ETF3231/5231 Individual Assignment 4

Darren Luwi (ID 29051754)

```
# Read in and tidy up your data
# Make sure you select the column with your student ID

# First three rows contain metadata, read them in separately
meta <- read_csv("Undergrad_Data.csv", col_names = TRUE, n_max = 3)

##
## -- Column specification -----------------------------------------------------
------
## cols(
##   .default = col_character()
## )
## i Use `spec()` for the full column specifications.

# meta
# The data follows after the third row, we skip the metadata and read the
data.
# Note: Skipping the first row skips the column names, we add them back from
the
#       metadata.
dat <- read_csv("Undergrad_Data.csv",
                # use column names from the metadata
                col_names = colnames(meta),
                # skip 4 rows as we also skip column names, specified above
                skip = 4,
                # The automatic column types correctly guess all columns but
the
                # date, we specify the date format manually here to correctly
                # get dates.
                col_types = cols("Student ID" = col_date("%b-%y")))

my_series <- dat %>%
  # feel free to rename your series appropriately
  rename(Month = "Student ID", y ="29452902") %>%
  select(Month, y) %>%
  mutate(Month=yearmonth(Month)) %>%
  as_tsibble(index = Month)

read_xlsx("C:\\Users\\user\\Desktop\\Monash Uni\\Semester 1 2021\\Business
Forecasting\\IA\\Undergrad_Data1.xlsx")

## # A tibble: 468 x 2
##    Months                  y
##    <dttm>              <dbl>
```

```
##   1 1982-04-01 00:00:00  303.
##   2 1982-05-01 00:00:00  298.
##   3 1982-06-01 00:00:00  298
##   4 1982-07-01 00:00:00  308.
##   5 1982-08-01 00:00:00  299.
##   6 1982-09-01 00:00:00  305.
##   7 1982-10-01 00:00:00  318
##   8 1982-11-01 00:00:00  334.
##   9 1982-12-01 00:00:00  390.
## 10 1983-01-01 00:00:00  311.
## # ... with 458 more rows
```

```
save.image("Undergrad_Data1.RData")
```

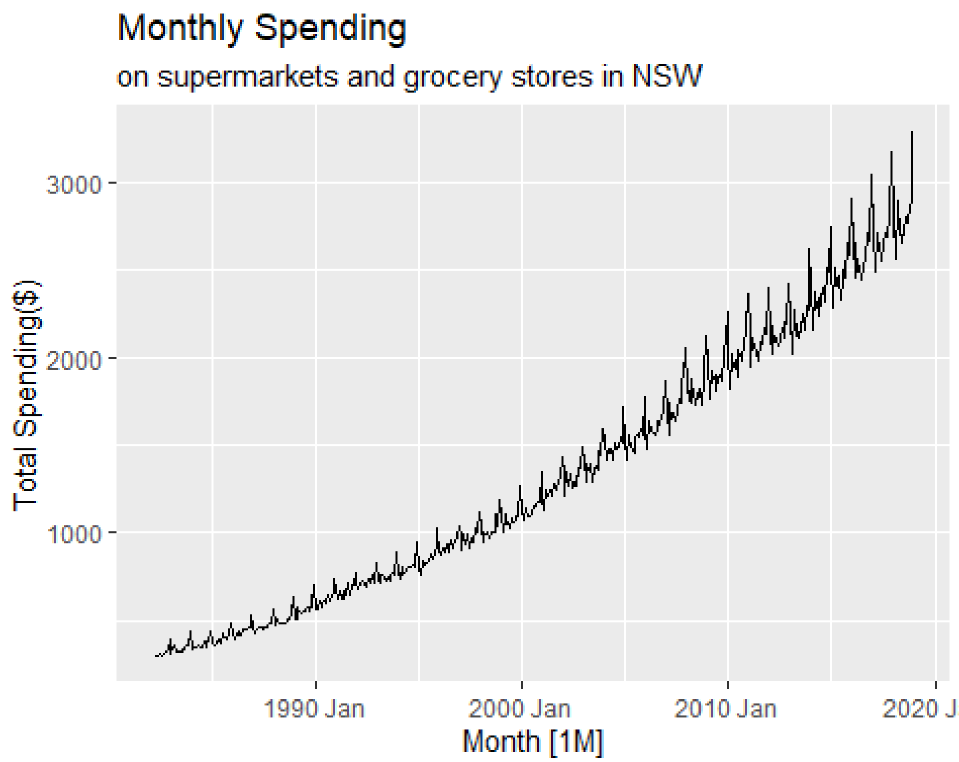
Your aim in the first part of the assignment is to build an ARIMA model and use this to forecast. Recall the first step in ARIMA modelling is to stabilize the variance of your data. If you decided that your data required a transformation in the previous assignments you will be required to use the same transformation for what follows (unless you have a reason to change your mind - please do so if you think it is necessary).

Question 1

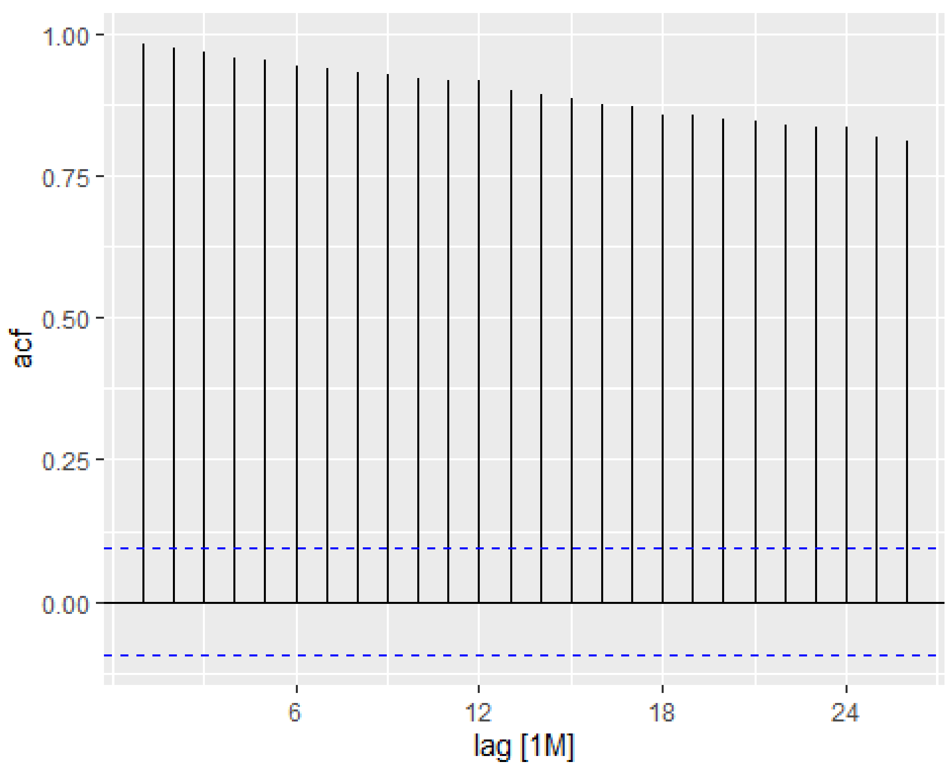
Visually inspect your data and decide on the transformation and what differencing is required to achieve stationarity. Plot the data at every step and comment on each plot justifying your actions. (No more than 50 words per plot). (6 marks)

```
autoplot(my_series) + labs(title = "Monthly Spending", subtitle = "on
supermarkets and grocery stores in NSW", y= "Total Spending($)")
```

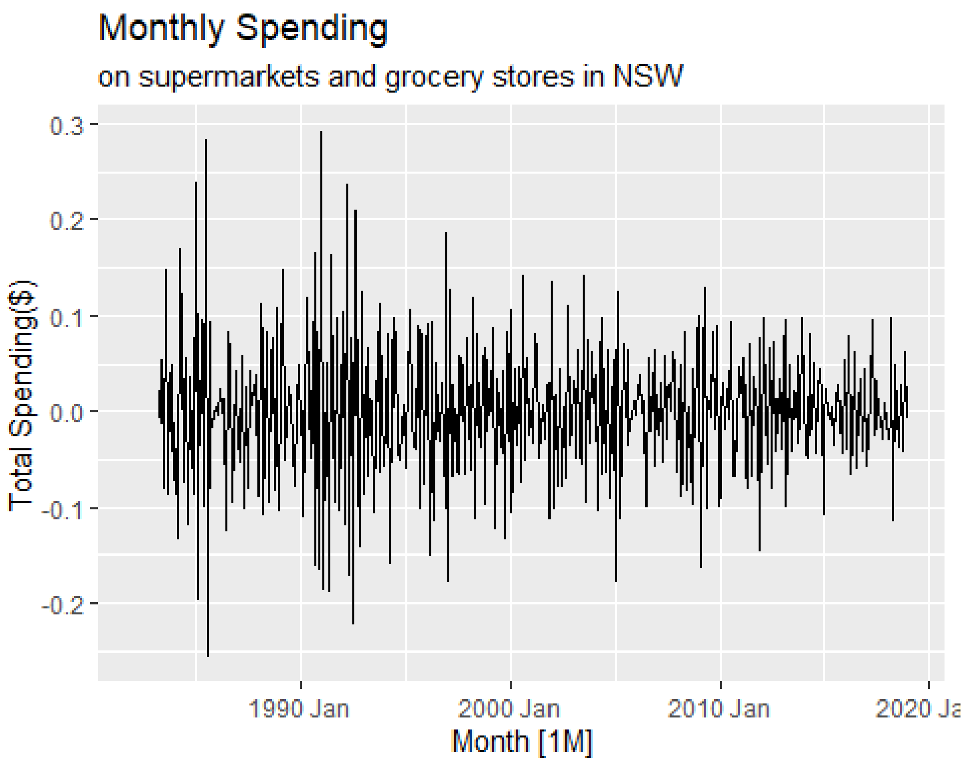
```
## Plot variable not specified, automatically selected `.vars = y`
```




```r
my_series %>%
  ACF(y) %>% autoplot()
```



```r
##There is an increasing trend in monthly spending, while the variance is
also not constant.
##Next, notice that the ACF spikes decreases slowly.
##Thus we assume non-stationarity.
##To confirm:

my_series %>%
  features(y, unitroot_kpss)

## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##       <dbl>       <dbl>
## 1      7.30        0.01

##as p-value < 0.05, we reject the null and conclude that the data is indeed
non-stationary.

lambda <- my_series %>%
  features(y, features = guerrero) %>%
  pull(lambda_guerrero)
lambda

## [1] 0.119595

my_series %>% features(box_cox(y,lambda),list(unitroot_nsdiffs,feat_stl))

## # A tibble: 1 x 10
##   nsdiffs trend_strength seasonal_strength_~ seasonal_peak_ye~
seasonal_trough_~
##     <int>          <dbl>               <dbl>             <dbl>
<dbl>
## 1       1          0.999               0.890                 9
11
## # ... with 5 more variables: spikiness <dbl>, linearity <dbl>, curvature
<dbl>,
## #   stl_e_acf1 <dbl>, stl_e_acf10 <dbl>

my_series %>%
  mutate(bc_y = difference(box_cox(y,lambda),12)) %>%
  features(bc_y, unitroot_ndiffs)

## # A tibble: 1 x 1
##   ndiffs
##    <int>
## 1      1

my_series %>%
  autoplot(box_cox(y,lambda) %>% difference(12) %>% difference(1)) +
labs(title = "Monthly Spending", subtitle = "on supermarkets and grocery
stores in NSW", y= "Total Spending($)")

## Warning: Removed 13 row(s) containing missing values (geom_path).
```




```
##unitroot_nsdiffs suggests that a seasonal difference is required.
##Simultaneosly, unitroot_ndiffs implies that a first-order diffence is also
needed.
##To verify if the now-transformed data is stationary:

my_series %>%
  features
((box_cox(y,lambda)%>%difference(12)%>%difference(1)),unitroot_kpss)

## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##       <dbl>       <dbl>
## 1    0.0118         0.1

##p-value is now greater than 0.05, hence we do not reject null and conclude
that the data is now stationary.
```

