Retail Assignment

Eshan Thakur

17/05/2022

This code chunk includes loading the fpp3 package for the assignment. I also created a myseries data according to my student ID number in this chunk.

```
library(fpp3)
```

```
## -- Attaching packages --------------------------------------- fpp3 0.4.0 --
```

```
## v tibble      3.1.6     v tsibble     1.1.1
## v dplyr       1.0.7     v tsibbledata 0.4.0
## v tidyr       1.1.4     v feasts      0.2.2
## v lubridate   1.8.0     v fable       0.3.1
## v ggplot2     3.3.5
```

```
## -- Conflicts ------------------------------------------------ fpp3_conflicts --
## x lubridate::date()    masks base::date()
## x dplyr::filter()      masks stats::filter()
## x tsibble::intersect() masks base::intersect()
## x tsibble::interval()  masks lubridate::interval()
## x dplyr::lag()         masks stats::lag()
## x tsibble::setdiff()   masks base::setdiff()
## x tsibble::union()     masks base::union()
```

```
# Use your student ID as the seed
set.seed(31118224)
myseries <- aus_retail %>%
  # Remove discontinued series
  filter(!(`Series ID` %in% c("A3349561R","A3349883F","A3349499L","A3349902A",
                        "A3349588R","A3349763L","A3349372C","A3349450X",
                        "A3349679W","A3349378T","A3349767W","A3349451A"))) %>%
  # Select a series at random
  filter(`Series ID` == sample(`Series ID`,1))
```

STATISTICAL FEATURES OF THE ORIGINAL DATA

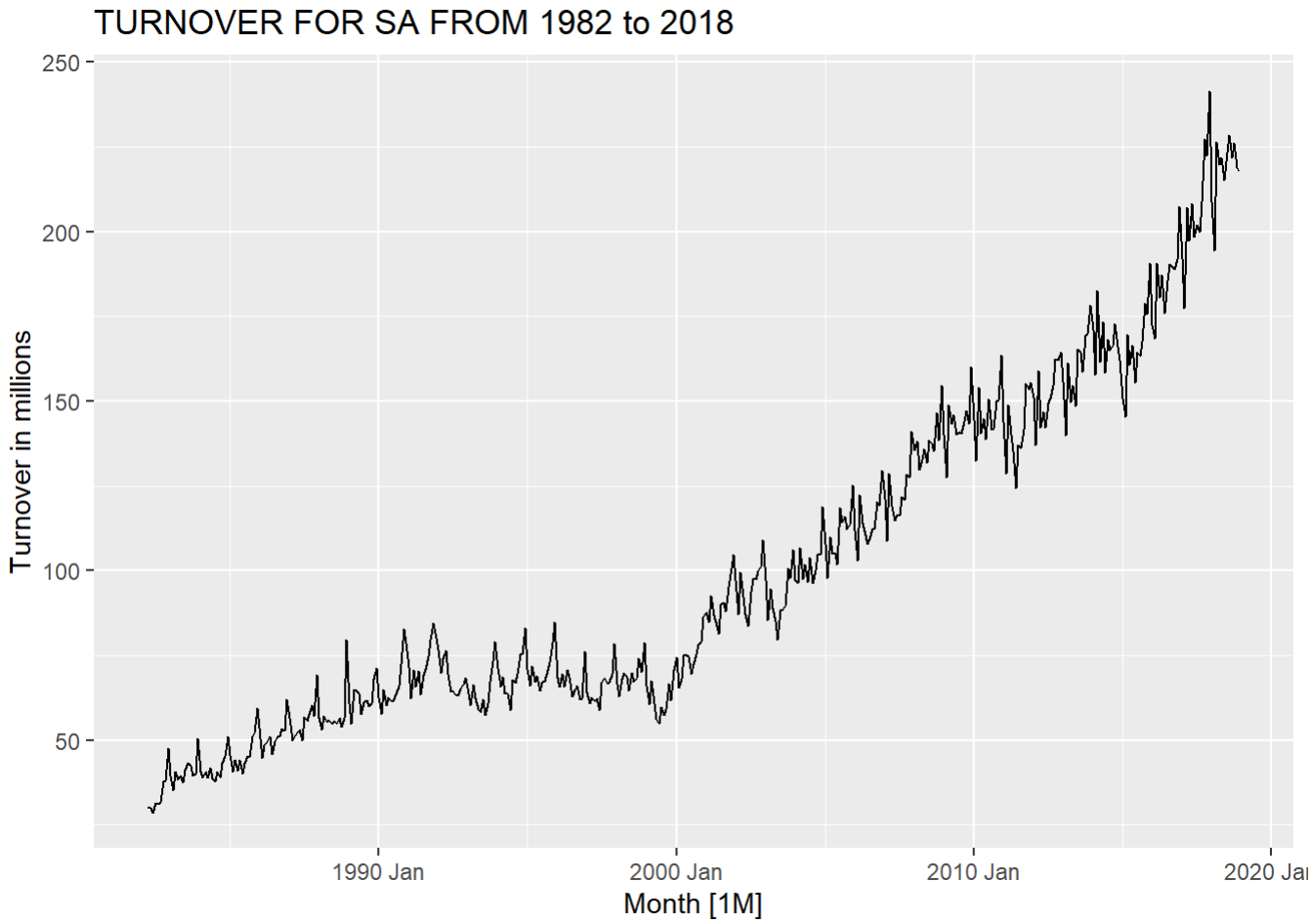
1)I generated myseries data, which is a table that includes turnovers for the Cafes, restaurants and takeaway food services in South Australia from 1982 April till 2018 December.

2)The autoplot function plots Turnover in millions on y axis and Months on X axis. We can see an upward moving trend throughout this period especially after 2000. We can also patterns of seasonality.

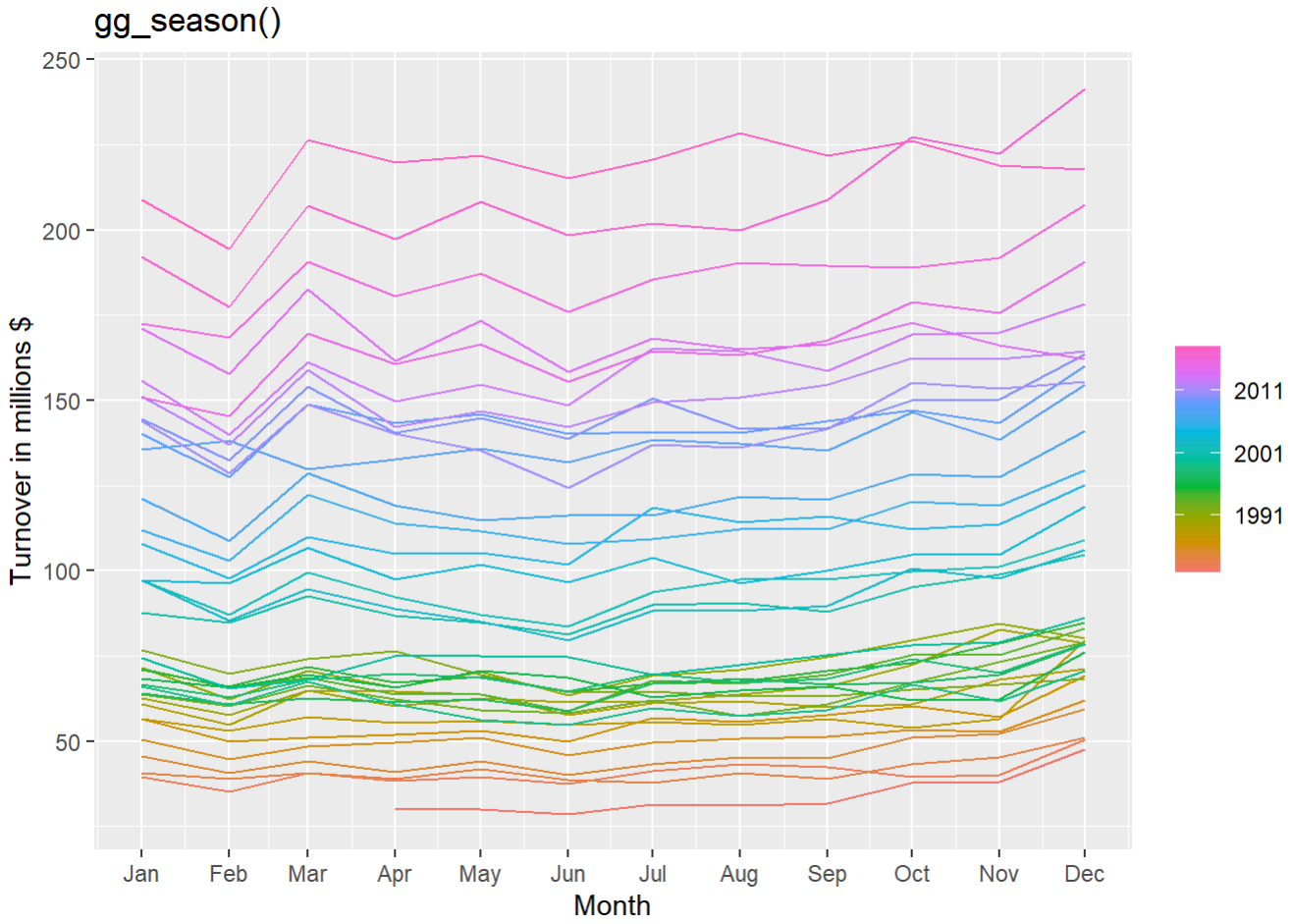
3)To examine more about seasonality, gg_season() is created. This plot confirms seasonality as we can see turnover being the highest in the month of December and then dropping in January. Seasonal patterns of high turnover are also seen the month of March.

