relative to 1961-1990")
```




```
sst.line1$coefficients

## (Intercept)       trend
## 0.090238827 0.001565913
```

This represents 0.0187 per year growth in the median anomaly.

The plots were all run again and exhibited much the same patterns as for the full series from 1950, so are not repeated. The seasonal fluctuations appear to be increasing over the series so the data could be multiplicative in nature. The seasonal fluctuations are emphasised in the ACF plot shown in Fig. 5.1.7.

```
sst<-window(sst.ts, start = c(1989,2))
ggAcf(sst)
```

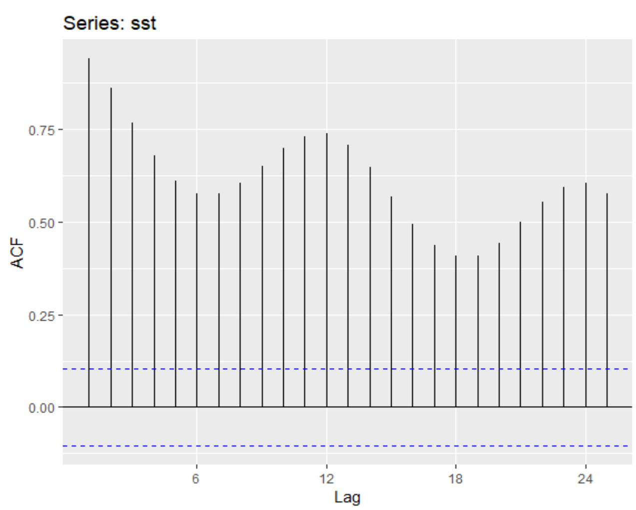


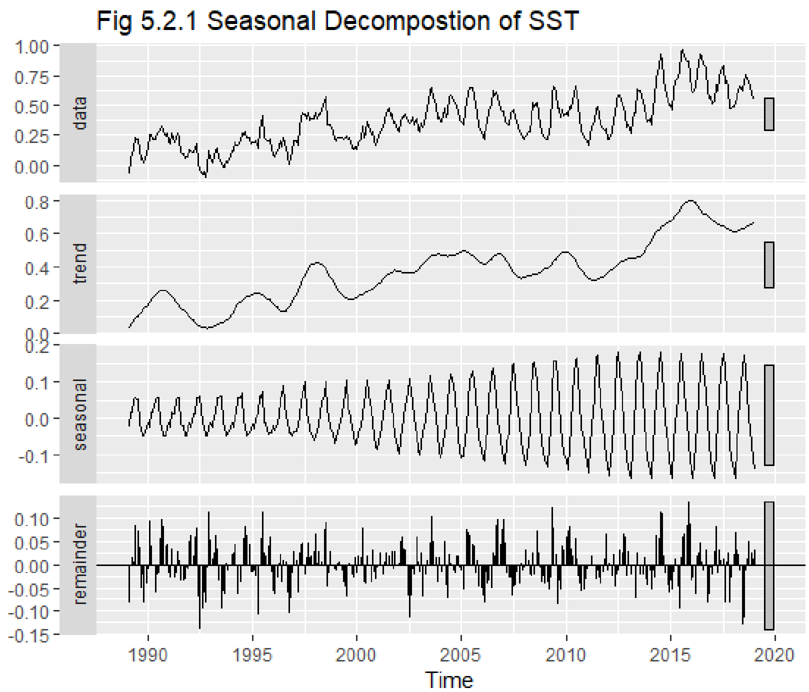
Fig. 5.1.7 ACF Plot for the Adjusted SST Series

Question 5.2

Forecasting - Basic Methods

Performing a stl decomposition, the various components of the time series can be seen.

```
sst.stl<-stl(sst[,1], s.window = 13)
autoplot(sst.stl) + ggtitle("Fig 5.2.1 Seasonal Decompostion of SST")
```



The increasing seasonal variance is clearly shown in Fig. 5.2.1 as well as an upward trend. There doesn't seem to be any pattern in the remainder. Forecasting methods that account for seasonal patterns need to be considered.

The data was split into a training and test set.

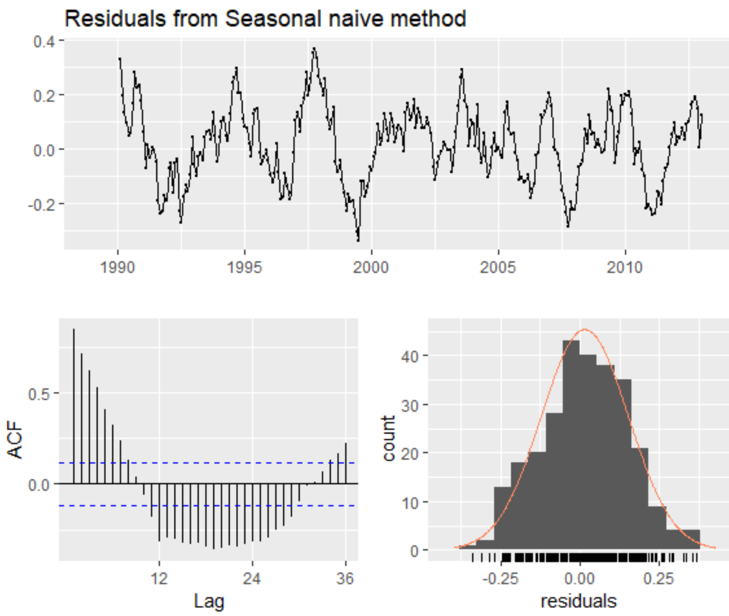
```
sst.train<-window(sst, end = c(2013,1))
sst.test<-window(sst, start = c(2013,2))
h=length(sst.test)
length(sst.train)

## [1] 288

length(sst.test)

## [1] 72
```

Using Seasonal naive on the training set:

```
sst.snaive <- snaive(sst.train,h=h)
```

Checking the residuals:

```
checkresiduals(sst.snaive)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Seasonal naive method
## Q* = 1063.6, df = 24, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 24
```

```
res.sst1<-residuals(sst.snaive)
shapiro.test(res.sst1)
```

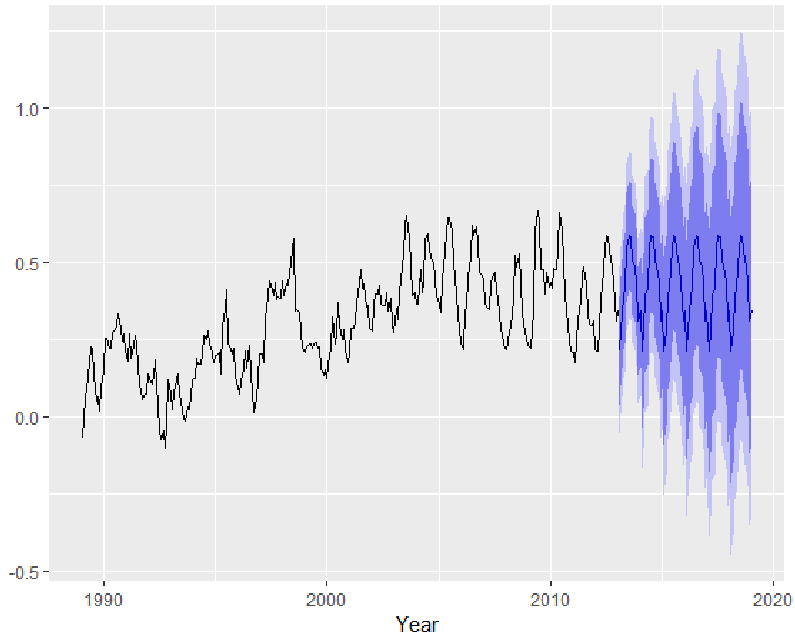
```
##
##  Shapiro-Wilk normality test
##
## data:  res.sst1
## W = 0.99501, p-value = 0.512
```

The ACF shows many lags outside the confidence bounds and the Ljung-Box test statistic is very small indicating that this method did not capture all the information in the data. The residuals have a near-zero mean but there seems to be some increased variance over the earlier part of the series. The histogram is normally distributed, supported by the Shapiro-Wilks test. The residuals do not resemble white noise.

Forecasting with seasonal naive:

```
sst.fc1<-(snaive(sst.train, h = h))
autoplot(sst.fc1) + ggtitle("Fig. 5.2.2 SST Seasonal Naive Forecast") + xlab(
"Year") + ylab("")
```

Fig. 5.2.2 SST Seasonal Naive Forecast



The forecast has picked up the seasonal pattern but does not account for trend.

Checking the accuracy against the test set:

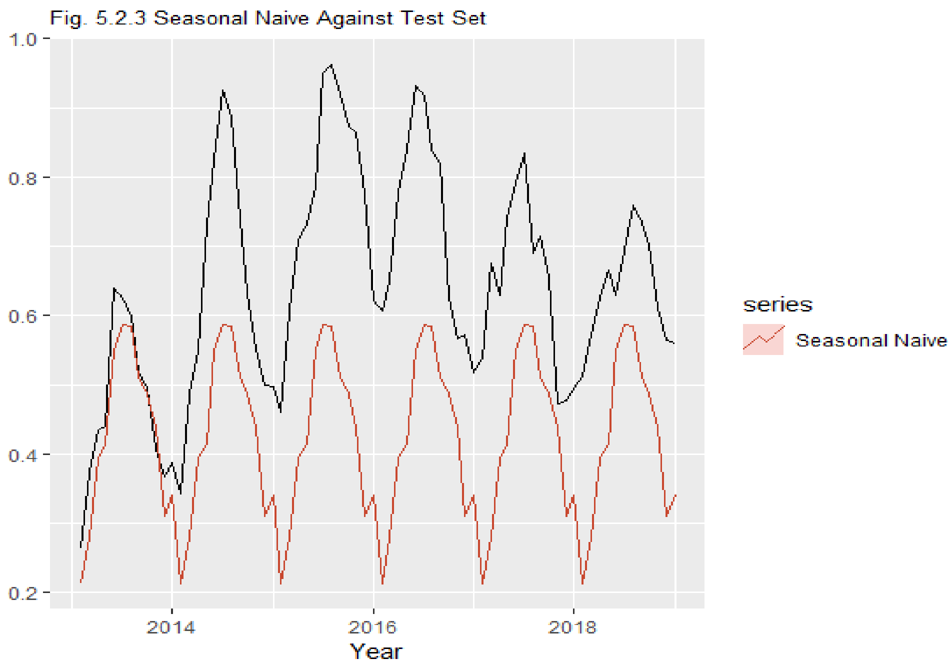
```
library("knitr")

## Warning: package 'knitr' was built under R version 3.5.3

sst.acc1<-accuracy(sst.fc1, sst.test)
d.table <- sst.acc1
row.names(d.table)<-c("SNaive Train","Snaive Test")
d.table<-as.data.frame(d.table)
kable(d.table, caption = "Table 1.10 Accuracy Measures for the Seasonal Naive
Method", digits = 2)
```

Table 1.10 Accuracy Measures for the Seasonal Naive Method

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
SNaive Train	0.01	0.14	0.11	-8.35	63.68	1.00	0.85	NA
Snaive Test	0.22	0.25	0.22	32.66	32.88	1.98	0.78	2.58

Plotting the forecast against the test set, clearly shows that the method simply repeats the pattern from the previous season and misses the rising peaks in the test set:

```
autoplot(sst.test) + autolayer(sst.fc1, PI = F, series = "Seasonal Naive") +
xlab("Year") + ylab("") + ggtitle("Fig. 5.2.3 Seasonal Naive Against Test Set
") + theme(plot.title = element_text(size = 10))
```



Question 5.3

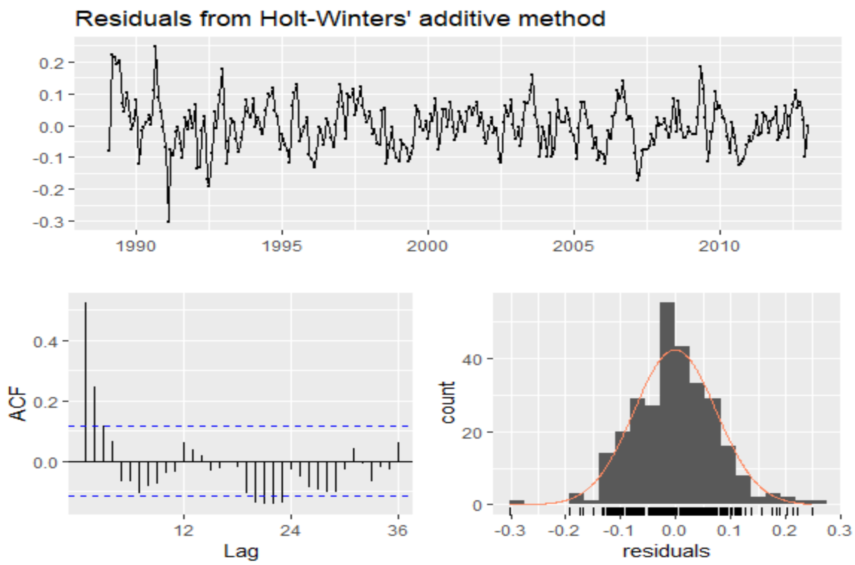
Holt Winters Methods

The Holt Winters methods account for both seasonality and trend. There are both additive and multiplicative seasonality methods and both can have a damped trend.

Note: *The Multiplicative methods require series that do not contain zero or negative values (Hyndman and Athanasopoulos, 2018, p.209), so only the additive methods were considered.*

Additive and Damped Additive:

```
sst.hwa<-hw(sst.train, seasonal = "additive", h = h)
sst.hwda<-hw(sst.train, seasonal = "additive", damped = T, h = h)

checkresiduals(sst.hwa)
```

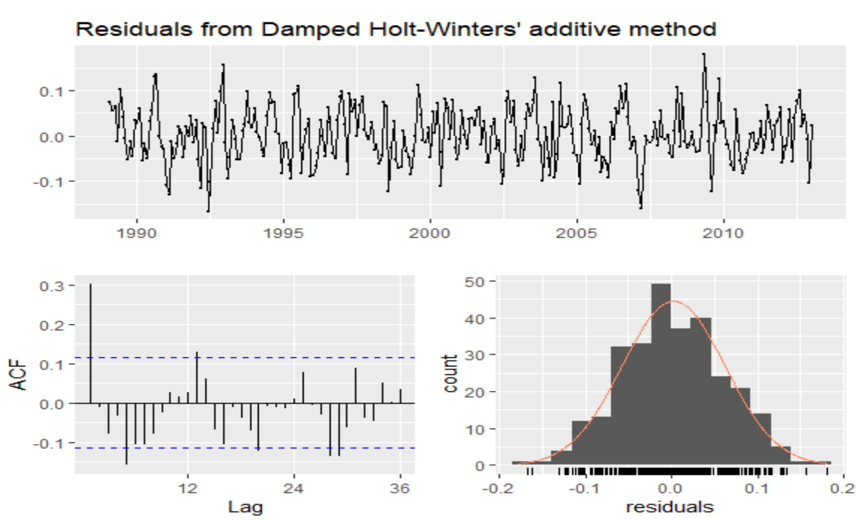



```
##
##  Ljung-Box test
##
## data:  Residuals from Holt-Winters' additive method
## Q* = 145.02, df = 8, p-value < 2.2e-16
##
## Model df: 16.   Total lags used: 24
```

```
res.sst2<-residuals(sst.hwa)
shapiro.test(res.sst2)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.sst2
## W = 0.98891, p-value = 0.02686
```

```
checkresiduals(sst.hwda)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from Damped Holt-Winters' additive method
## Q* = 63.513, df = 7, p-value = 2.99e-11
##
## Model df: 17.   Total lags used: 24
```

```
res.sst3<-residuals(sst.hwda)
shapiro.test(res.sst3)
```

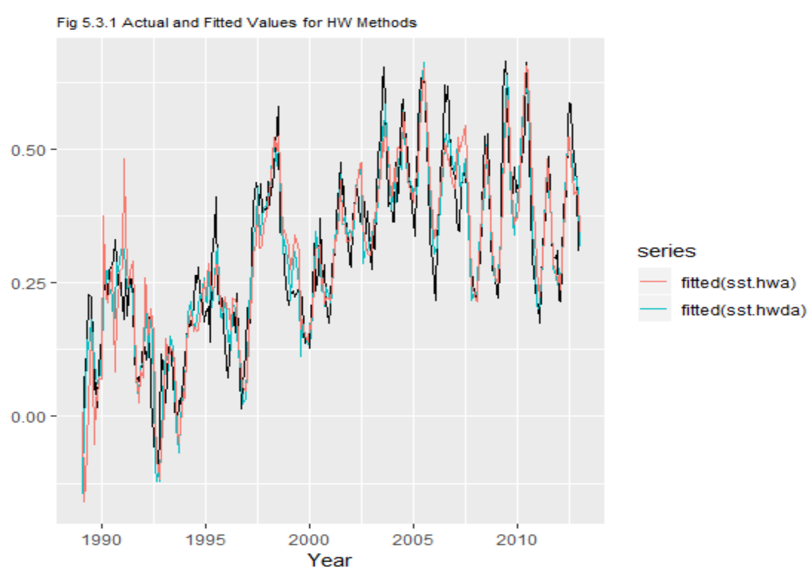
```
##
##  Shapiro-Wilk normality test
##
## data:  res.sst3
## W = 0.99785, p-value = 0.9722
```

Observations

Both models picked up the seasonal pattern quite well, but there is still some autocorrelation left in the residuals and the Ljung-Box test reflects this with very small p-values for both methods. The mean and variance for both appears constant and the mean is near-zero. The residuals for the additive damped method appear normally distributed but those for the un-damped are not.

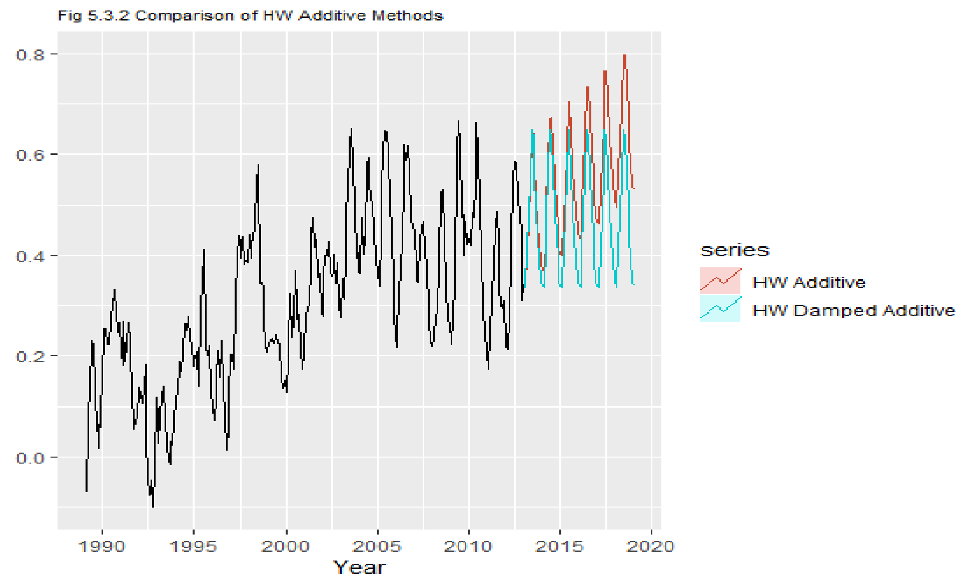
Plotting the fitted values over the training set (Fig. 5.3.1), the HW Additive method doesn't fit the series as well in the early years.