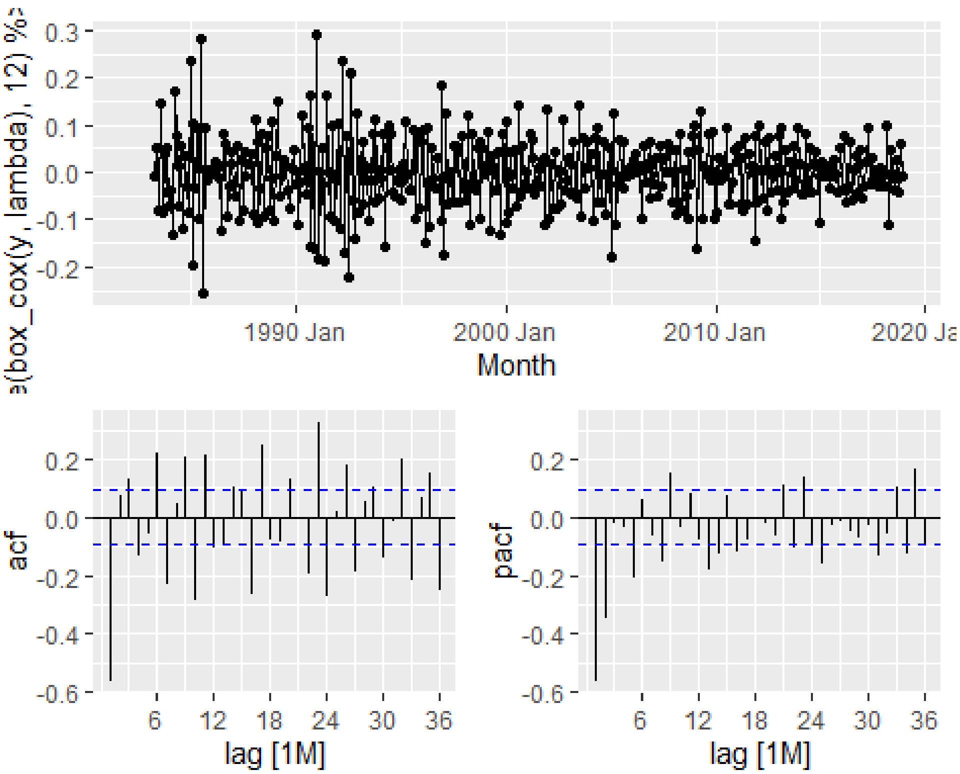
Question 2

Plot the ACF and PACF of the stationary data. Reading from these choose an appropriate ARIMA model. Make sure you justify your choice. (No more than 70 words in total – do not revise the theory – describe what you see in your plots and decide what ARIMA orders may be appropriate. Also note that it is highly likely that the ACF and PACF plots will be very messy. Do the best you can). (6 marks)

```
my_series %>%
  gg_tsdisplay(difference(box_cox(y,lambda),12)%>% difference(), lag_max= 36,
plot_type = 'partial')
```

```
## Warning: Removed 13 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 13 rows containing missing values (geom_point).
```



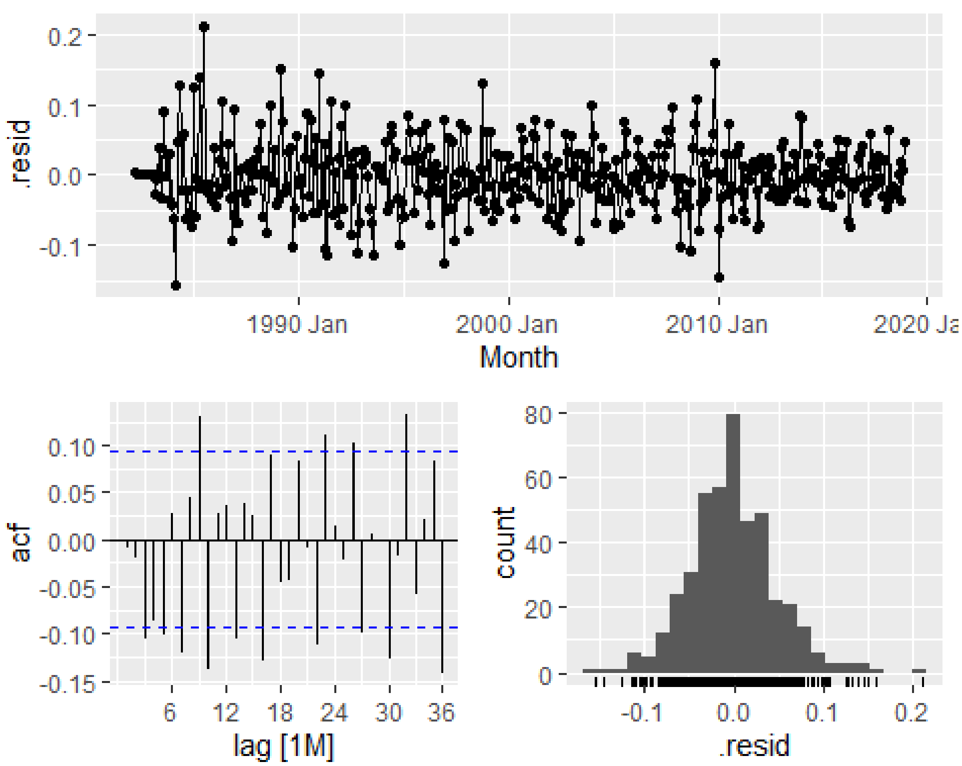

```
##From ACF:
##Significant Spike at lag 24 and 36= Seasonal MA(3).
##Significant Spikes at lag 1 and 3 = non-seasonal MA(3).
##From PACF:
##No seasonally Significant spikes = Seasonal AR(0)
##Significant spikes at lag 1 and 2 = non-seasonal AR(2).
##We have two candidates: ARIMA(0,1,3)(0,1,3) and ARIMA(2,1,0)(0,1,3)
fit <- my_series %>%
  model(
    arima013013 = ARIMA(box_cox(y,lambda)~0+ pdq(0,1,3)+PDQ(0,1,3)),
    arima210013 = ARIMA(box_cox(y,lambda)~0 + pdq(2,1,0)+PDQ(0,1,3))
```

```
  )
glance(fit)

## # A tibble: 2 x 8
##   .model       sigma2 log_lik    AIC   AICc    BIC ar_roots  ma_roots
##   <chr>         <dbl>   <dbl>  <dbl>  <dbl>  <dbl> <list>    <list>
## 1 arima013013 0.00246    674. -1334. -1334. -1306. <cpl [0]> <cpl [39]>
## 2 arima210013 0.00245    675. -1338. -1338. -1314. <cpl [2]> <cpl [36]>

##Since ARIMA(2,1,0)(0,1,3) have lower AICc value, we settle with it for the
time being.
arima210013 <- my_series %>%
  model(arima013013 = ARIMA(box_cox(y,lambda)~0+pdq(2,1,0)+PDQ(0,1,3)))
```

Question 3

Check the whiteness of the residuals from the fitted ARIMA model. Based on these evaluate and if necessary review the ARIMA model specified in Q2. (No more than 50 words). (4 marks)

```
report(arima210013)

## Series: y
## Model: ARIMA(2,1,0)(0,1,3)[12]
## Transformation: box_cox(y, lambda)
##
## Coefficients:
##           ar1      ar2      sma1     sma2    sma3
##       -0.7675  -0.3929  -0.5836  -0.3275  0.0790
## s.e.   0.0449   0.0464   0.0618   0.0534  0.0748
##
## sigma^2 estimated as 0.002445:  log likelihood=675.14
## AIC=-1338.27   AICc=-1338.07   BIC=-1313.92

arima210013 %>%
  gg_tsresiduals(lag_max=36)
```




```
##residuals distribution looks normal
##But ACF suggests non-white noise.
##To verify:
augment(arima210013) %>%
  features(.innov,ljung_box,lag=36,dof=5)
```

```
## # A tibble: 1 x 3
##   .model      lb_stat lb_pvalue
##   <chr>         <dbl>     <dbl>
## 1 arima013013    113.  2.95e-11

##dof = no.of coefficients = 5
##lag = 36
##Null = White Noise
##Since p-value = 0 (<0.05), we reject the Null and conclude non-white noise.
##Thus we need to review the selected ARIMA model.
```