4)gg_subseries() plot helps us to identify increase in turnover in each month throughtout the period. All the months show an increase turnover over the years with December being the highest followed by August.

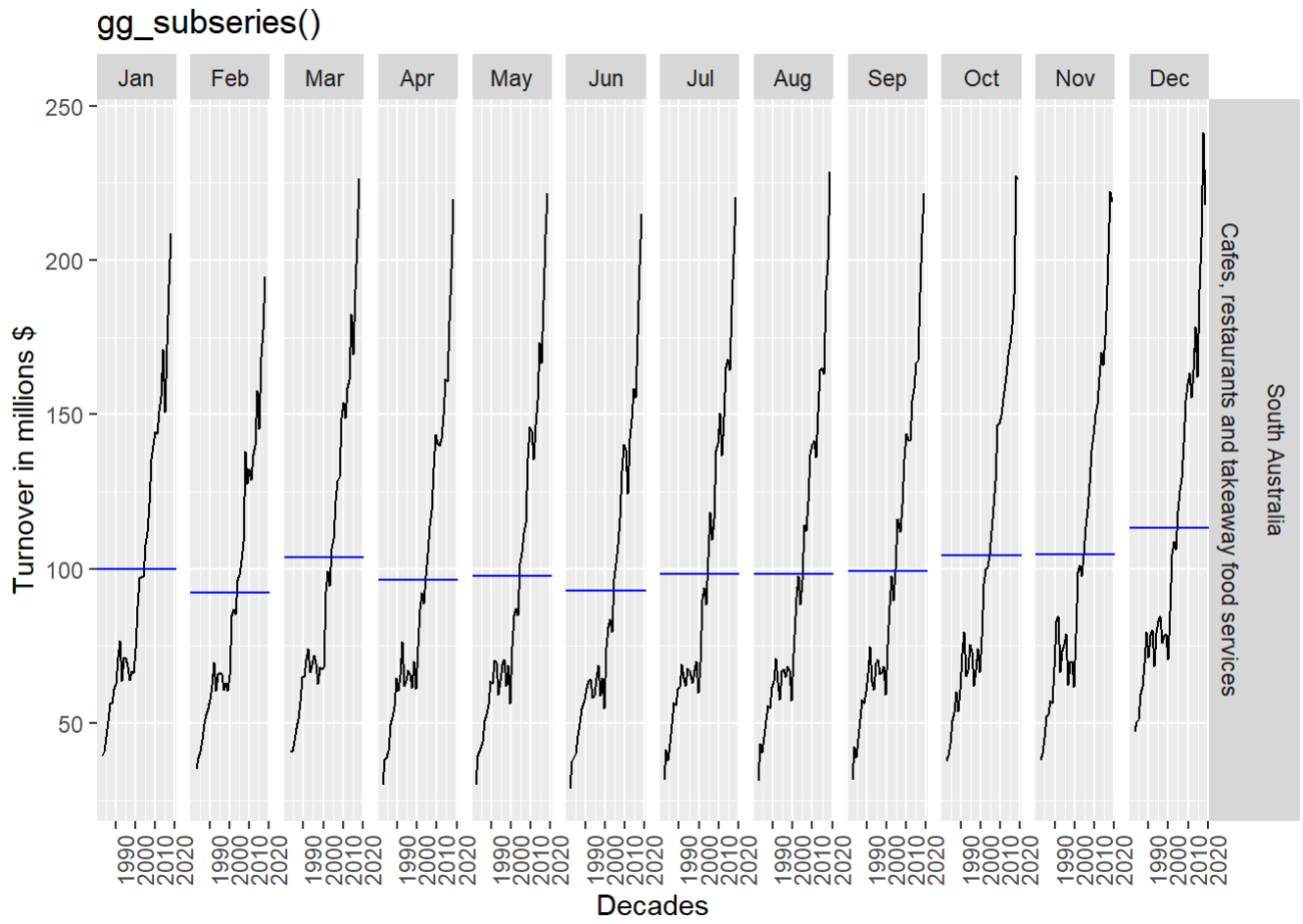
```
View(myseries)

myseries %>%
  autoplot(Turnover)+
  labs(title = "TURNOVER FOR SA FROM 1982 to 2018",
       y="Turnover in millions")
```




```
myseries %>%
   gg_season(Turnover)+
  labs(title = "gg_season() ",
       y="Turnover in millions $")
```




```
myseries %>%
  gg_subseries(Turnover)+
  labs(title = "gg_subseries() ",
       y="Turnover in millions $",
       x="Decades")
```



QUESTION 2

1)BOX_COX Transformation

i)The data has an upward trend which is steeper after year 2000. Also their is seems to be a little higher variance in the end of the dataset than in the beginning. To tackle this problem, we use box_cox transformation which helps us to make our data more linear and try to make the variance more constant.

ii)Using guerrero function to find the best lamba value for the box_cox transformation, I got the best lambda(myseries_lambda) as 0.4648823.

iii)Transforming our data and creating an autoplot of its shows significant changes. The graph is more stable now with the trend not increasing as much after 2000 as it was before. The variance is also more constant.

iv. In myseries_T,I stored the value of this transformed data in a new table which includes all variables from myseries and box_cox_ transformed data.

2. STATIONARITY

i)For stationary, we check the ACF plot. The ACF plot is decreasing slowly but it is not equal to 0 which suggests that we have do not have a stationary data.

ii. We do the unit root test to check if we need our data is stationary or not. The test shows us a pvalue of 0.01 which is less than 0.5 which results in rejecting the null that the data is stationary.

iii)Using the unitroot_nsdiffs feature in R, we get to know that we need 1 seasonal differencing.

iv. Checking for stationary again, I did a unit root test again but on Stationary_data_d1 this time. The unit root test showed a pvalue of 0.096 which is more than 0.05 and thus we do not reject null and conclude that the data is stationary.

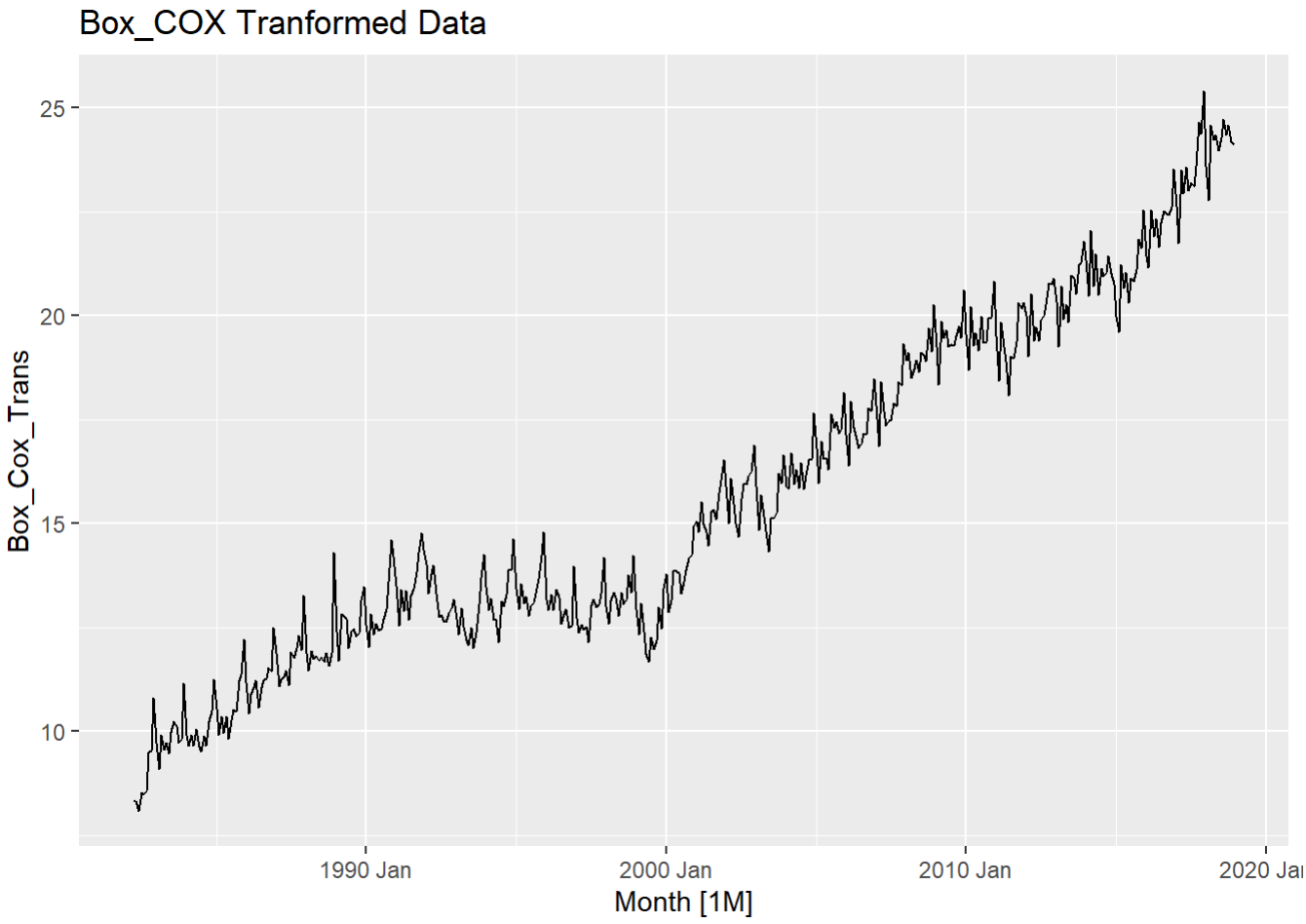
v. To be sure about any further differencing required, we use unitroot_ndiffs feature this time that tells if our data needs any second order differencing. The unitroot_ndiffs feature showed that I do not require any further differencing.

vi. The data looks stationary when plotted using autoplot.