```
autoplot(sst.train)+ autolayer(fitted(sst.hwda)) + autolayer(fitted(sst.hwa))
+ xlab("Year") + ylab("") + ggtitle("Fig 5.3.1 Actual and Fitted Values for H
W Methods") + theme(plot.title = element_text(size = 8))
```



Plotting the forecasts in Fig. 5.3.2, the additive method seems to trend up indefinitely, whereas the damped method contains the forecast to being level with the previous few years.

```r
autoplot(sst.train) + autolayer(sst.hwa, PI = F, series = "HW Additive") + au
tolayer(sst.hwda, PI = F, series = "HW Damped Additive") + ggtitle("Fig 5.3.2
Comparison of HW Additive Methods") + theme(plot.title = element_text(size =
8)) + xlab("Year") + ylab("")
```

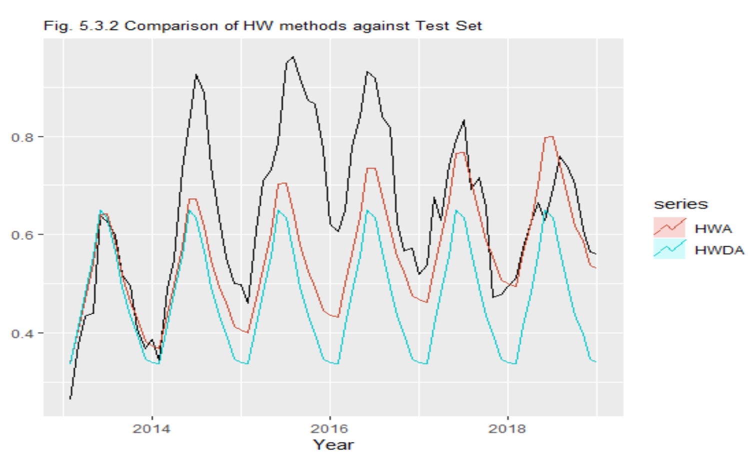


Checking the parameters of the damped method, the smoothing parameter phi is equal to 0.813. Damping has a strong effect for small values so this has affected the forecast quite significantly (Hyndman and Athanasopoulos, 2018)

```r
sst.hwda$model$par[1:4]

##      alpha       beta      gamma        phi
## 0.60073232 0.02246074 0.21388909 0.81291868
```

Plotting the two methods against the test set, neither method catches the peak SST readings during the El Nino event from 2014 to 2016 :

```r
autoplot(sst.test) + autolayer(sst.hwa, PI = F, series = "HWA") + autolayer(s
st.hwda, PI = F, series = "HWDA") + xlab("Year") + ylab("") + ggtitle("Fig. 5
.3.3 Comparison of HW methods against Test Set") + theme(plot.title = element
_text(size = 10))
```



Comparing the accuracy of the two methods against the test set:

```
sst.acc2<-accuracy(sst.hwa,sst.test)
sst.acc3<-accuracy(sst.hwda,sst.test)
d.table <- rbind(sst.acc2,sst.acc3)
row.names(d.table)<-c("HWA Train","HWA Test","HWDA Train", "HWDA Test")
d.table<-as.data.frame(d.table)
kable(d.table, caption = "Table 1.11 "Accuracy Measures for the Holt Winters
Additive Method", digits = 3)
```

Table 1.11 Accuracy Measures for the Holt Winters Additive Method

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
HWA Train	-0.001	0.075	0.058	-2.986	37.785	0.518	0.524	NA
HWA Test	0.085	$\boxed{0.143}$	0.107	10.702	15.175	0.958	0.838	1.264
HWDA Train	0.002	0.060	0.048	-8.358	29.942	0.430	0.301	NA
HWDA Test	0.175	0.214	0.183	24.717	26.940	1.647	0.842	2.017

The HW additive method has a lower RMSE and is the preferred method. However, looking at the forecast in Fig. 5.3.2 it seems to have an indefinite upward trend so may not be suitable for forecasting a long way beyond the test period.

Question 5.4

ETS Models

ETS models allow for forecasts and prediction intervals to be generated and are viewed as preferable to methods. Each method is associated with two models with an additional error term that can be multiplicative or additive.

Running the ets() function returned a model (A,Ad,A) with additive seasonality and errors and additive damped trend.

```
sst.ets<-ets(sst.train)
sst.ets

## ETS(A,Ad,A)
##
## Call:
##  ets(y = sst.train)
##
##   Smoothing parameters:
##     alpha = 0.6007
##     beta  = 0.0225
##     gamma = 0.2139
##     phi   = 0.8129
##
##   Initial states:
##     l = -0.1867
##     b = 0.0432
##     s = -0.0293 -0.0695 -0.0897 -0.0746 -0.0716 -0.0289
```

```
##            0.0543 0.0758 0.1022 0.0426 0.0835 0.005
##
##   sigma:  0.0616
##
##        AIC      AICc       BIC
##  44.42709  46.96984 110.36038
```

The parameters for the ETS(A,Ad,A) were the same as for the HW(Ad,A) method as they are associated.

```
sst.hwda$model$par[1:4]
```

```
##      alpha       beta      gamma        phi
## 0.60073232 0.02246074 0.21388909 0.81291868
```

Plotting the ets decompostion:

```
autoplot(sst.ets)
```

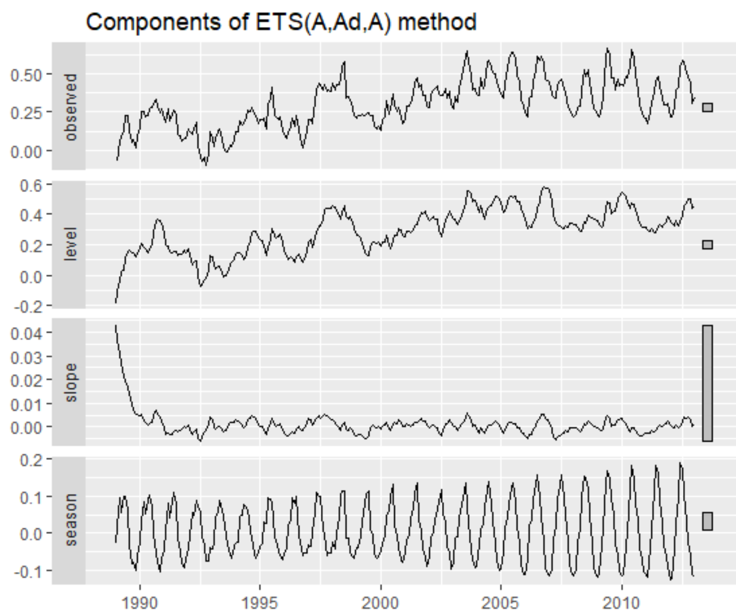
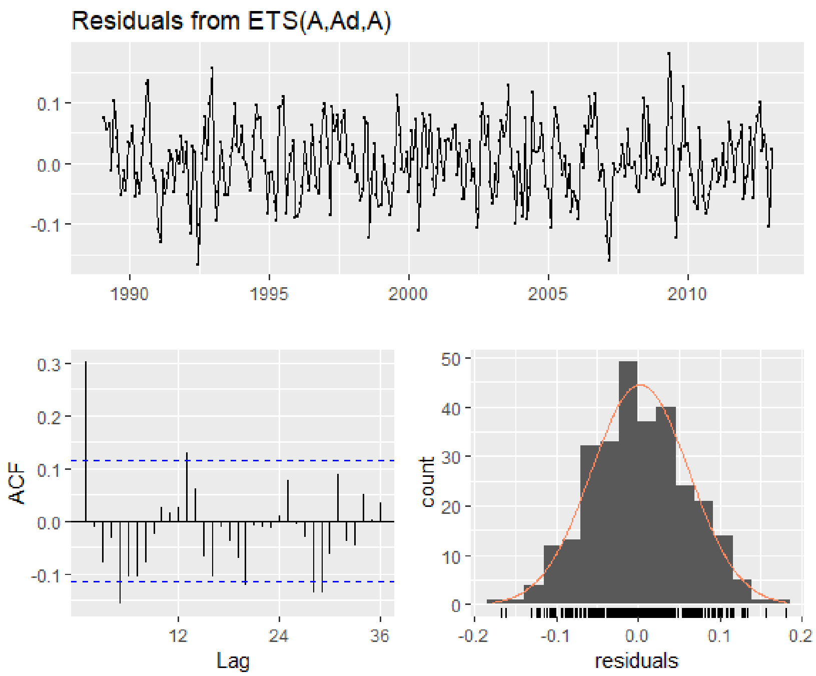


Fig. 5.4.1 ETS Model Components

The model picks up a seasonal pattern with increasing variability and the slope is relatively flat after accounting for the seasonal pattern.

The residuals of the model:

```
checkresiduals(sst.ets)
```



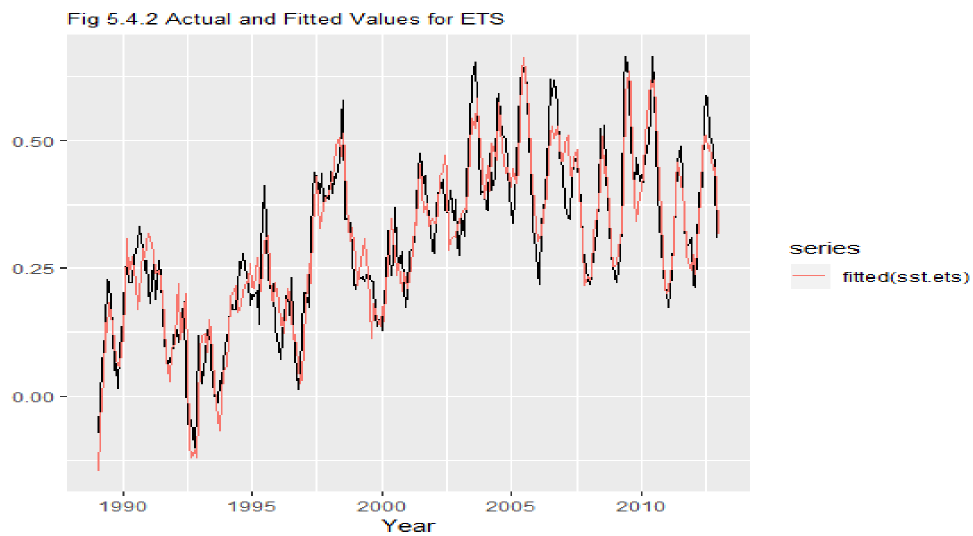

```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,Ad,A)
## Q* = 63.513, df = 7, p-value = 2.99e-11
##
## Model df: 17.   Total lags used: 24
```

```
res.sst4<-residuals(sst.ets)
shapiro.test(res.sst4)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.sst4
## W = 0.99785, p-value = 0.9722
```

There are several lags outside the confidence bounds and the Ljung-Box test is very small, so the model does not pick up all the information in the data. The mean and variance appear constant and the residuals are normally distributed so prediction intervals should be reliable. Plotting the actual and fitted values shows the model picks up the patterns reasonably well, missing just a few peaks:

```
autoplot(sst.train)+ autolayer(fitted(sst.ets)) + xlab("Year") + ylab("") + g
gtitle("Fig 5.4.2 Actual and Fitted Values for ETS") +  theme(plot.title = el
ement_text(size = 10))
```



Plotting the forecast based on the ETS model:

```
sst.fc.ets<-forecast(sst.ets, h = h)
autoplot(sst.fc.ets)
```

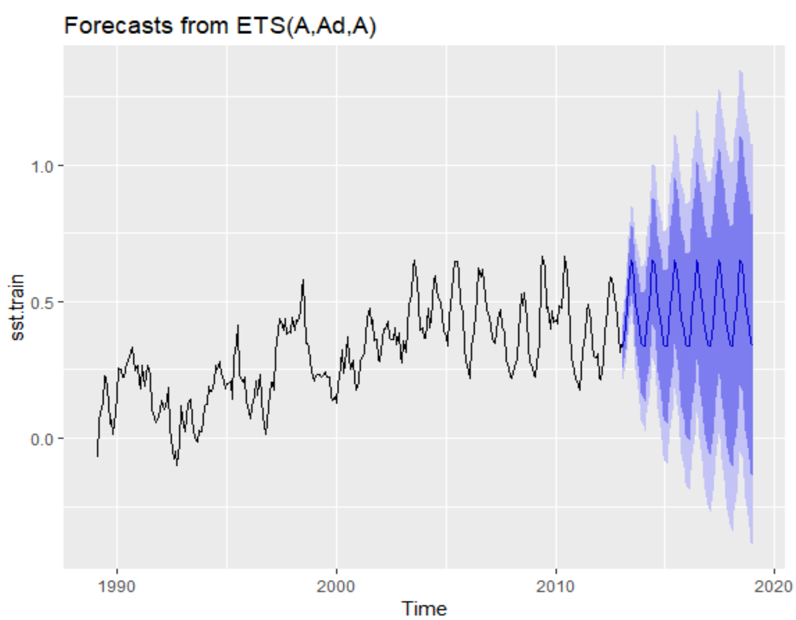


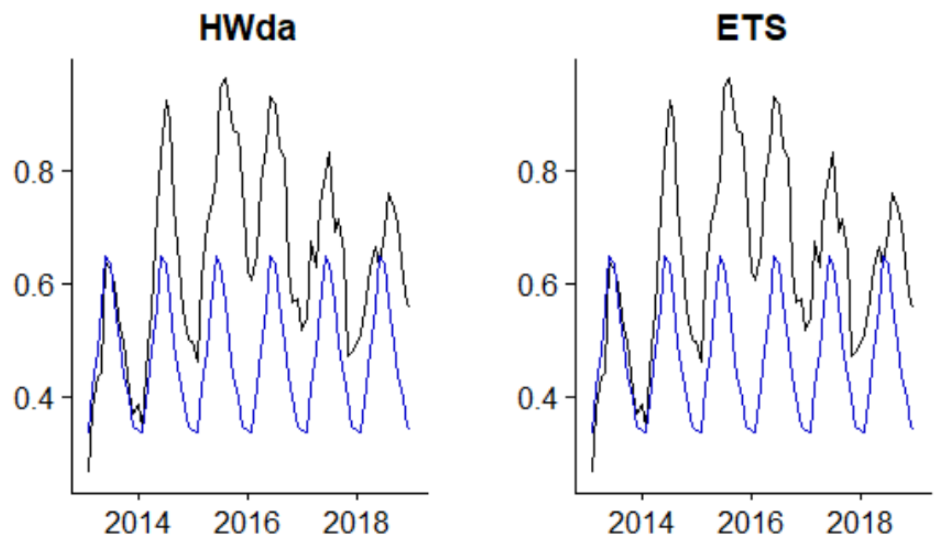
Fig. 5.4.3 ETS Forecast

The forecast matches the HW Damped Additive Method since it underpins the ETS model and the parameters are the same. The very small differences between the point forecasts and performance intervals of the two models represent the errors.

Fig. 5.4.4 Comparison of Forecasts for HWdA and ETS Against the Test Set

```
library("cowplot")
plot.1<-autoplot(sst.test) + autolayer(sst.hwda, PI = F) + ggtitle("HWda") +
xlab("Year") + ylab("")
plot.2<-autoplot(sst.test) + autolayer(sst.fc.ets, PI = F) + ggtitle("ETS") +
```

```
xlab("Year") + ylab("")
plot_grid(plot.1, plot.2)
```



Checking the accuracy against the test set, as expected the results are identical to the HW Damped Additive method:

```
sst.acc4<-accuracy(sst.fc.ets,sst.test)
d.table <- rbind(sst.acc4,sst.acc3)
row.names(d.table)<-c("ETS AAdA Train","ETS AAdA Test", "HW Add Damp Train",
"HW Add Damp Test")
d.table<-as.data.frame(d.table)
kable(d.table, caption = "Table 1.12 Accuracy Measures for the ETS and Associ
ated HW Method", digits = 3)
```

Table 1.12 Accuracy Measures for the ETS and Associated HW Method

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
ETS AAdA Train	0.002	0.060	0.048	-8.358	29.942	0.430	0.301	NA
ETS AAdA Test	0.175	0.214	0.183	24.717	26.940	1.647	0.842	2.017
HW Add Damp Train	0.002	0.060	0.048	-8.358	29.942	0.430	0.301	NA
HW Add Damp Test	0.175	0.214	0.183	24.717	26.940	1.647	0.842	2.017

Tweaking the ETS model

1. The ETS(AAdA) model was tweaked by removing the damping to see if the accuracy could be improved.