Question 4

Consider three (up to five if you think you need them) alternative ARIMA models based on your choice in Q2 and Q3. (Very briefly justify each choice with no more than 1 or 2 lines each). Use information criteria to choose the best model you have considered so far. (6 marks)

```
arima014013 <- my_series %>%
  model(arima014013 = ARIMA(box_cox(y,lambda)~0+pdq(0,1,4)+PDQ(0,1,3)))
##From the ACF, spike at lag 4 might be significant, thus we take non-
seasonal MA(4) into consideration

arima210011 <- my_series %>%
  model(arima210011 = ARIMA(box_cox(y,lambda)~0+pdq(2,1,0)+PDQ(0,1,1)))

arima013011<- my_series %>%
  model(arima013011 = ARIMA(box_cox(y,lambda)~0+pdq(0,1,3)+PDQ(0,1,1)))

arima014011<- my_series %>%
  model(arima014011 = ARIMA(box_cox(y,lambda)~0+pdq(0,1,4)+PDQ(0,1,1)))
##Again from the ACF, spike at lag 12 is slightly above the blue line, and
thus it might be more significant (ie we try seasonal ARIMA(0,1,1) on non-
seasonal ARIMA(2,1,0), (0,1,3) and (0,1,4))

fit2 <- my_series %>%
  model(
    arima014013 = ARIMA(box_cox(y,lambda)~0+pdq(0,1,4)+PDQ(0,1,3)),
    arima210011 = ARIMA(box_cox(y,lambda)~0+pdq(2,1,0)+PDQ(0,1,1)),
    arima013011 = ARIMA(box_cox(y,lambda)~0+pdq(0,1,3)+PDQ(0,1,1)),
    arima014011 = ARIMA(box_cox(y,lambda)~0+pdq(0,1,4)+PDQ(0,1,1))
  )

glance(fit2)

## # A tibble: 4 x 8
##   .model      sigma2 log_lik   AIC  AICc   BIC ar_roots   ma_roots
##   <chr>        <dbl>   <dbl> <dbl> <dbl> <dbl> <list>     <list>
## 1 arima014013 0.00239    681. -1346. -1346. -1313. <cpl [0]> <cpl [40]>
## 2 arima210011 0.00264    658. -1309. -1309. -1293. <cpl [2]> <cpl [12]>
## 3 arima013011 0.00272    652. -1294. -1294. -1274. <cpl [0]> <cpl [15]>
## 4 arima014011 0.00261    661. -1311. -1311. -1286. <cpl [0]> <cpl [16]>

##ARIMA(0,1,4)(0,1,3) has the lowest AICc, thus we go with this model.
```

Question 5

Let the `ARIMA()` function choose a model. How does this compare with your chosen model from Q4? If you need to, make `ARIMA()` search harder exploring all possible options within the function. Perform a residual diagnostics analysis for your chosen model. (No more than 100 words). (6 marks)

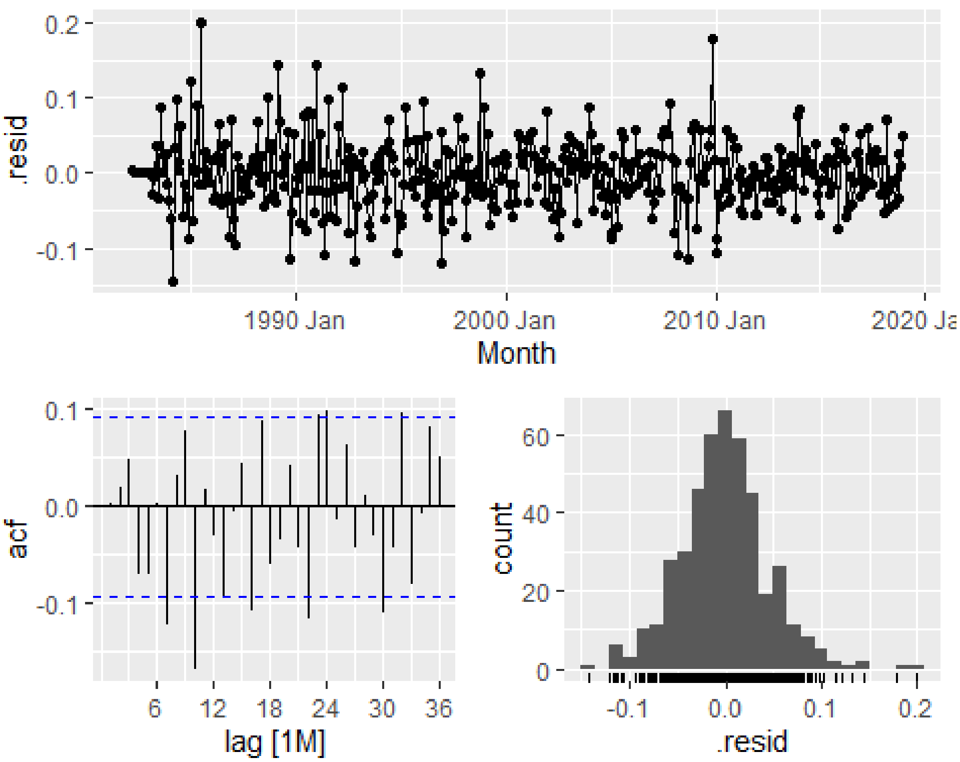
```
rmodel <- my_series %>%
  model(rmodel = ARIMA(box_cox(y,lambda)))
report(rmodel)

## Series: y
## Model: ARIMA(2,1,0)(2,1,2)[12]
## Transformation: box_cox(y, lambda)
##
## Coefficients:
##           ar1      ar2    sar1     sar2     sma1    sma2
##       -0.6926  -0.3130  0.7809  -0.5114  -1.3503  0.5361
## s.e.   0.0481   0.0496  0.0795   0.0576   0.0822  0.0726
##
## sigma^2 estimated as 0.002253:  log likelihood=691.4
## AIC=-1368.79   AICc=-1368.53   BIC=-1340.38

##r selected ARIMA(2,1,0)(2,1,2)

fit3 <- my_series %>%
  model(
    rmodel = ARIMA(box_cox(y,lambda)),
    arima014013 = ARIMA(box_cox(y,lambda)~0+pdq(0,1,4)+PDQ(0,1,3))
  )

glance(fit3)

## # A tibble: 2 x 8
##   .model       sigma2 log_lik    AIC   AICc    BIC ar_roots    ma_roots
##   <chr>         <dbl>   <dbl>  <dbl>  <dbl>  <dbl> <list>      <list>
## 1 rmodel      0.00225    691. -1369. -1369. -1340. <cpl [26]> <cpl [24]>
## 2 arima014013 0.00239    681. -1346. -1346. -1313. <cpl [0]>  <cpl [40]>

##the r selected model has lower AICc, and might be better.
##We try making R search harder:
rmodel <- my_series %>%
  model(rmodel = ARIMA(box_cox(y,lambda), stepwise = FALSE, approximation =
FALSE))

report(rmodel)

## Series: y
## Model: ARIMA(1,1,1)(2,1,2)[12]
## Transformation: box_cox(y, lambda)
```

```
##
## Coefficients:
##           ar1      ma1    sar1     sar2     sma1    sma2
##       -0.1366  -0.5949  0.8005  -0.5503  -1.3593  0.5497
## s.e.   0.0723   0.0642  0.0789   0.0515   0.0863  0.0748
##
## sigma^2 estimated as 0.002208:  log likelihood=694.91
## AIC=-1375.81   AICc=-1375.54   BIC=-1347.4
```

```
##This time r used ARIMA(1,1,1)(2,1,2)

fit4 <- my_series %>%
  model(
    rmodel = ARIMA(box_cox(y,lambda),stepwise=FALSE, approximation = FALSE),
    arima014013 = ARIMA(box_cox(y,lambda)~0+pdq(0,1,4)+PDQ(0,1,3))
  )
glance(fit4)
```

```
## # A tibble: 2 x 8
##   .model       sigma2 log_lik    AIC   AICc    BIC ar_roots   ma_roots
##   <chr>         <dbl>   <dbl>  <dbl>  <dbl>  <dbl> <list>     <list>
## 1 rmodel      0.00221    695. -1376. -1376. -1347. <cpl [25]> <cpl [25]>
## 2 arima014013 0.00239    681. -1346. -1346. -1313. <cpl [0]>  <cpl [40]>
```

```
##the r selected model still have lower AICc.
##Therefore the r model is still better, and we will go with this (as we assume letting R 'think harder' = better result)

rmodel %>%
  gg_tsresiduals(lag_max=36)
```




```
##residuals distribution looks normal
##but ACF suggests non-white noise.

augment(rmodel) %>%
  features(.innov,ljung_box,lag = 36, dof = 6)

## # A tibble: 1 x 3
##   .model lb_stat   lb_pvalue
##   <chr>    <dbl>       <dbl>
## 1 rmodel    83.9 0.000000534

##dof = no. of coefficients = 6
##Null = White Noise
##Since p-value = 0 (<0.05), we reject the Null and conclude non-white noise.
##This suggests that the model can be improved further.
```

