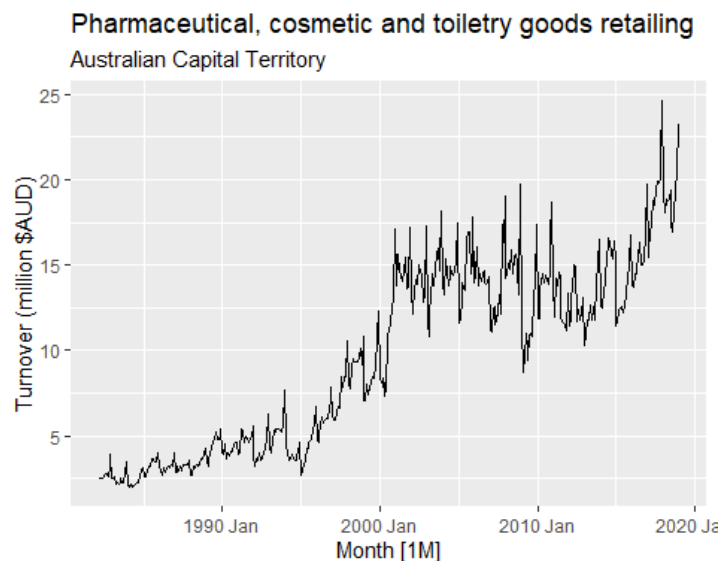**Final ECON2209 project**

0: Setup

```
library(fpp3)

set.seed(5207462)
myseries <- aus_retail %>%
  filter(`Series ID` == sample(aus_retail$`Series ID`,1))
```

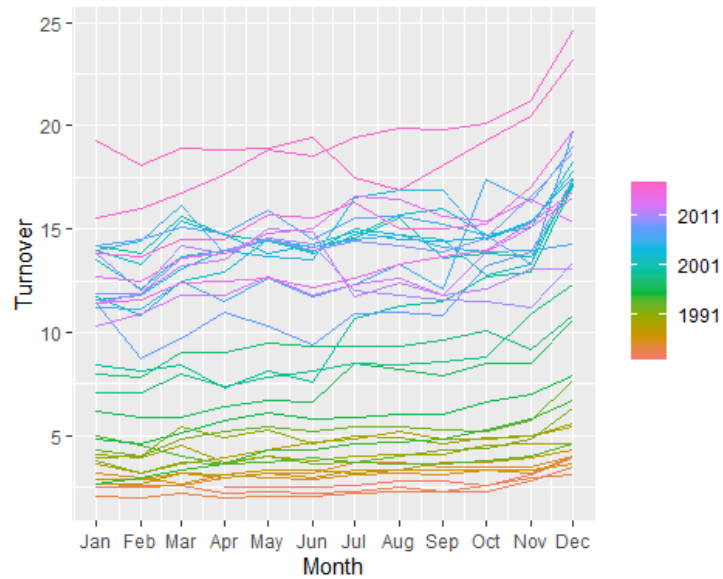**1: Data Exploration and Transformation**

a)

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


As seen above, the data has an upward trend with a cyclical nature i.e. during the GFC and early 2000's. However, seasonality is harder to see unless further analyzed.
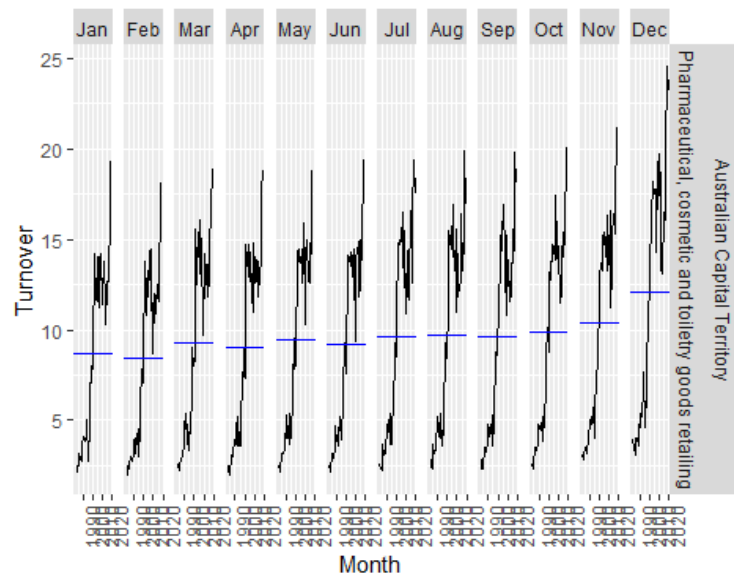
b)

```
gg_season(myseries, y = Turnover)
```
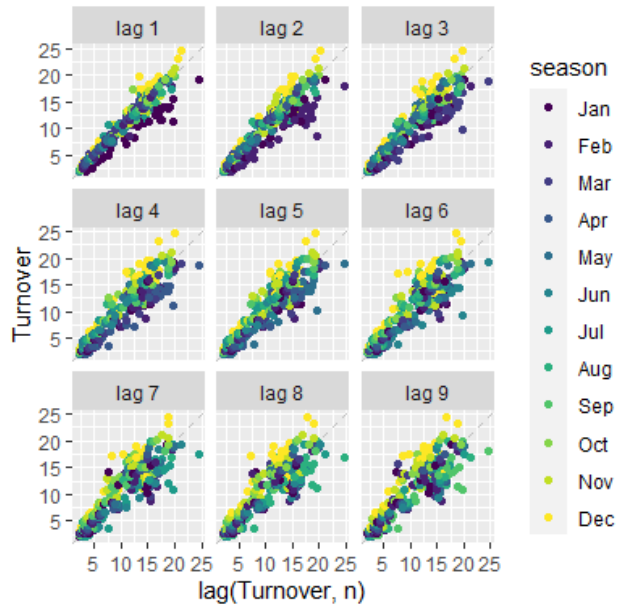
As seen above, the data does have seasonality due to variations within a year as well as general upward trend.

