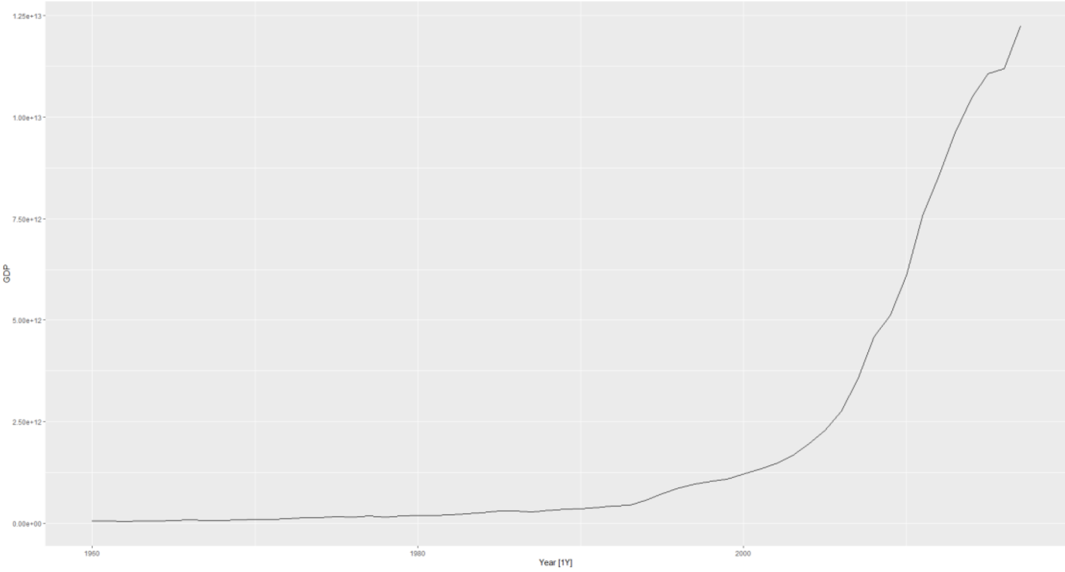
Problem 1

```
Library(fpp3)

chosen <- filter(global_economy, Code == "CHN")

autoplot(chosen, GDP)
```



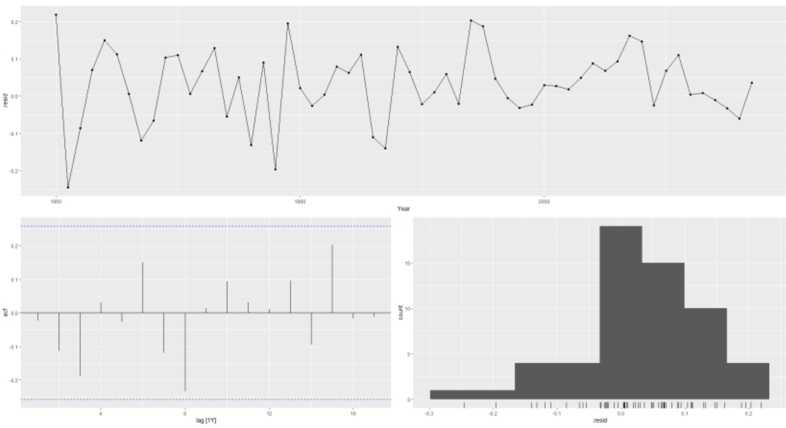
When the chosen data is autoplotted, the graph looks like the graph above.

```
fit1 <- chosen %>%
  model(
    G = ETS(GDP ~ error("M") + trend("A") + season("N")),
    Gdamp = ETS(GDP ~ error("M") + trend("Ad") + season("N")))
fit1 %>% glance()
```

```
# A tibble: 2 x 10
  Country .model sigma2 log_lik   AIC  AICc   BIC     MSE
  <fct>   <chr>   <dbl>   <dbl> <dbl> <dbl> <dbl>   <dbl>
1 China   G      0.0108  -1546. 3102. 3103. 3112. 4.00e22
2 China   Gdamp  0.0113  -1547. 3105. 3107. 3117. 3.98e22
# ... with 2 more variables: AMSE <dbl>, MAE <dbl>
```