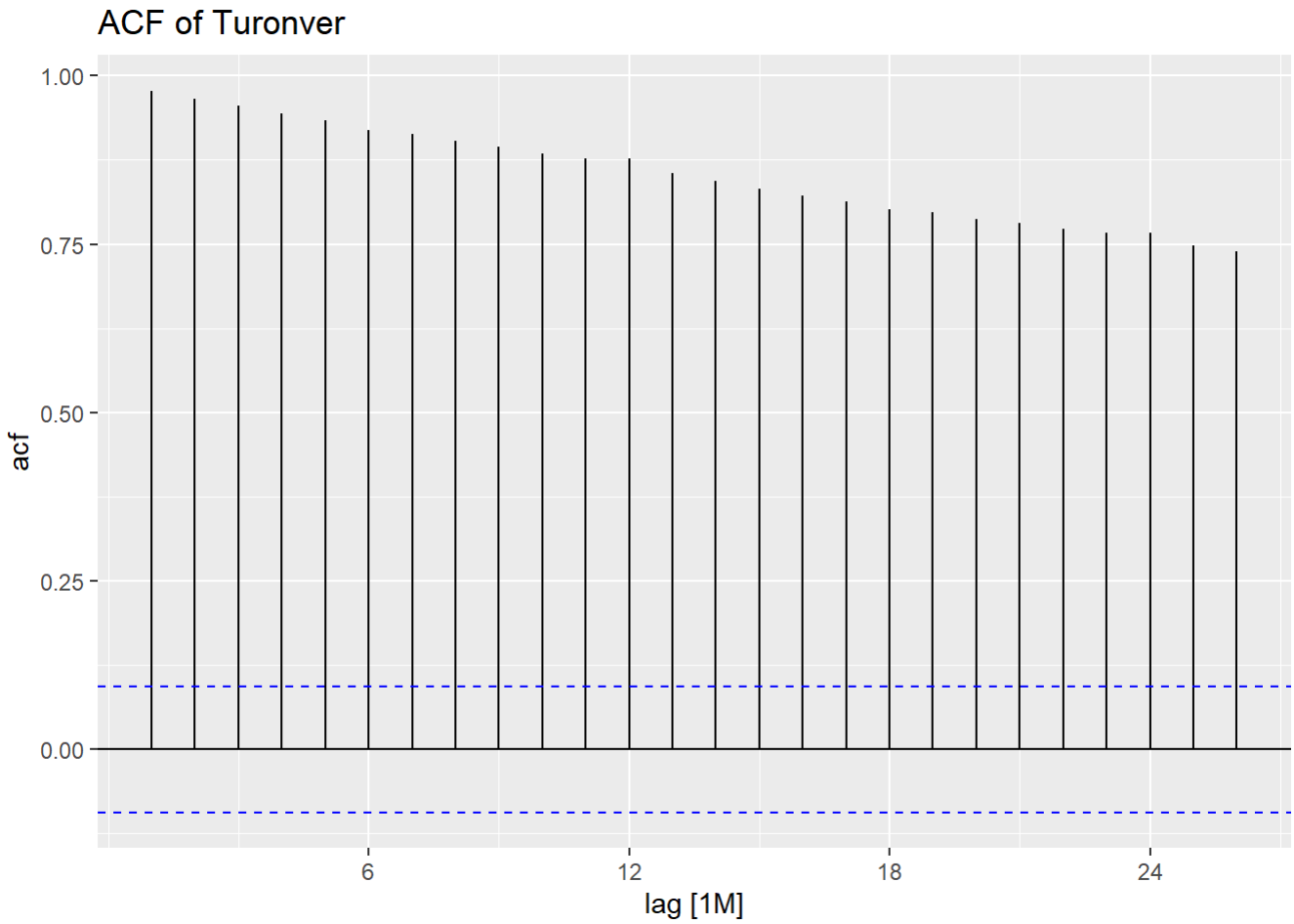
```
##BOX_COX TRANSFORMATION
myseries_lambda= myseries %>%
  features(Turnover, features = guerrero)%>%
  pull(lambda_guerrero)


myseries_lambda
```

```
## [1] 0.4648823
```

```
myseries_T = myseries %>%
  mutate(Box_Cox_Trans =box_cox(Turnover,myseries_lambda))

myseries_T %>%
  autoplot(Box_Cox_Trans)+
  labs(title = "Box_COX Tranformed Data")
```



```
##Removing stationarity

myseries_acf= myseries %>%
  ACF(Turnover) %>%
  autoplot()+
  labs(title = "ACF of Turonver")

myseries_acf
```




```
unit_root_test1=myseries_T %>%
  features(Box_Cox_Trans, unitroot_kpss)

unit_root_test1
```

```
## # A tibble: 1 x 4
##   State           Industry                                  kpss_stat kpss_pvalue
##   <chr>           <chr>                                         <dbl>       <dbl>
## 1 South Australia Cafes, restaurants and takeaway food se~      6.99        0.01
```

```
u_diffs=  myseries_T %>%
  features(Box_Cox_Trans, unitroot_nsdiffs) %>%
  pull(nsdiffs)

u_diffs
```

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

```
myseries_T= myseries_T %>%
  mutate(Stationary_data_d1 = difference(Box_Cox_Trans, 12))


unit_root_test2=myseries_T %>%
  features(Stationary_data_d1, unitroot_kpss)

unit_root_test2
```

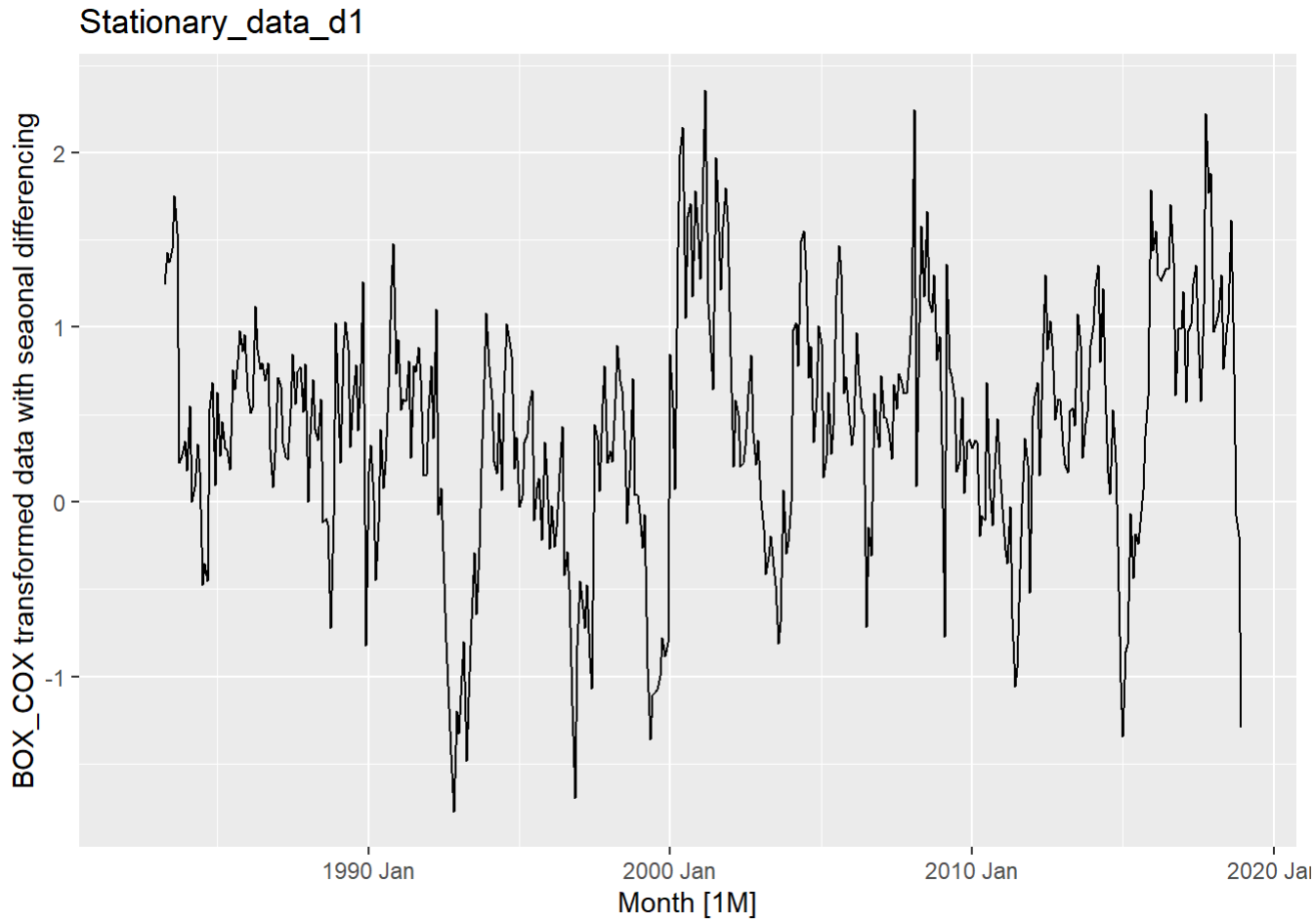
```
## # A tibble: 1 x 4
##   State           Industry                                  kpss_stat kpss_pvalue
##   <chr>           <chr>                                         <dbl>       <dbl>
## 1 South Australia Cafes, restaurants and takeaway food se~     0.356      0.0963
```

```
u_diffs2=  myseries_T %>%
  features(Stationary_data_d1, unitroot_ndiffs) %>%
  pull(ndiffs)

u_diffs2
```

```
## [1] 0
```

```
myseries_T%>%
  autoplot(Stationary_data_d1)+
  labs(title = "Stationary_data_d1",
       y="BOX_COX transformed data with seaonal differencing")
```

```
## Warning: Removed 12 row(s) containing missing values (geom_path).
```



QESTION3

SHORT LISTING THE ETS MODELS

1)I used the automated function of R to calculate the ETS model on Turnover. The function suggested to use ETS~M,N,A model.