gg_subseries(myseries, y = Turnover)

The seasonality is only further emphasized in the subseries graph with December sales being higher than the rest of the year potentially due to Christmas sales.

gg_lag(myseries, y = Turnover, geom = "point")

As seen above, the relationship between all the lags is strongly positive highlighting the strong seasonality of the data.
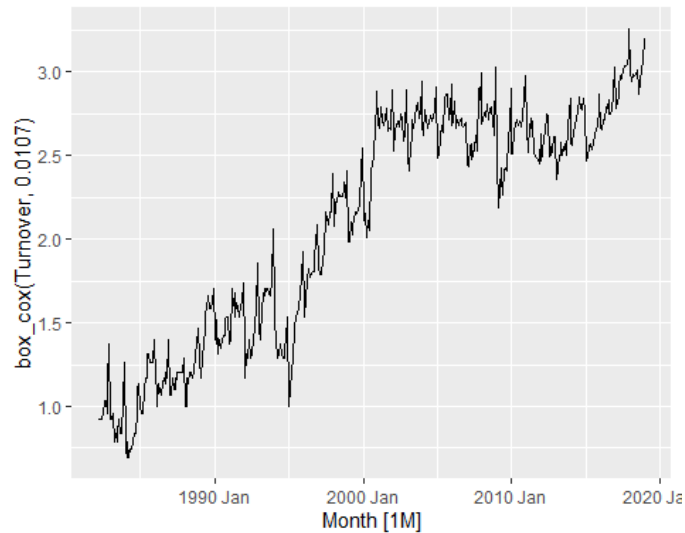
```
ACFeg <- myseries %>% ACF(Turnover)
autoplot(ACFeg)
```
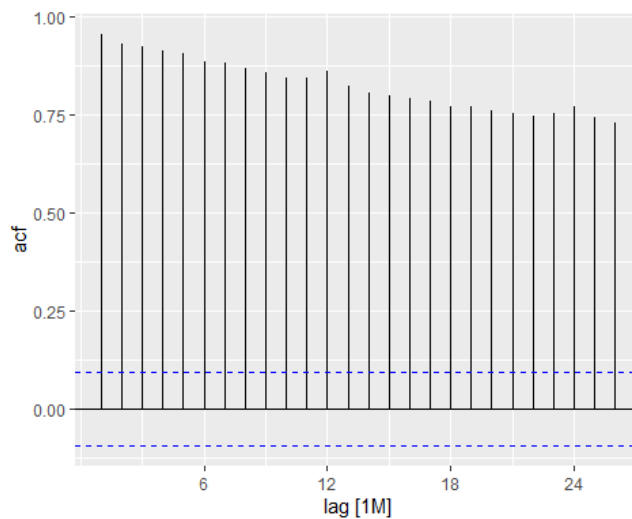

This graph highlights the seasonality due to the "scalloped" shape of the sales curve whilst the general down emphasizes the curves strong trend.

c)

```
## # A tibble: 1 x 3
##   State          Industry                lambda_guerrero
##   <chr>          <chr>                   <dbl>
## 1 Australian Capital Te~ Pharmaceutical, cosmetic and toiletry ~     0.0107
```

By using features, we can find the appropriate value for Box-Cox transformations. However, since the data does not have a heavy skew, there is no need for the data to be normalized and consequently no need for a transformation.
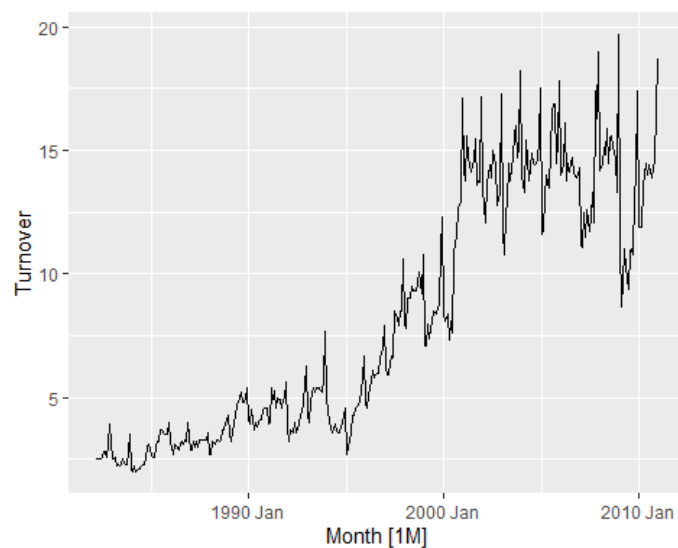
2: Forecasting