Question 6

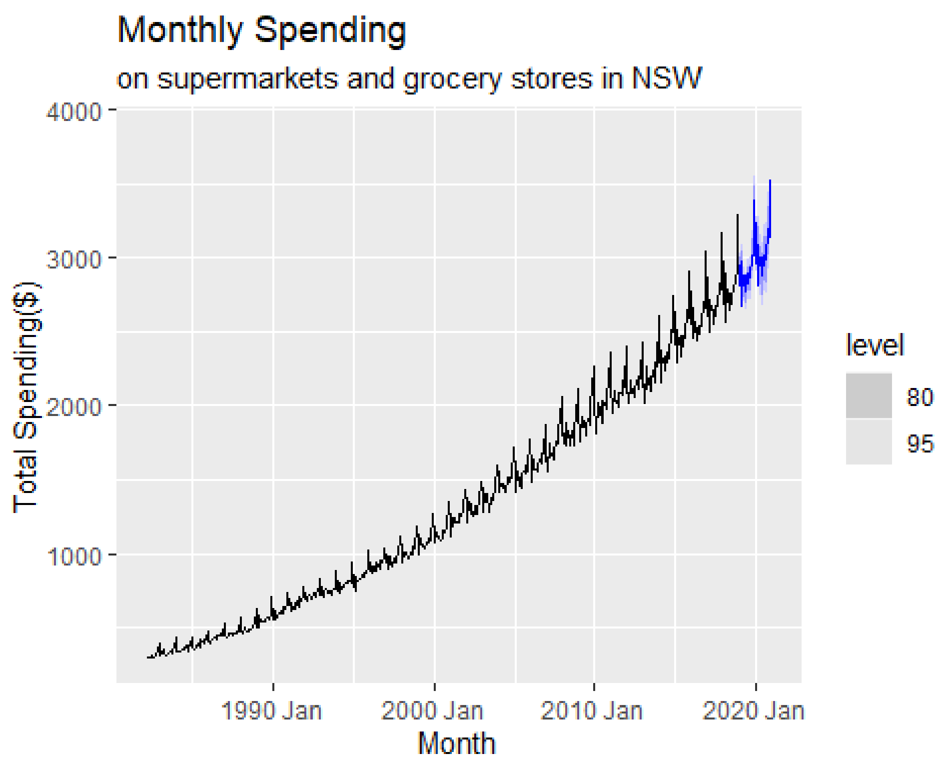
Remember that you cannot use information criteria to compare between models with different orders of differencing. **If necessary** *use an appropriate test set to choose the ARIMA model you want to use for forecasting. Which model have you selected and why? (No more than 50 words). (4 marks)*

```
test <- my_series %>%
  filter_index(~"2016 Jan") %>%
  model(
     arima014013 = ARIMA(box_cox(y,lambda)~0+pdq(0,1,4)+PDQ(0,1,3)),
    arima210011 = ARIMA(box_cox(y,lambda)~0+pdq(2,1,0)+PDQ(0,1,1)),
    arima013011 = ARIMA(box_cox(y,lambda)~0+pdq(0,1,3)+PDQ(0,1,1)),
    arima014011 = ARIMA(box_cox(y,lambda)~0+pdq(0,1,4)+PDQ(0,1,1)),
    rmodel = ARIMA(box_cox(y,lambda))
  )

test %>%
  forecast(h = "2 years") %>%
  accuracy(my_series)

## # A tibble: 5 x 10
##   .model      .type     ME  RMSE   MAE    MPE  MAPE  MASE RMSSE    ACF1
##   <chr>       <chr>  <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>   <dbl>
## 1 arima013011 Test  -49.3   59.8  53.7 -1.86   2.03 0.798 0.753 -0.0501
## 2 arima014011 Test  -42.4   54.3  48.5 -1.60   1.84 0.720 0.684 -0.0570
## 3 arima014013 Test  -42.3   53.3  47.4 -1.60   1.80 0.704 0.672 -0.0215
## 4 arima210011 Test  -48.0   58.9  52.7 -1.81   2.00 0.783 0.742 -0.0559
## 5 rmodel      Test   -4.38  33.1  27.3 -0.191  1.02 0.406 0.417 -0.0417

##The r-selected model have the lowest RMSE.
##Thus, among all the choices, this is the best model for forecasting.
```

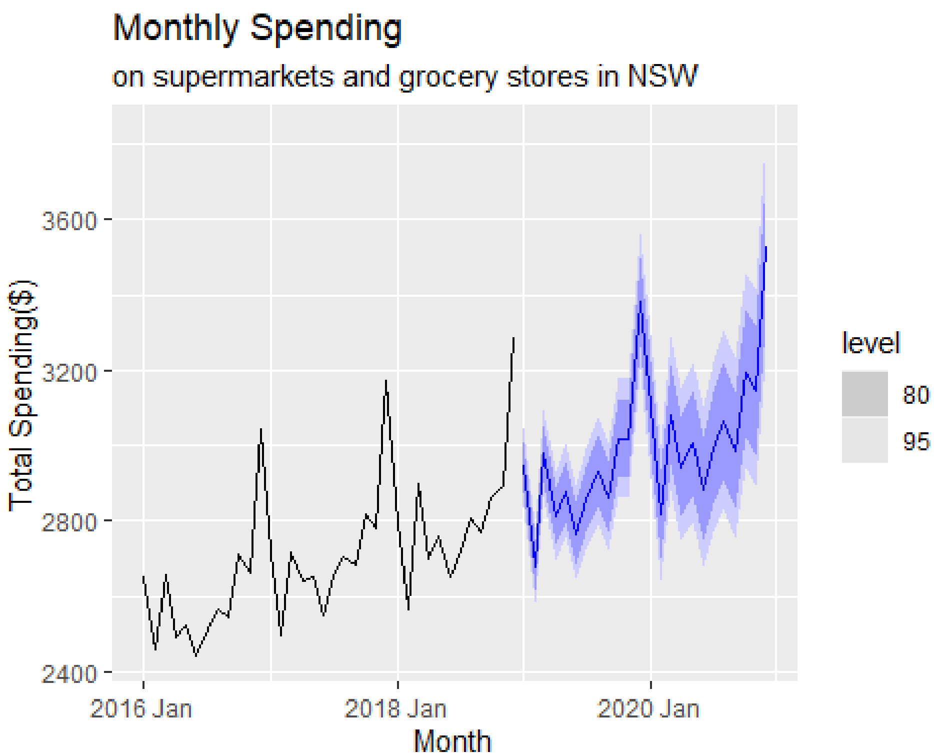
Question 7

Generate and plot forecasts and forecast intervals from your chosen ARIMA model for two years following the end of your sample. Comment on these. (No more than 50 words). (3 marks)

```
forecast(rmodel,h=24) %>%
  autoplot(my_series) + labs(title = "Monthly Spending", subtitle = "on
supermarkets and grocery stores in NSW", y= "Total Spending($)")
```




```
forecastinterval <- my_series %>%
  filter(year(Month) > 2015)

forecast(rmodel,h=24) %>%
  autoplot(forecastinterval) + labs(title = "Monthly Spending", subtitle =
"on supermarkets and grocery stores in NSW", y= "Total Spending($)")
```



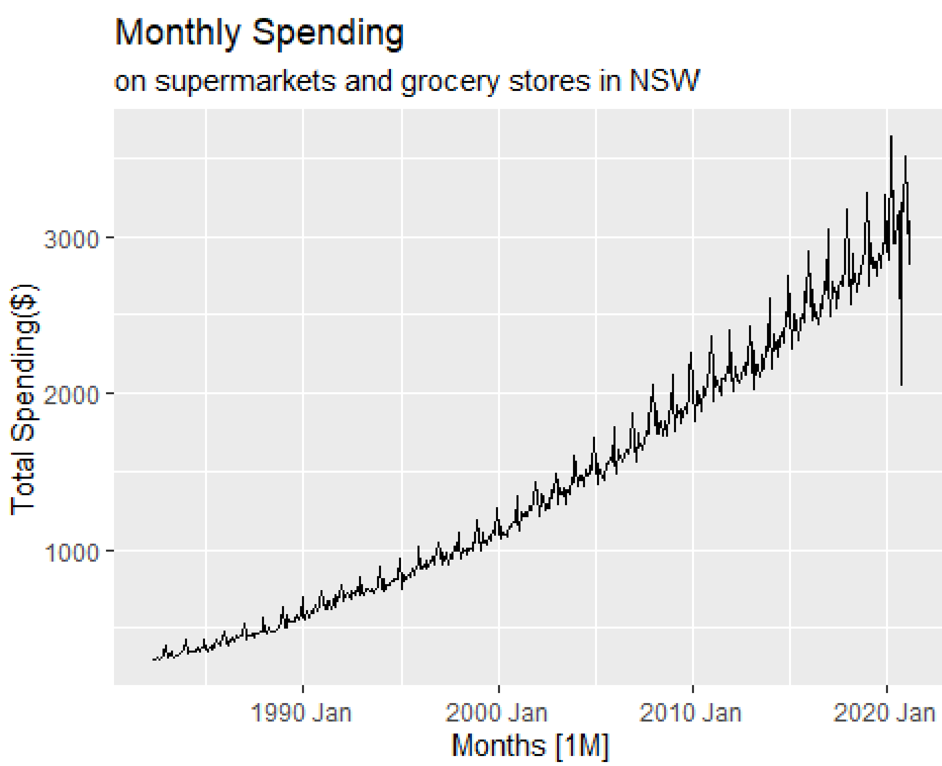

```
##The forecast suggests that seasonality and upward trend still exist for the
next 2 years.
##This is reasonable as demand for food and groceries will not die out & is
heavily influenced by festive seasons.
##Forecast intervals are wide given the uncertainties in the future
(e.g.COVID-19, inflation).
```

Question 8

You have now considered several modelling frameworks and built several models for your data set. In this part of the assignment you will evaluate these.