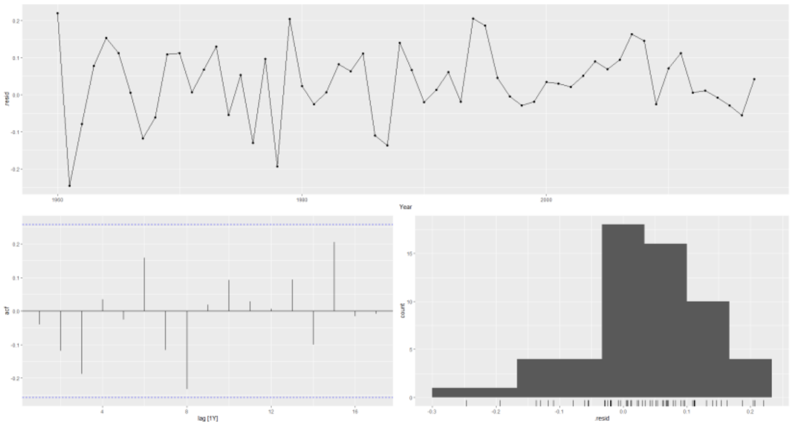
When we look at the normal and damped data, we can see that the normal data is doing slightly better.

```
fit1 %>% glance()

fit1 %>% tidy()

fit1 %>% select(G) %>% gg_tsresiduals()
```



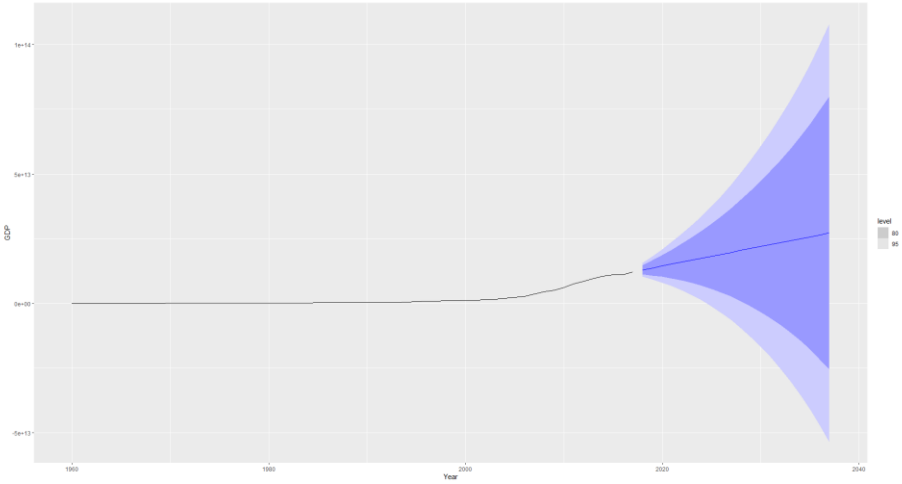
When we look at the residuals for normal, we can see that there is strong correlations with most data being captured. However there also appears to be heteroskedasticity with an uneven distribution.

```
fit1 %>% select(Gdamp) %>% gg_tsresiduals()
```



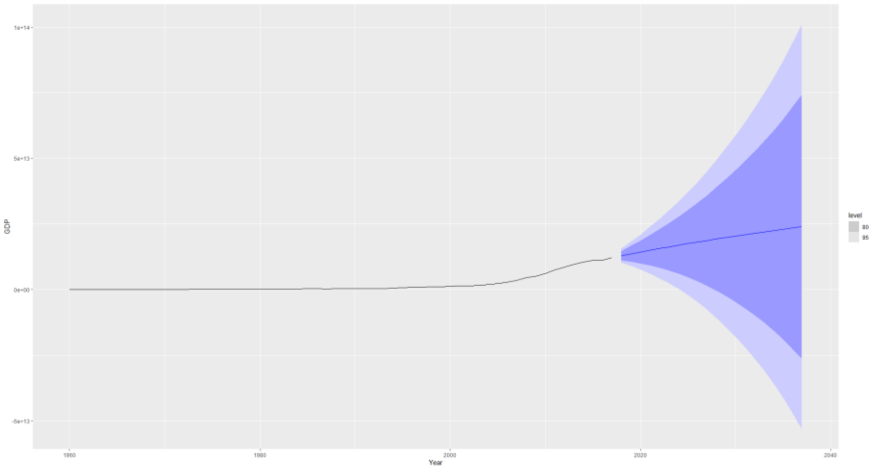
The residual data for damped is almost identical as expected.

```
fit1 %>% forecast(h = 20) %>% filter(.model == "G") %>% autoplot(chosen)
```