a)

```
myseries_train <- myseries %>%
 filter(Month <= max(Month) - 8*12) %>%
 summarise(Turnover = sum(Turnover))
```

b)

```
autoplot(myseries_train, .vars = Turnover)
```
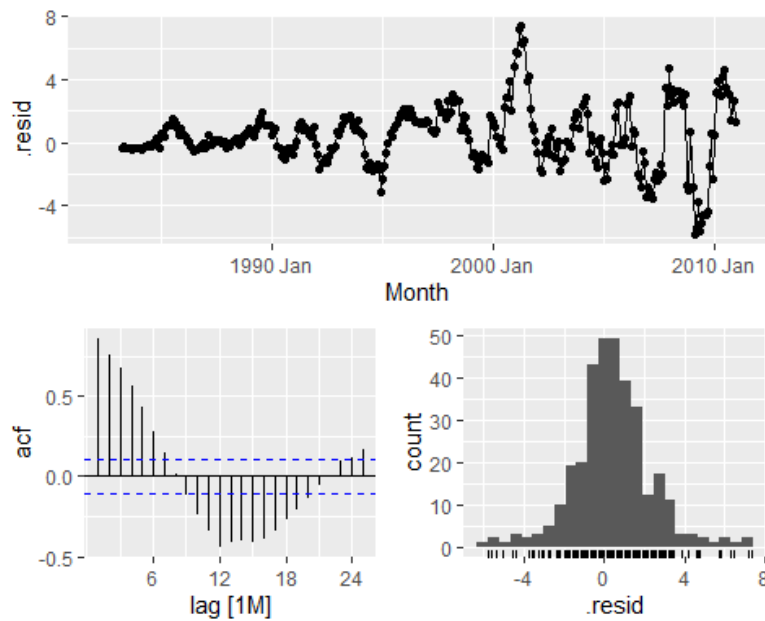
As seen above, the data is split appropriately and produced.

c)

```
fit <- myseries_train %>%
 model(
   snaive = SNAIVE(Turnover)
 )
fc <- fit %>% forecast(h = "8 years")
```

d)

```
fit %>% gg_tsresiduals()
```
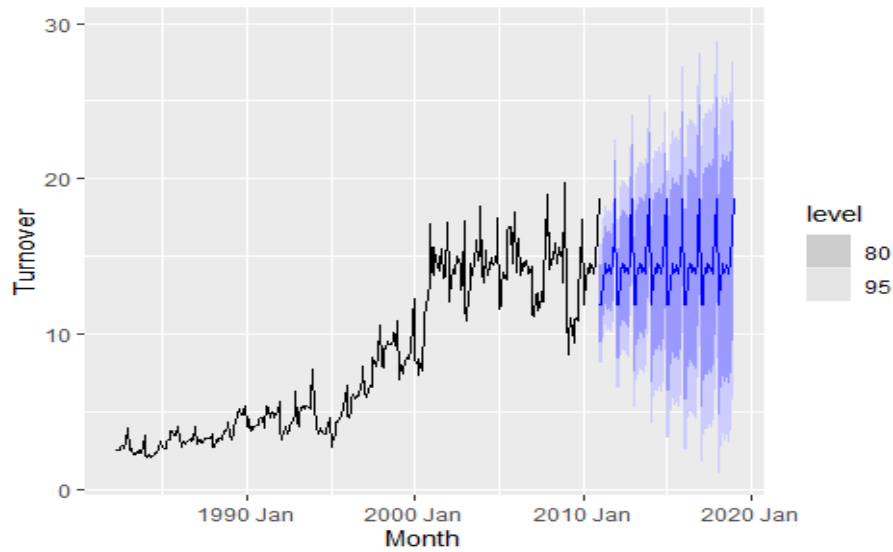

The mean of the residuals is not close to zero since it varies for the valued after 2000 Jan. The ACF has *strong trends and correlation* suggestion that the forecasts will not be that good. However, the histogram is quite like a *normal distribution* meaning that the prediction intervals computed might be accurate.