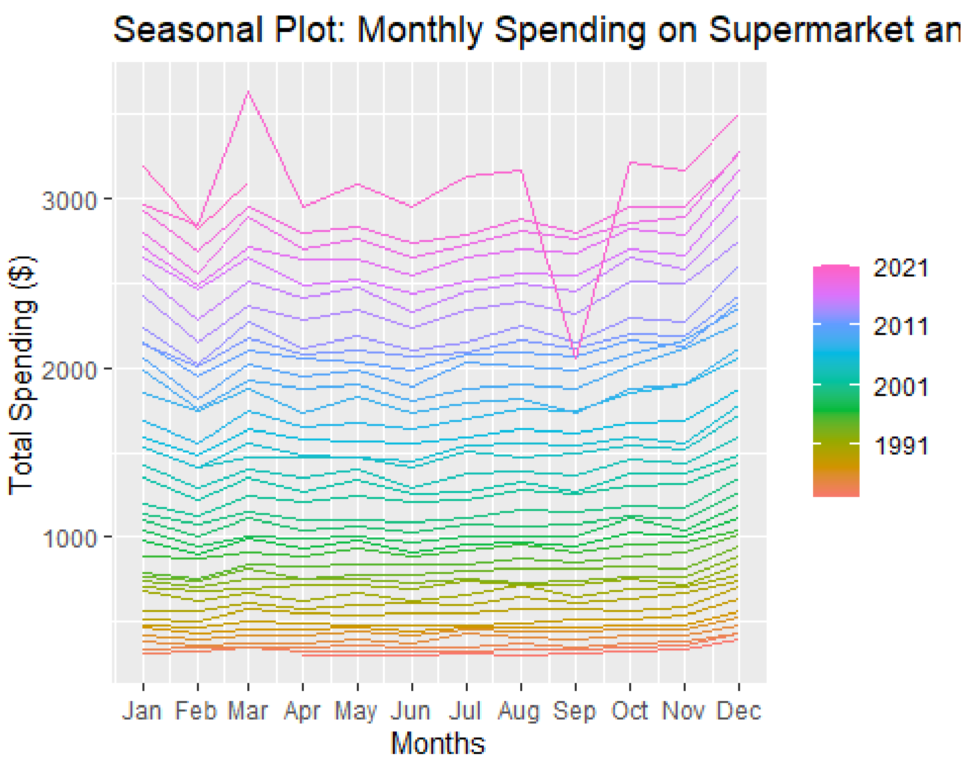
Find your data on the ABS website and update your series till the end of the currently available data. Explore your updated time series and comment on the effects of the COVID-19 pandemic. Provide any necessary plots to support your analysis. Some States and Industries are unfortunately affected more than others. (6 marks)

```
my_series1<-load("my_series1.RData")

my_series1 <- Undergrad_Data1 %>%
  select(Months,y) %>%
  mutate(Months = yearmonth(Months))%>%
  as_tsibble(index = Months)

autoplot(my_series1) + labs(title = "Monthly Spending", subtitle = "on
supermarkets and grocery stores in NSW", y= "Total Spending($)")

## Plot variable not specified, automatically selected `.vars = y`
```

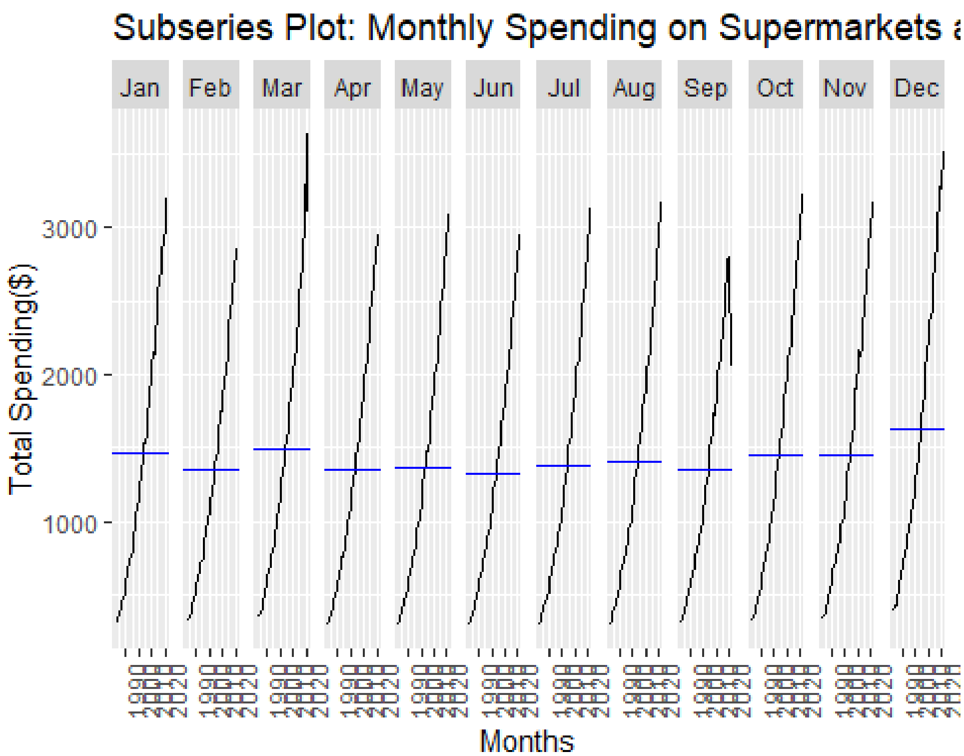



```
##Upward trend and seasonality still exists post 2018. This is reasonable as
again, demand for groceries will never die out.
##Simultaneously, there is a a huge jump in Total spending on March 2020,
which is coincidentally the month where COVID was declared as a Pandemic.
This rapid increase is likely the result of the first Australian Lockdown,
```

```r
where lots of people decided to "Panic Buy".
##Next, there was a rapid drop on September 2020. A sensible explanation for
this is the Economic Recession, where unemployment rate increases, businesses
collapsed and economic activities were hindered.
##Fortunately it didnt took long for the Australian economy to recover, as
evidenced by the rise in total spending from October 2020 onwards.

gg_season(my_series1) + labs(title = "Seasonal Plot: Monthly Spending on
Supermarket and Grocery Stores in NSW", y = "Total Spending ($)")

## Plot variable not specified, automatically selected `y = y`
```




```r
gg_subseries(my_series1) + labs(title = "Subseries Plot: Monthly Spending on
Supermarkets and Grocery Stores in NSW", y= "Total Spending($)")

## Plot variable not specified, automatically selected `y = y`
```

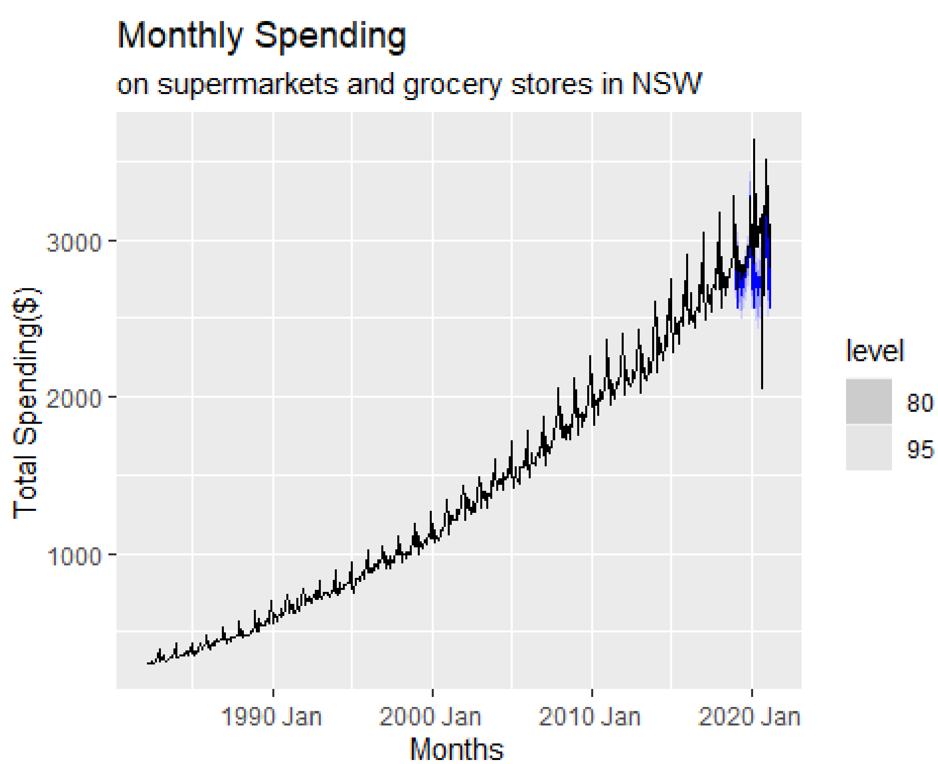



```
##Apart from the unexpected jump on March 2020, Monthly spending is still the
highest on December (due to Christmas and New Year Celebrations), followed by
January, and March, which might be affected by the "Panic Buy" in 2020.
##Also notice the drop in the September subseries (again due to recession)
```

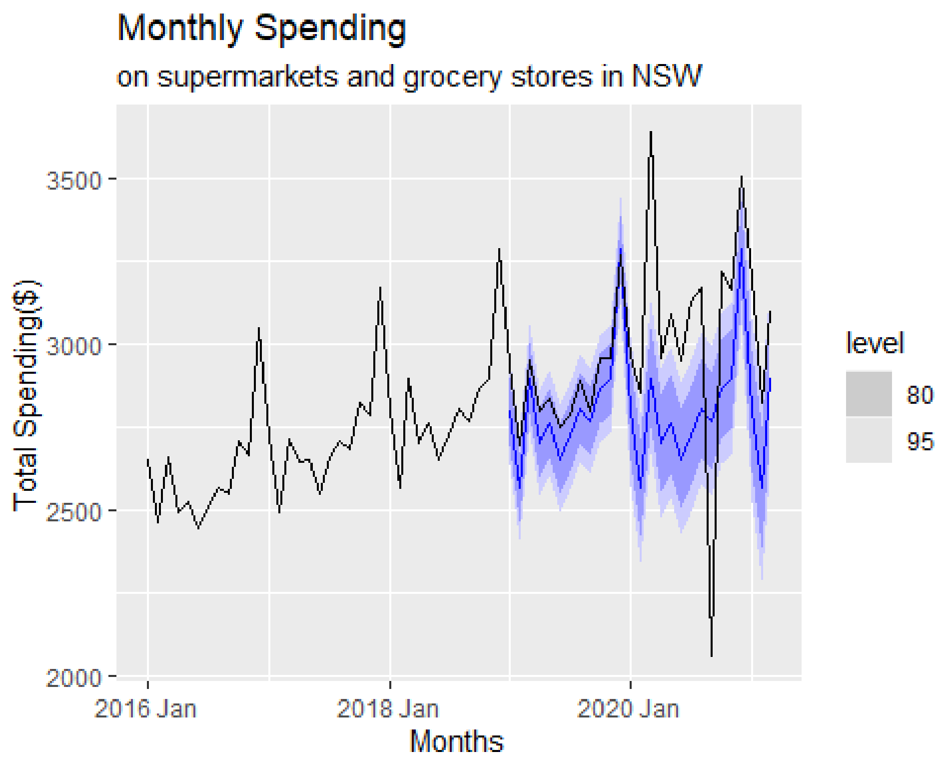
Question 9

Generate forecasts for the period post 2019 until the end of your sample, from the models considered "best" in all assignments. More specifically, generate forecasts from the best benchmark, the best ETS and best ARIMA model. Plot the forecasts (both point forecasts and prediction intervals) together with the observed data and comment on these. (Make sure you can clearly visualise these. You may choose to plot on multiple graphs.) (6 marks)

