ECON2209 Final Project

Setup

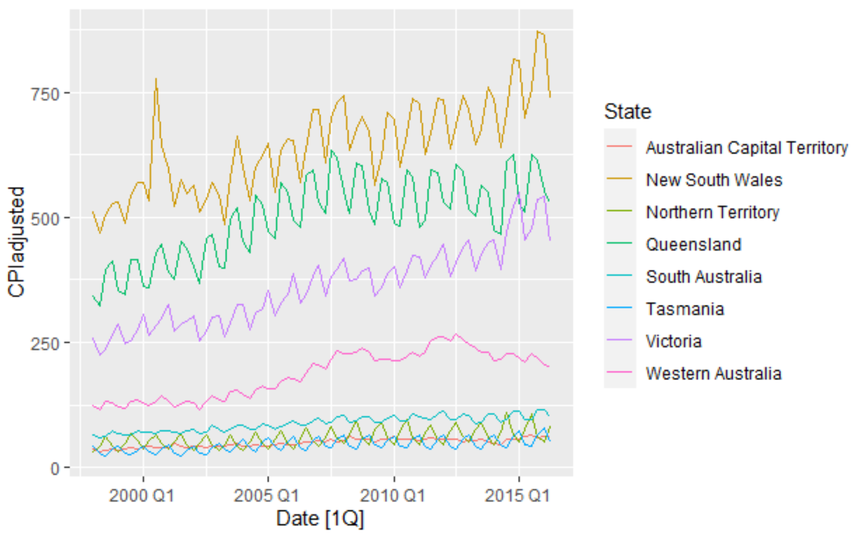
```
library(fpp3)
library(readabs)
library(pander)
```

Setup for the project where required libraries are called.

Question 1

a)

```
aus_accommodation$CPIadjusted <- (aus_accommodation$Takings / aus_accommodation$CPI) * 100
autoplot(aus_accommodation, .vars = CPIadjusted)
```



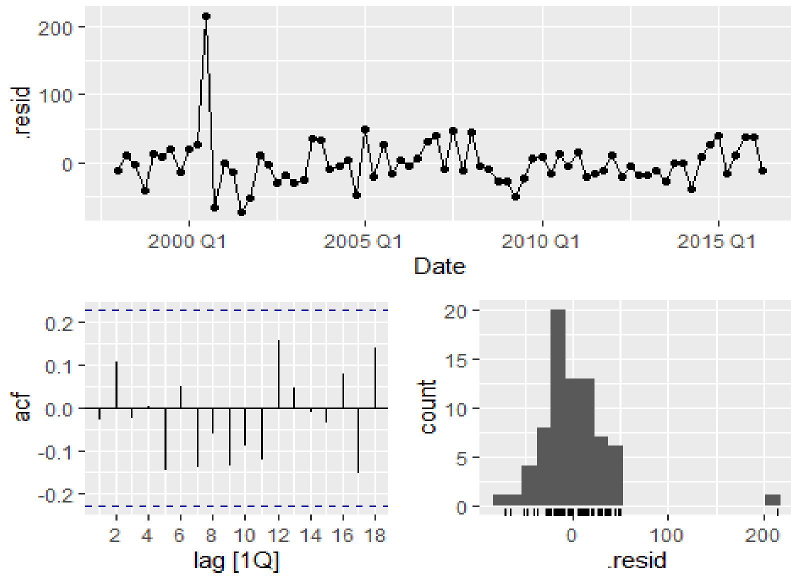
The graph shows strong seasonality in the data with variations within each quarter in most states. The data in general also has a positive trend with time with this being more obvious for ACT, NSW and Vic due to their relative sizes. There is also a slight cyclical trend which can be observed the WA value which had a small drop around 2008 as expected due to GFC conditions.

b) i)

```
chosen <- aus_accommodation %>% filter(State == "New South Wales")

fit <- chosen %>% model( a = ARIMA(CPIadjusted ~ fourier(K = 1) + trend(yearquarter(c("2008 Q1")))), MAM = ETS(CPIadjusted ~ error("M") + trend("A") + season("M")))
```

ii and iii)

```
gg_tsresiduals(fit)
```



As seen above the seasonality as represented through the ACF graph is below the blue significant values. The histogram also has values that are somewhat normally distributed except for the slight rightward weighting and outlier. Therefore by using fourier with K = 1 as a dummy variable substitute, the lags were reduced and the seasonality was somewhat adequately handled.

augment(fit) %>% features(.resid, ljung_box, dof = 1, lag = (4/2))

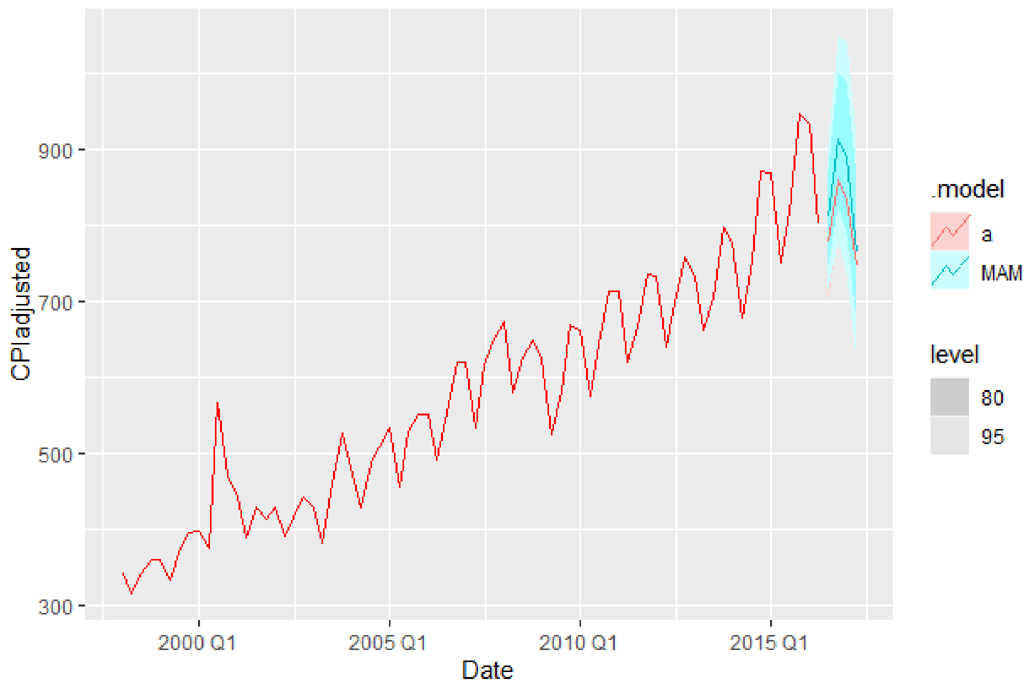
```
  State           .model                                                               lb_stat lb_pvalue
  <chr>           <chr>                                                                  <dbl>     <dbl>
1 New South Wales "ARIMA(CPIadjusted ~ fourier(K = 1) + trend(yearquarter(c(\"2008 Q1\"))))"   0.968     0.325
```