e)

```
fit %>% forecast(h = 96) %>% autoplot(myseries_train)
```

f)

```
fit %>% forecast(h = 96) %>% autoplot(myseries_train) + autolayer(myseries)
```
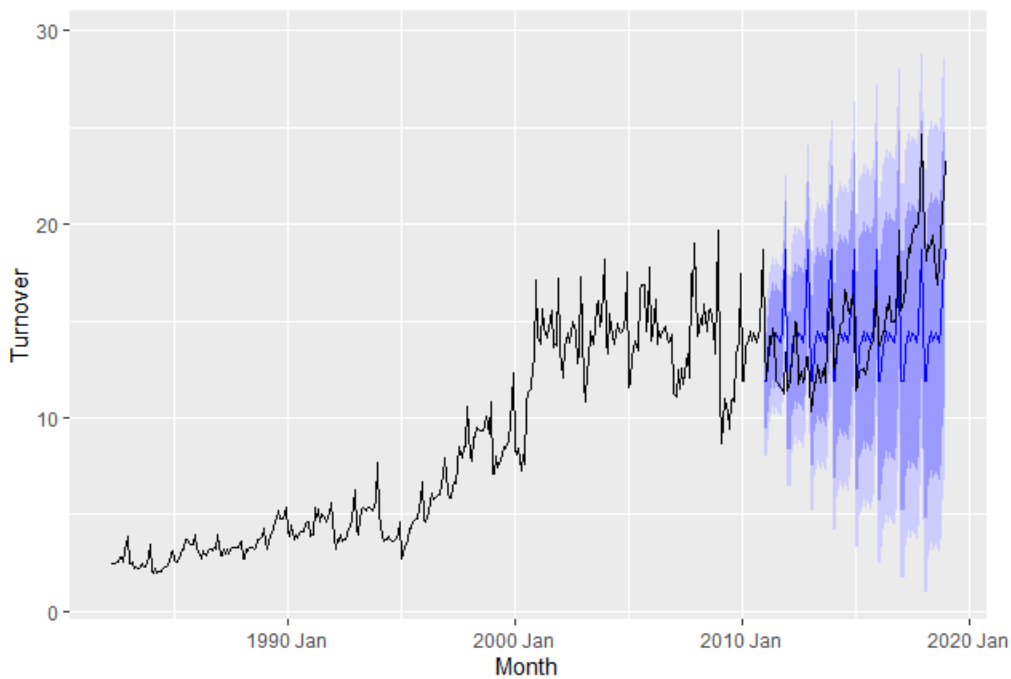

As seen above, the autoplot of the trained series underestimates the actual values since there is no multiplicative forecasting in the trained set. However, the actual values did stay within the 80-95% confidence intervals most of the time owing some credibility to the data set.

3: Exponential Smoothing

a)

Multiplicative seasonality method is more appropriate for this series due to the seasonality present in the data, allowing for the width of the seasonal patterns to be proportional to the changes. If additive was used, the results would be less accurate.

```
fit <- myseries_train %>%
 model(
   G = ETS(Turnover ~ error("M") + trend("A") + season("M")),
   Gdamp = ETS(Turnover ~ error("M") + trend("Ad") + season("M"))
   )
```

b)

The multiplicative method and the damped version are created above.

```
fit %>%
 forecast(h = 36) %>% autoplot(myseries_train)
```
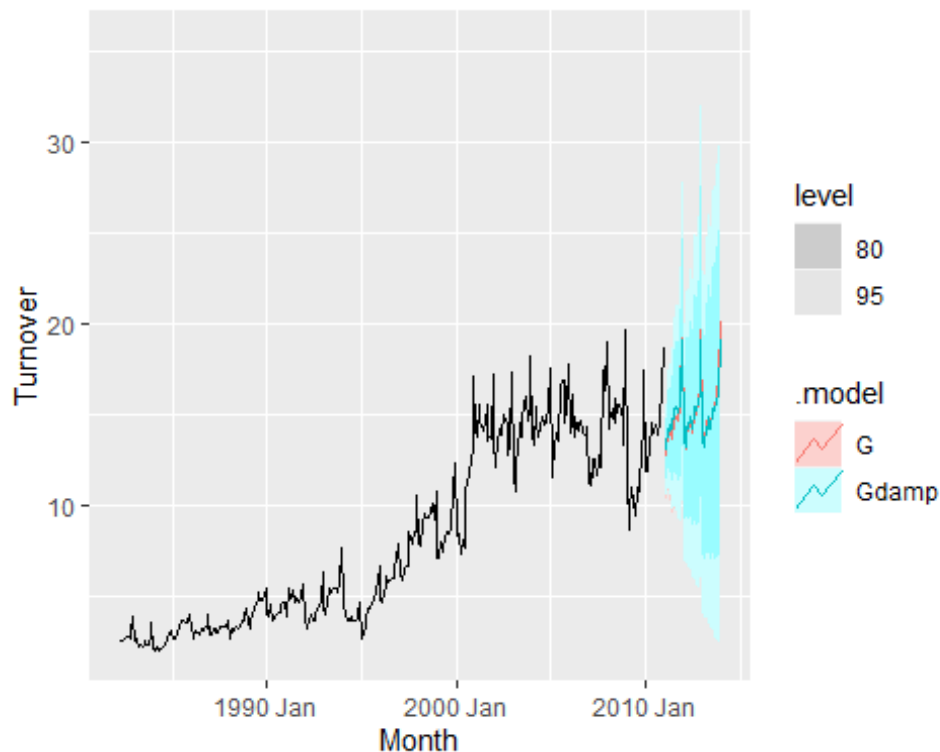

Both the damped and normal forecasts and prediction intervals are plotted above. As we can see, the G model exaggerates the highs and lows a bit more than the damped model.

```
accuracy(fit)
```

```
## # A tibble: 2 x 10
##   .model .type      ME  RMSE   MAE   MPE  MAPE  MASE RMSSE   ACF1
##   <chr>  <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 G      Training 0.00757 0.728 0.462 -0.324  5.72 0.322 0.371 0.0576
## 2 Gdamp  Training 0.0480  0.727 0.458  0.437  5.77 0.320 0.370 0.118
```

As seen above, the normal version has a bit higher of a RMSE compared to the damped model. However, the normal method is better since it has lower values for ME, MPE, MAPE and ACF1 than the damped version. Consequently, using the damped version will provide better values.