2. If we look at the data, we see that their is not a very large difference between the seasonality patterns in our data. This suggests our to use Additive model.
3. Also, the trend is quite positive. It might not be right to short list damped trend for our dataset.
4. So, I short listed the MNA model(auto), MAA model and AAA model.

5)** Checking the report of these models, we will select auto(MNA) model as it has the lowest AICc value. For selecting the ETS models, we ets models

SHORT LISTING ARIMA MODELS

1. The data is seasonal, so we would use ACF/PACF seasonality models to decide what ARIMA models should we take. Also, I have used R function Arima() to calculate the best ARIMA model by stepwise and search methods.
2. We will use Stationary_data_d1 of myseries_T data to do the analysis of ACF and PACF of our model. by using this data we know taht our value for D is equal to 1.

NON SEASONAL 3) Using gg_tdisplay(), we observe the PACF which has a significant spike at lag 3(p) but no other significant spikes. Our ACF does not tells us anything about the non seasonal MR model. So, we short like 3,0,0 for the non seasonal part of the data.

SEASONAL 4) We observe 2 significant spikes at lag 12 and 24 in ACF with very few significant except them. This advices us to use a model 0,1,2 for the seasonal component.

Also, if we start from PACF, after 4 lags we do not see any significant spikes suggesting us to use model 4,1,0.

The two models that we have have short listed are ARIMA(3,0,0)(0,1,2) and ARIMA(3,0,0)(4,1,0)

5. Applying the models tells us that the search and stepwise models gives us the same result and are the best models as they have the lowest AICc values.ARIMA(1,0,1)(0,1,2) is the model.

```
ets_model = myseries_T%>%
  model(
    auto=ETS(Turnover)
  )
ets_model%>%
  report()
```

```
## Series: Turnover
## Model: ETS(M,N,A)
##   Smoothing parameters:
##     alpha = 0.6558654
##     gamma = 0.1183063
##
##   Initial states:
##      l[0]      s[0]     s[-1]     s[-2]    s[-3]    s[-4]    s[-5]     s[-6]
##  32.49669 0.5893261 -3.977381 0.7352613 10.64829 2.086571 1.383442 -1.811429
##       s[-7]     s[-8]     s[-9]   s[-10]    s[-11]
##  -0.8546505 -1.656636 -3.744226 -1.34685 -2.051722
##
##   sigma^2:  0.0022
##
##      AIC     AICc      BIC
## 3948.448 3949.577 4009.783
```

```
ets_models = myseries_T%>%
  model(
    auto=ETS(Turnover),
    etsMAA=ETS( Turnover ~ error("M")+ trend("A")+ season("A")),
    etsAAA= ETS( Turnover ~ error("A")+ trend("A")+ season("A"))

  )

ets_models%>%
  report()
```

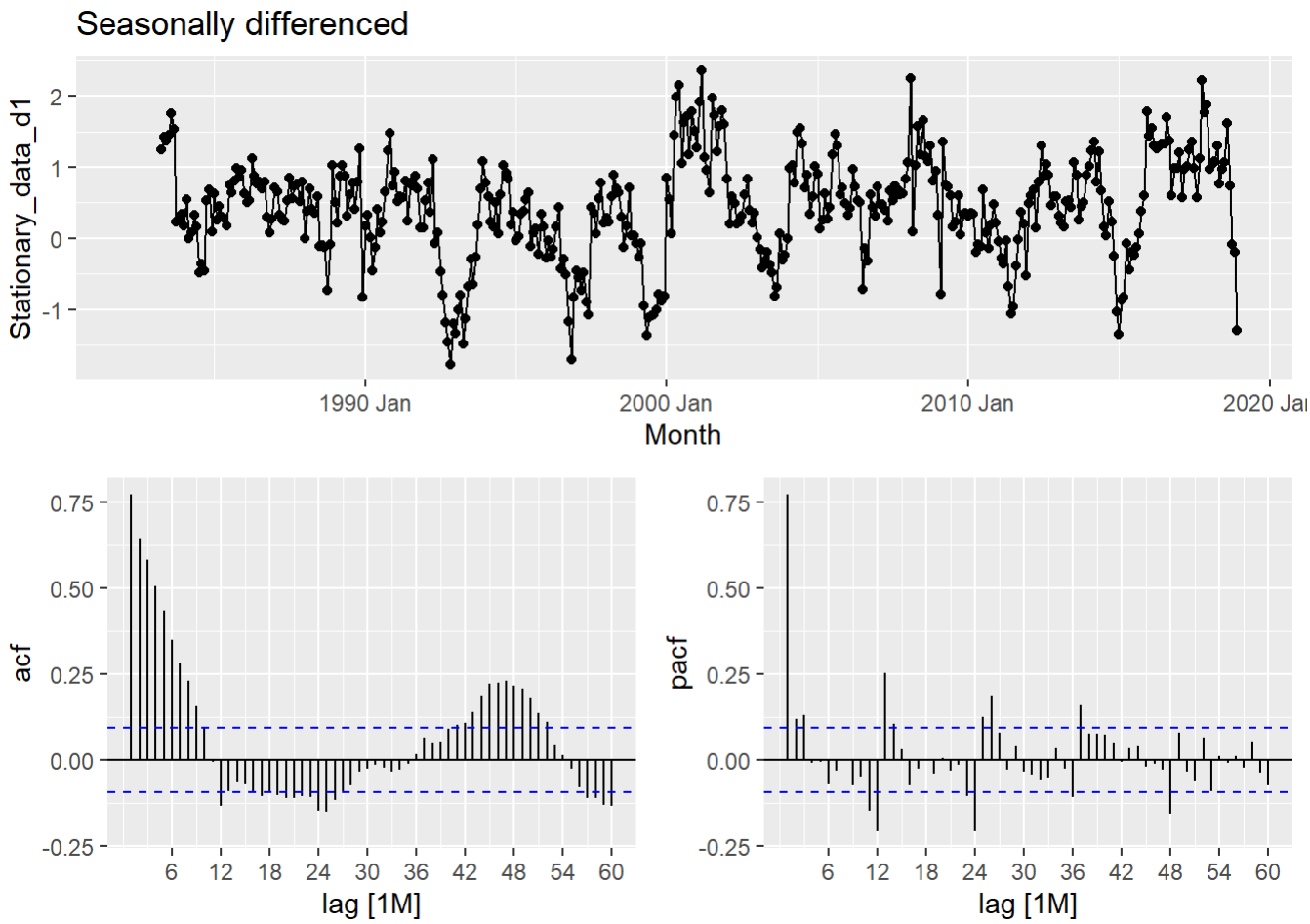
```
## Warning in report.mdl_df(.): Model reporting is only supported for individual
## models, so a glance will be shown. To see the report for a specific model, use
## `select()` and `filter()` to identify a single model.
```

```
## # A tibble: 3 x 11
##   State     Industry .model  sigma2 log_lik   AIC  AICc   BIC   MSE  AMSE    MAE
##   <chr>     <chr>    <chr>    <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 South Au~ Cafes, ~ auto   2.17e-3  -1959. 3948. 3950. 4010.  19.8  26.9 0.0348
## 2 South Au~ Cafes, ~ etsMAA 2.21e-3  -1965. 3964. 3966. 4034.  19.3  25.5 0.0351
## 3 South Au~ Cafes, ~ etsAAA 1.99e+1  -1994. 4022. 4024. 4092.  19.2  25.3 3.28
```

```
##ARIMA MODELS

myseries_T %>%
  gg_tsdisplay(Stationary_data_d1,
               plot_type='partial',lag=60) +
  labs(title="Seasonally differenced")
```

```
## Warning: Removed 12 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 12 rows containing missing values (geom_point).
```




```
arima_models <- myseries_T %>%
  model(
    arima300012 = ARIMA(Box_Cox_Trans ~ 0+pdq(3,0,0) + PDQ(0,1,2)),
    arima300410= ARIMA(Box_Cox_Trans ~ 0 +pdq(3,0,0) + PDQ(4,1,0)),
    stepwise = ARIMA(Box_Cox_Trans),
    search = ARIMA(Box_Cox_Trans, stepwise=FALSE)
  )


arima_models%>%
  report()
```

```
## Warning in report.mdl_df(.): Model reporting is only supported for individual
## models, so a glance will be shown. To see the report for a specific model, use
## `select()` and `filter()` to identify a single model.
```

```
## # A tibble: 4 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 South Aust~ Cafes, ~ arima~  0.132   -180.  372.  373.  397. <cpl>    <cpl>
## 2 South Aust~ Cafes, ~ arima~  0.143   -193.  402.  402.  435. <cpl>    <cpl>
## 3 South Aust~ Cafes, ~ stepw~  0.130   -177.  365.  365.  390. <cpl>    <cpl>
## 4 South Aust~ Cafes, ~ search  0.130   -177.  365.  365.  390. <cpl>    <cpl>
```

```
arima_models%>%
  select(search)%>%
  report()
```

```
## Series: Box_Cox_Trans
## Model: ARIMA(1,0,1)(0,1,2)[12] w/ drift
##
## Coefficients:
##          ar1      ma1     sma1     sma2  constant
##       0.9686  -0.3689  -0.7582  -0.1001    0.0123
## s.e.  0.0139   0.0527   0.0536   0.0521    0.0018
##
## sigma^2 estimated as 0.1302:  log likelihood=-176.64
## AIC=365.27   AICc=365.47   BIC=389.64
```