```
sst.ets2<-ets(sst.train, damped = FALSE)
sst.fc.ets2<-forecast(sst.ets2, h = h)
sst.fc.ets2$model

## ETS(A,N,A)
##
## Call:
##  ets(y = sst.train, damped = FALSE)
```

```
##
##   Smoothing parameters:
##     alpha = 0.6179
##     gamma = 0.2981
##
##   Initial states:
##     l = -0.079
##     s = 0.0141 -0.0064 -0.1558 -0.0496 -0.056 0.0182
##            0.0495 0.0606 0.019 0.0616 0.0639 -0.0191
##
##   sigma:  0.0623
##
##       AIC      AICc       BIC
##  47.40942  49.17413 102.35383
```

This returns a model with no trend. Accuracy is not improved as shown in Table below:

```
sst.acc5<-accuracy(sst.fc.ets2,sst.test)
d.table <- rbind(sst.acc5)
row.names(d.table)<-c("ETS ANA Train","ETS ANA Test")
d.table<-as.data.frame(d.table)
kable(d.table, caption = "Table 1.13 Accuracy Measures for the ETS ANA Model"
, digits = 3)
```

Table 1.13 Accuracy Measures for the ETS ANA Model

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
ETS ANA Train	0.003	0.061	0.048	-3.578	30.645	0.434	0.258	NA
ETS ANA Test	0.182	0.219	0.189	26.031	27.687	1.693	0.835	2.089

2. Then trying an "AAA" model so the trend dampening is supressed:

```
sst.ets3<-ets(sst.train, model = "AAA", damped = NULL)
sst.ets3
```

```
## ETS(A,Ad,A)
##
## Call:
##  ets(y = sst.train, model = "AAA", damped = NULL)
##
##   Smoothing parameters:
##     alpha = 0.6007
##     beta  = 0.0225
##     gamma = 0.2139
##     phi   = 0.8129
##
##   Initial states:
##     l = -0.1867
##     b = 0.0432
##     s = -0.0293 -0.0695 -0.0897 -0.0746 -0.0716 -0.0289
##            0.0543 0.0758 0.1022 0.0426 0.0835 0.005
```

```
##
##   sigma:  0.0616
##
##        AIC       AICc        BIC
##  44.42709  46.96984 110.36038
```

The accuracy does not improve on the HW AA method.

```
sst.fc.ets3<-forecast(sst.ets3, h = h)
sst.acc6<-accuracy(sst.fc.ets3,sst.test)
d.table <- rbind(sst.acc6,sst.acc2)
row.names(d.table)<-c("ETS AAA Train","ETS AAA Test","HW AA Train", "HW AA Te
st")
d.table<-as.data.frame(d.table)
kable(d.table, caption = "Table 1.14 Accuracy Measures for the ETS AAA and HW
AA", digits = 3)
```

Table 1.14 Accuracy Measures for the ETS AAA and HW AA

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
ETS AAA Train	0.002	0.060	0.048	-8.358	29.942	0.430	0.301	NA
ETS AAA Test	0.175	0.214	0.183	24.717	26.940	1.647	0.842	2.017
HW AA Train	-0.001	0.075	0.058	-2.986	37.785	0.518	0.524	NA
HW AA Test	0.085	0.143	0.107	10.702	15.175	0.958	0.838	1.264

Check the residuals of ETS (AAA):

```
checkresiduals(sst.ets3)
```


```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,Ad,A)
## Q* = 63.513, df = 7, p-value = 2.99e-11
```

```
##
## Model df: 17.   Total lags used: 24
```

```
res.sst5<-residuals(sst.ets3)
shapiro.test(res.sst5)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.sst5
## W = 0.99785, p-value = 0.9722
```

The ACF has a couple of larger and a few smaller spikes and the Ljung-Box is still very small. The residuals apppear to have constant mean and variance with some 'choppiness' at the start of the series. The residuals are not normally distributed.

Comparing the ETS models:

```
d1<-accuracy(sst.fc.ets,sst.test)
d2<-accuracy(sst.fc.ets2,sst.test)
d3<-accuracy(sst.fc.ets3,sst.test)
library("knitr")
a.table <- rbind(d1,d2,d3)
row.names(a.table)<-c("ETS (AAdA) Train","ETS(AAdA) Test","ETS (ANA) Train","
ETS(ANA) Test", "ETS(AAA) Train", "ETS(AAA) Test")
a.table<-as.data.frame(a.table)
kable(a.table, caption= "Table 1.15 Summary of Forecast Accuracy of ETS Model
s and HW Methods",digits = 3 )
```

Table 1.15 Summary Forecast Accuracy of ETS Models and HW Methods

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
ETS (AAdA) Train	0.002	0.060	0.048	-8.358	29.942	0.430	0.301	NA
ETS(AAdA) Test	0.175	0.214	0.183	24.717	26.940	1.647	0.842	2.017
ETS (ANA) Train	0.003	0.061	0.048	-3.578	30.645	0.434	0.258	NA
ETS(ANA) Test	0.182	0.219	0.189	26.031	27.687	1.693	0.835	2.089
ETS(AAA) Train	0.002	0.060	0.048	-8.358	29.942	0.430	0.301	NA
ETS(AAA) Test	0.175	0.214	0.183	24.717	26.940	1.647	0.842	2.017
HWDA Train	0.002	0.060	0.048	-8.358	29.942	0.430	0.301	NA
HWDA Test	0.175	0.214	0.183	24.717	26.940	1.647	0.842	2.017
HW AA Train	0.001	0.075	0.058	-2.986	37.785	0.518	0.524	NA
HW AA Test	0.085	0.143	0.107	10.702	15.175	0.958	0.838	1.264

The HW Method with additive trend and seasonality performs the best against the test set compared to all the ETS models. However, as the ETS models also provide performance intervals, helpful for forecasting, the ETS(AAA) may be preferable.

Question 5.5

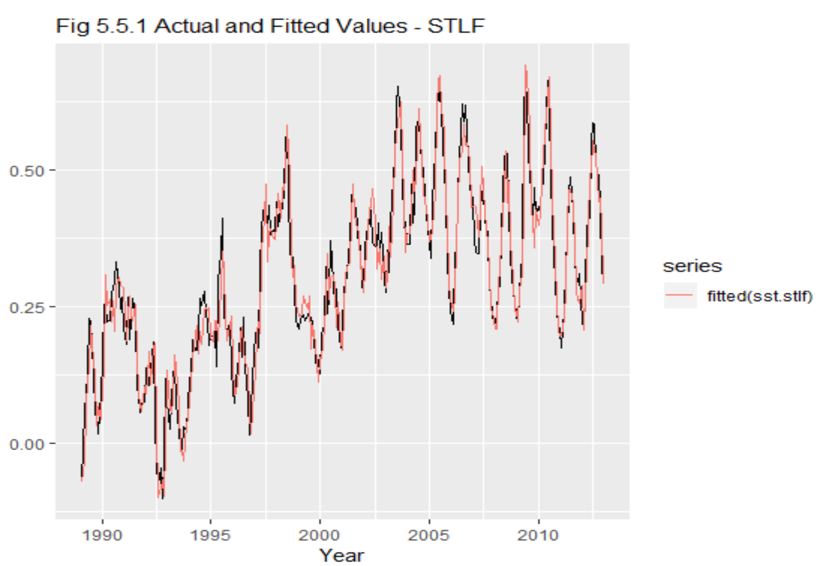
Seasonal and Trend decomposition using Loess

A seasonal decomposition returns a model with no trend or seasonality, so the forecast is based purely on additive errors:

```
sst.stlf<-stlf(sst.train, lambda = "auto", s.window = 13,method = "ets", h =
72)
sst.stlf$model
```

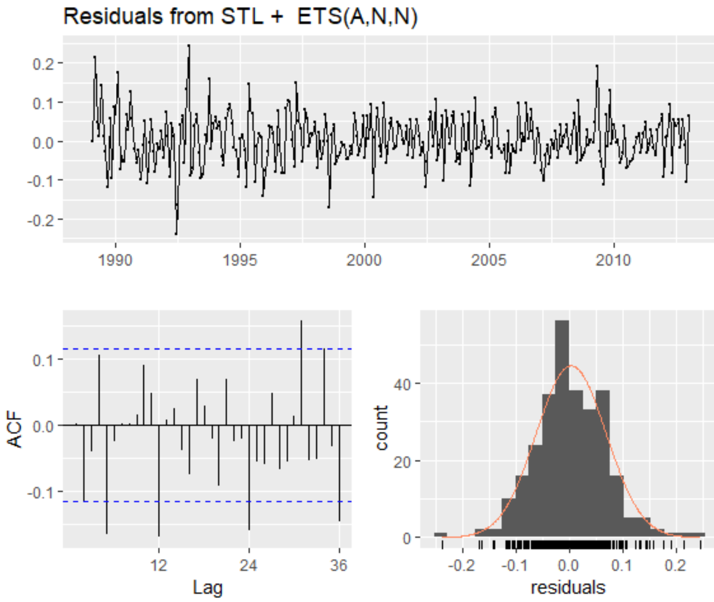
```
## ETS(A,N,N)
##
## Call:
##  ets(y = x, model = etsmodel, allow.multiplicative.trend = allow.multiplic
ative.trend)
##
##   Smoothing parameters:
##     alpha = 0.9999
##
##   Initial states:
##     l = -1.3702
##
##   sigma:  0.0658
##
##      AIC     AICc      BIC
## 67.94197 68.02647 78.93085
```

```
autoplot(sst.train)+ autolayer(fitted(sst.stlf)) + xlab("Year") + ylab("") +
ggtitle("Fig 5.5.1 Actual and Fitted Values - STLF") + theme(plot.title = ele
ment_text(size = 12))
```




```
checkresiduals(sst.stlf)
```

```
## Warning in checkresiduals(sst.stlf): The fitted degrees of freedom are bas
ed
## on the model used for the seasonally adjusted data.
```




```
##
##  Ljung-Box test
##
## data:  Residuals from STL +  ETS(A,N,N)
## Q* = 44.299, df = 22, p-value = 0.003253
##
## Model df: 2.   Total lags used: 24
```

```
res.sst5<-residuals(sst.stlf)
shapiro.test(res.sst5)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.sst5
## W = 0.98938, p-value = 0.03381
```

Observations

There is still autocorrelation in the residuals, the mean and variance appear stable and the residuals are not normally distributed.