When the normal data is forecasted we get this.

```
fit1 %>% forecast(h = 20) %>% filter(.model == "Gdamp") %>% autoplot(chosen)
```

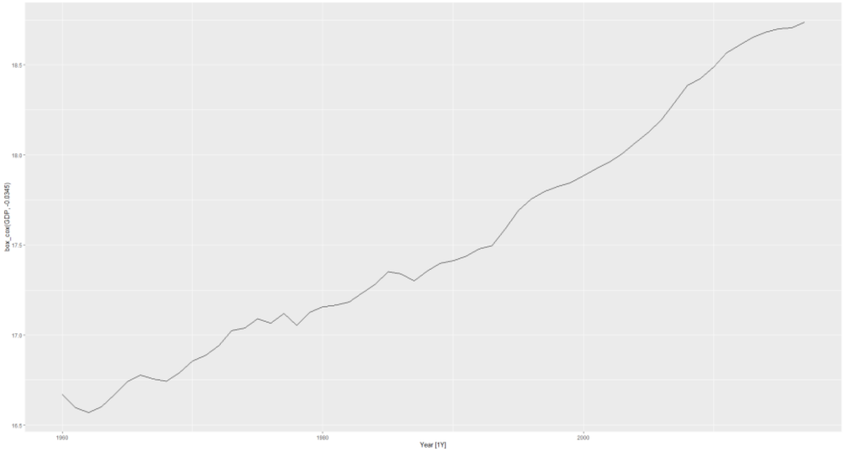


The damped data is also extremely similar although a bit more accurate since it has a smaller range of forecasts.

```
chosen %>% features(GDP, features = guerrero)
```

```
# A tibble: 1 x 2
  Country lambda_guerrero
  <fct>             <dbl>
1 China           -0.0345
```

```
chosen %>% autoplot(box_cox(GDP, -0.0345))
```



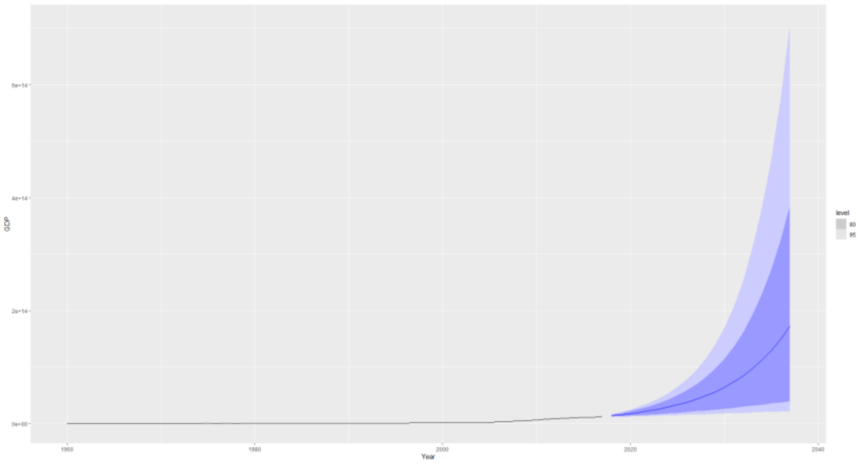
When the boxcox transformation is plotted however it looks like this. By following the rest of the basic model steps we get

```
fit <- chosen %>% model( G = ETS(log(GDP) ~ error("A") + trend("A") + season("N")), Gdamped =
ETS(log(GDP) ~ error("A") + trend("Ad") + season("N")))

fit %>% glance()
```