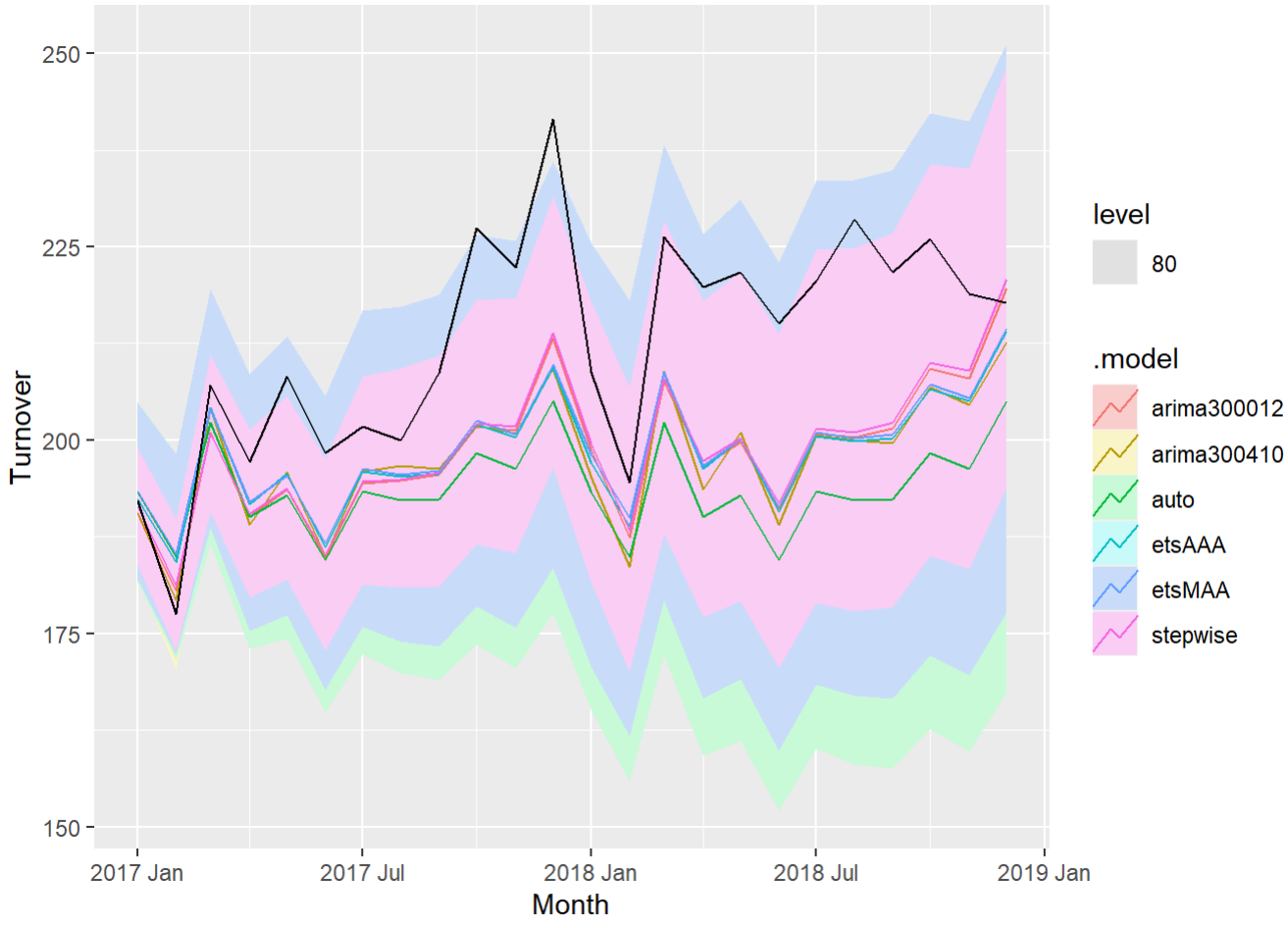
3b) FITTING THE MODELS IN TRAINING SET AND THEN APPLYING IT ON TEST SET.

1. I used all_models variables to create a training set from 1981 Apr to 2016 Dec. Then, the short listed ETS and ARIMA models were applied to this training set.As the search and stepwise had similar results in 3a, we will be using just one model ARIMA(1,0,1)(0,1,2) for our analysis.

2)Use the information from the training set and applying this models on the test set shows us that all the models are catching the pattern of the data but 80% confidence intervals of ETS (M,A,A) and ARIMA(1,0,1)(0,1,2) are the closest to the test data.

3. Further comparing the forecast of each model and calculating its accuracy according the test data tells us that stepwise i.e. ARIMA(1,0,1)(0,1,2) has the lowest RMSE of all the ARIMA models and ETS(M,A,A) has the lowest RMSE with auto being the highest.

```
all_models= myseries_T %>%
  filter(yearmonth(Month) <= yearmonth("2016 DEC"))%>%
  model(
    auto=ETS(Turnover ~ error("M")+ trend("N")+ season("A")),
    etsMAA=ETS( Turnover ~ error("M")+ trend("A")+ season("A")),
    etsAAA= ETS( Turnover ~ error("A")+ trend("A")+ season("A")),
    arima300012 = ARIMA(box_cox(Turnover,myseries_lambda) ~ 0+pdq(3,0,0) + PDQ(0,1,2)),
    arima300410= ARIMA(box_cox(Turnover,myseries_lambda) ~ 0 +pdq(3,0,0) + PDQ(4,1,0)),
    stepwise = ARIMA(box_cox(Turnover,myseries_lambda)~ 0 +pdq(1,0,1) + PDQ(0,1,2))
  )


test_data = myseries_T %>%
  filter(yearmonth(Month) > yearmonth("2016 DEC"))

all_models %>%
  forecast(h="2years")%>%
  autoplot(test_data,level=80)
```




```
all_models %>%
  forecast(h="2years")%>%
  accuracy(test_data)%>%
  arrange(RMSE)
```

```
## # A tibble: 6 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 stepwise  Sout~ Cafes, ~ Test   13.6  16.3  14.1  6.18  6.47   NaN   NaN 0.529
## 2 arima300~ Sout~ Cafes, ~ Test   14.1  16.8  14.5  6.44  6.66   NaN   NaN 0.542
## 3 etsMAA    Sout~ Cafes, ~ Test   13.7  16.9  14.5  6.21  6.63   NaN   NaN 0.594
## 4 etsAAA    Sout~ Cafes, ~ Test   14.2  17.2  14.8  6.42  6.75   NaN   NaN 0.605
## 5 arima300~ Sout~ Cafes, ~ Test   15.1  17.6  15.2  6.87  6.95   NaN   NaN 0.601
## 6 auto      Sout~ Cafes, ~ Test   18.8  22.1  19.5  8.52  8.92   NaN   NaN 0.683
```

Question 4)

I have decided to choose ARIMA(1,0,1)(0,1,2) from all the ARIMA models as it has the lowest AICc and RMSE of all the other ARIMA models and also the 80% Prediction Intervals of this models fits the data well.

For the ETS model, I will be going with ETS(M,A,A) as we can it has the lowest RMSE and its prediction itervals did the best against the test data. Also, the auto generated ETS model on the full data had the highest RMSE.

1)PARAMETER ESTIMATES

Using the report() function,we get the parameter estimates for both of these models.

2. Residual DIAGONOSTICS

ARIMA If we see the ACF of ARIMA(1,0,1)(0,1,2) model, it only has 2 significant spikes. Using the Ljung test, we get a pvalue of .23 which is higher than 0.05 suggesting that the data is not autocorrelated. The innov graph also shows stationary behaviour

ETS If we see our ETS model, it shows 4 significant spikes. The innov residuals looks stationary. Performing the Ljung test, we get a pvalue of 0.0002 which is less than 0.05 suggesting that there may be some autocorrelated residuals in the data.

3. Forecast and prediction Intervals

Using the test data set and the chosen model, I created two graphs to distinctly see how they performed against the test data.

The ETSMAA model is really close to the test data set and alot of the actual values lie in its prediction intervals. The ARIMA model did a good job as well. It captures the trend and the data well.

