ECON2209 – Course Project 2021

Important:

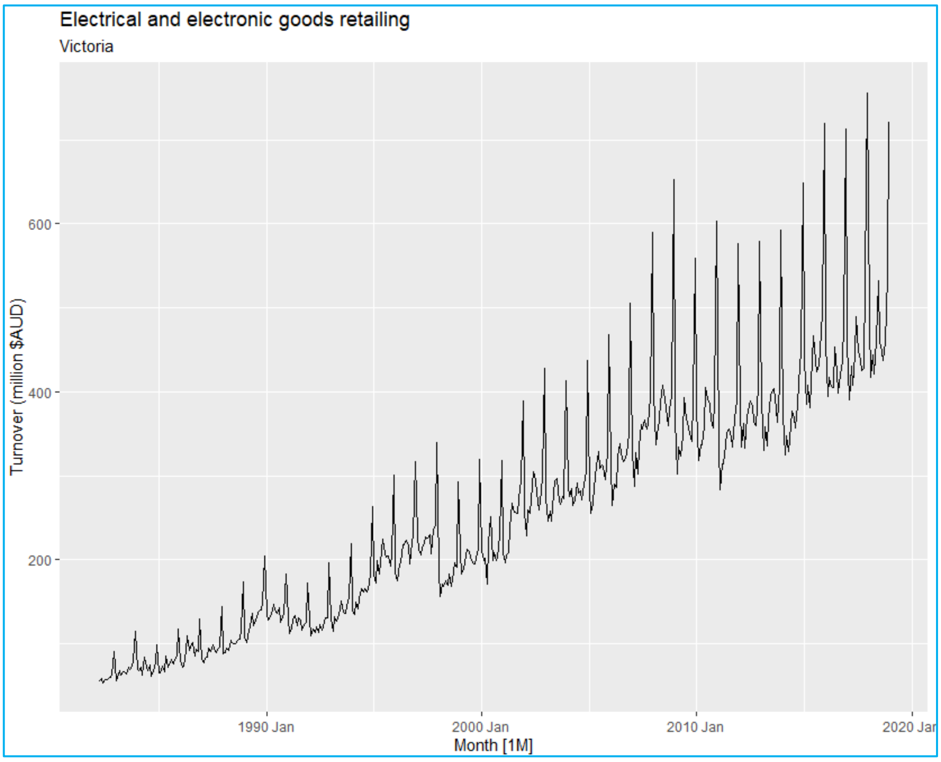
- **Red outlined indicates R code**
- **Blue outlined indicates R output**

Part 1: Data Exploration and Transformation

a. **Plot your data using the following code:**

```
myseries %>%
  autoplot(Turnover) +
  labs(y = "Turnover (million $AUD)",
       title = myseries$Industry[1],
       subtitle = myseries$State[1])
```

Figure 1: Electrical and electronic goods retailing in Victoria (1982 – 2018)



From the plot in figure 1, there is a clear indication of an increasing trend of electrical goods retailing in Victoria between 1982 and 2018. The variation in the plot is increasing, specifically, a noticeable increase between 1990 and 2010. A transformation is required to organise the data set and to understand the variation, we shall start with a log transformation.

```
myseries %>%
  autoplot(log(Turnover)) +
  labs(y = "Log Turnover (million $AUD)",
       title = myseries$Industry[1],
       subtitle = "log transformation on aus_retail")
```

Figure 2: Log transformation on Electrical and electronic goods retailing in Victoria (1982 – 2018)

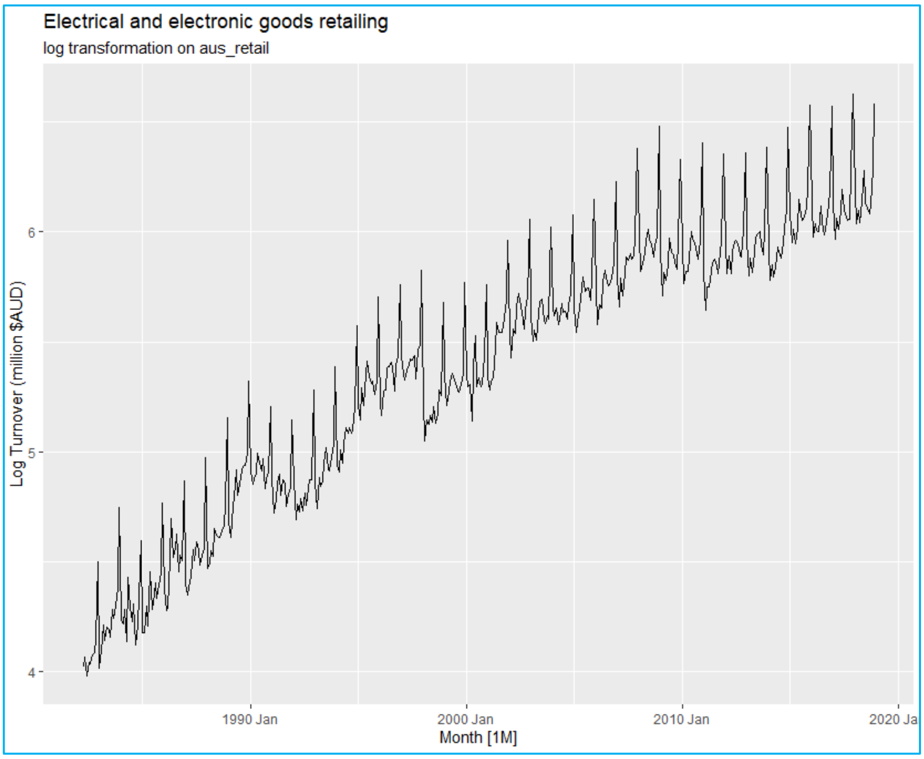
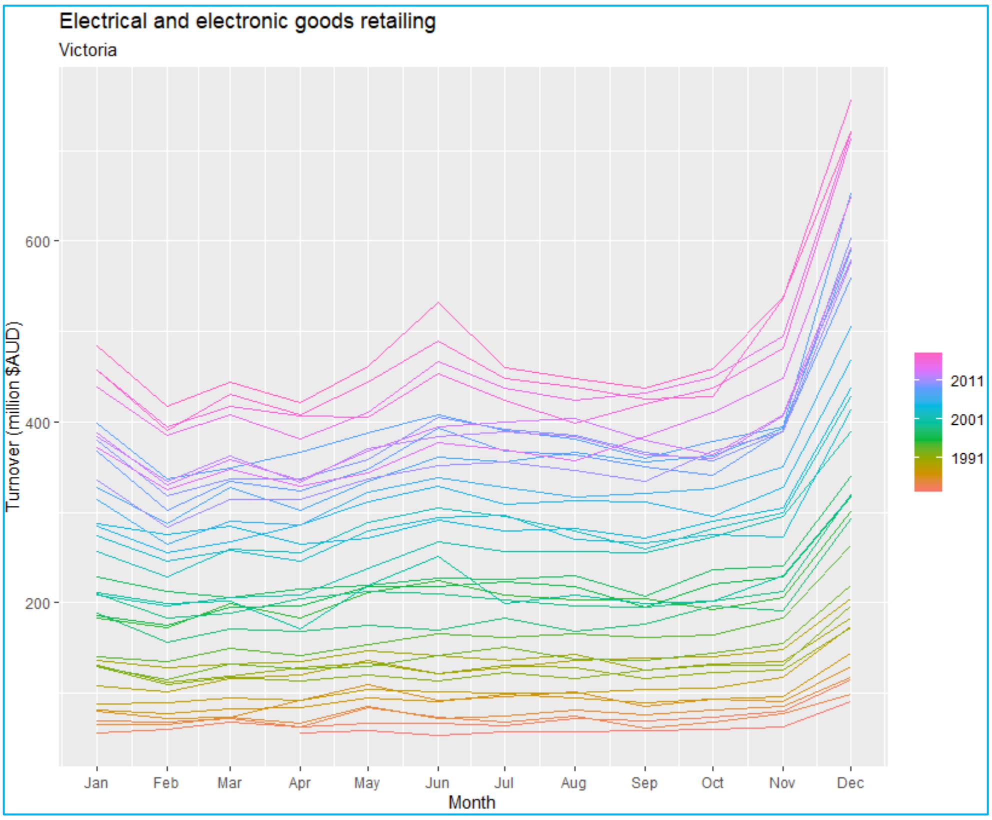