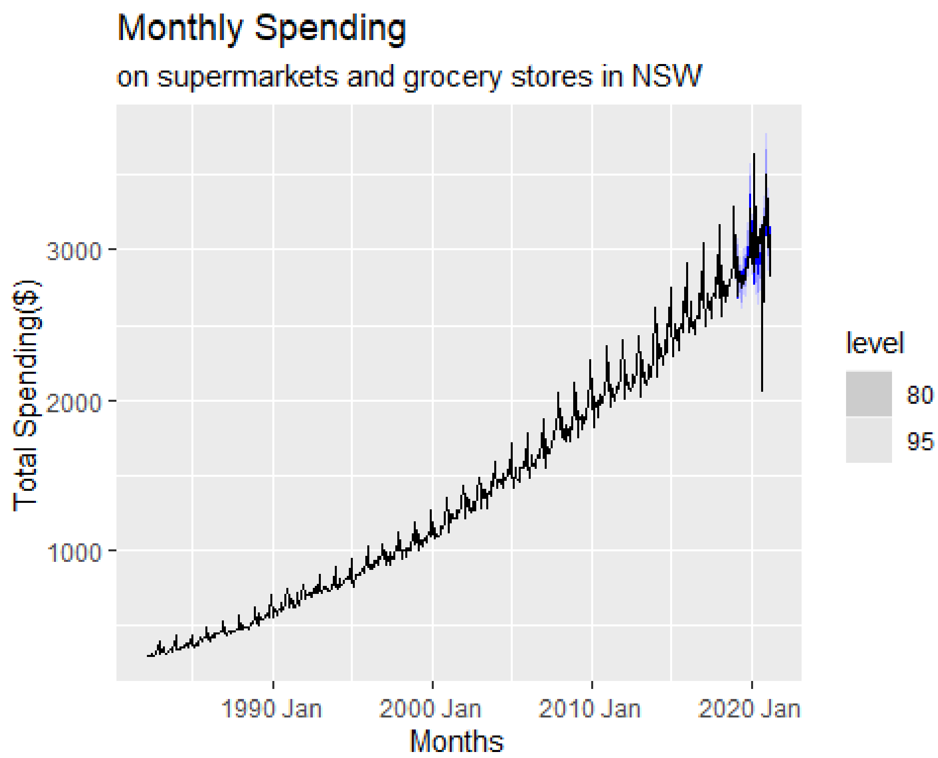
```
fit <- my_series1 %>%
  filter(year(Months) < 2019) %>%
  model(SNAIVE(y))

fc <- fit %>%
  forecast(h=27)

fc %>%
  autoplot(my_series1) + labs(title = "Monthly Spending", subtitle = "on
supermarkets and grocery stores in NSW", y= "Total Spending($)")
```



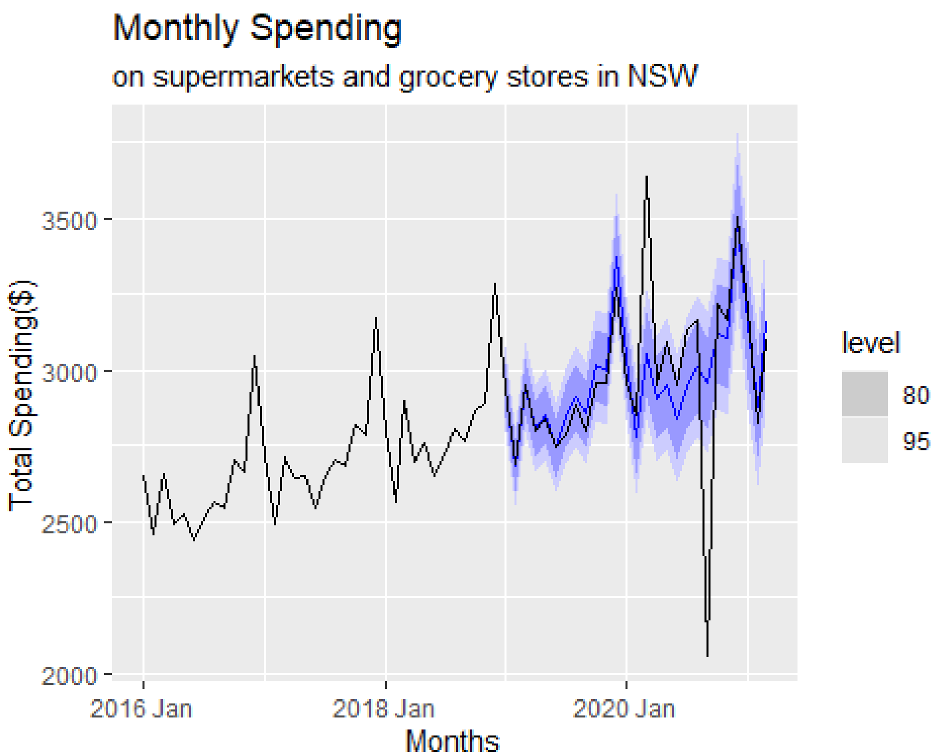

```
fc %>%
  autoplot(my_series1%>%filter(year(Months)>2015)) + labs(title = "Monthly
Spending", subtitle = "on supermarkets and grocery stores in NSW", y= "Total
Spending($)")
```



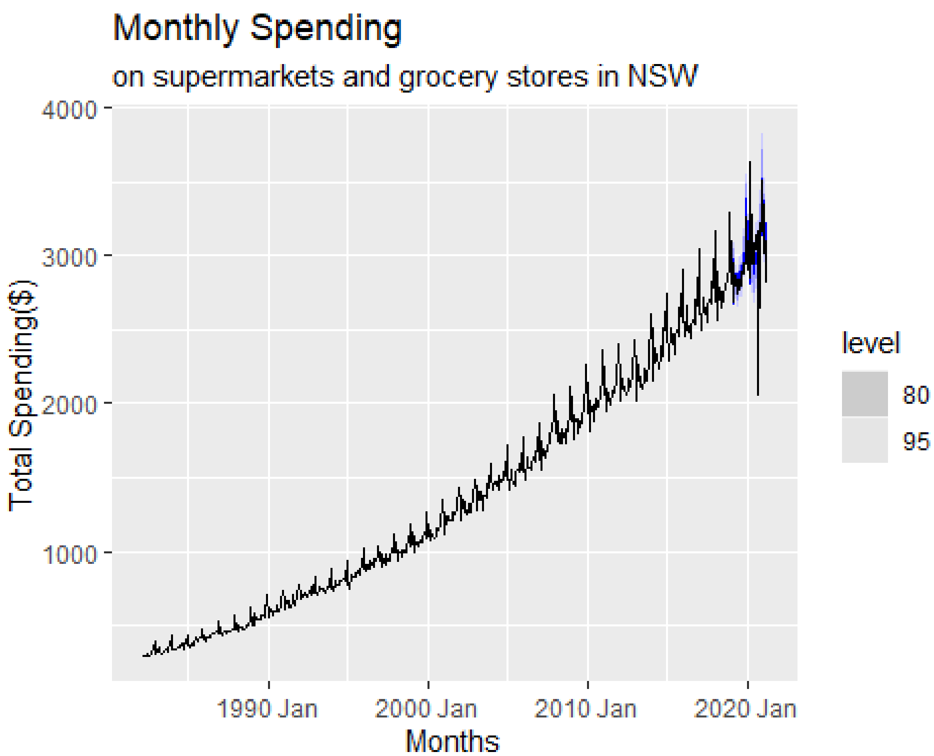

```r
##The snaive model forecasted that 27 months onwards the increasing trend
will die out, turning flatter.
##Seasonality is also weaker here.
##The snaive method fitted the data relatively well, and managed to predict
the rise and fall on March and September 2020 (although not as extreme).
##Forecast Intervals are relatively wide, considering the uncertainties and
population growth in the future.

fit2 <- my_series1 %>%
  filter(year(Months) < 2019) %>%
  model(
    MAM = ETS(y ~ error("M") + trend("A") + season("M")))

fc2 <- fit2 %>%
  forecast(h=27)

fc2 %>%
  autoplot(my_series1) + labs(title = "Monthly Spending", subtitle = "on
supermarkets and grocery stores in NSW", y= "Total Spending($)")
```