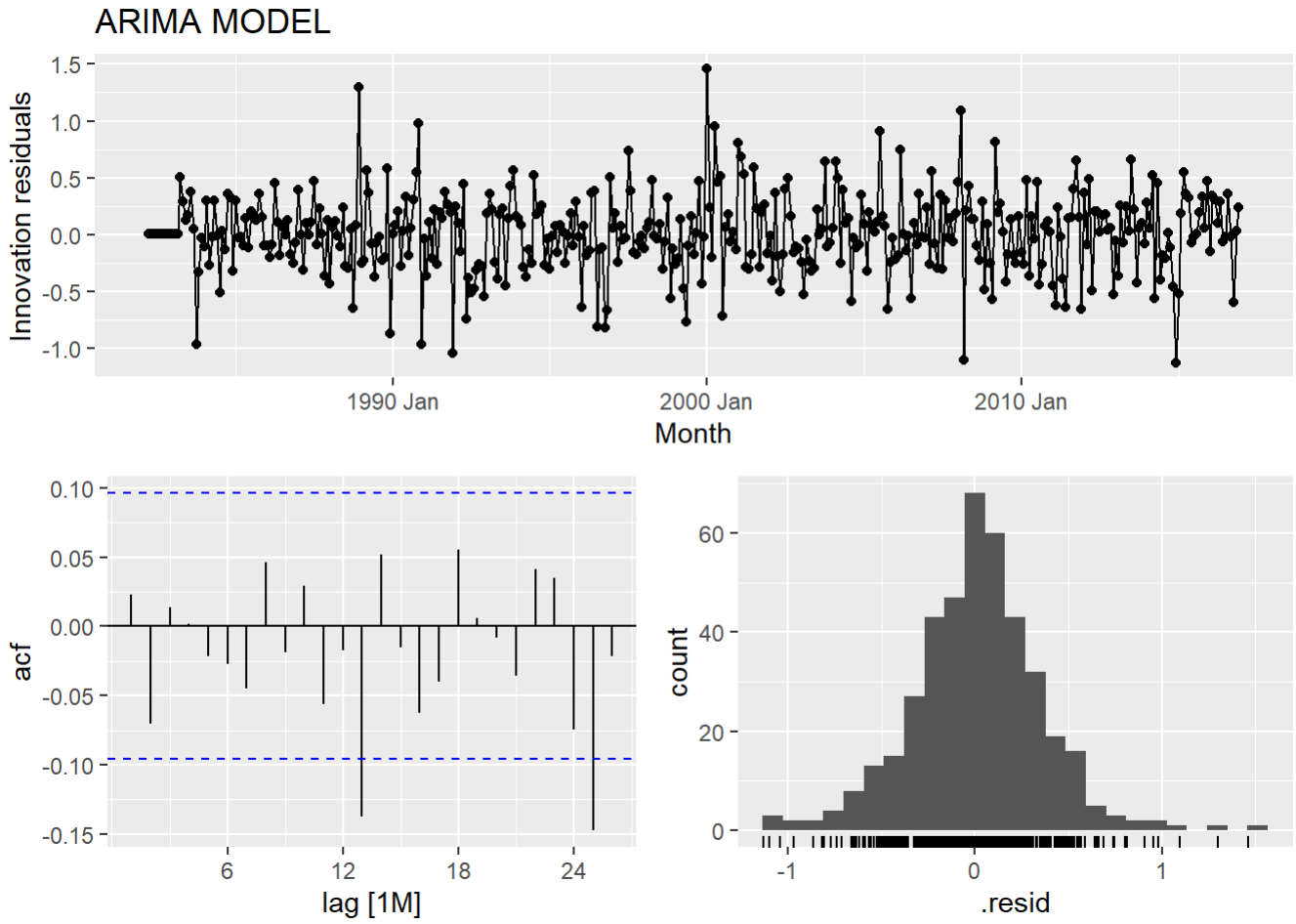
```
##PARAMTER ESTIMATES
all_models %>%
  select(stepwise) %>%
  report()
```

```
## Series: Turnover
## Model: ARIMA(1,0,1)(0,1,2)[12]
## Transformation: box_cox(Turnover, myseries_lambda)
##
## Coefficients:
##          ar1      ma1     sma1     sma2
##       0.9984  -0.4091  -0.7574  -0.1249
## s.e.  0.0023   0.0486   0.0539   0.0536
##
## sigma^2 estimated as 0.1275:  log likelihood=-164.48
## AIC=338.95   AICc=339.11   BIC=358.97
```

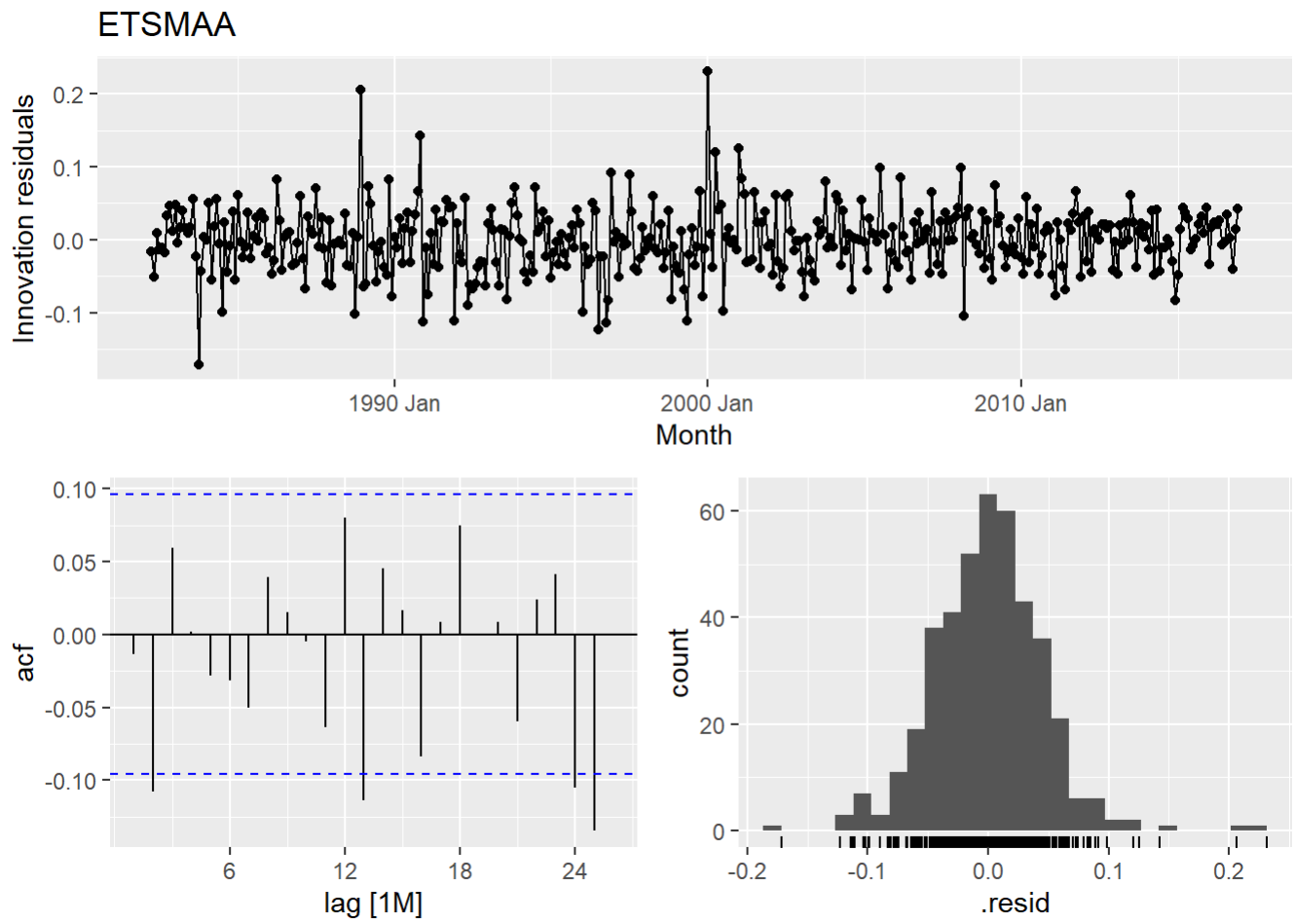
```
all_models %>%
  select(etsMAA) %>%
  report()
```

```
## Series: Turnover
## Model: ETS(M,A,A)
##   Smoothing parameters:
##     alpha = 0.6012073
##     beta  = 0.0001045992
##     gamma = 0.1304033
##
##   Initial states:
##      l[0]      b[0]    s[0]    s[-1]     s[-2]    s[-3]    s[-4]    s[-5]
##  32.92044 0.3914572 1.30319 -3.23975 0.6959263 10.75279 1.620183 2.814159
##      s[-6]    s[-7]     s[-8]     s[-9]    s[-10]    s[-11]
##  -1.447275 -2.06689 -1.550314 -4.396503 -1.780487 -2.705032
##
##   sigma^2:  0.0021
##
##      AIC     AICc      BIC
## 3670.979 3672.513 3739.541
```

```
##RESIDUAL DIAGONOSTICS

all_models %>%
  select(stepwise) %>%
  gg_tsresiduals()+
  labs(title = "ARIMA MODEL")
```




```
all_models %>%
  select(etsMAA) %>%
  gg_tsresiduals()+
  labs(title="ETSMAA")
```