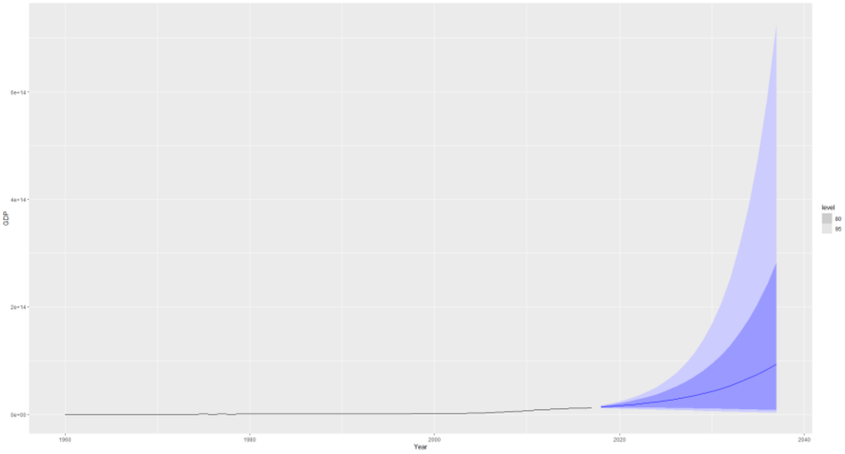
```
  Country .model  sigma2 log_lik   AIC  AICc   BIC     MSE
  <fct>   <chr>    <dbl>   <dbl> <dbl> <dbl> <dbl>   <dbl>
1 China   G      0.00881    21.5 -33.1 -31.9 -22.8 0.00820
2 China   Gdamp~ 0.00907    21.2 -30.5 -28.8 -18.1 0.00829
```

Which lets us know that damped performs a bit better.

```
fit %>% tidy()

fit %>% forecast(h = 20) %>% filter(.model == "G") %>% autoplot(chosen)
```



When the undamped version is autoplotted, we get this compared to

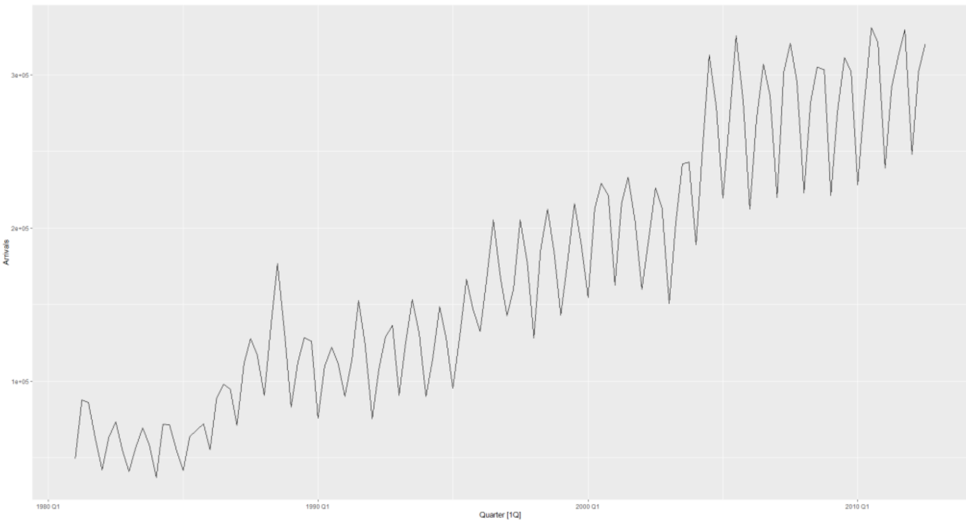


This shows us that the forecast ranges increased when the box cox transformation was used compared to when the ETS model was only used. This could be due to the normalizing of the data leading to a bigger skew in the forecasts.

Problem 2

a)

```
aus_arrivals

chosen <- filter(aus_arrivals, Origin == "NZ")

autoplot(chosen)
```



There is a general positive trend over time potentially due to the easier access to flights and international travel over time along with the growing population.

There also seems to be strong seasonality with increased arrivals at the beginning and end of each year.

The data is also very cyclical as seen above with variation within each period. However, this is harder to see from the data.