As the pvalue is below the critical value, we accept the H0 hypothesis, and the values are not distinguishable from a white noise series.

<u>c)</u>

fc <- fit %>% forecast(h = "1 year")

fc %>% autoplot() + autolayer(filter_index(chosen), colour = "red")



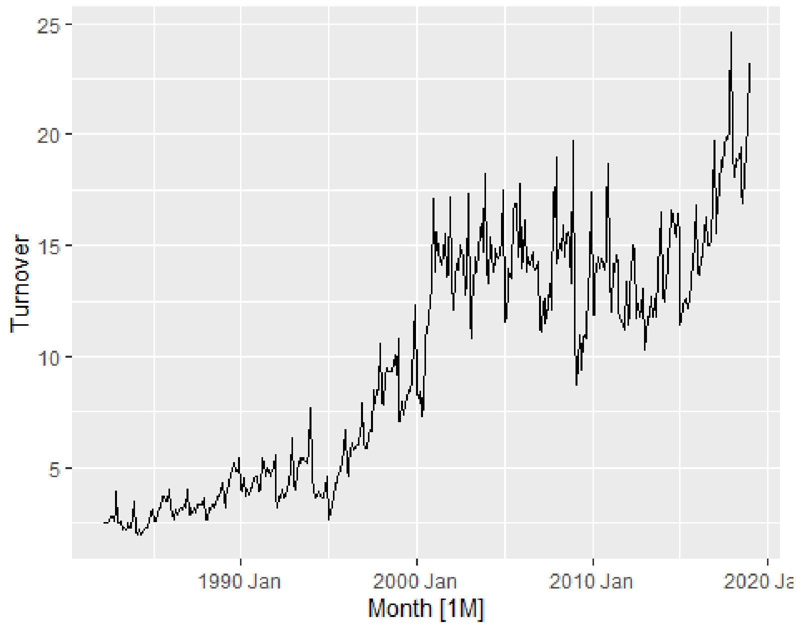
As seen above the adjusted takings for NSW to the end of 2017 are quite like the ARIMA and the MAM models except MAM had higher values at peak compared to ARIMA. The 80% confidence intervals ranged from 800 to 1000 suggesting big variability in potential movements.

d)

The wide range of the confidence interval suggests that variations in cyclical movements such as recession as in the case of 2000 and 2008 will have not been appropriately forecasted. This is because outlier values are hard for ARIMA to capture as in the case of cyclical factors.

<u>Question 2</u>

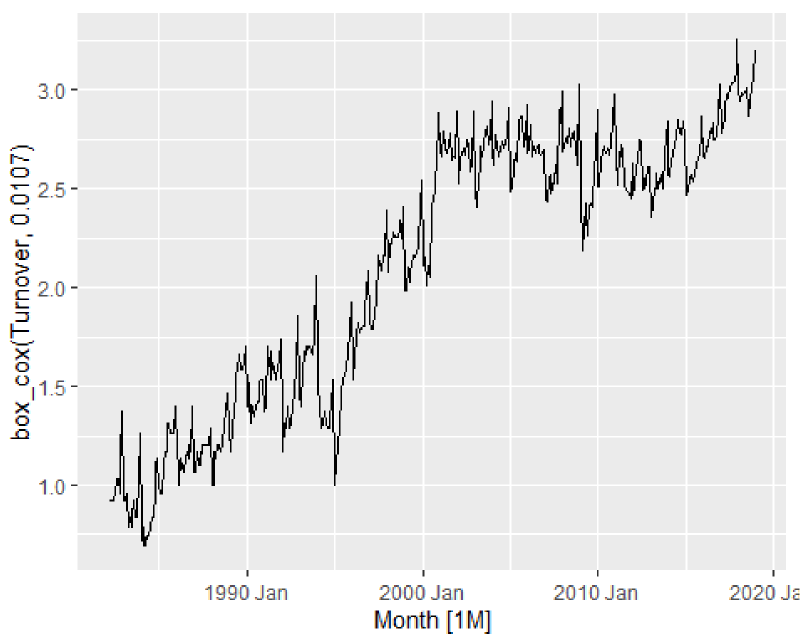
```
set.seed(5207462)
myseries <- aus_retail %>% filter(`Series ID` == sample(aus_retail$`Series ID`,1))

autoplot(myseries)
```



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

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

myseries %>% autoplot(box_cox(Turnover, 0.0107))



ZID is used to set the seed var and autoplot "myseries" with default values above. As seen the data might benefit form a Box-Cox transformation and the consequent use of auto-BoxCox suggest