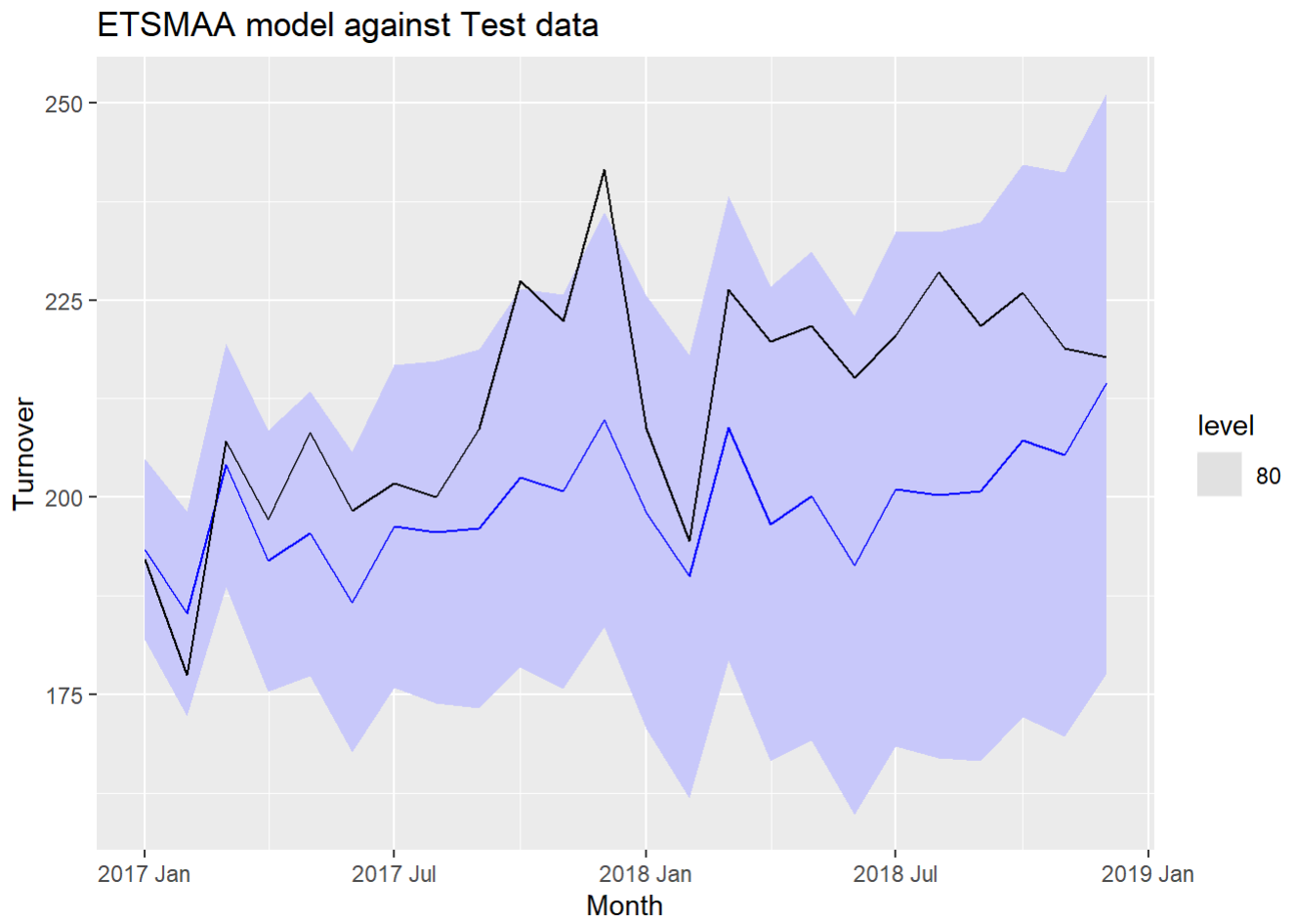

```
augment(all_models) %>%
  filter(.model=='stepwise') %>%
  features(.innov, ljung_box, lag = 24, dof = 4 )
```

```
## # A tibble: 1 x 5
##   State           Industry                                  .model lb_stat lb_pvalue
##   <chr>           <chr>                                     <chr>    <dbl>     <dbl>
## 1 South Australia Cafes, restaurants and takeaway food~ stepw~    24.2     0.233
```

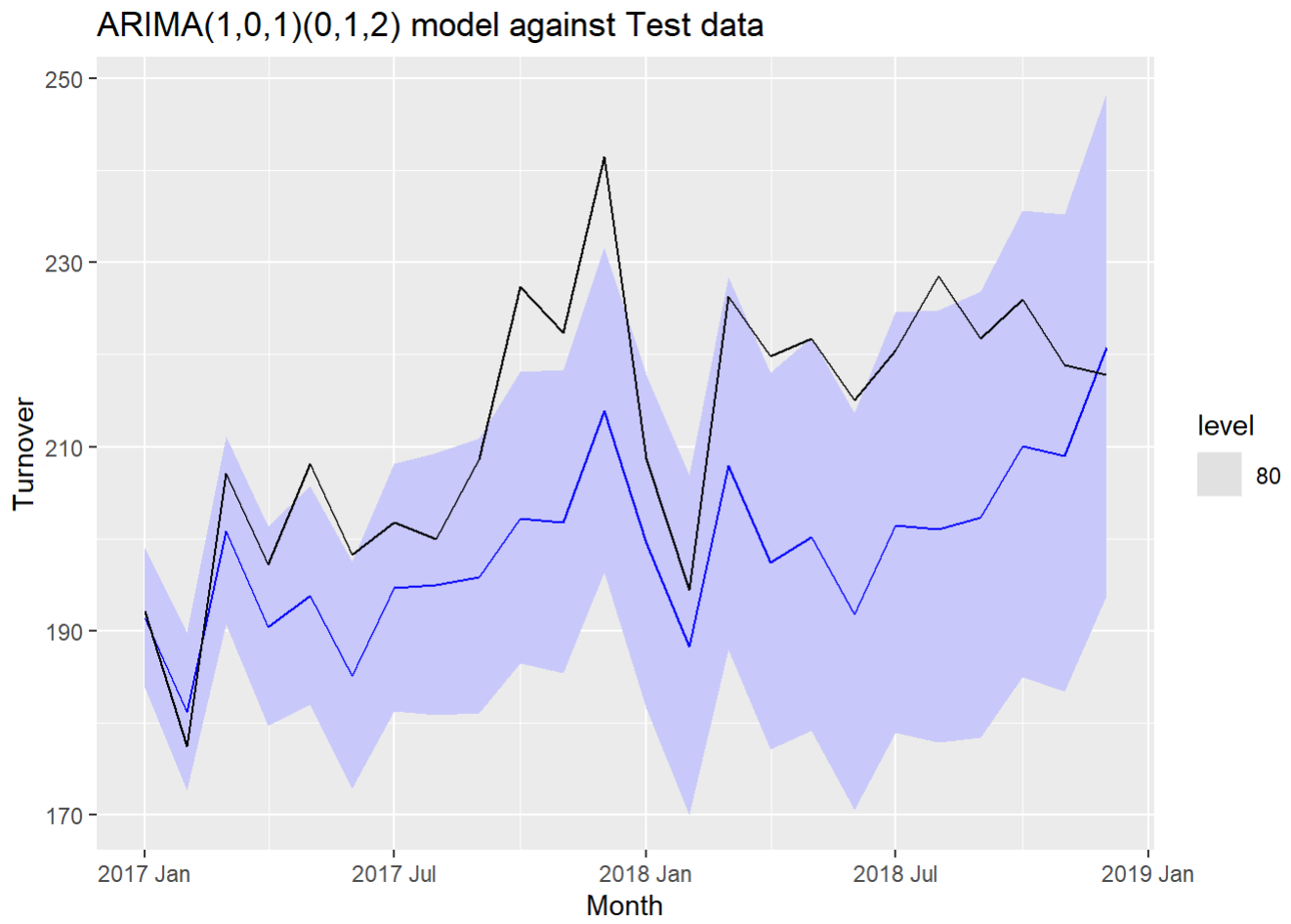
```
augment(all_models) %>%
  filter(.model=='etsMAA') %>%
  features(.innov, ljung_box, lag=24, dof = 14)
```

```
## # A tibble: 1 x 5
##   State           Industry                                  .model lb_stat lb_pvalue
##   <chr>           <chr>                                     <chr>    <dbl>     <dbl>
## 1 South Australia Cafes, restaurants and takeaway food~ etsMAA    33.2  0.000249
```

```
##Forecast and Prediction Intervals


all_models %>%
  forecast(h="2years")%>%
  filter(.model=='etsMAA')%>%
  autoplot(test_data,level=80)+
  labs(title = "ETSMAA model against Test data")
```




```
all_models %>%
  forecast(h="2years")%>%
  filter(.model=='stepwise')%>%
  autoplot(test_data,level=80)+
  labs(title = "ARIMA(1,0,1)(0,1,2) model against Test data")
```