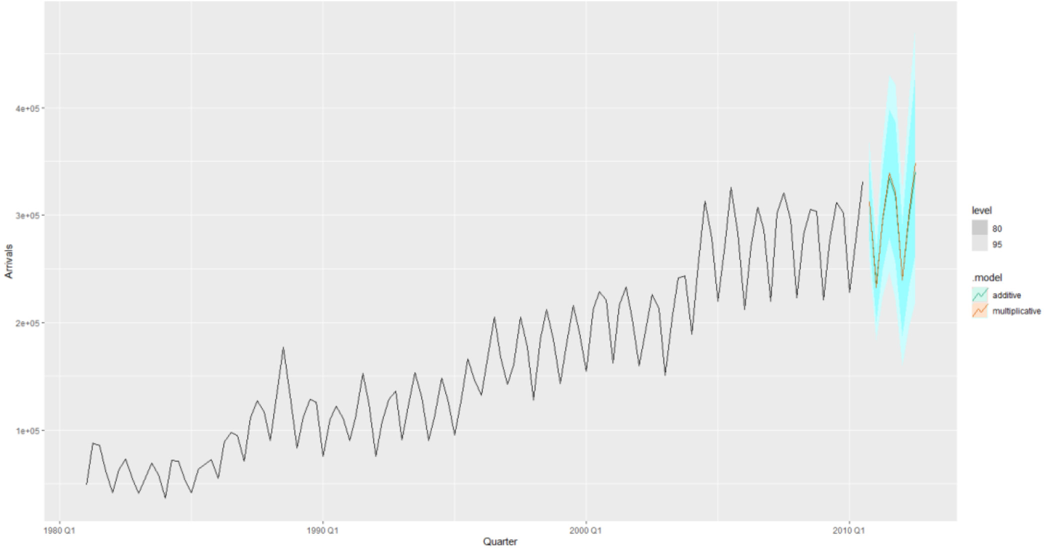
b)

```
chosen2 <- chosen %>% filter(Quarter < max(Quarter - 7))

tail(chosen2)
```

```
# A tsibble: 6 x 3 [1Q]
# Key:       Origin [1]
  Quarter Origin Arrivals
    <qtr> <chr>     <int>
1 2009 Q2 NZ       275761
2 2009 Q3 NZ       311431
3 2009 Q4 NZ       301983
4 2010 Q1 NZ       228162
5 2010 Q2 NZ       281791
6 2010 Q3 NZ       330812
```

So now since the data is only from 1981- 2010 q3, we can perform H-W multiplicated method to forecast.

```
fit <- chosen2 %>% model(additive = ETS(Arrivals ~ error("A") + trend("A") + season("A")),
multiplicative = ETS(Arrivals ~ error("M") + trend("A") + season("M")))

fc <- fit %>% forecast()

fc %>% autoplot(chosen2) + scale_color_brewer(type = "qual", palette = "Dark2")
```



When the data is forecasted for the chosen dataset, the graph above is outputted. As we can see both the multiplicate and additive results are quite similar. However, only the multiplicative results follow the uptrend in values.

c)

```
accuracy(fit)
```

```
# A tibble: 2 x 11
  Origin .model         .type       ME   RMSE    MAE    MPE  MAPE  MASE RMSSE    ACF1
  <chr>  <chr>          <chr>    <dbl>  <dbl>  <dbl>  <dbl> <dbl> <dbl> <dbl>   <dbl>
1 NZ     additive       Training  805. 12873. 10200. -0.174  8.17 0.686 0.665  0.0759
2 NZ     multiplicative Training -576. 11152.  8883. -1.07   6.65 0.597 0.576 -0.103
```

As seen above, the multiplicate results resulted in lower and therefore more accurate values.

This could be due to the level of the series increase over time.

Consequently, multiplicative seasonality displays larger and increasing seasonal variation as the level of the forecasts increased compared to the forecasts generated by the method with additive seasonality.