c)

```
fit <- myseries_train %>%
 model(
   G = ETS(Turnover ~ error("M") + trend("A") + season("M")),
 ) %>%
 select(G)
fit %>% gg_tsresiduals()
```


As seen above, the residuals from the best method does look like white noise since the:

- histogram looks very similar to a normal distribution.
- the mean of the residuals is close to 0
- most of the ACF spikes are within range of +-2/sqrt(T)

d)

```
fit %>%
 forecast(h = 96)

accuracy(fit)
```

```
## # A tibble: 1 x 10
##   .model .type      ME  RMSE  MAE   MPE  MAPE MASE RMSSE  ACF1
##   <chr>  <chr>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 G      Training 0.00757 0.728 0.462 -0.324  5.72 0.322 0.371 0.0576
```
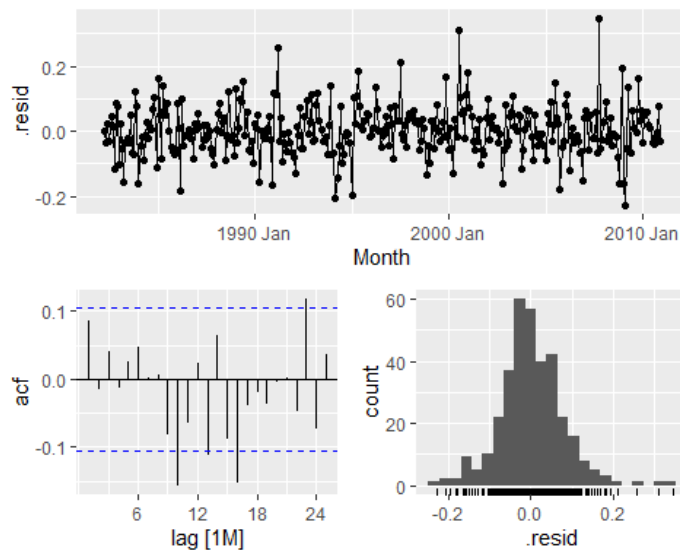
The RMSE for the normal (undamped version) using MAM is 0.728 compared to 1.96 for snaive (from previous part). This means that the normal and undamped version is the better method to use in this regard. Similarly, all the values are lower for the normal, undamped version when compared to snaive suggesting that we should use this.

e)

```
fit <- myseries_train %>%
 model(
   G = ETS(Turnover ~ error("M") + trend("A") + season("M")),
   stl = decomposition_model(STL(log(Turnover)), ETS(season_adjust))
 )
accuracy(fit)

## # A tibble: 2 x 10
##  .model .type       ME RMSE  MAE   MPE MAPE MASE RMSSE   ACF1
##  <chr>  <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 G    Training 0.00757 0.728 0.462 -0.324  5.72 0.322 0.371  0.0576
## 2 stl  Training 0.0476  0.694 0.429  0.341  5.15 0.299 0.354 -0.0612
```
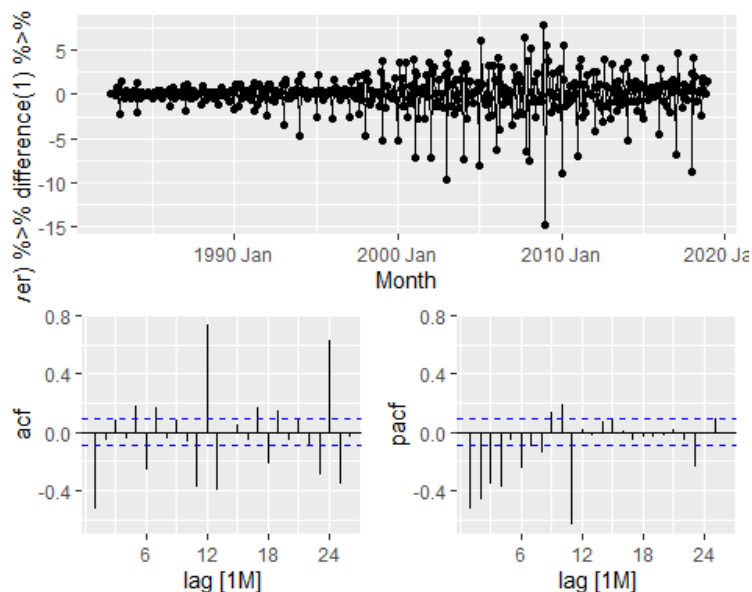
As seen above, the RMSE and MAE are lower for the STL decomposition than for the MAM. When compared to the previous comparison of MAM is lower for RMSE and MAE. Consequently, since more values are lower for STL decomposition compared to MAM, STL decomposition should be used.