Figure 2 shows that there seems to be a constant variation across different levels of the series, a log transformation seems to allow for a more constant variation throughout the data series. We will investigate other transformations in part c of this problem.

b. Now explore your retail time series using the following functions, being sure to discuss what you find: gg_season(), gg_subseries(), gg_lag(), ACF() %>% autoplot()

```
myseries %>%
  gg_season(Turnover) + labs(y = "Turnover (million $AUD)",
                             title = myseries$Industry[1],
                             subtitle = myseries$State[1])
```

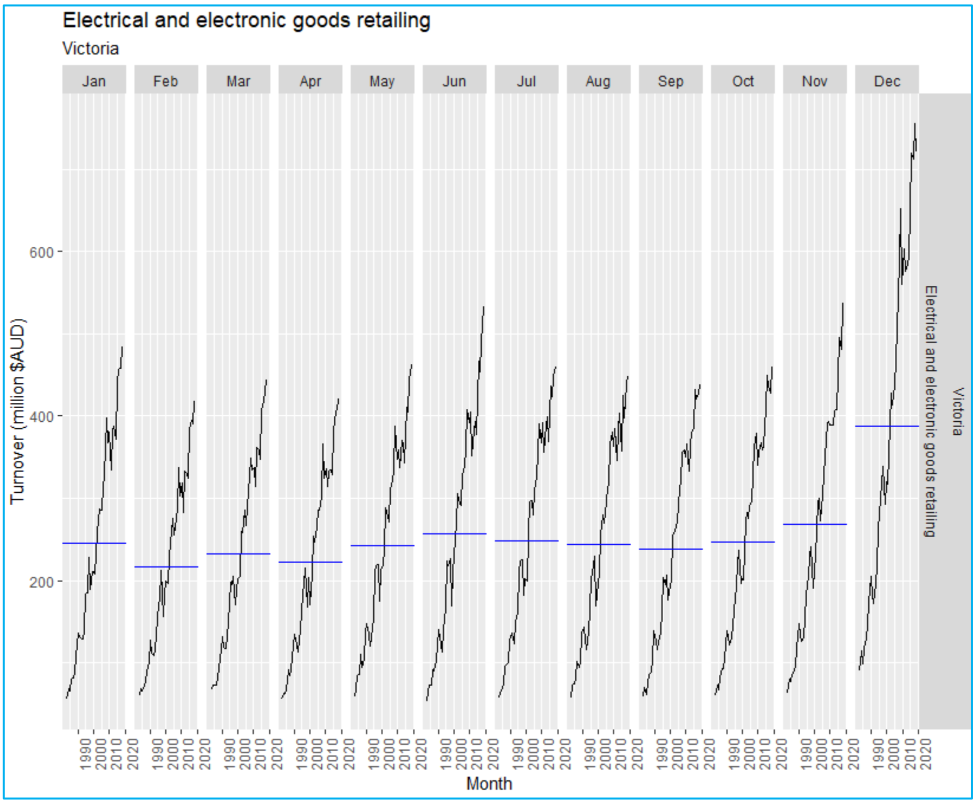
Figure 3: Seasonal plot of electrical and electronic goods retailing in Victoria (1982 – 2018)



The seasonal plot in figure 3, depicts seasonal effects. We can see that there is a dramatic increase in December which may be due to the introduction of summer sales which caused the population to purchase more electrical goods. Another noticeable aspect is in June, as there is a spike possibly due to mid-season sales. Overall, there is evidence of seasonality and in general throughout the years, turnover increases.

```
myseries %>%
  gg_subseries(Turnover) + labs(y = "Turnover (million $AUD)",
                               title = myseries$Industry[1],
                               subtitle = myseries$State[1])
```

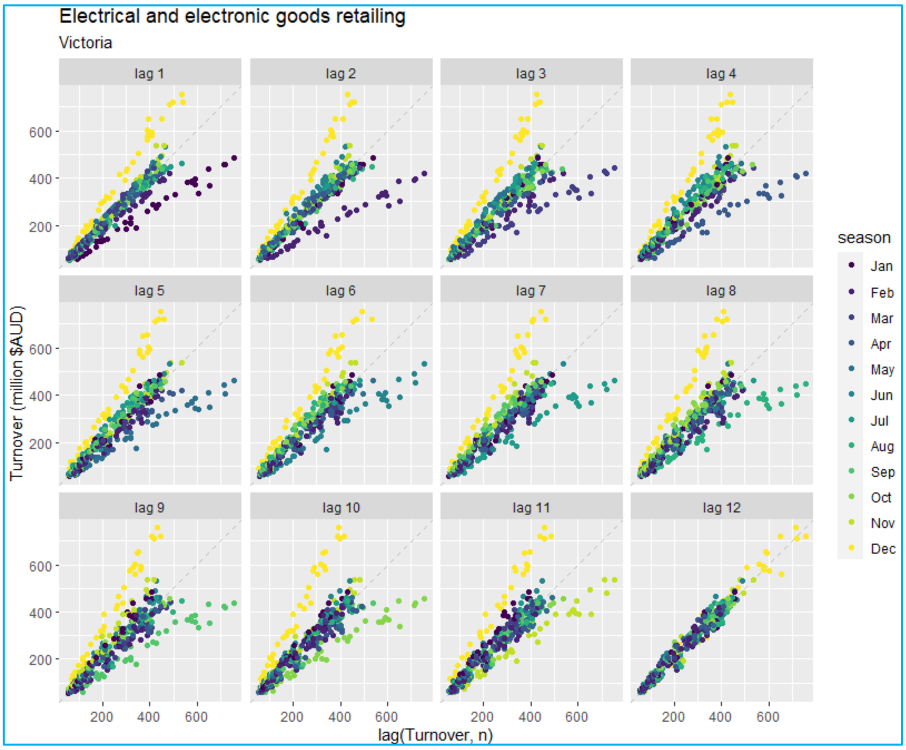
Figure 4: Subseries plot of electrical and electronic goods retailing in Victoria (1982 – 2018)



The subseries output displayed in figure 4, exemplifies that there is increasing seasonal variation because the trends seem to be greater in the higher peaking months. Another noticeable feature is that average (indicated by the blue horizontal lines in each month) increases and decreases slightly throughout the months. However, in December, the average increases close to 400 million dollar which may be due to holiday sales on electronic goods in Victoria.

```
myseries %>%
  gg_lag(Turnover, geom = 'point', lags = 1:12) + labs(y = "Turnover (million $AUD)",
                                                      title = myseries$Industry[1],
                                                      subtitle = myseries$State[1])
```

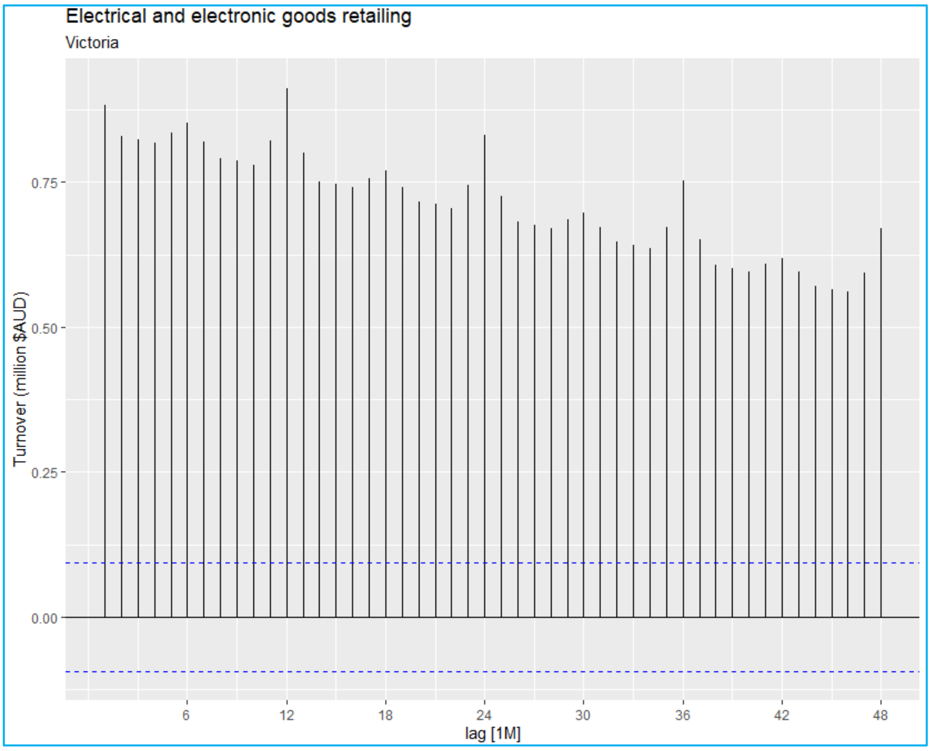
Figure 5: Lag plot of electrical and electronic goods retailing in Victoria (1982 – 2018)