0.0107 as a lambda. This value greatly normalizes the data as seen the subsequent graph and therefore log transformations will be sued in the following dynamic regression models.

```
fit <- myseries %>% model(
  `K = 1.a` = ARIMA(log(Turnover) ~ fourier(K = 1) + PDQ(0,0,0)),
  `K = 1.b` = ARIMA(log(Turnover) ~ fourier(K = 1) + pdq(0:2, 0, 0:2) + PDQ(0:1, 0, 0:1)),
  `K = 1.c` = ARIMA(log(Turnover, base = exp(0.0107)) ~ fourier(K = 1) + PDQ(0,0,0)),
  `K = 1.d` = ARIMA(log(Turnover, base = exp(0.0107)) ~ fourier(K = 1) + pdq(0:2, 0, 0:2) +
PDQ(0:1, 0, 0:1)),

  `K = 2.a` = ARIMA(log(Turnover) ~ fourier(K = 2) + PDQ(0,0,0)),
  `K = 2.b` = ARIMA(log(Turnover) ~ fourier(K = 2) + pdq(0:2, 0, 0:2) + PDQ(0:1, 0, 0:1)),
  `K = 2.c` = ARIMA(log(Turnover, base = exp(0.0107)) ~ fourier(K = 2) + PDQ(0,0,0)),
  `K = 2.d` = ARIMA(log(Turnover, base = exp(0.0107)) ~ fourier(K = 2) + pdq(0:2, 0, 0:2) +
PDQ(0:1, 0, 0:1)),

  `K = 3.a` = ARIMA(log(Turnover) ~ fourier(K = 3) + PDQ(0,0,0)),
  `K = 3.b` = ARIMA(log(Turnover) ~ fourier(K = 3) + pdq(0:2, 0, 0:2) + PDQ(0:1, 0, 0:1)),
  `K = 3.c` = ARIMA(log(Turnover, base = exp(0.0107)) ~ fourier(K = 3) + PDQ(0,0,0)),
  `K = 3.d` = ARIMA(log(Turnover, base = exp(0.0107)) ~ fourier(K = 3) + pdq(0:2, 0, 0:2) +
PDQ(0:1, 0, 0:1)),

  `K = 4.a` = ARIMA(log(Turnover) ~ fourier(K = 4) + PDQ(0,0,0)),
  `K = 4.b` = ARIMA(log(Turnover) ~ fourier(K = 4) + pdq(0:2, 0, 0:2) + PDQ(0:1, 0, 0:1)),
  `K = 4.c` = ARIMA(log(Turnover, base = exp(0.0107)) ~ fourier(K = 4) + PDQ(0,0,0)),
  `K = 4.d` = ARIMA(log(Turnover, base = exp(0.0107)) ~ fourier(K = 4) + pdq(0:2, 0, 0:2) +
PDQ(0:1, 0, 0:1)),

  `K = 5.a` = ARIMA(log(Turnover) ~ fourier(K = 5) + PDQ(0,0,0)),
  `K = 5.b` = ARIMA(log(Turnover) ~ fourier(K = 5) + pdq(0:2, 0, 0:2) + PDQ(0:1, 0, 0:1)),
  `K = 5.c` = ARIMA(log(Turnover, base = exp(0.0107)) ~ fourier(K = 5) + PDQ(0,0,0)),
  `K = 5.d` = ARIMA(log(Turnover, base = exp(0.0107)) ~ fourier(K = 5) + pdq(0:2, 0, 0:2) +
PDQ(0:1, 0, 0:1)),

  `K = 6.a` = ARIMA(log(Turnover) ~ fourier(K = 6) + PDQ(0,0,0)),
  `K = 6.b` = ARIMA(log(Turnover) ~ fourier(K = 6) + pdq(0:2, 0, 0:2) + PDQ(0:1, 0, 0:1)),
  `K = 6.c` = ARIMA(log(Turnover, base = exp(0.0107)) ~ fourier(K = 6) + PDQ(0,0,0)),
  `K = 6.d` = ARIMA(log(Turnover, base = exp(0.0107)) ~ fourier(K = 6) + pdq(0:2, 0, 0:2) +
PDQ(0:1, 0, 0:1)),

  fourier1 = TSLM(log(Turnover) ~ trend() + fourier(K = 1)),
  fourier2 = TSLM(log(Turnover) ~ trend() + fourier(K = 2)),
  fourier3 = TSLM(log(Turnover) ~ trend() + fourier(K = 3)),
  fourier4 = TSLM(log(Turnover) ~ trend() + fourier(K = 4)),
  fourier5 = TSLM(log(Turnover) ~ trend() + fourier(K = 5)),
  fourier6 = TSLM(log(Turnover) ~ trend() + fourier(K = 6))
)
```

Above, numerous ARIMA models utilizing log and Fourier models with PDQ of (0,0,0), the suggested values from the assessment specifications and once again the same models with log transformations which use the lambda value of 0.0107 as seen through guerrero. K was set to 6 which is half the seasonal periods given that the data is monthly. Below that TSLM with log transformations were also modeled.

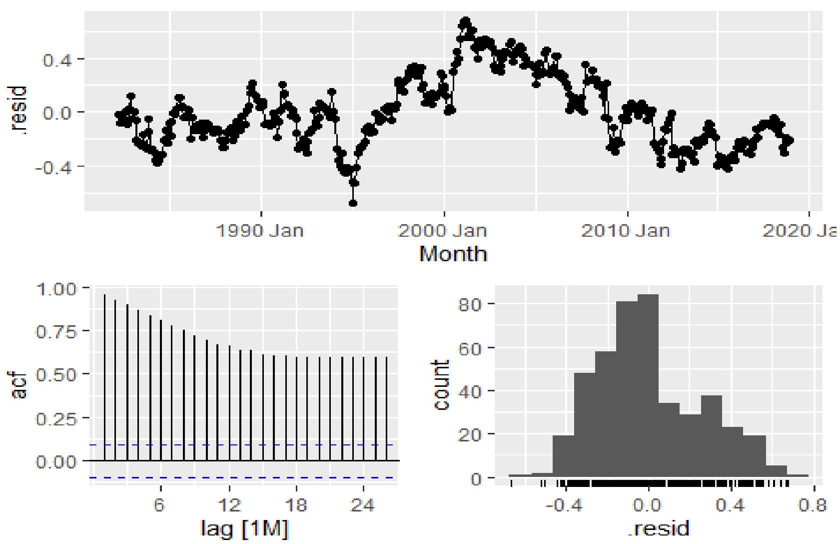
pander(glance(fit) %>% arrange(AICc) %>% select(.model, AICc))

.model	AICc
fourier5	-1177
fourier4	-1176
fourier6	-1176

...