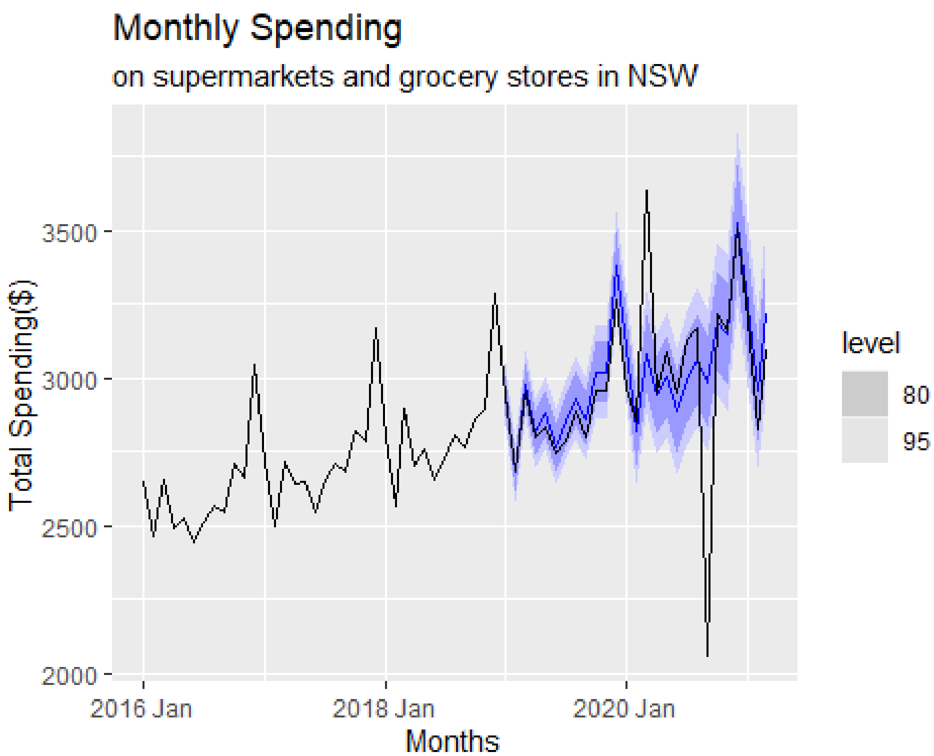

```
fc2 %>%
  autoplot(my_series1%>%filter(year(Months)>2015)) + labs(title = "Monthly
Spending", subtitle = "on supermarkets and grocery stores in NSW", y= "Total
Spending($)")
```




```
##Meanwhile, the ETS model forecasted a continuously increasing trend in
Monthly Spending.
##The model also fitted the actual data well, and in fact, better than the
previous snaive method.
##Forecast intervals here are also narrower.

fit3 <- my_series1 %>%
  filter(year(Months)<2019)%>%
   model(
    ARIMA = ARIMA(box_cox(y,lambda),stepwise=FALSE, approximation = FALSE))

fc3 <- fit3 %>%
  forecast(h=27)

fc3 %>%
  autoplot(my_series1) + labs(title = "Monthly Spending", subtitle = "on
supermarkets and grocery stores in NSW", y= "Total Spending($)")
```




```
fc3 %>%
  autoplot(my_series1%>%filter(year(Months)>2015)) + labs(title = "Monthly
Spending", subtitle = "on supermarkets and grocery stores in NSW", y= "Total
Spending($)")
```




```
##The ARIMA model also forecasted a continuously increasing trend.
##Plot also fitted the actual data well, although the previous ETS model
arguably fitted better.
##However, Forecast intervals here are narrower than the ETS'.
```

Question 10

Evaluate the accuracy of the point forecasts over the period post 2019. A table with accuracy measures will be necessary to be presented here. Comment on which forecasts are the most accurate. (4 marks)

```
forecast <- my_series1 %>%
  filter(year(Months) < 2019) %>%
  model(
    SNAIVE = SNAIVE(y),
    MAM = ETS(y ~ error("M") + trend("A") + season("M")),
    ARIMA = ARIMA(box_cox(y,lambda),stepwise=FALSE, approximation = FALSE)
  )

forecast %>%
  forecast(h=27)%>%
  accuracy(my_series1)
```

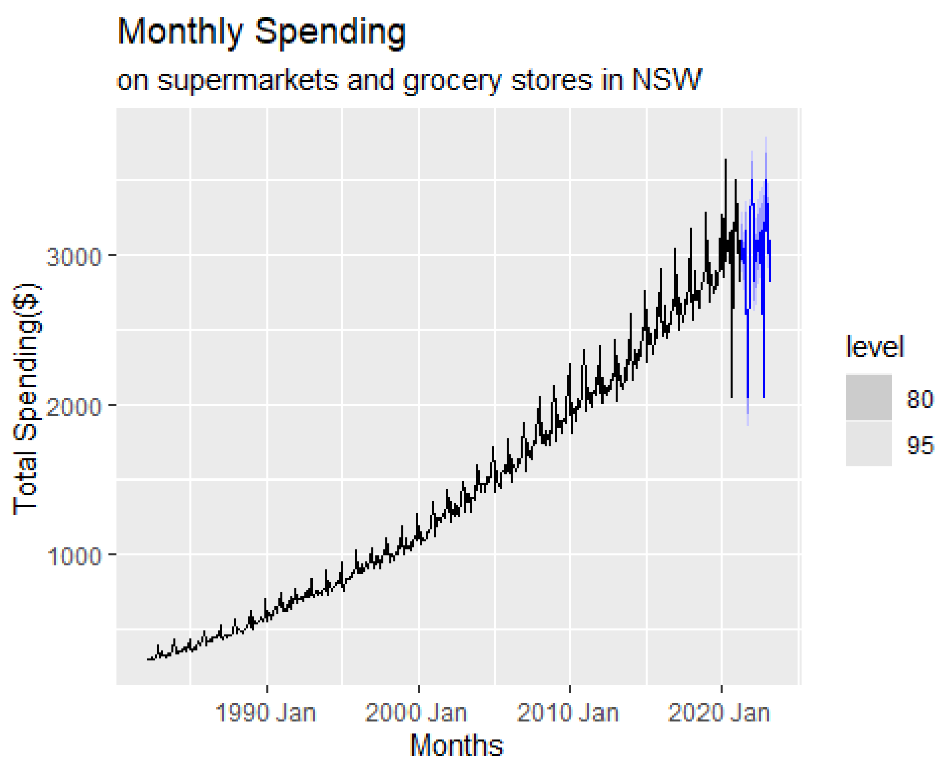
```
## # A tibble: 3 x 10
##   .model .type     ME  RMSE   MAE    MPE  MAPE  MASE RMSSE    ACF1
##   <chr>  <chr>  <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>   <dbl>
## 1 ARIMA  Test  -32.0   219.  110. -1.74   4.06  1.57  2.68 -0.0267
## 2 MAM    Test    2.13  219.  111. -0.605  4.05  1.59  2.68 -0.0498
## 3 SNAIVE Test  176.    293.  229.  5.23   7.82  3.28  3.58  0.0335
```

```
##The ETS(MAM) model have the lowest RMSE, while ARIMA have lower MAE.
##However, since we consider RMSE a better measure of accuracy (can be used
to compare any models), we assume that the ETS model is better.
##Thus the ETS model is the most accurate.
```

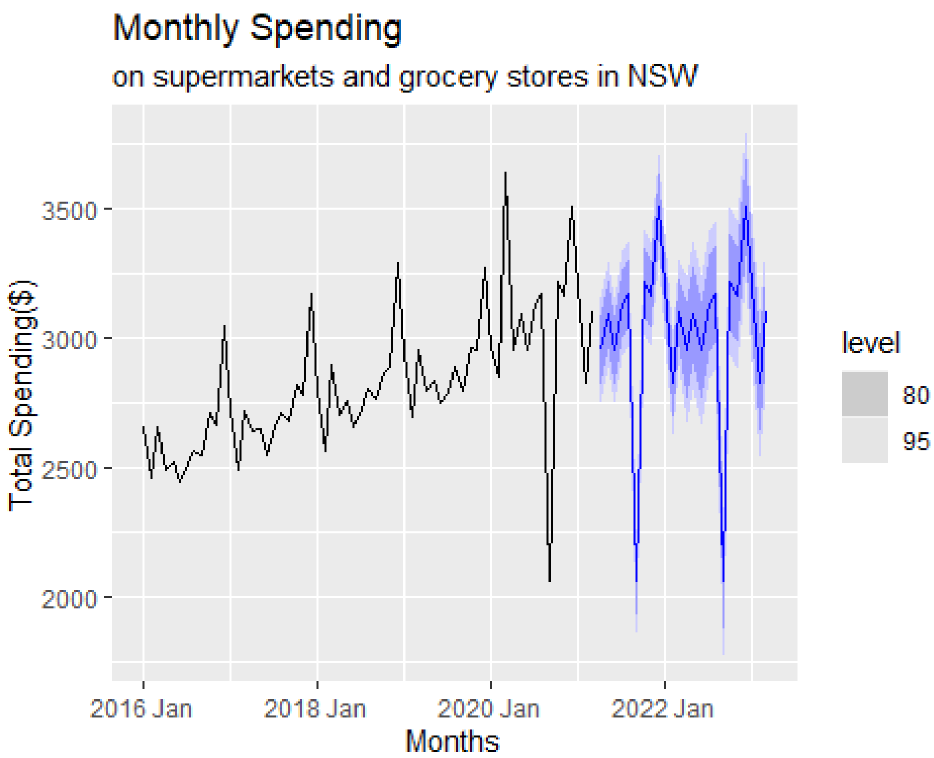
Question 11

Use all three models to forecast the next 24-months of your updated series. Generate the necessary plots and comment on the forecasts. Make sure you can clearly visualise these. How have your models faired amidst the effects of COVID-19? (9 marks)

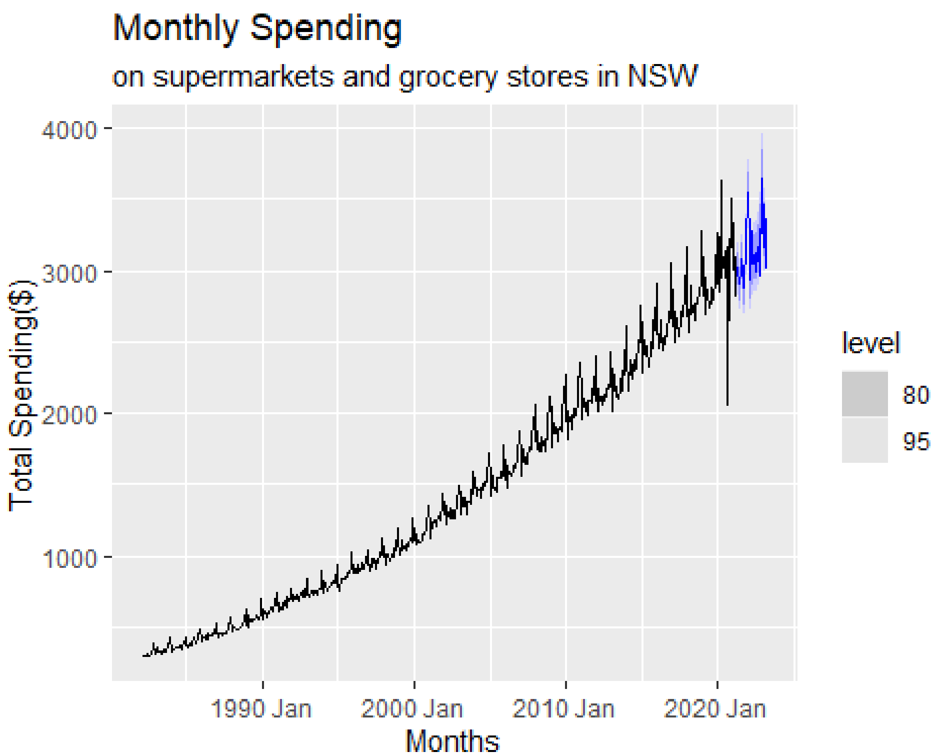
```
fit <- my_series1 %>%
  model(SNAIVE(y))

fc <- fit %>%
  forecast(h=24)

fc %>%
  autoplot(my_series1) + labs(title = "Monthly Spending", subtitle = "on
supermarkets and grocery stores in NSW", y= "Total Spending($)")
```