The lag plot produced in figure 5, contains data points in colours which indicate the different months for each lag. The relationship is strongly positive in lag 12, which is also the most linearly plotted lag compared to the others. The other lags seem to be quite similar in appearance, with no other notable feature in the lag plot of worth mentioning.

```
myseries %>%
  ACF(Turnover, lag_max = 48) %>%
  autoplot() + labs(y = "Turnover (million $AUD)",
                    title = myseries$Industry[1],
                    subtitle = myseries$State[1])
```

Figure 6: Autocorrelation Function plot of electrical and electronic goods retailing in Victoria (1982 – 2018)



In the Autocorrelation Function (ACF) output of figure 6, we can see that there is a steady decreasing relationship between y_t and the previous month. The ACF shows evidence of seasonality, as seen from lag 12, 24, 36, and 48, showing that the month of December had large sales throughout the years.

c. What Box-Cox transformation, if any, would you select for your data, and why?

To stabilise the variance within the data series, we need to apply a box-cox transformation.

```
myseries %>%
  features(Turnover, features = guerrero)
```

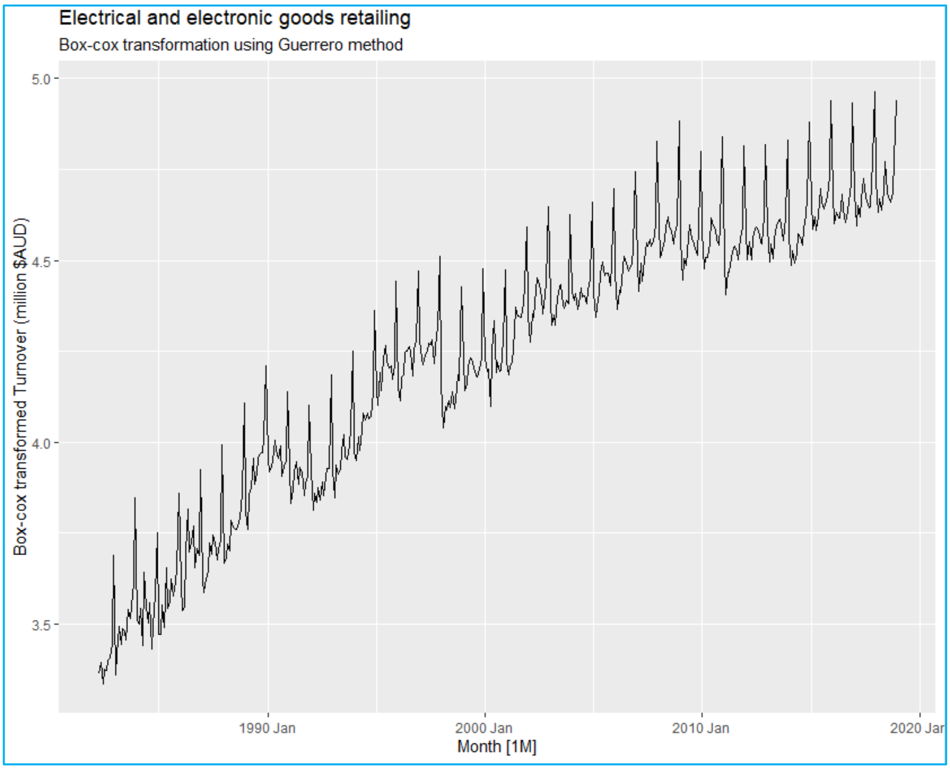
```
# A tibble: 1 x 4
  State    Industry                                  lambda_guerrero .name_repair
  <chr>    <chr>                                               <dbl> <chr>
1 Victoria Electrical and electronic goods retailing         -0.0918 minimal
```

Using the guerrero method, we obtain a lambda value of -0.0918 which is an inverse transformation. Let us plot this and see if the variance has been stabilised. Remember, that the guerrero method is a method to obtain a lambda value for a box-cox transformation, but it is not necessarily the best value to utilise.

```
myseries %>%
  features(Turnover, features = guerrero)

myseries_lambda <- myseries %>%
  features(Turnover, guerrero)
myseries %>%
  autoplot(box_cox(Turnover, -0.0918)) +
  labs(y = "Box-cox transformed Turnover (million $AUD)",
       title = myseries$Industry[1],
       subtitle = "Box-cox transformation using Guerrero method")
```

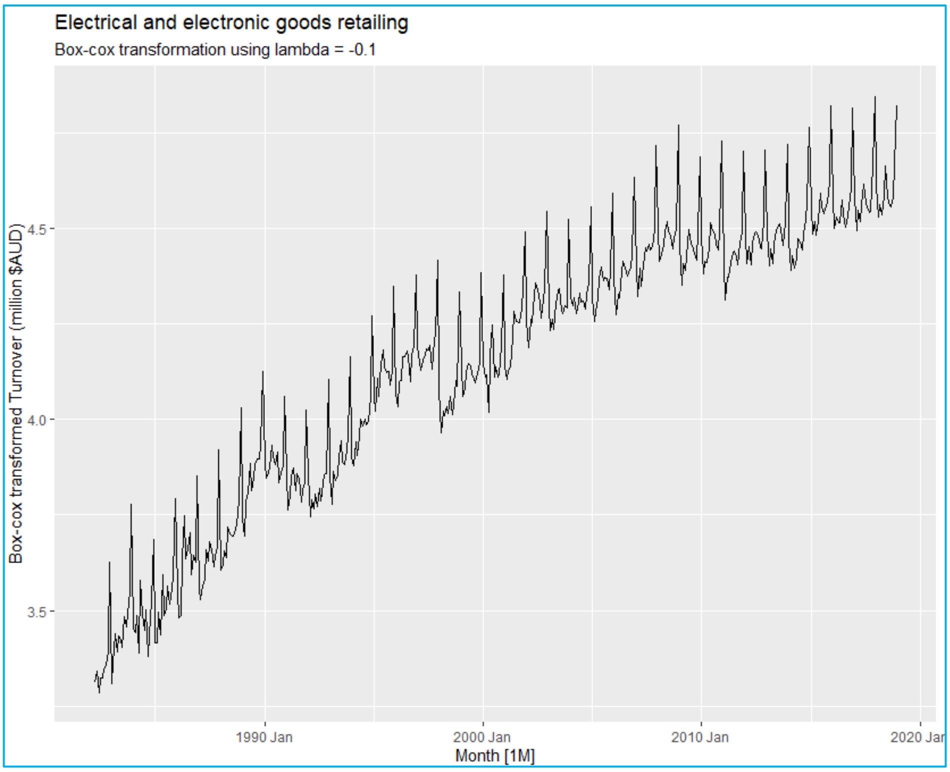
Figure 7: Box-cox transformation ($\lambda = -0.0918$) on electrical and electronic goods retailing in Victoria (1982 – 2018)



According to the box-cox transformed plot above, we can see that the variance has been relatively stable throughout the plot. Let us try a lambda value of -0.1 to see if the variance is more stable.

```
myseries %>%
  autoplot(box_cox(Turnover, -0.1)) +
  labs(y = "Box-cox transformed Turnover (million $AUD)",
       title = myseries$Industry[1],
       subtitle = "Box-cox transformation using lambda = -0.1")
```

Figure 8: Box-cox transformation ($\lambda = -0.1$) on electrical and electronic goods retailing in Victoria (1982 – 2018)



The variance within the data series is relatively stable as well with a lambda of -0.1. There seems to be almost no difference between this plot and the guerrero generated box-cox transformed plot in figure 7. Therefore, we can conclude that a box-cox transformation of lambda -0.1 would also be appropriate to stabilise the variance.