As seen above the first few lowest AICc values were plotted and fourier5 was seen to have the lowest AICc value as per the question.

fit %>% select(fourier5) %>% gg_tsresiduals()



As seen above, the residuals are not normalized with the histogram showing leftward lea. The ACF model is also above the blue significance line for all the lags represented with a strong seasonal variation being presented. Lastly the. resid plot also has significant variation and does not vary around a mean of 0. Therefore, further adjustments and testing is necessary to find a model with lower AICc scores and more normalized data.

augment(fit) %>% filter(.model == "fourier5") %>% features(.innov, ljung_box, dof = 1, lag = 12)

```
  State                   Industry                                .model  lb_stat lb_pvalue
  <chr>                   <chr>                                   <chr>     <dbl>     <dbl>
1 Australian Capital ~ Pharmaceutical, cosmetic and toile~ fourie~   3451.         0
```

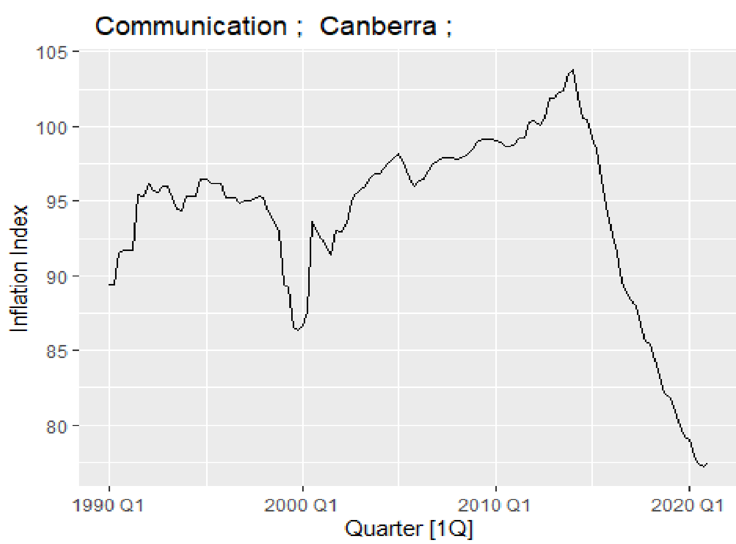
To test for white noise the usage of LJungBox with dof of 1 since the model tests for one parameter and a lag of 12 since the data is monthly. The pvalue is less than the critical value consequently the null hypothesis is not rejected and consequently the residuals are not distinguishable from white noise. However as seen above, ACF values are above the blue line which suggests an error in the creation of the model or in testing for white noise.

<u>Question 3</u>

cpidata <- read_abs("6401.0", tables=5, check_local=FALSE) %>% mutate (Quarter = yearquarter (date)) %>%
 as_tsibble (index = Quarter, key = c (series_id))

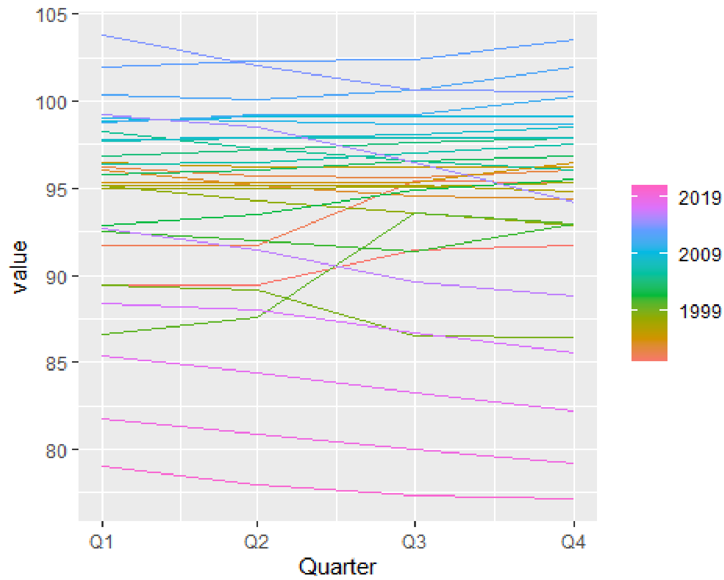
set.seed(5207462)
select_series <- cpidata %>% filter (`series_id` == sample (cpidata$`series_id` , 1), year(Quarter)>=1990)

myseries <- select_series %>% replace("series", sub("Index Numbers ;", "", select_series$series))
myseries %>% autoplot(value) + labs (y = "Inflation Index" , title = myseries$series[1])



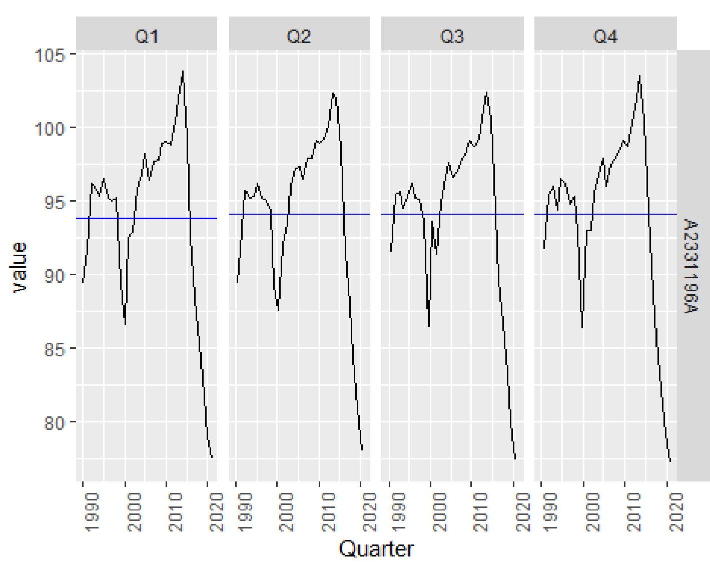
The code above is setup for this question.