4: ARIMA Modeling

```
myseries %>%
 gg_tsdisplay((Turnover) %>% difference(1) %>% difference(1), plot_type = "partial")
```


When the differences of 1 and 1 was applied on the myseries set due to PACF having a significant peak at 12 after the first difference followed by a peak at 6 in the ACF curve, the values of (1,0,1) seemed optimal. As a result, the differences are closed to the mean of 0 but the ACF and PACF still have significant outliers.

```
fit <- myseries %>%
 model(
   guess = ARIMA(Turnover ~ 1 + PDQ(1,1,0)),
 )
fit %>%
 gg_tsresiduals()
```
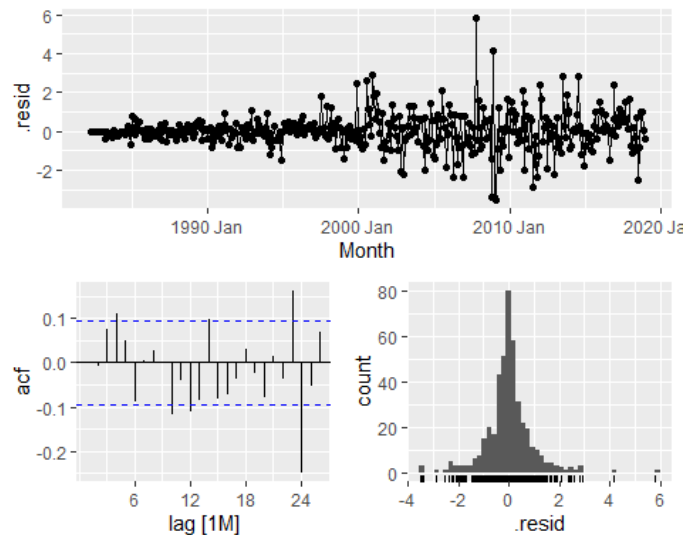

When the PDQ of (1,1,0) was applied, the data output was quite close to the mean of 0 for most residuals. Similarly, the histogram is normalized. Nevertheless, there are two values which are outliers in the ACF. However, this application of the PDQ resulted in quite tidy results.

b)

```
fit %>%
 report()
```

```
## Series: Turnover
## Model: ARIMA(1,0,1)(1,1,0)[12] w/ drift
##
## Coefficients:
##        ar1     ma1    sar1  constant
##     0.8916  -0.1260  -0.3939   0.0684
## s.e.  0.0250   0.0553   0.0450    0.0391
##
## sigma^2 estimated as 0.8831:  log likelihood=-581.66
## AIC=1173.31   AICc=1173.45   BIC=1193.62
```
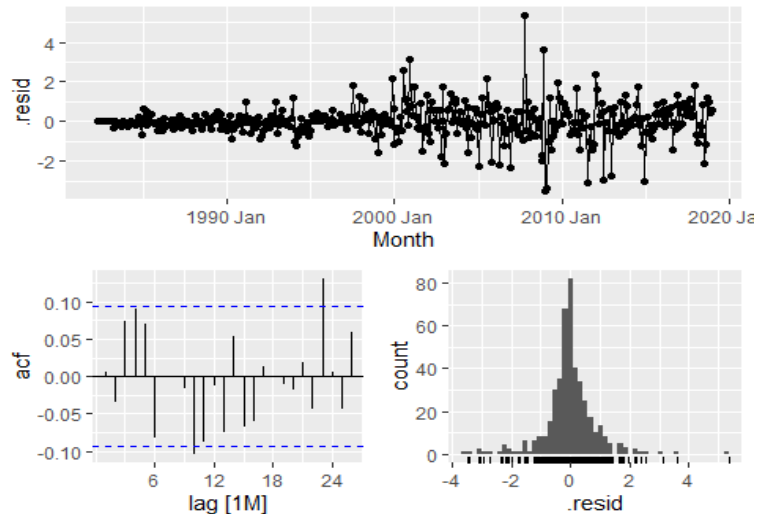
```
myseries %>%
 model(ARIMA(Turnover))
```

```
## # A mable: 1 x 3
## # Key:    State, Industry [1]
##   State      Industry                     ARIMA(Turnover)`
```

```
##  <chr>       <chr>                          <model>
## 1 Australian Cap~ Pharmaceutical, cosmetic a~ <ARIMA(1,0,1)(0,1,2)[12] w/ drift>

fit <- myseries %>%
 model(
   auto = ARIMA(Turnover ~ pdq(1,0,1) + PDQ(0,1,2)),
 )

fit %>%
 gg_tsresiduals()
```



```
fit %>%
 report()

## Series: Turnover
## Model: ARIMA(1,0,1)(0,1,2)[12] w/ drift
##
## Coefficients:
##      ar1    ma1    sma1    sma2  constant
##    0.9320 -0.1549 -0.6479 -0.1245   0.0303
## s.e. 0.0201  0.0548  0.0502  0.0482   0.0087
##
## sigma^2 estimated as 0.7402:  log likelihood=-547.14
## AIC=1106.28  AICc=1106.48  BIC=1130.65
```

By using the chosen values, we received the results of (1,0,1)(1,1,0) with a 12 value drift. This also resulted in an AICc of 1173.45. The mean is also close to zero although variance existed, and the histogram is also normalized. The ACF values are also accurate and valid since there are only two significant values resulting in the data being mostly accurate.