Part 2: Forecasting

For your untransformed data series:

a. **Create a training dataset (myseries_train) consisting of observations before 2011.**

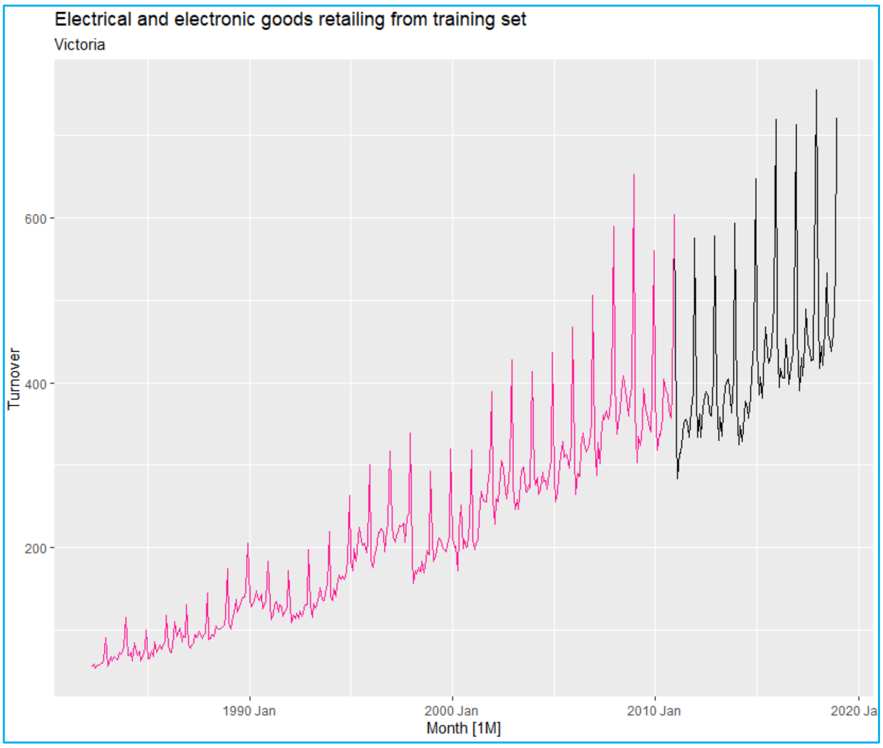
```
myseries_train <- myseries %>%
  filter(year(Month) < 2011)
myseries_train
```

```
A tsibble: 345 x 5 [1M]
Key:       State, Industry [1]
  State    Industry                              `Series ID`    Month Turnover
  <chr>    <chr>                                 <chr>          <mth>    <dbl>
  Victoria Electrical and electronic goods retailing A3349564W   1982 Apr     55.8
  Victoria Electrical and electronic goods retailing A3349564W   1982 May     58.4
  Victoria Electrical and electronic goods retailing A3349564W   1982 Jun     5[illegible]
  Victoria Electrical and electronic goods retailing A3349564W   1982 Jul     5[illegible]
  Victoria Electrical and electronic goods retailing A3349564W   1982 Aug     5[illegible]
  Victoria Electrical and electronic goods retailing A3349564W   1982 Sep     58.9
  Victoria Electrical and electronic goods retailing A3349564W   1982 Oct     59.6
  Victoria Electrical and electronic goods retailing A3349564W   1982 Nov     63.2
  Victoria Electrical and electronic goods retailing A3349564W   1982 Dec     90.3
  Victoria Electrical and electronic goods retailing A3349564W   1983 Jan     55.5
... with 335 more rows
```

b. **Check that your data have been split appropriately by producing a plot of myseries_train**

```
autoplot(myseries, Turnover) +
  autolayer(myseries_train, Turnover, colour = 'deeppink') +
  labs(title = "Electrical and electronic goods retailing from training set",
       subtitle = "Victoria")
```

Figure 9: Electrical and electronic goods retailing in Victoria before 2011 (1982 – 2018)



From figure 9, we can confirm that the training set consists of the data points before 2011, indicated by the pink line. The pink section of the plot starts from 1982 to 2010.

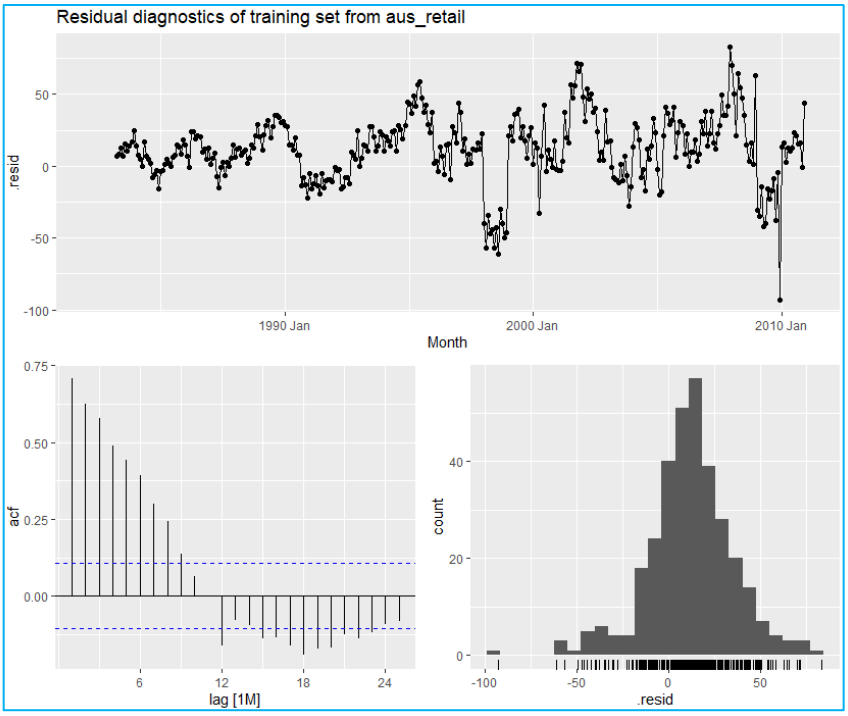
c. Calculate seasonal naive forecasts using SNAIVE() applied to your training data.

```
fit_new <- myseries_train %>%
  model(SNAIVE(Turnover))
```

d. Check the residuals. Do the residuals appear to be uncorrelated and normally distributed?

```
fit_new %>% gg_tsresiduals() +
  labs(title = "Residual diagnostics of training set from aus_retail")
```

Figure 10: Residual diagnostics of electrical and electronic goods retailing in Victoria before 2011

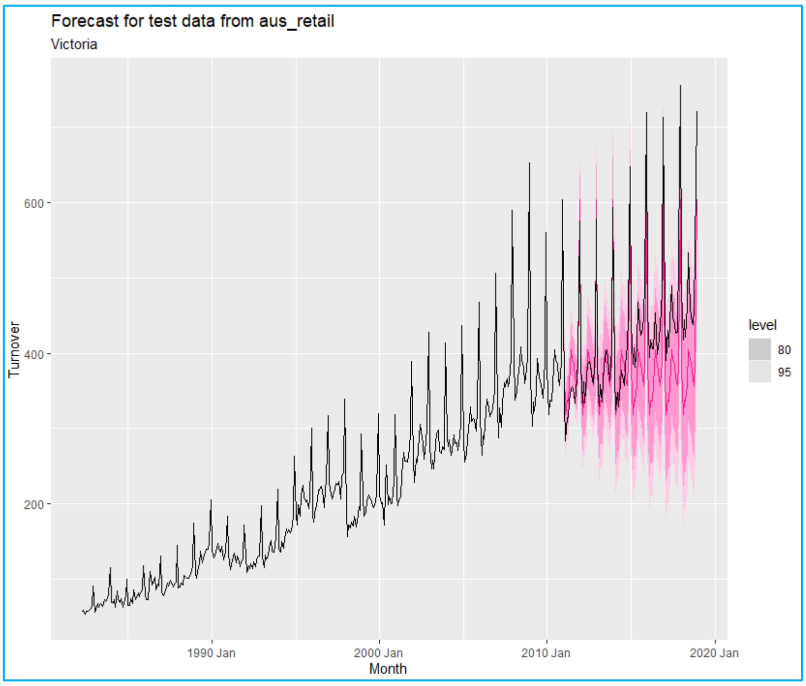


The residual diagnostics of the seasonally naïve model for electrical and electronic goods retailing in Victoria before 2011 is shown in figure 10. Some of the information or data points are captured in the error term, indicating that the residuals appear to be correlated. The histogram above is slightly left tailed, which means it is not quite normally distributed as we are not observing a perfectly bell-shaped histogram.