gg_season(myseries, y = value)



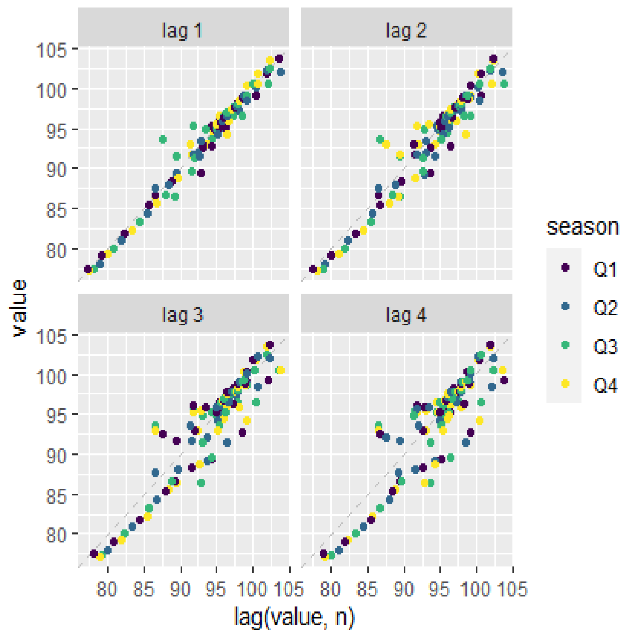
As we can see above, there is not a strong seasonality pattern across time periods. In the year pre-2010, the variations within a year stayed quite consistent. However, since 2010, the variations within a year have been negatively correlated with timed as illustrated by the plot of the graph. Even though the data does not have a strong seasonality, a negative trend is present in the data after 2010.

gg_subseries(myseries, y = value)



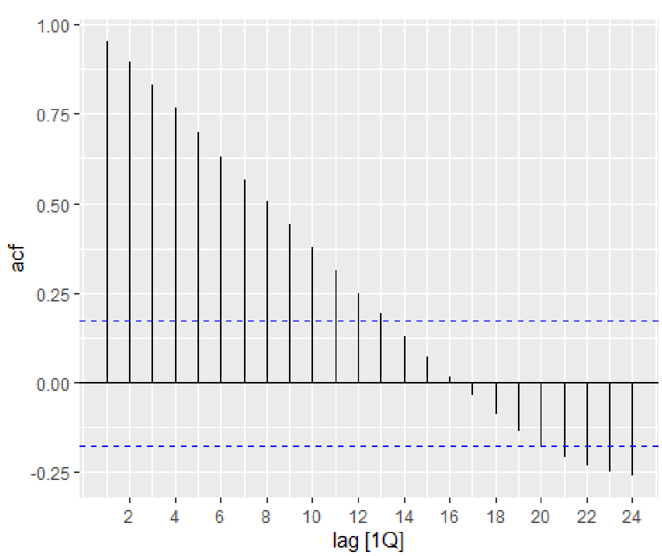
The subseries shows consistency across each quarter over the chosen time with the plots quite similar in shape with exceptions at year ~2003 in the Q4 when compared to the Q1. The median values stayed quite consistent over the period showing no major fluctuations. Therefore, there was not a strong trend across the quarters.

gg_lag(myseries, y = value, lags = 1:4, geom = 'point')



The lag plots above display the value for 4 chosen lags to display variations within each quarter along with point descriptors instead of lines. As seen above, the values for lag 1 is strongly positively correlated followed by lag 2 which is still strongly correlated although less. As time passes, the values become less positively correlated. This highlights the lack of strong seasonality with the data as seen by the autoplot.

myseries %>% ACF(value, lag_max = 24) %>% autoplot()

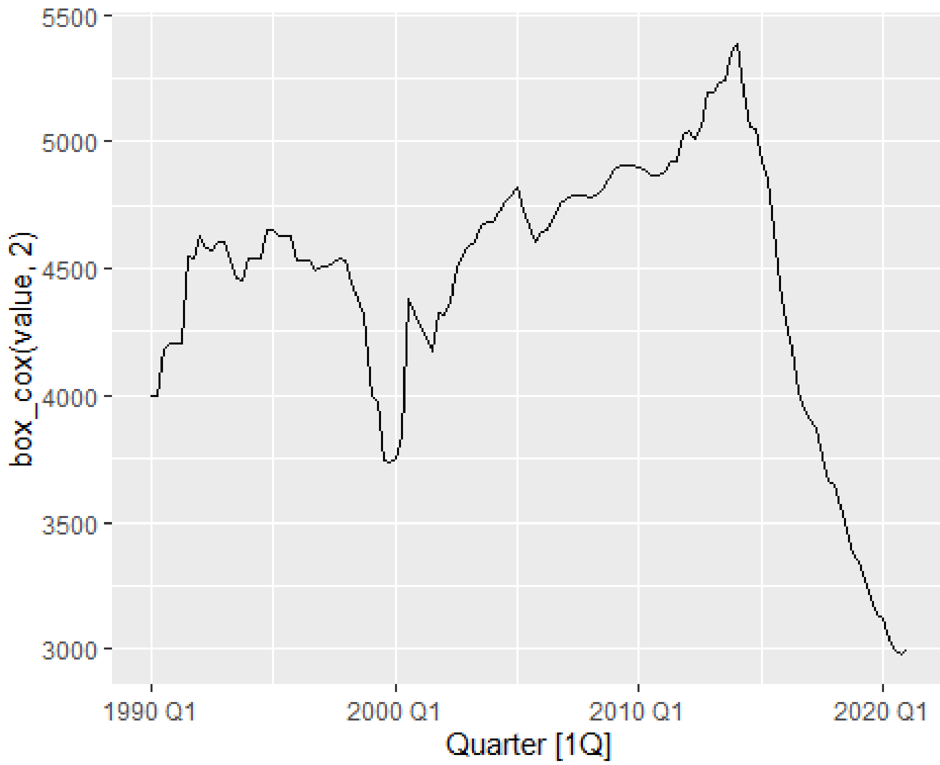


The ACF graph shows the value with a lag max of 24 for 6 years of values at quarterly. As seen, the value up until the 14th lag were significant since they were above the blue line and lags are again present from 20 to 24 This illustrates that there is not strong seasonality present in the data especially when analyzed over an extended period of time but rather a trend in the data.