```
autoplot(sst.stlf)
```

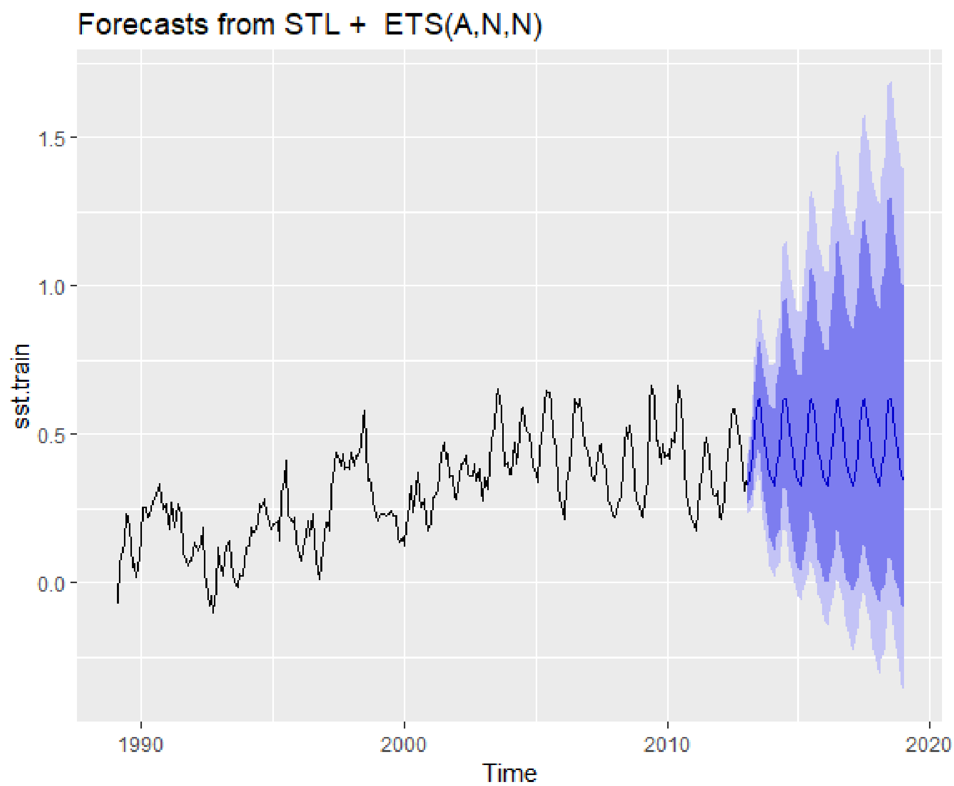


Fig. 5.5.2 STL + ETS(ANN) Forecast

```
sst.acc8<-round(accuracy(sst.stlf,sst.test),3)
a.table <- rbind(sst.acc8)
row.names(a.table)<-c("STLF Train","STLF Test")
a.table<-as.data.frame(a.table)
kable(a.table, caption= "Table 1.16 Forecast accuracy of STLF Model",digits =
5)
```

Table 1.16 Forecast accuracy of STLF Model

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
STLF Train	0.002	0.048	0.038	-5.030	23.087	0.340	-0.03	NA
STLF Test	0.185	0.220	0.189	26.219	27.556	1.698	0.84	2.097

Forecast accuracy has not been improved with STLF.

Question 5.6

ARIMA models

The series was transformed to stabilise the variance.

```
sst.trans<-BoxCox(sst.train, lambda = "auto")
```

Testing for stationarity:

```r
library("urca")
summary(ur.kpss(sst.trans))

##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 5 lags.
##
## Value of test-statistic is: 2.7473
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

The null hypothesis of stationarity can be rejected as the test statistic is greater than 1% significance level. Using the nsdiffs() function, one seasonal difference is required.

```r
nsdiffs(sst.trans)

## [1] 1
```

Checking for a first order difference:

```r
sst.diff<-diff(sst.trans, frequency = 12)
ndiffs(sst.diff)

## [1] 0
```

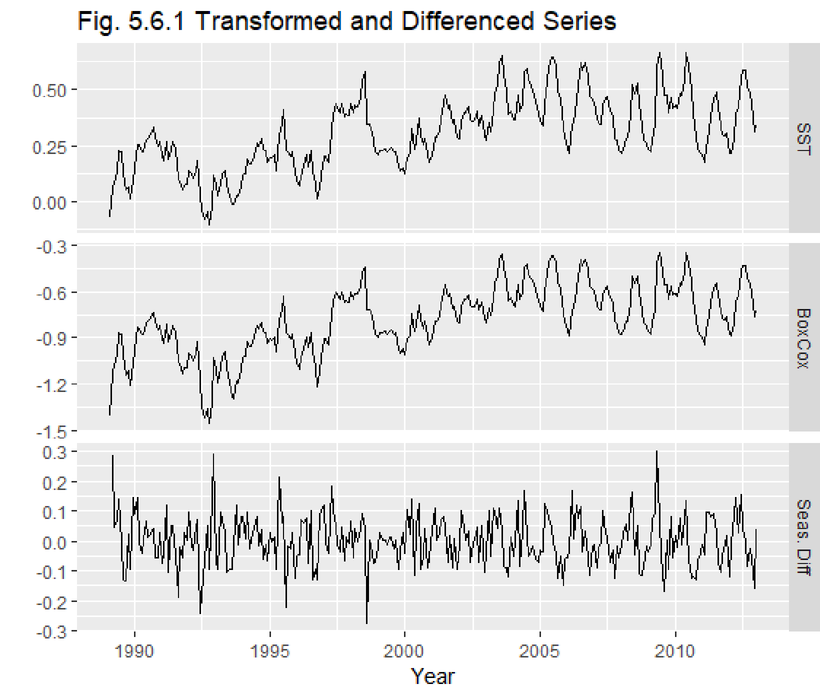
Only one order of seasonal difference is required. Checking with the Kpss test:

```r
library("urca")
summary(ur.kpss(sst.diff))

##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 5 lags.
##
## Value of test-statistic is: 0.0457
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

We can accept the null hypothesis of stationary data. Plotting the differenced series, it appears stationary.

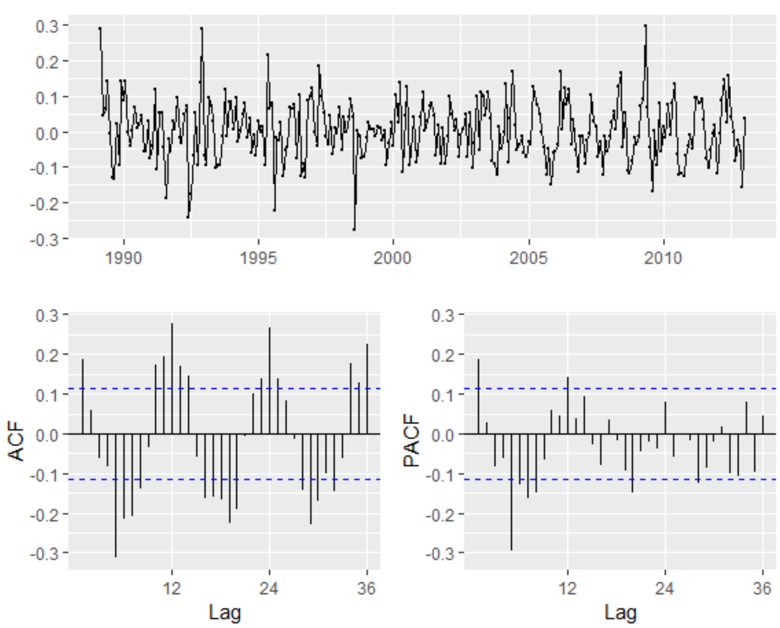
```r
cbind("SST" = sst.train,
      "BoxCox" = sst.trans,
      "Seas. Diff" = sst.diff) %>%
  autoplot(facets=TRUE) +
```

```
    xlab("Year") + ylab("") +
    ggtitle("Fig. 5.6.1 Transformed and Differenced Series")
```



Deriving the ACF and PACF plots:

```
ggtsdisplay(sst.diff)
```



A (p,d,q)(P,1,Q) model is required as there is one order of seasonal differencing.

Non-seasonal part:

The ACF has a sinusoidal pattern indicating AR terms. Looking at the PACF, there is one significant spike at lag one then drops away, so AR(1) model could be appropriate. Looking at the seasonal part the PACF has a significant spike at lag 12 with decay indicating an MA(1) term.

The following models were investigated:

```
sst.ar1 <- Arima(sst.train, order=c(1,0,0), seasonal=c(0,1,0), lambda="auto")
sst.ar2<-Arima(sst.train, order = c(2,0,0), seasonal = c(0,1,0), lambda= "aut
o")
sst.ar3<-Arima(sst.train, order = c(1,0,0), seasonal = c(1,1,0),lambda= "auto
")
sst.ar4<-Arima(sst.train, order = c(1,0,1), seasonal = c(1,1,0),lambda= "auto
")
sst.ar5<-Arima(sst.train, order = c(2,0,0), seasonal = c(1,1,0),lambda= "auto
")
sst.ar6<-Arima(sst.train, order = c(1,0,1), seasonal = c(0,1,1),lambda= "auto
")

sst.ar1$aicc

## [1] -501.6659

sst.ar2$aicc

## [1] -500.6926

sst.ar3$aicc

## [1] -565.3231

sst.ar4$aicc

## [1] -564.9843

sst.ar5$aicc

## [1] -564.848

sst.ar6$aicc

## [1] -620.2766
```

Using auto.arima() the following model was returned:

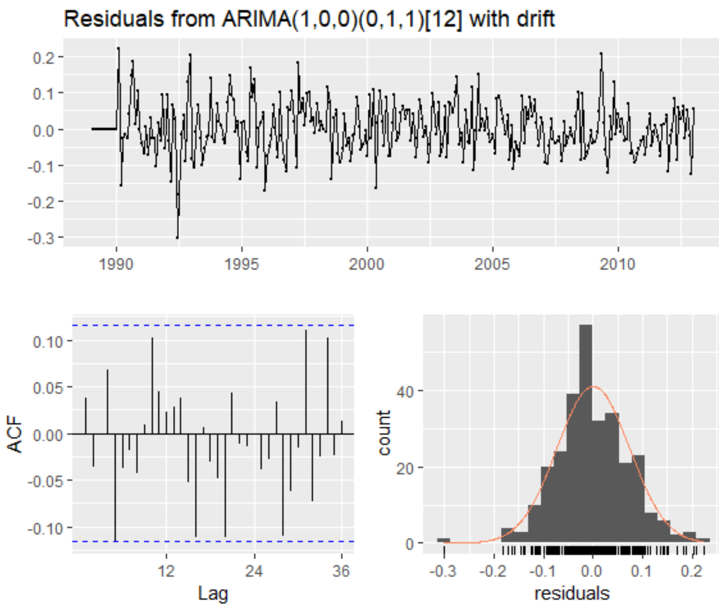
```
sst.ar7<-auto.arima(sst.train, lambda = "auto", stepwise = F, approximation =
F)
sst.ar7

## Series: sst.train
## ARIMA(1,0,0)(0,1,1)[12] with drift
## Box Cox transformation: lambda= 0.7980202
##
## Coefficients:
##          ar1     sma1   drift
##       0.8864  -0.8367  0.0018
```

```
## s.e.  0.0292   0.0453  0.0007
##
## sigma^2 estimated as 0.005656:  log likelihood=316.26
## AIC=-624.53   AICc=-624.38   BIC=-610.05
```

The auto.arima has the lowest AICc so is the preferred model.

Checking the residuals:

```
checkresiduals(sst.ar7)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0)(0,1,1)[12] with drift
## Q* = 22.332, df = 21, p-value = 0.3806
##
## Model df: 3.   Total lags used: 24
```

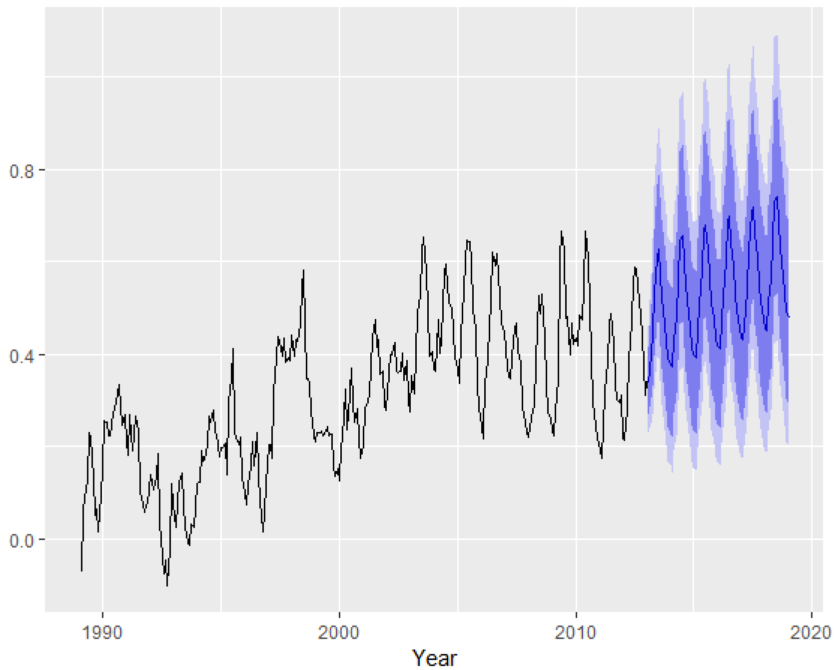
```
res.sst7<-residuals(sst.ar7)
shapiro.test(res.sst7)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.sst7
## W = 0.98997, p-value = 0.04506
```

Observations

The Arima model captured the information in the data well. There are no significant spikes on the ACF, and the Ljung-Box test statistic is above 0.05. The residuals plot has constant mean and variance, but histogram has a negative skew, so the residuals are not normally distributed. For forecasting, bootstrapping should be used.

```r
sst.fc.ar7<-forecast(sst.ar7, h=h, biasadj = TRUE, bootstrap = T)
autoplot(sst.fc.ar7) + ggtitle("Fig. 5.6.2 ARIMA SST Forecast")+ xlab("Year")
+ ylab("")
```



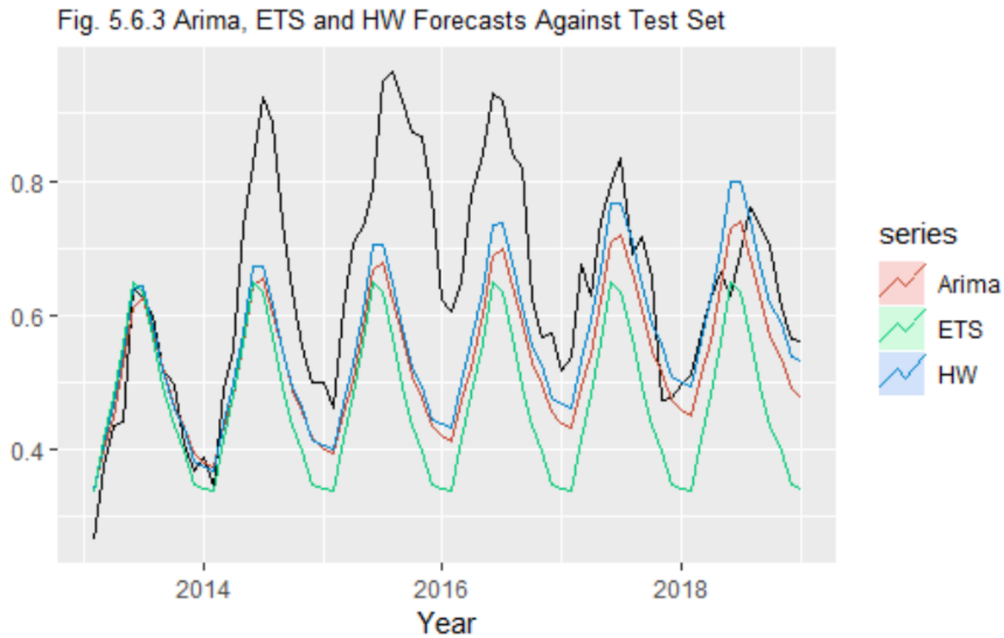

```r
sst.acc7<-accuracy(sst.fc.ar7,sst.test)
a.table <- rbind(sst.acc7)
row.names(a.table)<-c("Arima Train","Arima Test")
a.table<-as.data.frame(a.table)
kable(a.table, caption= "Table 5.5.1 Forecast accuracy of Arima Model",digits
= 4 )
```

Table 1.17 Forecast accuracy of Arima Model

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Arima Train	0.0009	0.0552	0.0430	-5.8623	23.7061	0.3859	0.0230	NA
Arima Test	0.1127	0.1585	0.1256	14.8056	17.8588	1.1277	0.8318	1.4232

Fig. 5.6.3 Compares the HW, ARIMA and ETS forecasting methods.