e. Produce forecasts for the test data.

```
fc <- fit_new %>%
  forecast(new_data = anti_join(myseries, myseries_train))
fc %>% autoplot(myseries, colour = 'deeppink')
```

Figure 11: Forecast of Electrical and electronic goods retailing in Victoria before 2011 (1982 – 2018)



f. Compare the accuracy of your forecasts against the actual values.

```
fit_new %>% accuracy()
fc %>% accuracy(myseries)
```

```
# A tibble: 1 x 12
  State    Industry                 .model      .type     ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
  <chr>    <chr>                    <chr>       <chr>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 Victor~ Electrical and electr~ SNAIVE(Tu~ Train~  11.7  26.2  20.4  5.75  10.4     1     1 0.707
```

```
# A tibble: 1 x 12
  .model       State    Industry                 .type     ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
  <chr>        <chr>    <chr>                    <chr>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 SNAIVE(Tur~ Victo~ Electrical and electro~ Test    38.4  62.4  49.3  7.92  11.0  2.42  2.38 0.870
```

The accuracy of the seasonal naïve and the test set can be interpreted from the table above. We can see that the RMSE (Root Mean Squared Error) is smaller for the seasonal-naive model, 26.2 compared to the test set, 62.4. This indicates to us that the seasonal naïve forecasting model is a better model compared to the test set. Overall, the training set seems to have better forecasting performance.

Part 3: Exponential Smoothing

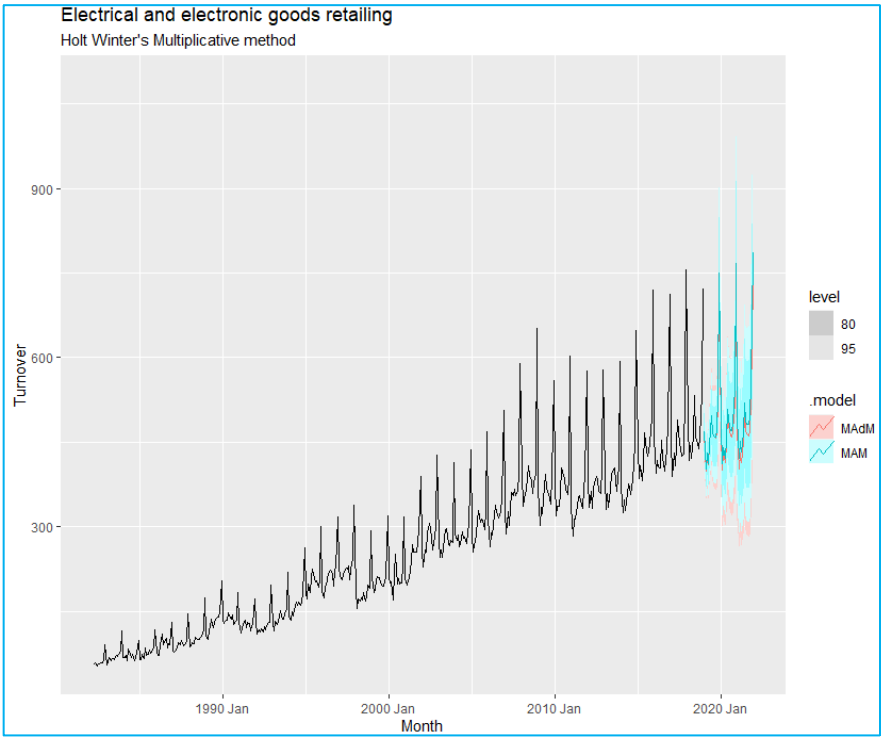
a. Is a multiplicative seasonality method appropriate for this series?

The plot in figure 1 showcases an apparent seasonal variation increase with proportion to time. This means that a multiplicative seasonality method would be appropriate for this series.

b. Apply Holt Winter's multiplicative trend method to the data. Then try the method making the trend damped. Plot and compare the 36-period ahead point forecasts and prediction intervals for both methods. Using accuracy(), compare the RMSE of the two methods; this comparison is based on one-step-ahead in-sample forecasts. Which method do you prefer and why?

```
holtFc <-myseries %>%
  model(
    MAM = ETS(Turnover ~error("M") + trend("A") + season("M")),
    MAdM = ETS(Turnover ~error("M") + trend("Ad") + season("M"))
  )
auto <- holtFc %>% forecast(h = 36)
auto %>% autoplot(myseries) + labs(title = myseries$Industry[1],
                                   subtitle = "Holt Winter's Multiplicative method")
```

Figure 12: Forecast of electrical and electronic goods retailing in Victoria using Holt Winter's Multiplicative method (1982 – 2018)



From figure 12, Holt-Winter's multiplicative method has been utilised in conjunction with a damped trend point forecast. The forecasts seem reasonable and very similar with the prediction intervals reaching lower levels due to the COVID-19 pandemic in 2020 affecting electrical good retailing. Since the forecasts and prediction intervals seem largely the same, we need to look at the RMSE values of the in-sample forecasts.

```
holtFc %>% accuracy()
```

```
A tibble: 2 x 12
State    Industry                     .model .type      ME  RMSE   MAE    MPE  MAPE  MASE RMSSE  ACF1
<chr>    <chr>                        <chr>  <chr>   <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
Victor~ Electrical and electron~ MAM    Train~ 0.0128  13.8  9.96 -0.441  4.34 0.458 0.494 0.208
Victor~ Electrical and electron~ MAdM   Train~ 1.29    13.6  9.57  0.159  4.12 0.440 0.489 0.108
```

The RMSE of the damped trend as shown above is 13.6 which is slightly lower than the model comprised of Holt's Linear Method, with an RMSE of 13.8. Although this difference is very insignificant, because Holt's Linear Method can tend to over-forecast with longer forecasts horizons, damping the trend allows for better forecasting performance. Indeed, other measurements of errors such as MAE, MAPE and MASE also tends to be lower than Holt's Linear Method. Therefore, the damped ETS model is preferred.

c. Check that the residuals from the best method look like white noise.

```
best_model <- myseries %>%
  model(MAdM = ETS(Turnover ~error("M") + trend("Ad") + season("M"))
  )
best_model %>% gg_tsresiduals + labs(title = "Residual diagnostics with damped trend")
```

Figure 13: Residual diagnostics of data series with Holt Winter's Multiplicative Method (damped trend)

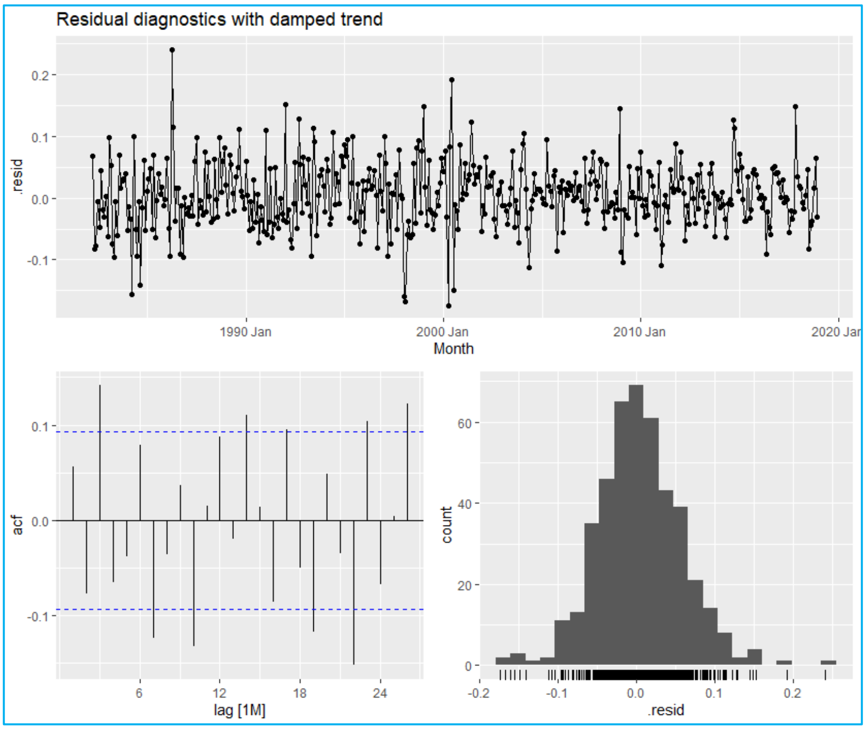


Figure 13 depicts a residual plot for the best model (damped method) which shows that there is no white noise evident. ACF plot contains number of spikes which lie outside the blue dotted line to be more than 5% for all the lags. Let us conduct a Ljung box test to further emphasise the possibility of white noise.

```
best_model <- myseries %>%
  model(MAdM = ETS(Turnover ~ error("M") + trend("Ad") + season("M")))
augment(best_model) %>%
  features(.resid, ljung_box, lag = 24, dof = 18)
```