myseries %>% features(value, features = guerrero)

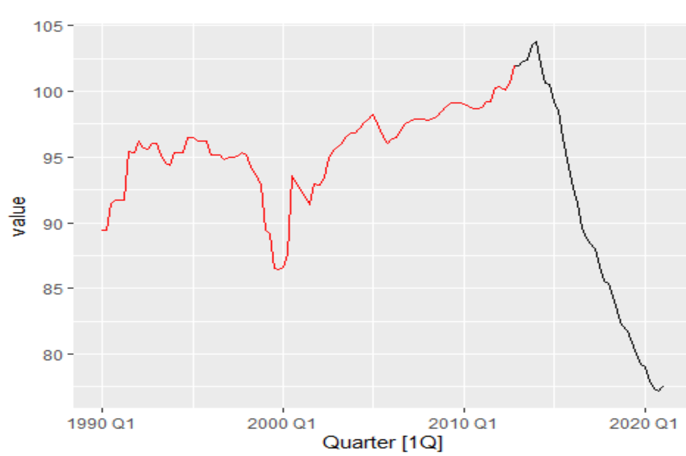
```
  series_id lambda_guerrero
  <chr>               <dbl>
1 A2331196A            2.00
```

myseries %>% autoplot(box_cox(value, 2))



As seen above, the automatic Box-Cox transformation value is 2 and once tested the resulting graph is like the original graph. Therefore, a transformation is not necessary and useful since no normalization of the data occurred.

myseries_train <- myseries %>% filter(year(Quarter) < 2013)
autoplot(myseries, value) + autolayer(myseries_train, value, colour = "red")

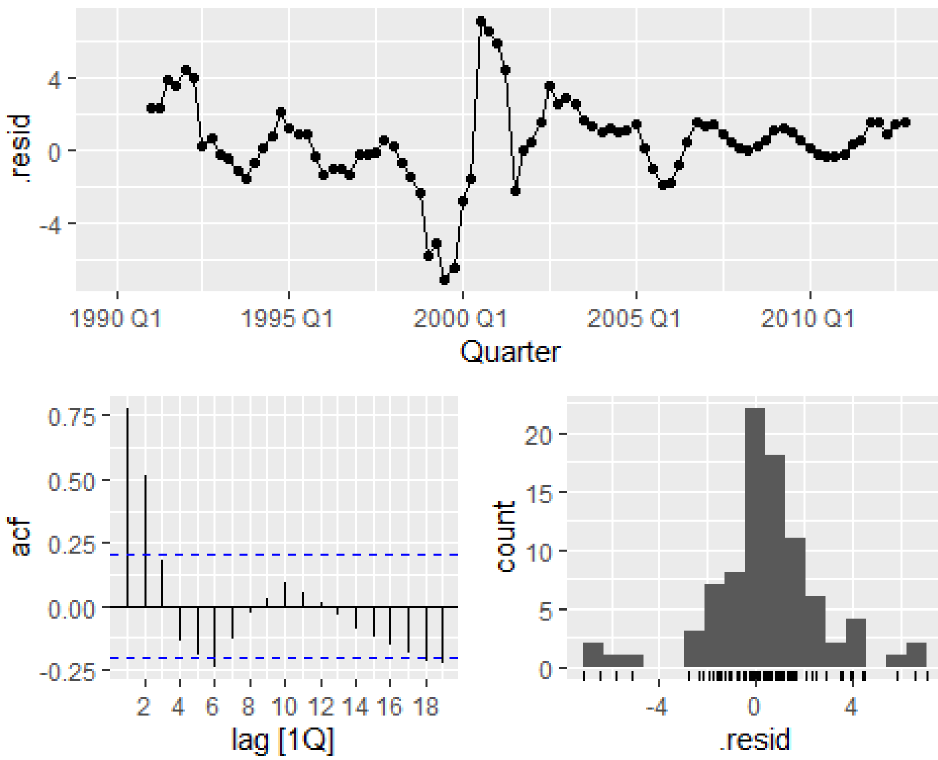


"myseries_train" was filtered to include observations before 2013 and the values are autoplotted in red below. The values in black illustrate the values after 2013.

b)

fit <- myseries_train %>% model(SNAIVE(value))

fit %>% gg_tsresiduals()



The model is then trained on seasonal naive method and the residuals are plotted above. As we can see, the distribution is somewhat normal although it has a small shift to the right. Outliers are also present on the graph as illustrated by the value above 4 and below -4. The residual graph also does not have a mean of zero with large variations such as in 2000 Q1. The ACF values are also significant before the 3rd lag and once at lag 6 and 18/19. This illustrates that the values show significant autocorrelation and do not capture the changing trends in the data.

fc <- fit %>% forecast(new_data = anti_join(myseries, myseries_train))

```
> fit %>% accuracy()
# A tibble: 1 x 11
  series_id .model         .type        ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
  <chr>     <chr>          <chr>     <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 A2331196A SNAIVE(value) Training 0.466  2.38  1.65 0.454  1.76     1     1 0.776
> fc %>% accuracy(myseries)
# A tibble: 1 x 11
  .model         series_id .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
  <chr>          <chr>     <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 SNAIVE(value) A2331196A Test  -11.0  14.3  11.7 -13.3  14.1  7.10  6.01 0.932
```

AntiJoin is used to remove differences in the data and then "myseries", which is the actual value, and "myseries_train" are then compared. As seen above the RMSE for the trained model is 2.38

whilst 14.3 for the untrained model. Therefore, the untrained model is substantially better with respect to RMSE and is the model I prefer for this case.