d)

i)

```
fit <- chosen2 %>% model(ETS(Arrivals))

fit
```

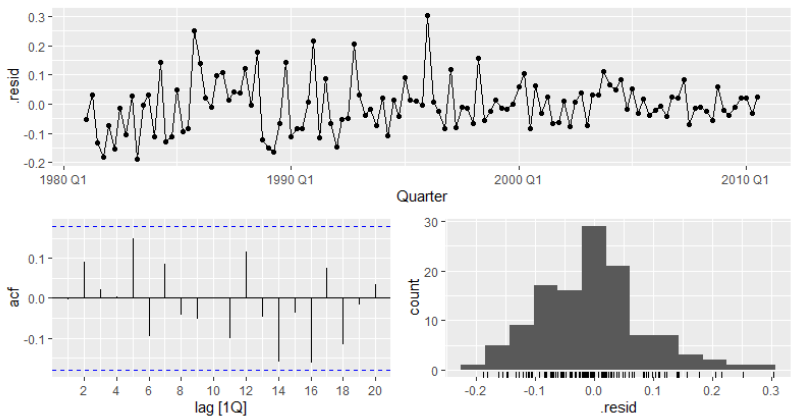
```
# A mable: 1 x 2
# Key:     Origin [1]
  Origin `ETS(Arrivals)`
  <chr>          <model>
1 NZ        <ETS(M,A,M)>
```

```
fit <- chosen2 %>% model(G = ETS(Arrivals ~ error("M") + trend("A") + season("M")), Gdamped =
ETS(Arrivals ~ error("M") + trend("Ad") + season("M")))

fit %>% glance()
```

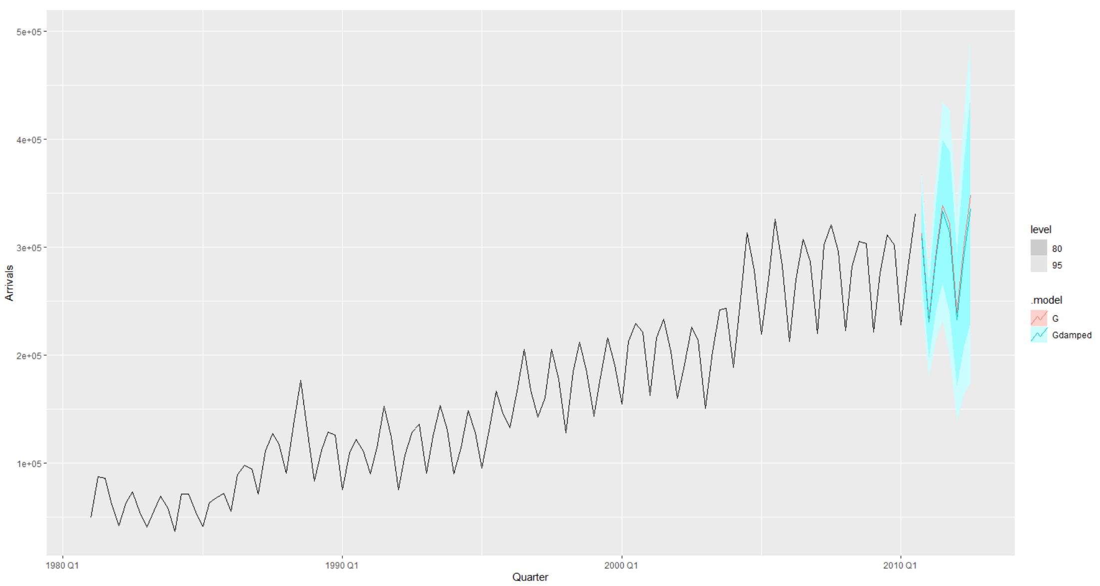
```
# A tibble: 2 x 10
  Origin .model   sigma2 log_lik   AIC  AICc   BIC        MSE
  <chr>  <chr>     <dbl>   <dbl> <dbl> <dbl> <dbl>      <dbl>
1 NZ     G       0.00822  -1406. 2831. 2833. 2856. 124365408.
2 NZ     Gdamped 0.00892  -1409. 2838. 2840. 2865. 125073838.
# ... with 2 more variables: AMSE <dbl>, MAE <dbl>
```

Therefore, as seen above, MAM is the best ETS model in this case which when graphed looks like the following.

```
fit %>% tidy()

fit %>% select(G) %>% gg_tsresiduals()
```




```
fit %>% forecast(h=8) %>% autoplot(chosen2)
```