A tibble: 1 x 6

State	Industry	.model	lb_stat	lb_pvalue	.name_repair
<chr>	<chr>	<chr>	<dbl>	<dbl>	<chr>
Victoria	Electrical and electronic goods retailing	MAdM	77.5	1.18e-14	minimal

According to the Ljung box test, the p-value is 1.18e-14 which is highly insignificant at the 5% level of significance. Thus, the null hypothesis, which is that the residuals are indistinguishable from white noise, is rejected. This indicates that there is autocorrelation between the residuals of the seasonal naïve model. Thus, there seems to be no evidence of white noise for the damped method based on residuals.

d. **Train the model to the end of 2010 and find the test set RMSE. Can you beat the seasonal naïve approach from part 2?**

```
myseries_train <- myseries %>%
  filter(year(Month) < 2011)

holt_damped_fit <- myseries_train %>%
  model(
    HWmd = ETS(Turnover ~ error("M") + trend("Ad") + season("M"))
  )

fit_train <- holt_damped_fit %>%
  forecast(h = '8 years')

accuracy(fit_train,myseries)
```

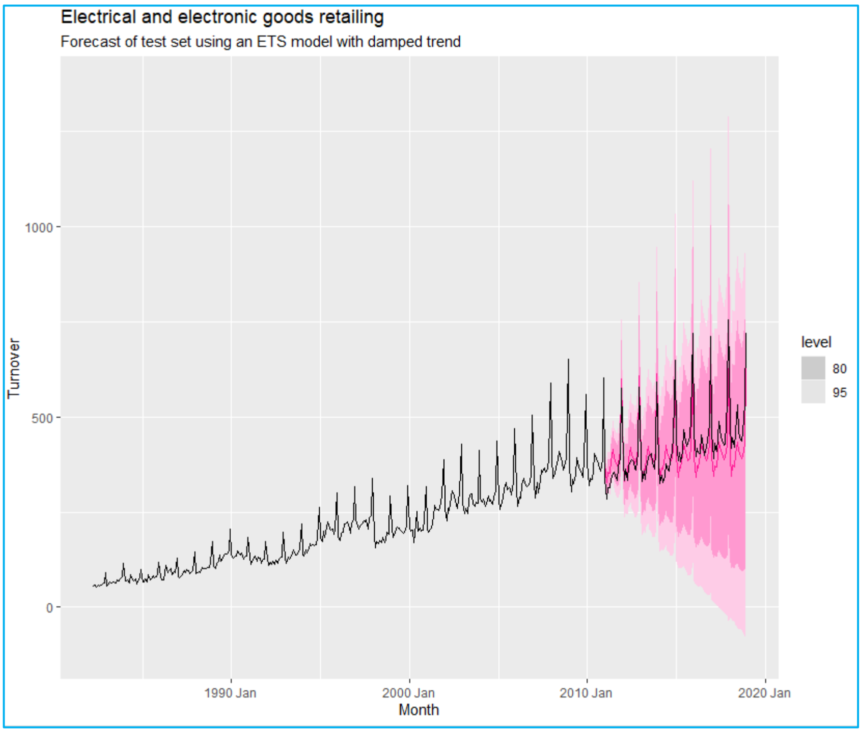
A tibble: 1 x 12

.model	State	Industry	.type	ME	RMSE	MAE	MPE	MAPE	MASE	RMSSE	ACF1
<chr>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	[illegible]
HWmd	Victor~	Electrical and electronic ~	Test	16.6	46.3	37.8	2.70	8.62	1.85	1.77	[illegible]

The test set RMSE of Holt Winter's Multiplicative damped trend model is 46.3. Compared to the RMSE value of the seasonal naïve model from part 2f which is 62.4, clearly the test set containing damped trend performs better than SNAIVE approach. Thus, we have beaten the seasonal naïve approach from part 2 in terms of RMSE. Now, let us produce a forecast of the test set:

```
autoplot(fit_train,myseries, colour = 'deeppink') + labs(title = myseries$Industry[1],
                                   subtitle = "Forecast of test set using an ETS model with damped trend")
```

Figure 14: Forecast of test set using an ETS model with damped trend



The forecast in figure 14 from the end of 2010 to 2018 seems reasonable however the prediction intervals are quite large and leads towards negative predictions, which seems quite unlikely. The point forecast, indicated by the pink line, is slowly increasing with a more constant variation compared to the rest of the data series.

e. **Try an STL decomposition applied to the Box-Cox transformed series, followed by ETS on the seasonally adjusted data. How does that compare with your best previous forecasts on the test set?**

```
lambda <- myseries %>%
  features(Turnover, features = guerrero) %>%
  pull(lambda_guerrero)

stl_model <- myseries_train %>%
  model (
    stlm = decomposition_model(STL(box_cox(Turnover,lambda)),ETS(season_adjust))
  )
stl_modelfc <- stl_model %>%
  forecast(h = '8 years')

accuracy(stl_modelfc,myseries)
autoplot(stl_modelfc,myseries, colour = 'deeppink') +
  labs(title = "Forecast using STL decomposition",
       subtitle = "Victoria")
```

```
A tibble: 1 x 12
.model State    Industry                      .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
<chr>  <chr>    <chr>                         <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
stlm   Victor~ Electrical and electronic  Test   -132.  152.  132. -30.0  30.0  6.45  5.79 0.712
```

Figure 15: Forecast of electrical and electronic retailing using STL decomposition on box-cox transformed data series

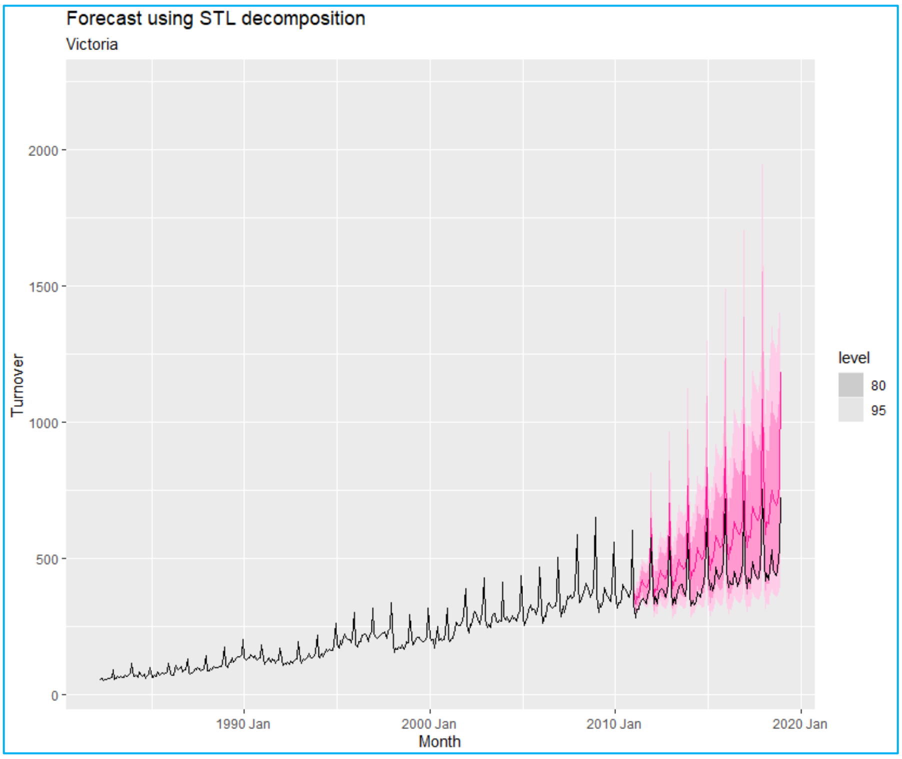


Figure 15 displays the forecast of electrical and electronic retailing using STL decomposition on the box-cox transformed data series. The point forecast is increasing through time, with high prediction intervals. In comparison to figure 14, which is our best model (according to RMSE), turnover increases through time whereas in figure 14 the forecasts seem to slowly decrease with prediction intervals located in the negative values.