However by using the autovalues of (1,0,1)(0,1,2) with a 12 drift the AICc is also lower. This means we should use the auto produced values.
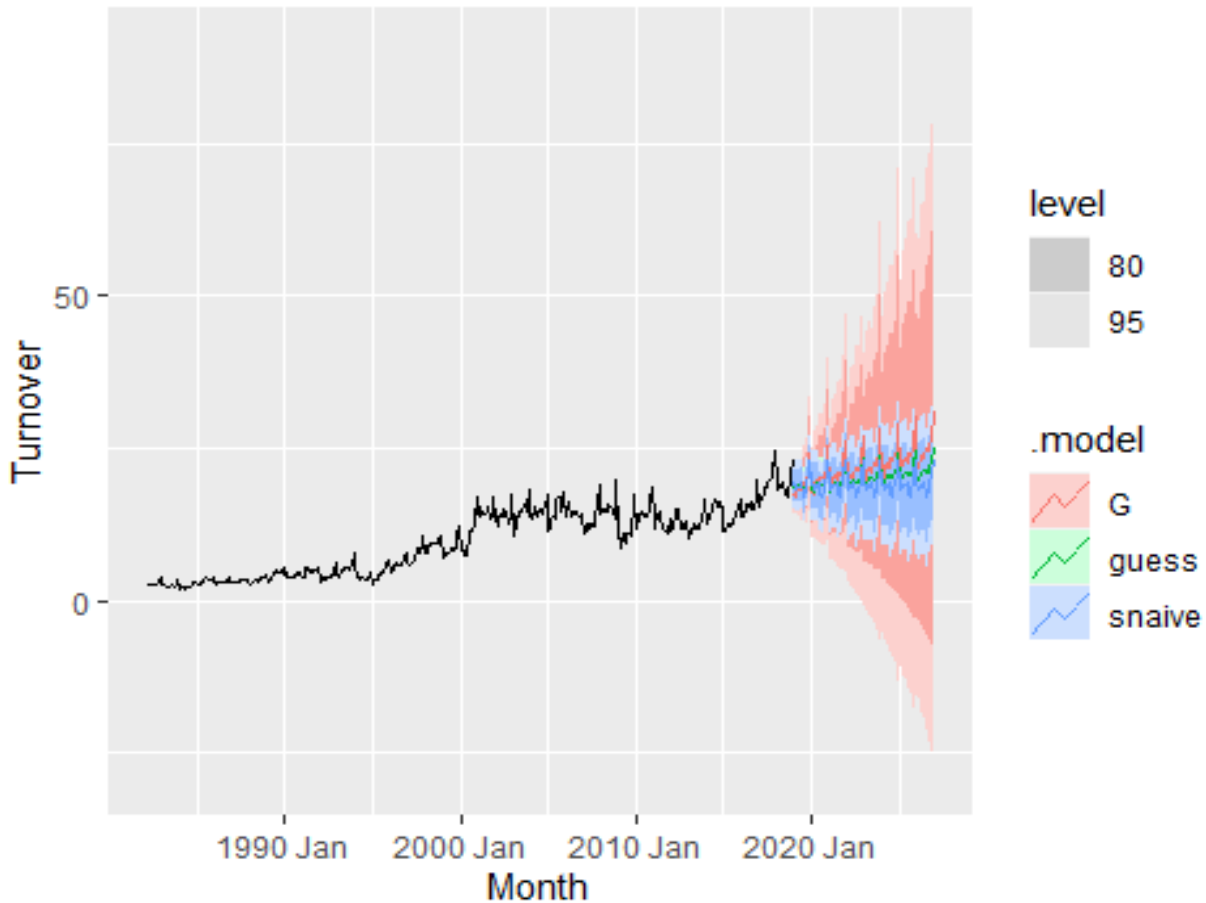
c)

```
fit <- myseries %>%
 model(
   snaive = SNAIVE(Turnover),
   G = ETS(Turnover ~ error("M") + trend("A") + season("M")),
   guess = ARIMA(Turnover ~ pdq(1,0,1) + PDQ(0,1,2)),
 )

fit %>%
 forecast(h = 96) %>% autoplot(myseries)
```


When the results from part 2 and 3 are auto plotted along with the ARIMA model, the graph above is presented. As we van see the G normal method results in values that exaggerate the data the most followed by the guess (auto-ARIMA) values.