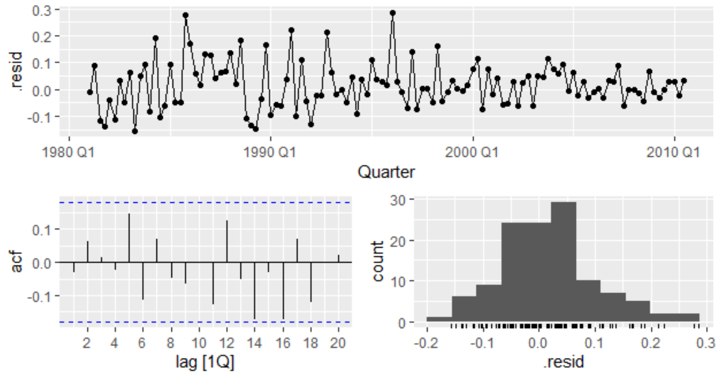
ii)

```
chosen2 %>% autoplot(log(Arrivals))

fit2 <- chosen2 %>% model(ETS(log(Arrivals)))

fit2
```

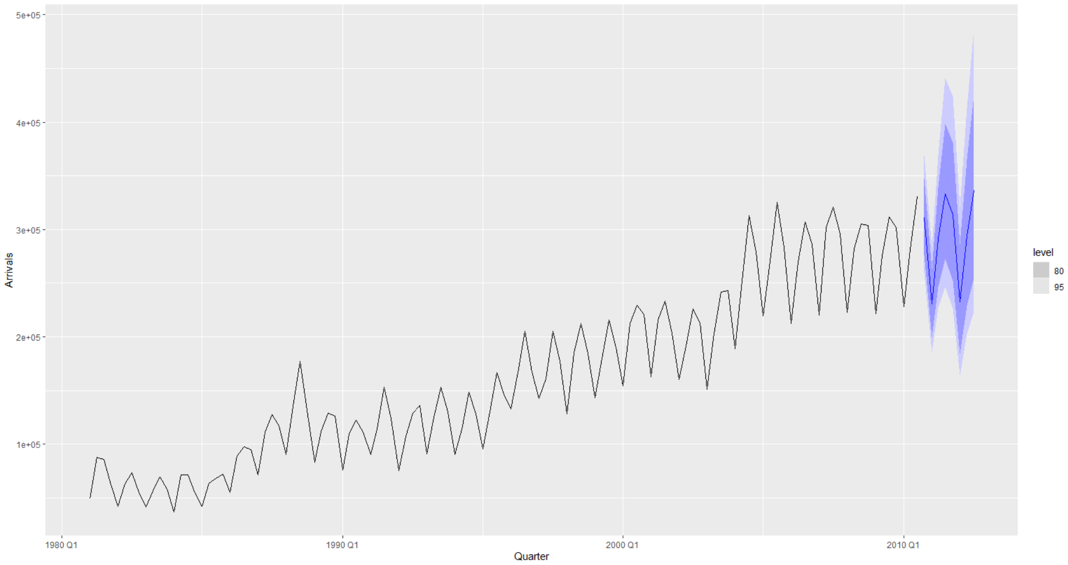
```
# A mable: 1 x 2
# Key:     Origin [1]
  Origin `ETS(log(Arrivals))`
  <chr>                <model>
1 NZ              <ETS(A,N,A)>
```

```
fit2 <- chosen2 %>% model( G = ETS(log(Arrivals) ~ error("A") + trend("N") + season("A")))

fit2 %>% tidy()
```

```
fit2 %>% gg_tsresiduals()
```