c)

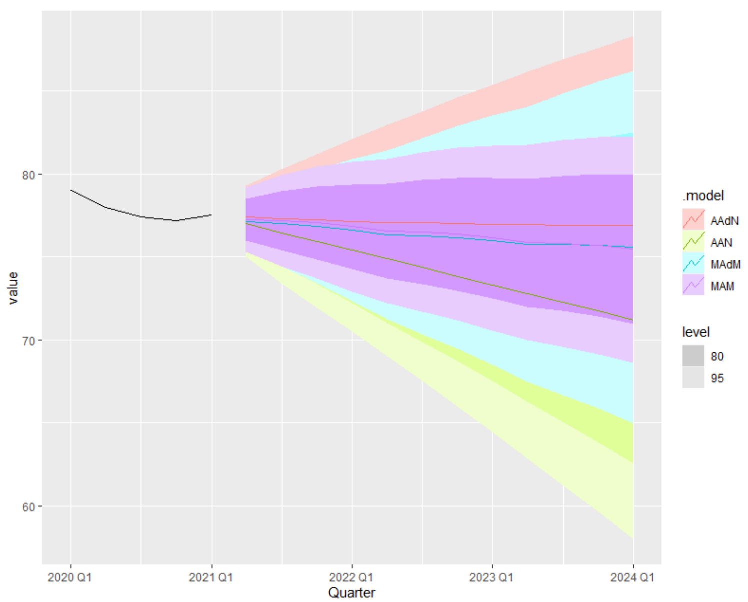
myseries %>% model(ETS(value))

```
  series_id  `ETS(value)`
  <chr>           <model>
1 A2331196A <ETS(A,Ad,N)>
```

```
fit <- myseries %>% model(
  MAM = ETS(value ~ error("M") + trend("A") + season("M")),
  MAdM = ETS(value ~ error("M") + trend("Ad") + season("M")),

  AAN = ETS(value ~ error("A") + trend("A") + season("N")),
  AAdN = ETS(value ~ error("A") + trend("Ad") + season("N"))
 )
fc <- fit %>% forecast(h = 12)
```

Firstly, ETS is run to show the automatically chosen most appropriate ETS value which is seen to be A,Ad,N in this case. This value is then also used in the model along with MAM and MAdM and a forecast of 12 periods is done.

fc %>% autoplot(filter(myseries, year(Quarter) > 2019))



As seen above, the graph (which starts from 2019 for the sake of legibility) forecasts all four models. MAdM and MAM are seen to have pretty much the same paths with MAM having slightly greater values than MAdM. However, the confidence interval ranges significantly more for MAdM than MAM.

fit %>% accuracy()

	series_id	.model	.type	ME	RMSE	MAE	MPE	MAPE	MASE	RMSSE	ACF1
	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	A2331196A	MAM	Training	-0.0290	1.14	0.679	-0.0499	0.732	0.321	0.405	0.227
2	A2331196A	MAdM	Training	-0.0577	0.964	0.600	-0.0635	0.645	0.284	0.344	0.0337
3	A2331196A	AAN	Training	-0.0653	0.993	0.606	-0.0632	0.651	0.287	0.354	0.0886
4	A2331196A	AAdN	Training	-0.0632	0.959	0.589	-0.0725	0.633	0.278	0.342	-0.00519

As seen above, the RMSE between the two multiplicative models is lower for the damped method. However, the lowest RMSE was from the AAdN model as suggested by autoETS a few code blocks above. As a result, MAdM is the better of the two methods due to a low RMSE and consistently low set of values for other tests.

<u>d)</u>

```
myseries_train <- myseries %>% filter(year(Quarter) < 2013)

stlarima <- decomposition_model(
  STL(box_cox(value, 2)),
  ARIMA(season_adjust))

fit <- myseries_train %>%
  model(
    MAM = ETS(value ~ error("M") + trend("A") + season("M")),
    MAdM = ETS(value ~ error("M") + trend("Ad") + season("M")),
    SNaive = SNAIVE(value), DRIFt = SNAIVE(value ~ drift()), stlarima = stlarima)

fit %>% accuracy()
```

	series_id	.model	.type	ME	RMSE	MAE	MPE	MAPE	MASE	RMSSE	ACF1
	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	A2331196A	MAM	Traini~	1.11e- 1	1.11	0.676	0.117	0.722	0.411	0.467	0.109
2	A2331196A	MAdM	Traini~	-8.69e- 2	1.02	0.645	-0.0999	0.686	0.391	0.429	0.115
3	A2331196A	SNaive	Traini~	4.66e- 1	2.38	1.65	0.454	1.76	1	1	0.776
4	A2331196A	DRIFt	Traini~	-2.75e-15	2.33	1.59	-0.0326	1.70	0.966	0.981	0.776
5	A2331196A	stlarima	Traini~	1.92e- 1	0.846	0.552	0.193	0.583	0.335	0.356	-0.0327

A model is chosen based on the filter of it being to the end of 2012. STL-ARIMA is also set with a BoxCox value of 2 even though transformations don't have a big impact on the overall graph. Once all four models were graphed, the lowest RMSE was from the STL-ARIMA with a value of 0.846 compared to the other above 1. STL-ARIMA also tested the lowest for all the other tests suggesting that it is the most appropriate model in this case.

<u>Question 5</u>

EC ~ Temp where temp is 18. If x > 18 -> heating elif x < 18 -> cooling else 0.

Yt: monthly total kw/h of elec used

X1,t: monthly total of heating deg

X2,t: monthly total of cooling deg

Yt* = B1x1,t* + B2x2,t* + nt

Where

(1-B)(1-B^12)nt = (1-Theta1B) / (1-ph12*B^12 – phi24*B^24) * epsilon t

And

yt* = log(yt), x1,t* = sqrt(x1,t) and x2,t* = sqrt(x2,t)

Therefore

log(yt) = B1sqrt(x1,t) + B2sqrt(x2,t) + nt

<u>a)</u>

We can see that AR(12) and AR(24) are used in the equation along with 12 differences and MA(2) as per the textbook (Non-seasonal ARIMA models, 2021). Therefore, ACF lags would need to be at 12, 24 and 1 to normalize the data since monthly data. Since the data is seasonal, we would need both pdq and PDQ values.