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

stlarima <- decomposition_model(
 STL(box_cox(Turnover, lambda)),
 ARIMA(season_adjust)
)
```
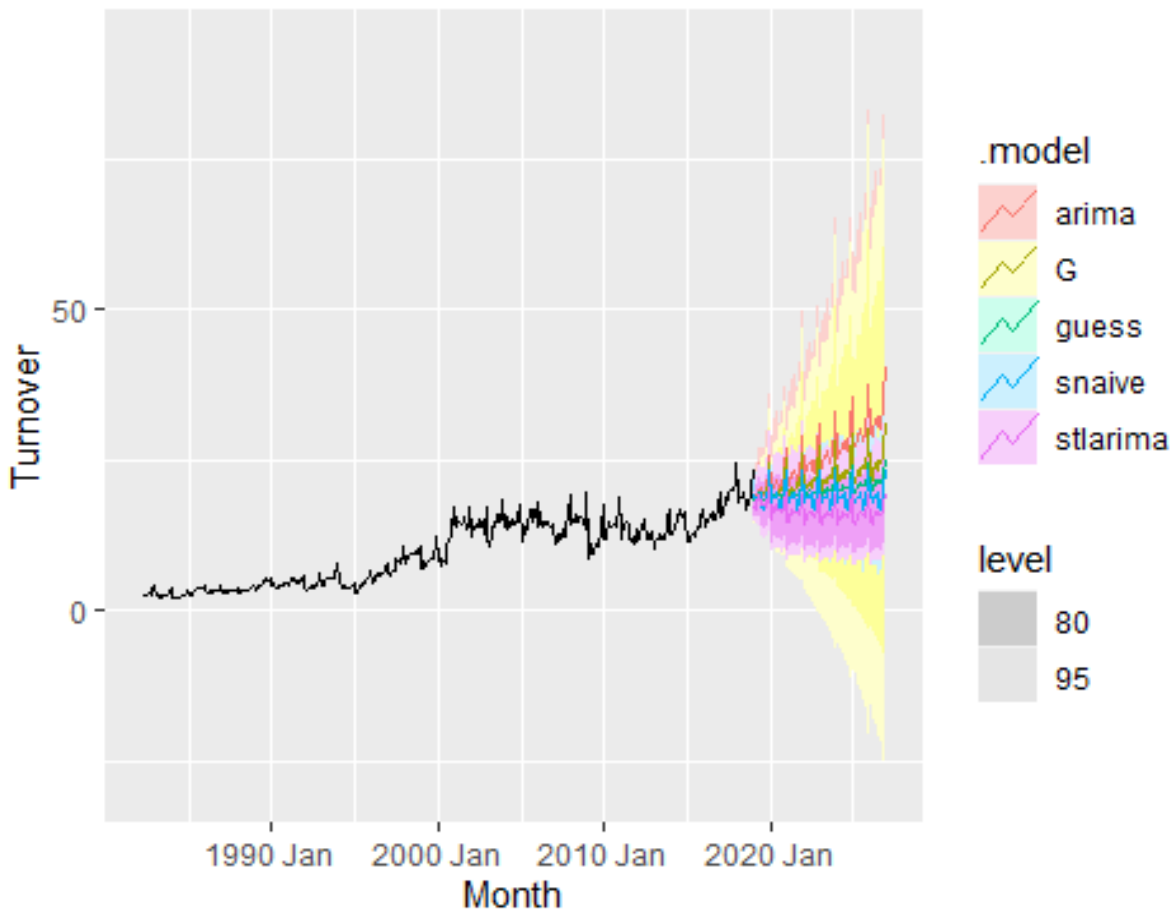
```
fit <- myseries %>%
 model(
  snaive = SNAIVE(Turnover),
  G = ETS(Turnover ~ error("M") + trend("A") + season("M")),
  guess = ARIMA(Turnover ~ pdq(1,0,1) + PDQ(0,1,2)),
  arima = ARIMA(box_cox(Turnover, lambda)),
  stlarima = stlarima
 )

fit %>%
 forecast(h = 96) %>% autoplot(myseries)
```


As seen above, the values for the ARIMA overestimate the values the most followed by guess (Auto-ARIMA), G (undamped), naïve method and STL ARIMA method. The ARIMA method also has the biggest range for forecasts compared to the other methods.

5: References- (font size: 11, font: Cambria, spacing: 1.0)

Moodle.telt.unsw.edu.au. 2021. *ARIMA models continued*. [online] Available at: <https://moodle.telt.unsw.edu.au/pluginfile.php/6603230/mod_resource/content/3/ECON2209_Lec6_OnlineSession.pdf> [Accessed 23 April 2021].

Moodle.telt.unsw.edu.au. 2021. *ARIMA session*. [online] Available at: <https://moodle.telt.unsw.edu.au/pluginfile.php/6603228/mod_resource/content/4/ECON2209_Lec5_OnlineSession.pdf> [Accessed 23 April 2021].

Moodle.telt.unsw.edu.au. 2021. *Lec 3 Online session*. [online] Available at: <https://moodle.telt.unsw.edu.au/pluginfile.php/6602637/mod_resource/content/3/ECON2209_Lec3_OnlineSession.pdf> [Accessed 23 April 2021].

Otexts.com. 2021. *Chapter 5 The forecaster's toolbox | Forecasting: Principles and Practice (3rd ed)*. [online] Available at: <https://otexts.com/fpp3/toolbox.html> [Accessed 23 April 2021].

Otexts.com. 2021. *Chapter 9 ARIMA models | Forecasting: Principles and Practice (3rd ed)*. [online] Available at: <https://otexts.com/fpp3/arima.html> [Accessed 23 April 2021].

Moodle.telt.unsw.edu.au. 2021. *Problem set 2 solutions*. [online] Available at: <https://moodle.telt.unsw.edu.au/pluginfile.php/6711331/mod_resource/content/1/ECON2209_Assessment_ProblemSet2_Solutions.pdf> [Accessed 23 April 2021].

Moodle.telt.unsw.edu.au. 2021. *Tutorial set 6 solutions*. [online] Available at: <https://moodle.telt.unsw.edu.au/pluginfile.php/6634731/mod_resource/content/5/ECON2209_Tutorial6_Solutions.pdf> [Accessed 23 April 2021].

Moodle.telt.unsw.edu.au. 2021. *Tutorial solutions wk4*. [online] Available at: <https://moodle.telt.unsw.edu.au/pluginfile.php/6634729/mod_resource/content/6/ECON2209_Tutorial4_Solutions.pdf> [Accessed 23 April 2021].