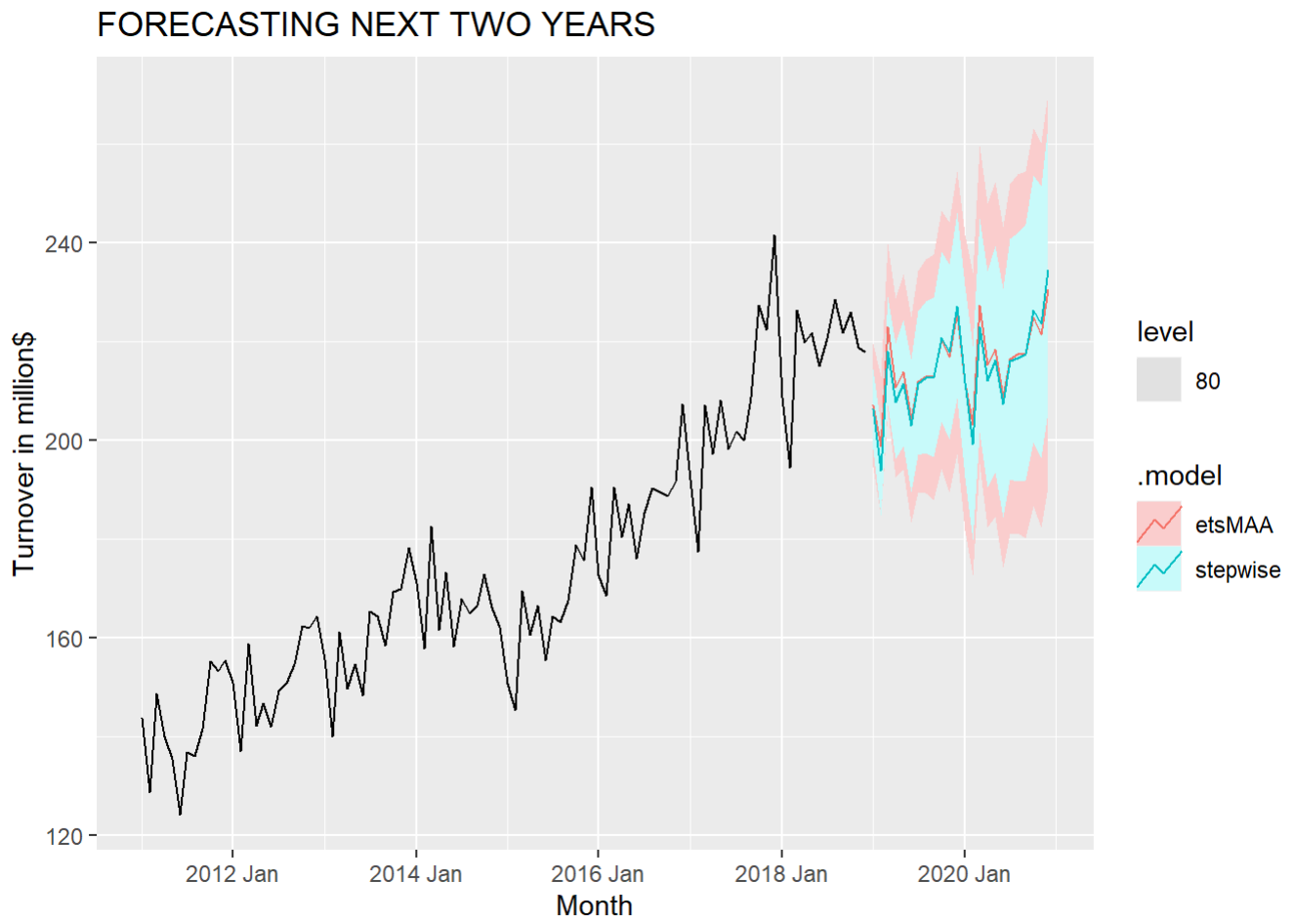


Question 5)

Comparing the two models, I think ARIMA is better model than ETSMAA model. Although ETSMAA model fits well to the test data set it fails to pass the Ljung Box test. On the other hand, ARIMA did well on plotting the test data set values and also passed the residual diagonostic tests.

question 6 I forecast for 2 years using both the models. To have a better view we are seeing the data from from 2010. The ETS model has a wider prediction intervals than the ARIMA model.

```
final_models= myseries_T%>%
  model(
    etsMAA=ETS( Turnover ~ error("M")+ trend("A")+ season("A")),
    stepwise = ARIMA(box_cox(Turnover,myseries_lambda)~ 0 +pdq(1,0,1) + PDQ(0,1,2))
  )


final_models %>%
  forecast(h="2years")%>%
  autoplot(myseries_T%>% filter(yearmonth(Month) > yearmonth("2010 DEC")),level=80)+
  labs(title="FORECASTING NEXT TWO YEARS",y="Turnover in million$")
```



Question 7

This part is to apply our forecasts model to the actual values observed. Using the accuracy function, we get ARIMA(1,0,1)(0,1,2) model has lower RMSE values than ETS MAA model.

Plotting the forecast against the actual numbers show little difference between the ETS MAA model and the ARIMA (1,0,1)(0,1,2) as it is the covid period which is like a cyclic(rare event).

THe models peroformed really well in 2019 but did not do well in 2020.

```
abs_data <- readabs::read_abs(series_id = myseries$`Series ID`[1]) %>%
  mutate(
    Month = yearmonth(date),
    Turnover = value
  ) %>%
  select(Month, Turnover) %>%
  filter(Month > max(myseries_T$Month)) %>%
  as_tsibble(index=Month)
```

```
## Finding URLs for tables corresponding to ABS series ID
```

```
## Attempting to download files from series ID , Retail Trade, Australia
```

```
## Downloading https://www.abs.gov.au/statistics/industry/retail-and-wholesale-trade/retail-trade-australia/latest-release/8501011.xlsx
```

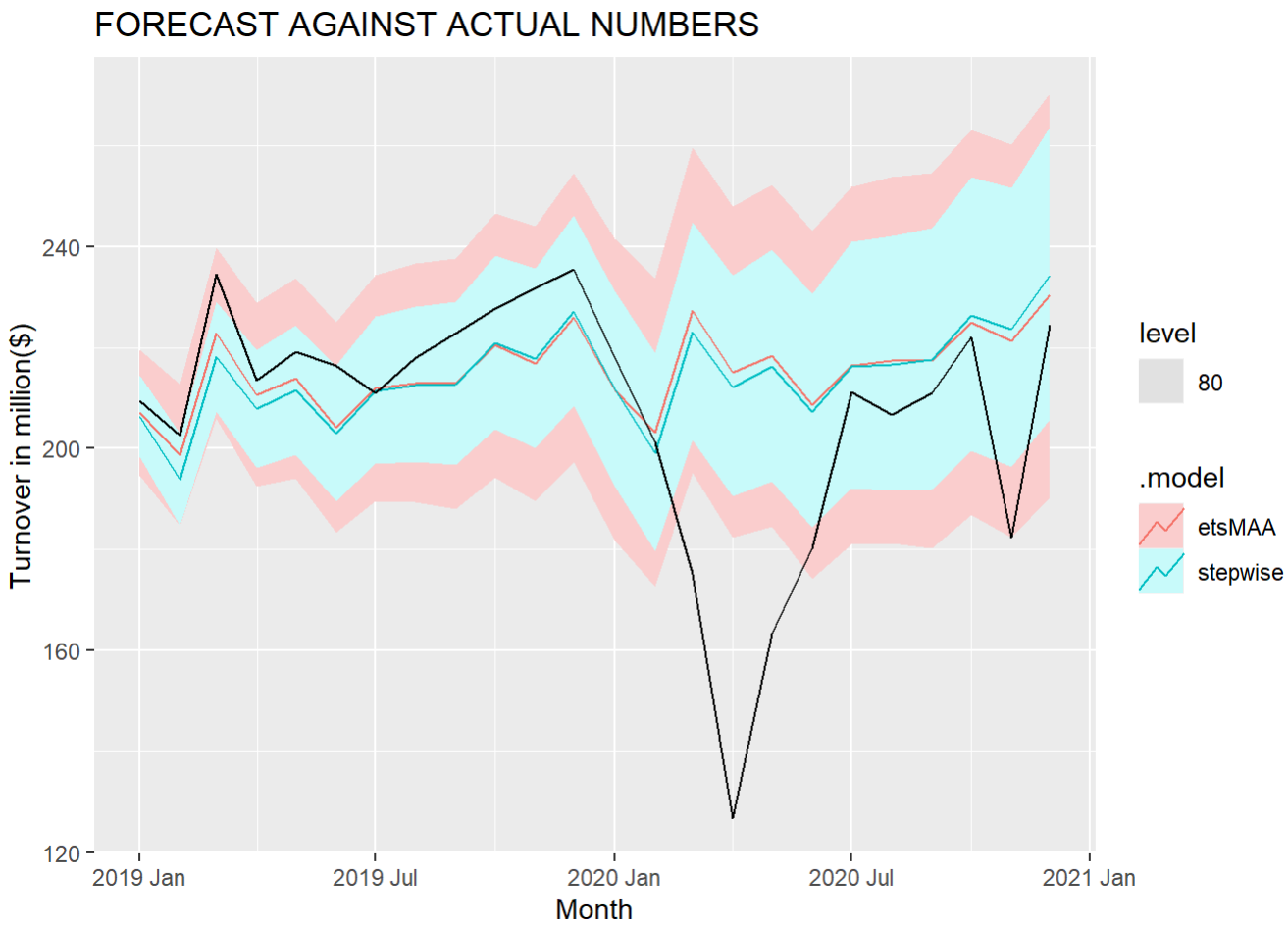
```
## Extracting data from downloaded spreadsheets
```

```
## Tidying data from imported ABS spreadsheets
```

```
fc = final_models%>%
  forecast(h=24)
fc %>%
  accuracy(abs_data)
```

```
## # 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 etsMAA   Test  -8.57  26.6  16.2 -6.06  9.46   NaN   NaN 0.693
## 2 stepwise Test  -7.56  25.9  16.6 -5.51  9.58   NaN   NaN 0.689
```

```
data= abs_data%>%
  filter(Month <= yearmonth("2020 Dec"))

forecast_a_abs = myseries_T%>%
  select(Month, Turnover)%>%
  model(
    etsMAA=ETS( Turnover ~ error("M")+ trend("A")+ season("A")),
    stepwise = ARIMA(box_cox(Turnover,myseries_lambda)~ 0 +pdq(1,0,1) + PDQ(0,1,2))
  )%>%
  forecast(h="2 years")%>%
  autoplot(data,level=80)+
  labs(title="FORECAST AGAINST ACTUAL NUMBERS", y = "Turnover in million($)")

forecast_a_abs
```



Question 8

ETS MAA

BENEFITS The model had the least RMSE of all the models and is fitting the dataset quite well.It has quite big prediction intervals that covered a majority of the test data values. Until the start of 2020, this model was really good in predicting the movement the turnover and did better than the arima model. Its prediction intervals had the actual values in them.

Limitation It failed to pass the Ljung Box test.

ARIMA(1,0,1)(0,1,2)

BENFITS This model fitted had a really low AICc value and also a low RMSE value. The model did well against the test data. It passed all the residuals diagonostics as well.

Limitation It is not fitting the actual values in 2019 as good as the ETS MAA model.