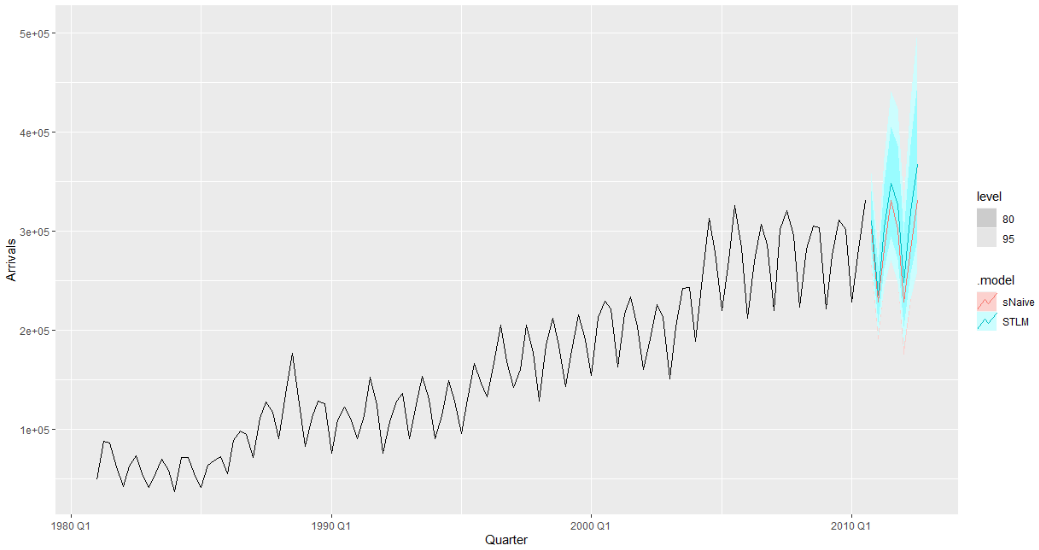

```
fit2 %>% forecast(h= 8) %>% autoplot(chosen2)
```



Therefore when the ANA model is used on the log data, the graph looks like this.

iii) and iv)

```
fit <- chosen2 %>% model(sNaive = SNAIVE(Arrivals), STLM =
decomposition_model(STL(log(Arrivals)), ETS(season_adjust))) %>% forecast(h = 8)

fit %>% autoplot(Arrivals)
```



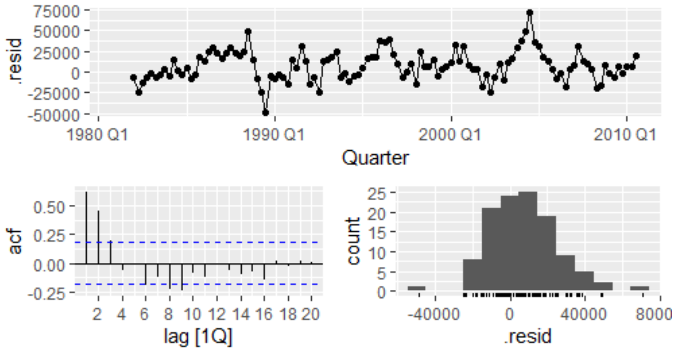
When the naïve and STLM is graphed along with the seasonally adjusted ETS data, the graph from above is created.

e)

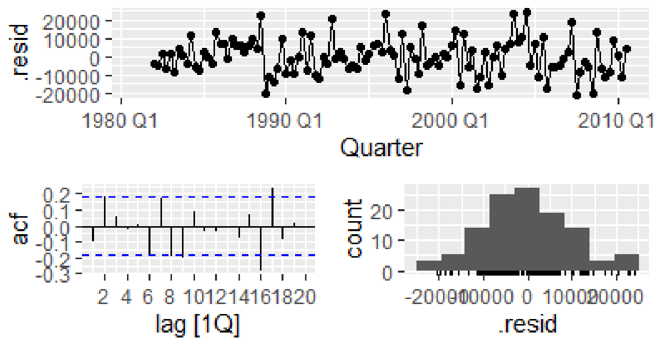
For all qs till iii and iv), I attached the residual charts.

The residual charts for iii) is found by doing

```
fit3 <- chosen2 %>% model(sNaive = SNAIVE(Arrivals))

fit3 %>% gg_tsresiduals()
```



```
fit4 <- chosen2 %>% model(STLM = decomposition_model(STL(log(Arrivals)), ETS(season_adjust)))

fit4 %>% gg_tsresiduals()
```



Since the residuals for the ETS model, additive ETS model both have lags which are below the blue significant value line, this lets us know that those values aren’t strong enough to be used thus can be rejected.

In contrast, the first few values for the ACF lag in the seasonal naïve method are above the significant value line. However, the residual chart is not a normal distribution with a bit more distribution to the left.

In contrast to the naïve method, the STL decomposition method has an ACF lag with consistent values that are above the lag and the residual chart is the most like a normal distribution when compared to the other forecasts.

Therefore, we can conclude that the STL decomposition gives us the best result in this case!