In terms of performance, based on RMSE, clearly the ETS damped method from part d performs better as this model comprises of a lower RMSE value, 46.3, compared to the STL decomposed function on the box-cox transformed data series which is 152.

Part 4: ARIMA Modelling

a. For your data series, find the appropriate order of differencing, after transformation if necessary, to obtain stationary data.

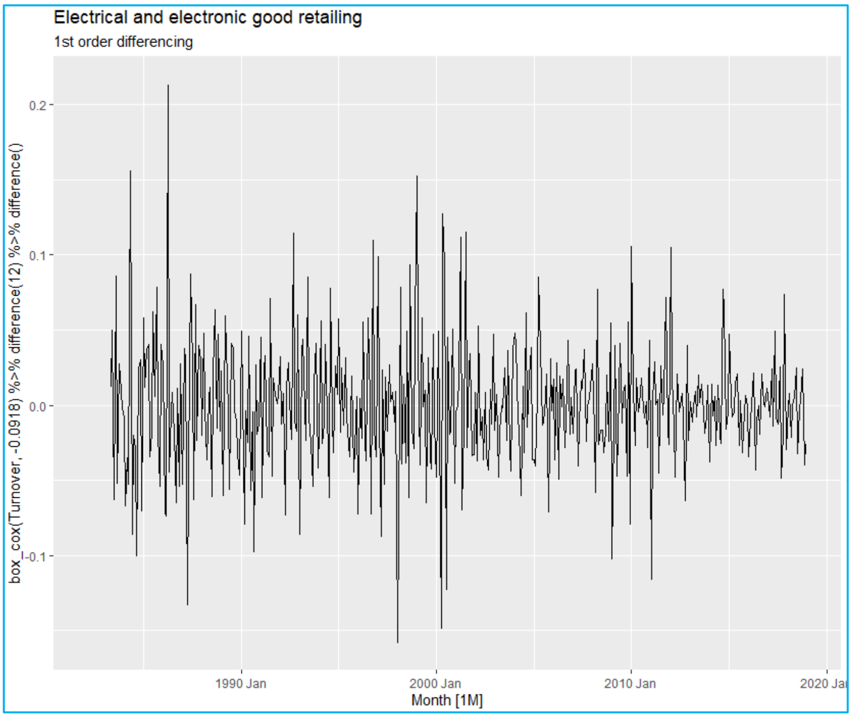
```
myseries %>%
  features(Turnover, unitroot_kpss)
```

```
# A tibble: 1 x 5
  State    Industry                                     kpss_stat kpss_pvalue .name_repair
  <chr>    <chr>                                            <dbl>       <dbl> <chr>
1 Victoria Electrical and electronic goods retailing         7.26        0.01 minimal
```

The data series contains an increasing trend, seasonality and increasing variance. From the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test, the KPSS p-value is 0.01, meaning we reject the null hypothesis that the data series is stationary and non-seasonal. Therefore, we require an appropriate order of differencing to make the data series stationary.

```
data_series <- myseries %>%
  autoplot(box_cox(Turnover, -0.0918) %>% difference(12) %>% difference()) +
  labs(title = "Electrical and electronic good retailing",
       subtitle = "1st order differencing")
data_series
```

Figure 16: First order differencing on Electrical and electronic goods retailing in Victoria (1982 – 2018)



The plot in figure 16 has been manipulated with first order differencing and the data series seems to be relatively stationary in comparison to the plot in figure 1. We can confirm using the unitroot_nsdiffs() command to check if further differencing is required.

```
myseries %>%
  features(Turnover, list(unitroot_nsdiffs), feat_stl)
```

```
# A tibble: 1 x 4
  State    Industry                                      nsdiffs .name_repair
  <chr>    <chr>                                           <int> <chr>
1 Victoria Electrical and electronic goods retailing           1 minimal
```

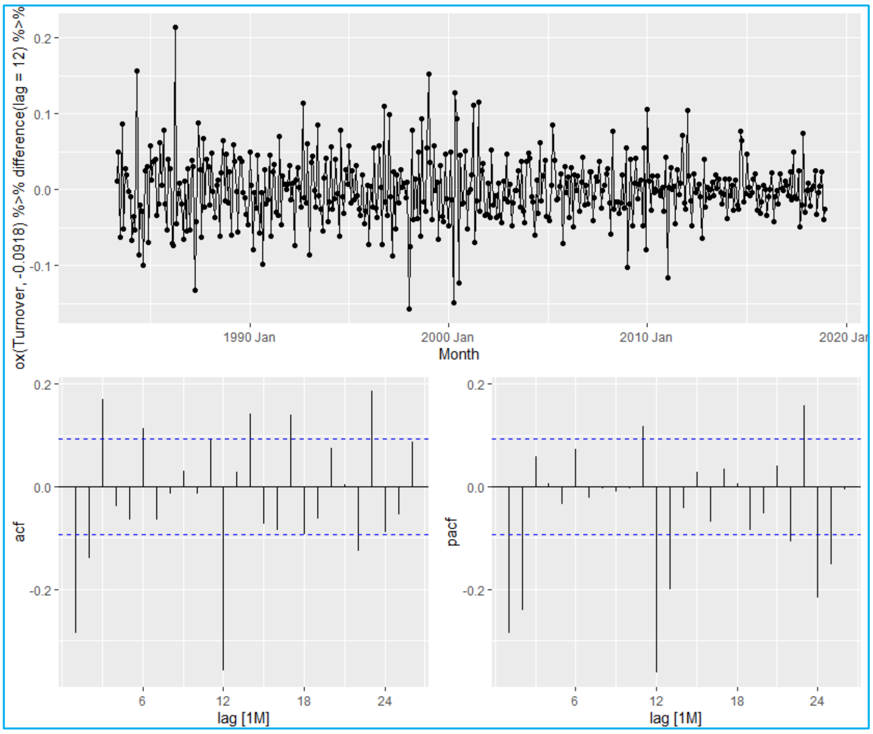
From this output, we can confirm that only a 1st order differencing is appropriate for this data set.

b. Select an appropriate seasonal ARIMA model. Explain your choice and report the results

```
myseries %>%
  gg_tsdisplay(box_cox(Turnover, -0.0918) %>% difference(lag = 12) %>% difference(), plot_type = "partial") +
```

Before selecting an appropriate seasonal ARIMA model, we require to check the residuals of the model with first order differencing.

Figure 17: Residual diagnostics of ARIMA model with first order differencing on Electrical and electronic goods retailing in Victoria (1982 – 2018)



Our main objective is to determine an appropriate ARIMA model based on the ACF and PACF plot shown in figure 17 (Hyndman, 2021). In the residual plots of the seasonally differenced data, the PACF plot contains spikes at lags 12 and 24, but no seasonal lags for lag 24 in the ACF plot. This could suggest a seasonal AR(2) term. On the other hand, there are 3 significant spikes in the PACF plot at lags 1,2 and 12, which may be suggestive of an AR(3) term. The pattern in the ACF plot does not indicate any possibility of a simple model (Hyndman, 2021). Since the data contains first order differencing of 1, we observe a I(1) term. In addition, the ACF and PACF shows a sinusoidal pattern which could indicate an MA(0) term.