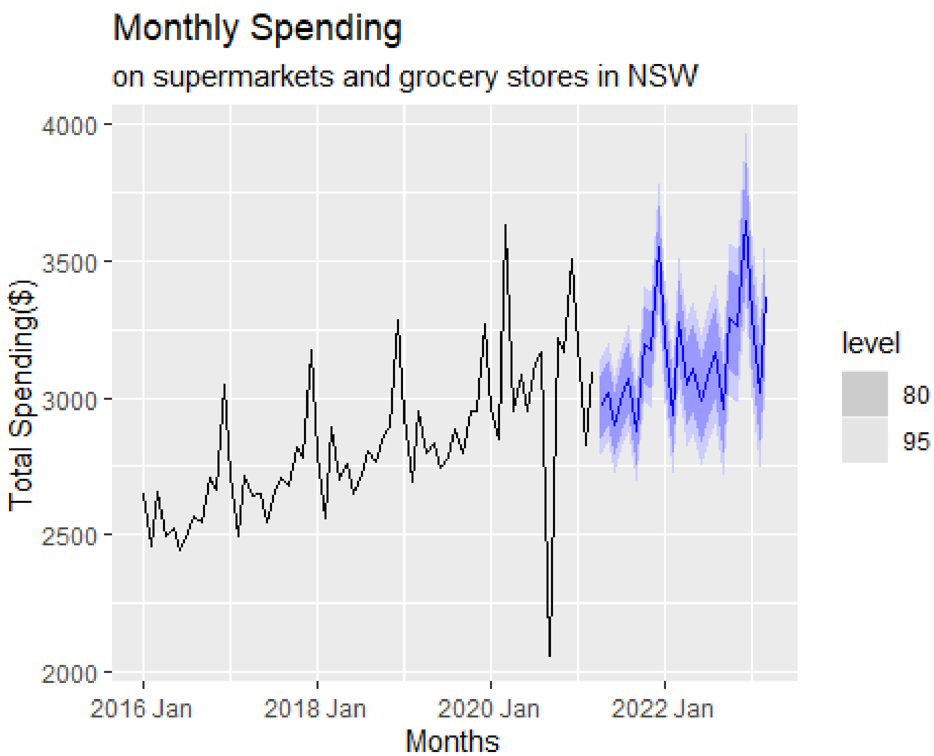

```
fc %>%
  autoplot(my_series1%>%filter(year(Months)>2015)) + labs(title = "Monthly
Spending", subtitle = "on supermarkets and grocery stores in NSW", y= "Total
Spending($)")
```



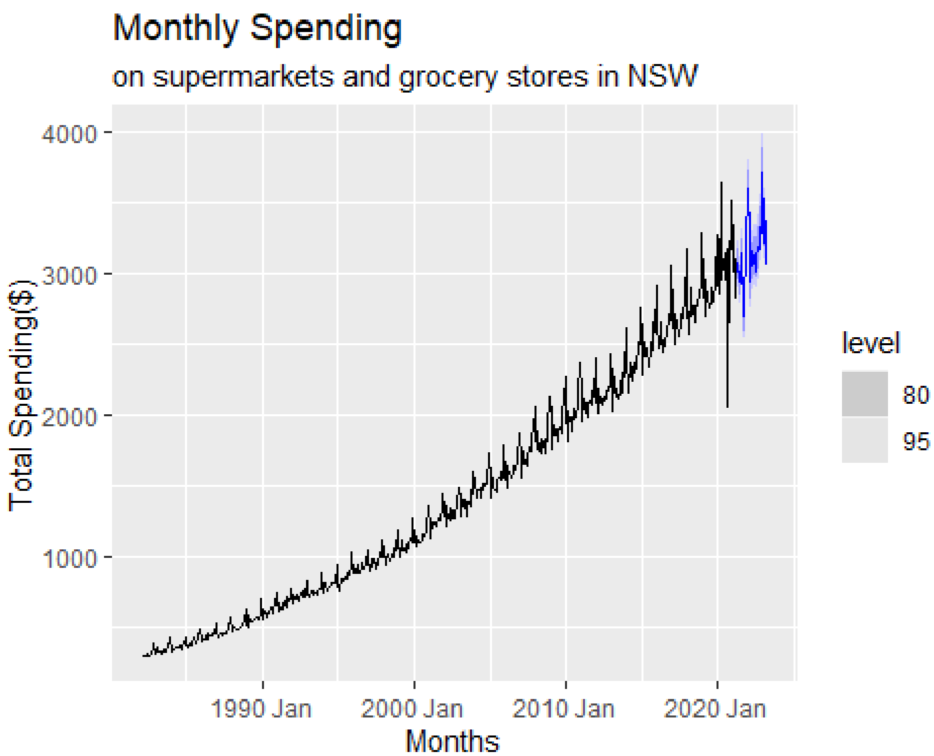

```
##The pandemic heavily affected the snaive forecast, as now it follows the seasonality and trend of 2020, and thus have lower magnitudes.
##This is no surprise as this method follows the most reason seasonal pattern (in this case 2020).
##Forecast Intervals are wide given the uncertainties and population growth in the future.

fit2 <- my_series1 %>%
  model(MAM = ETS(y ~ error("M") + trend("A") + season("M")))

fc2 <- fit2 %>%
  forecast(h=24)

fc2 %>%
  autoplot(my_series1) + labs(title = "Monthly Spending", subtitle = "on supermarkets and grocery stores in NSW", y= "Total Spending($)")
```



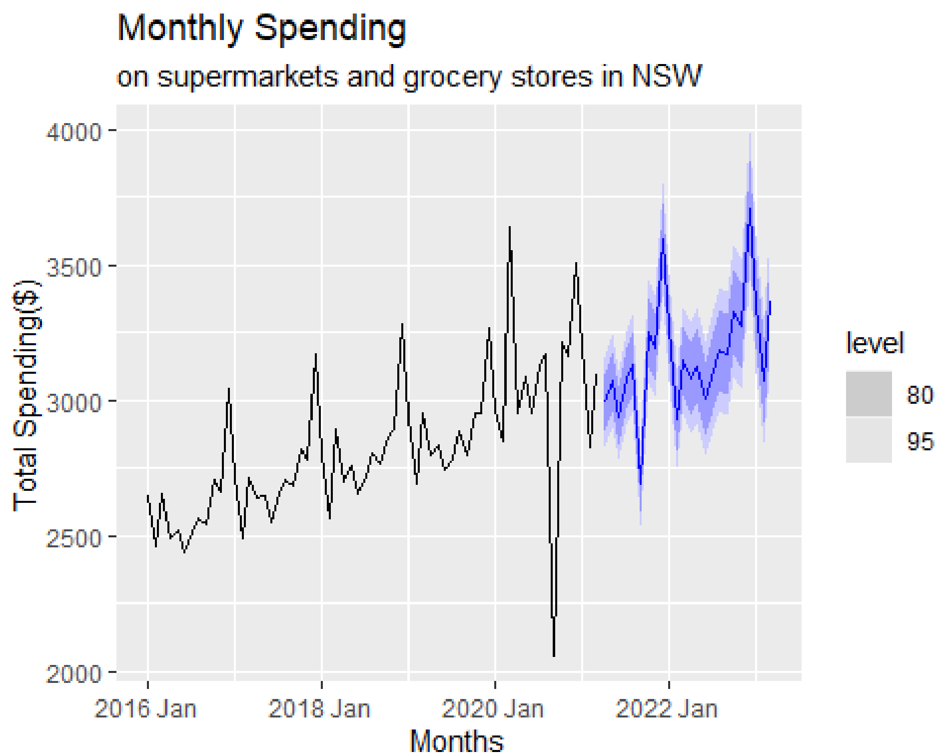

```
fc2 %>%
  autoplot(my_series1%>%filter(year(Months)>2015)) + labs(title = "Monthly
Spending", subtitle = "on supermarkets and grocery stores in NSW", y= "Total
Spending($)")
```




```
##ETS(MAM) forecasts that total spending will recover in the future, as
proven by the continuously increasing trend and existence of seasonality. In
fact, the forecast behaved similarly to the data pre-COVID, suggesting that
the ETS model is somewhat unaffected by the pandemic.
##This may be the result of the ETS model taking into consideration the
overall increase in seasonal magnitude and increasing trend.
##However, Forecast Intervals here are wider here.

fit3 <- my_series1 %>%
  model(ARIMA = ARIMA(box_cox(y,lambda),stepwise=FALSE, approximation =
FALSE))

fc3 <- fit3 %>%
  forecast(h=24)

fc3 %>%
  autoplot(my_series1) + labs(title = "Monthly Spending", subtitle = "on
supermarkets and grocery stores in NSW", y= "Total Spending($)")
```




```
fc3 %>%
  autoplot(my_series1%>%filter(year(Months)>2015)) + labs(title = "Monthly
Spending", subtitle = "on supermarkets and grocery stores in NSW", y= "Total
Spending($)")
```




```
##Similarly, ARIMA forecasts that total spending will recover, and an increasing trend with seasonality will remain existent in the future.
##However, notice that significant drops on September 2021, implicitly implying that the ARIMA model is affected by the pandemic to a certain extent.
##Interestingly, Forecast Intervals here are the narrowest.
```

You have now completed all tasks for this unit if you are enrolled in ETF3231 (one more to go for ETF5231). I hope you have learnt a lot from your hard work throughout the semester completing the assignments.

Many congratulations.
Cheers,
George