```r
autoplot(sst.test) + autolayer(sst.fc.ar7, PI = F, series = "Arima") + xlab("
Year") + ylab("") + ggtitle("Fig. 5.6.3 Arima, ETS and HW Forecasts Against T
est Set") + autolayer(sst.fc.ets3, PI = F, series = "ETS") + theme(plot.title
= element_text(size = 10))
```




```r
f.ets <- function(x, h) {
  forecast(ets(x), h = h)
}


f.arima <- function(x, h) {
  forecast(auto.arima(x), h=h)
}

e.1 <- tsCV(sst.train, f.ets, h=1)
e.2 <- tsCV(sst.train, f.arima, h=1)

mean(e.1^2, na.rm=TRUE)

## [1] 0.003868146

mean(e.2^2, na.rm=TRUE)

## [1] 0.003765318
```

Conclusions

Comparing all the models, the Seasonal Decompostion STL + ETS (ANN) modelled the training data best.

When run against the test set, the HW AAA model returned the best accuracy measures, closely followed by the Arima model which produced very similar forecasts.

Looking at the forecast compared to the test set in Fig. 5.6.3, the ETS model misses two of the larger peaks during the exceptional years associated with El Nino, which would be difficult for any model to predict but does manage to capture the seasonality and trend in the data reasonably well.

REFERENCES

Arndt, E. (2019). *Did You Know? | Monitoring References | National Centers for Environmental Information (NCEI)*. [online] Ncdc.noaa.gov. Available at: https://www.ncdc.noaa.gov/monitoring-references/dyk/anomalies-vs-temperature [Accessed 21 May 2019].

Climate.gov. (2019). *2015 State of the Climate: El Niño came, saw, and conquered | NOAA Climate.gov*. [online] Available at: https://www.climate.gov/news-features/understanding-climate/2015-state-climate-el-niño-came-saw-and-conquered [Accessed 21 May 2019].

Cowpertweit, P. and Metcalfe, A. (2009). *Introductory time series with R*. Dordrecht: Springer.

En.wikipedia.org. (2019). *Sea surface temperature*. [online] Available at: https://en.wikipedia.org/wiki/Sea_surface_temperature [Accessed 21 May 2019].

Hyndman, R. and Athanasopoulos, G. (2018). *Forecasting*. [Heathmont, Vic.]: OTexts.

Jmp.com. (2019). *35406 - How do I interpret the Shapiro-Wilk test for normality in JMP®?*. [online] Available at: http://www.jmp.com/support/notes/35/406.html [Accessed 19 May 2019].

Metoffice.gov.uk. (2019). *Met Office Hadley Centre observations datasets*. [online] Available at: https://www.metoffice.gov.uk/hadobs/hadsst3/data/download.html [Accessed 19 May 2019].

Ncdc.noaa.gov. (2019). *Global Surface Temperature Anomalies | Monitoring References | National Centers for Environmental Information (NCEI)*. [online] Available at: https://www.ncdc.noaa.gov/monitoring-references/faq/anomalies.php [Accessed 21 May 2019].

Newonlinecourses.science.psu.edu. (2019). *14.1 - Autoregressive Models | STAT 501*. [online] Available at: https://newonlinecourses.science.psu.edu/stat501/node/358/ [Accessed 19 May 2019].

Ons.gov.uk. (2019). *Internet retail sales, £ millions. All retailing - Office for National Statistics*. [online] Available at: https://www.ons.gov.uk/businessindustryandtrade/retailindustry/timeseries/je2j/drsi [Accessed 19 May 2019].

People.duke.edu. (2019). *Introduction to ARIMA models*. [online] Available at: https://people.duke.edu/~rnau/411arim.htm [Accessed 19 May 2019].

Robjhyndman.com. (2019). *Why doesn't auto.arima() return the model with the lowest AICc value? | Rob J Hyndman*. [online] Available at: https://robjhyndman.com/hyndsight/badarima/ [Accessed 23 May 2019].

Shmueli, G. and Lichtendahl, K. (n.d.). *Practical time series forecasting with R*.

test, I. and Hardy, R. (2019). *Interpretation of critical values of KPSS test*. [online] Cross Validated. Available at: https://stats.stackexchange.com/questions/160810/interpretation-of-critical-values-of-kpss-test [Accessed 19 May 2019].

We Make Websites. (2019). *16 important e-commerce shopping dates for your 2019 diary | We Make Websites*. [online] Available at: https://wemakewebsites.com/blog/16-important-e-commerce-shopping-dates-for-your-2019-diary [Accessed 26 May 2019].

APPENDIX 1

Wine Industry Report from the Commodity Review and Outlook 1989-1990

95

ports from Brazil, particularly during the second half of the year. The factors responsible for the reduction in 1988 included smaller requirements by the United States, where demand had also weakened because of high prices. Import demand in western Europe, accounting for about 50 percent of global net imports, was expected to strengthen in 1989 as a result of lower prices in the second half of the year. In Japan, imports of processed citrus were likely to expand gradually as a result of the agreement with the United States which calls for a progressive increase in the import quota of FCOJ to complete liberalization from quantitative restrictions in 1992.

Prices of FCOJ averaged around US$2 000/ton during the first part of 1989. Prices weakened during the second half, principally due to plentiful supply with contracts at the US$1 400/ton level being sought at the end of the year.

WINE

Highlights

World wine production recovered in 1989/90 from the low output of the previous year. With exports also expected to make some recovery, it was anticipated that prices would remain firm. However, there are long-term trends, including declining consumption and increasing self-sufficiency, which could limit the scope for expanded world trade in the future.

Some recovery in 1989/90 after small vintages in 1988/89

World wine production in 1989/90 increased by about 10 percent from the low output of the previous year. Nevertheless, it was expected to be some 7 percent less than the average of the mid-1980s as measures to restrict supplies continued to be in effect in many major producing areas. Although production recovered in the European Community, this was principally because of substantially improved output by Spain and Portugal following their weather-damaged 1988/89 crops. Increases in production were also forecast for Italy and the Federal Republic of Germany. Preliminary reports from Italy indicated that production of wine of controlled appellations of origin increased by 11 percent, which was sufficient to offset the fall in production of table wines. However, the 3 percent increase in production of wine of controlled appellations of origin in France did not counterbalance the 5 percent fall in production of table wines.

Among other major producing countries, moderate increases in output in 1989/90 were anticipated in Greece and the USSR, whereas in Australia production of wine rose by about 20 percent because of an exceptionally large wine grape harvest. Crops in Austria, Chile, Hungary, Yugoslavia and the United States were expected to be below those of the previous year.

World wine production in 1988/89 was almost 20 percent less than output in the previous year principally because of sharp declines in production in the European Community following large crop losses in Spain and Portugal and substantially reduced crops in France and Italy. Exceptionally cold weather in 1988 followed by excessive rainfall and hailstorms and a severe outbreak of mildew reduced the vintage of Spain, the lowest for over ten years, to a little over half the 1987/88 volume. Output in France of 56.6 million hectolitres was reduced following cold weather and excessive rainfall. The lower production was primarily in the output of table wines and wines for the production of cognac, whereas production of wines of controlled appellations of origin suffered comparatively less. The reduction in the output of Italy was confined to table wines, with no decline occurring in the production of wine of controlled appellation of origin.

In the rest of the world, increased output was experienced by Argentina and the United States, where favourable weather in California compensated for reduced crops elsewhere in the country. Production in the USSR showed some increase from the previous year but remained almost a quarter below the average of the mid-1980s following efforts to discourage the production and consumption of alcoholic beverages. Production in Australia recovered considerably to levels that approximated average output during the first half of the 1980s.

In world terms, there has been a reduction of over 10 percent in the area planted to vines since the peak of the late 1970s. Reductions in the order of 20 percent have occurred in France and Italy, the world's largest producing countries, reflecting the European Community's efforts to rationalize the wine sector. Substantial reductions have also occurred in North Africa with a more moderate decline in the USSR.

Some export recovery expected in 1989

World wine exports were expected to make some recovery in 1989 following reduced trade in the previous two years. This occurred despite a continuing reduction in world consumption because of the need to supplement domestic availabilities in certain producing countries. Preliminary estimates have indicated a substantial increase in shipments by the European Community. Much of this is intra-Community trade since France, Italy and Spain are the principal suppliers of wine to other members of the Community, to restore stocks, supplement domestic availabilities from the reduced 1988/89

1988 was a good weather year for Australian wineries and Penfold Grange 88 Bin 95 Shiraz was an award-winning vintage.

As the grapes are picked in February to April and wines are ready much earlier, the problems in European supply with picking later in the year could have accounted for the spike in July 1988 sales .Wines from Australia also increased in popularity worldwide from this time, possibly after more people had had the chance to try them.

Note. The data shows total red wine sales and does not discriminate between wines with different fermentation times

(Google Books, 2019)

APPENDIX 2

Background Information on Sea Surface Temperature Measurement

- ***Measurements of SST have had inconsistencies over the last 130 years due to the way they were taken.*** In the nineteenth century, measurements were taken in a bucket off of a ship. However, there was a slight variation in temperature because of the differences in buckets. Samples were collected in either a wood or an uninsulated canvas bucket, but the canvas bucket cooled quicker than the wood bucket.
- Weather satellites have been available to determine sea surface temperature information since 1967, with the first global composites created during 1970.[10] Since 1982,[11] satellites have been increasingly utilized to measure SST and have allowed its spatial and temporal variation to be viewed more fully. Satellite measurements of SST are in reasonable agreement with in situ temperature measurements.

Source: https://en.wikipedia.org/wiki/Sea_surface_temperature

Temperature Anomalies

In climate change studies, temperature **anomalies** are more important than absolute temperature. A temperature anomaly is the difference from an average, or *baseline*, temperature. The baseline temperature is typically computed by averaging 30 or more years of temperature data. A **positive anomaly** indicates the observed temperature was **warmer** than the baseline, while a **negative anomaly** indicates the observed temperature was **cooler** than the baseline.

When calculating an average of absolute temperatures, things like station location or elevation will have an effect on the data (ex. higher elevations tend to be cooler than lower elevations and urban areas tend to be warmer than rural areas). However, when looking at anomalies, those factors are less critical. For example, a summer month over an area may be cooler than average, both at a mountain top and in a nearby valley, but the absolute temperatures will be quite different at the two locations.

Using anomalies also helps minimize problems when stations are added, removed, or missing from the monitoring network.

https://www.ncdc.noaa.gov/monitoring-references/dyk/anomalies-vs-temperature