Therefore, in this case the values will be (0,1,0) (0,1,2) [12] since the numerator has a theta of 1 and the denominator has phi at 12 and 24 which, when divided by 12 months, is 1 and 2. The 12 is due to there being 12 monthly periods as seen by the B^12.

<u>b) and c)</u>

By substituting the value we can see that the equation would be

$$\log(y_t) = 0.0077\sqrt{x_{1,t}} + 0.0205\sqrt{x_{2,t}} + n_t$$

$$(1-B)(1-B^{12})n_t = \frac{(1-0.5830 \cdot B)}{(1-(0.5373)B^{12}-(-0.4667)B^{24})}\varepsilon_t$$

Since the pvalue is less than 0.05, it tells us that the values are significant. This also tells us that since both betas are positive as temperature increases or decreases depending on heating or cooling, so does the electrical consumption and. Beta2 is larger than beta 1 telling us that cooling costs more electricity. The slighter larger standard error for beta2 also shows the slightly larger variance.

The results below show how to get a more suitable equation for forecasting.

$$(1-B)(1-B^{12})x_t = \frac{(0.417 \cdot B)}{1 + 0.5373B^{12} + 0.4667B^{24}}\varepsilon_t$$

Expanding this we get;

$$((1-B^{12}) - B + B^{13})x_t =$$

$$(1 - B - B^{12} + B^{13})x_t = \frac{0.417 \cdot B}{1 + 0.5373B^{12} + 0.4667B^{24}}\varepsilon_t$$

Applying Backshift we get;

$$x_t - B_{x_{t-1}} - B^{12}_{x_{t-12}} + B^{13}_{x_{t-13}} = \frac{0.417\, B_{x_t}}{x_t - 0.5373B^{12}_{t-12} + 0.4667B^{24}_{t-24}}\varepsilon_t$$

$\therefore$

$$x_t = \frac{0.417\, B_{x_t}}{x_t - 0.5373B^{12}_{t-12} + 0.4667B^{24}_{t-24}} \cdot \varepsilon_t + B_{x_{t-1}} + B^{12}_{x_{t-12}} - B^{13}_{x_{t-13}}$$

$\therefore$

$$\log(y_t) = 0.0077\sqrt{x_{1,t}} + 0.0208\sqrt{x_{2,t}} + \frac{0.417\, B_{x_t}}{x_t - 0.5373B^{12}_{t-12} + 0.4667B^{24}_{t-24}} \cdot \varepsilon_t + B_{x_{t-1}} + B^{12}_{x_{t-12}} - B^{13}_{x_{t-13}}$$

d)

The will allow us to find the values into the future by working on each set of data one iteration at a time. The lagged variables show that the effect of electricity on temperature is a continuous function with each period before impacting the current seasons electric usage and consequently a good indicator of current electricity usage.

e)

OLS is used for linear regression models which analyze patterns which are linear in nature. However ARIMA is useful because it accounts for seasonality and small variations in the data (except for cyclical situations) allowing us to have more normalized residuals and consequently more accurate outputs. This will allow us to have valid regression results through ARIMA than OLS.

Since nt is somewhat like an error term, by using ARIMA we will be able to properly account for the heteroscedasticity present in the data allowing us to have standard estimates that are also more accurate than in the case of OLS regression.

Bibliography

Otexts.com. 2021. *9.8 Forecasting | Forecasting: Principles and Practice (3rd ed)*. [online] Available at: <https://otexts.com/fpp3/arima-forecasting.html> [Accessed 6 May 2021].

Otexts.com. 2021. *9.9 Seasonal ARIMA models | Forecasting: Principles and Practice (3rd ed)*. [online] Available at: <https://otexts.com/fpp3/seasonal-arima.html> [Accessed 6 May 2021].

Otexts.com. 2021. *Decomposition*. [online] Available at: <https://otexts.com/fpp3/forecasting-decomposition.html> [Accessed 6 May 2021].

Robjhyndman.com. 2021. *Dynamic regression*. [online] Available at: <https://robjhyndman.com/nyc2018/3-1-Dynamic-Regression.pdf> [Accessed 6 May 2021].

Florian Hartig, U., 2021. *DHARMa: residual diagnostics for hierarchical (multi-level/mixed) regression models*. [online] Cran.r-project.org. Available at: <https://cran.r-project.org/web/packages/DHARMa/vignettes/DHARMa.html> [Accessed 6 May 2021].

Forecast, A., 2021. *Arima , weekday dummy vars*. [online] Cross Validated. Available at: <https://stats.stackexchange.com/questions/205970/arima-model-with-weekday-dummy-variables-forecast> [Accessed 6 May 2021].

Otexts.com. 2021. *Fourier*. [online] Available at: <https://otexts.com/fpp3/useful-predictors.html#> [Accessed 6 May 2021].

Otexts.com. 2021. *Non-seasonal ARIMA models*. [online] Available at: <https://otexts.com/fpp3/non-seasonal-arima.html> [Accessed 6 May 2021].

Otexts.com. 2021. *Residual diagnostics*. [online] Available at: <https://otexts.com/fpp3/diagnostics.html> [Accessed 6 May 2021].

Otexts.com. 2021. *STL-Arima*. [online] Available at: <https://otexts.com/fpp3/forecasting-decomposition.html> [Accessed 6 May 2021].

Otexts.com. 2021. *Time series patterns*. [online] Available at: <https://otexts.com/fpp3/tspatterns.html> [Accessed 6 May 2021].

Rdocumentation.org. 2021. *yearquarter function - RDocumentation*. [online] Available at: <https://www.rdocumentation.org/packages/tsibble/versions/1.0.1/topics/yearquarter> [Accessed 6 May 2021].