From our analysis, a possible model for this data series is ARIMA(3,1,1)(2,1,0)$_{12}$. To choose an appropriate ARIMA model, we fit this model and account for some variation on it, compute the AICc values for each model and determine the best model (indicated by the lowest AICc value).

```
arima_fit <- myseries %>%
  model(arima1 = ARIMA(box_cox(Turnover, -0.0918) ~ pdq(3,1,1) + PDQ(2,1,0)),
        arima2 = ARIMA(box_cox(Turnover, -0.0918) ~ pdq(3,1,1) + PDQ(1,1,1)),
        arima3 = ARIMA(box_cox(Turnover, -0.0918) ~ pdq(3,1,0) + PDQ(2,1,0)),
        arima4 = ARIMA(box_cox(Turnover, -0.0918) ~ pdq(3,1,1) + PDQ(0,1,1)),
        arima5 = ARIMA(box_cox(Turnover, -0.0918) ~ pdq(3,1,0) + PDQ(1,1,1)),
        auto = ARIMA(box_cox(Turnover, -0.0918))
        )

glance(arima_fit)
```

Figure 18: Accuracy table for selected ARIMA and auto ARIMA models

```
# A tibble: 6 x 10
  State   Industry                  .model  sigma2 log_lik    AIC   AICc    BIC ar_roots  ma_roots
  <chr>   <chr>                     <chr>    <dbl>   <dbl>  <dbl>  <dbl>  <dbl> <list>    <list>
1 Victor~ Electrical and electron~ arima1 0.00131    814. -1614. -1614. -1586. <cpl [27~ <cpl [1]>
2 Victor~ Electrical and electron~ arima2 0.00111    845. -1676. -1676. -1648. <cpl [15~ <cpl [13~
3 Victor~ Electrical and electron~ arima3 0.00131    814. -1616. -1615. -1591. <cpl [27~ <cpl [0]>
4 Victor~ Electrical and electron~ arima4 0.00111    844. -1676. -1676. -1652. <cpl [3]> <cpl [13~
5 Victor~ Electrical and electron~ arima5 0.00111    845. -1678. -1678. -1653. <cpl [15~ <cpl [12~
6 Victor~ Electrical and electron~ auto   0.00110    846. -1678. -1678. -1650. <cpl [27~ <cpl [12~
```

```
myseries %>%
  model(autoarima = ARIMA(box_cox(Turnover, -0.0918))) %>%
  report()
```

```
Series: Turnover
Model: ARIMA(3,1,0)(2,1,1)[12]
Transformation: box_cox(Turnover, -0.0918)

Coefficients:
          ar1      ar2     ar3    sar1     sar2     sma1
      -0.3890  -0.2584  0.0855  0.0579  -0.0956  -0.8071
s.e.   0.0489   0.0505  0.0486  0.0619   0.0579   0.0411

sigma^2 estimated as 0.001103:  log likelihood=846.24
AIC=-1678.48   AICc=-1678.21   BIC=-1650.06
```

According to figure 18 above, the AICc value is the lowest for ARIMA(3,1,0)$(1,1,1)_{12}$ and the automatically computed ARIMA model, which is ARIMA(3,1,0)$(2,1,1)_{12}$. We observe that the BIC and AIC values are lower for **ARIMA(3,1,0)$(1,1,1)_{12}$**, indicating an appropriate model for this data series.

c. Using the test data set as before, compare the forecast performance with the models you obtained in Part 2 and Part 3. Try also an STL decomposition applied to the Box-Cox transformed series, followed by ARIMA on the seasonality adjusted data; that is, an STL-ARIMA model rather than the STL-EST model used in Part 3(e).

There are 2 parts to this question:

Part 1: Forecast performance of models in previous parts

```
arima_fc_test <- myseries_train %>%
  model(snaive = SNAIVE(Turnover),
        MAM = ETS(Turnover ~ error("M") + trend("A") + season("M")),
        MAdM = ETS(Turnover ~ error("M") + trend("Ad") + season("M")),
        stlm = decomposition_model(STL(box_cox(Turnover, lambda)), ETS(season_adjust)),
  )

arima_fc <- arima_fc_test %>%
  forecast(new_data = anti_join(myseries, myseries_train))

arima_fc %>% accuracy(myseries)
```

A tibble: 4 x 12

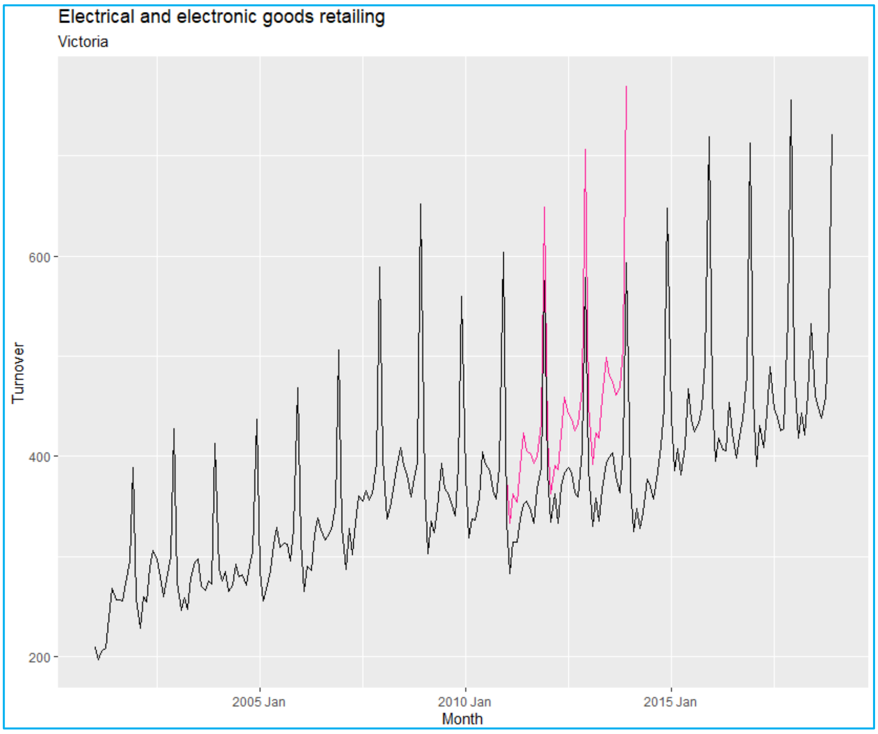
.model	State	Industry	.type	ME	RMSE	MAE	MPE	MAPE	MASE	RMSSE	ACF1
<chr>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
MAdM	Victor~	Electrical and electroni~	Test	16.6	46.3	37.8	2.70	8.62	1.85	1.77	0.868
MAM	Victor~	Electrical and electroni~	Test	-13.5	29.9	24.4	-3.96	6.13	1.19	1.14	0.738
snaive	Victor~	Electrical and electroni~	Test	38.4	62.4	49.3	7.92	11.0	2.42	2.38	0.870
stlm	Victor~	Electrical and electroni~	Test	-132.	152.	132.	-30.0	30.0	6.45	5.79	0.712

Using the test data set for all the models explored in part 2 and 3, we have obtained the accuracy tables for each of the models. In terms of RMSE, the model with the best forecast performance, 29.9 is Holt Winter's Multiplicative method. The seasonal naïve model and the damped trend model come close in terms of forecasting performance. On the other hand, the STL decomposition model produces a poor model in comparison, with an RMSE value of 152 for the test set data. Therefore, based on the forecast performance of the test data, when comparing the models from part 2 and part 3, the best model is Holt Winter's Multiplicative method on the test data set.

<u>Part 2: STL decomposition applied to Box-cox transformed data series followed by ARIMA on the seasonality adjusted data</u>

```
fc <- myseries_train %>%
  model(stl_boxcox = decomposition_model(STL(box_cox(Turnover, -0.0918)),
                                         ARIMA(season_adjust))) %>% forecast(h = "3 years")

fc %>% autoplot(level = NULL, colour = 'deeppink') +  autolayer(filter(myseries, year(Month) > 2000), Turnover) +
  labs(title = myseries$Industry[1], subtitle = myseries$State[1])
```



<u>Reference</u>

Hyndman, R.J., & Athanasopoulos, G. (2001) *Forecasting: principles and practice*, 3rd edition, OTexts: Melbourne, Australia. OTexts.com/fpp3. Accessed on <16.04.21>