ECON2209 – Problem Set 2

Problem 1:

Figure 1: Autoplot of Chinese GDP with no transformations

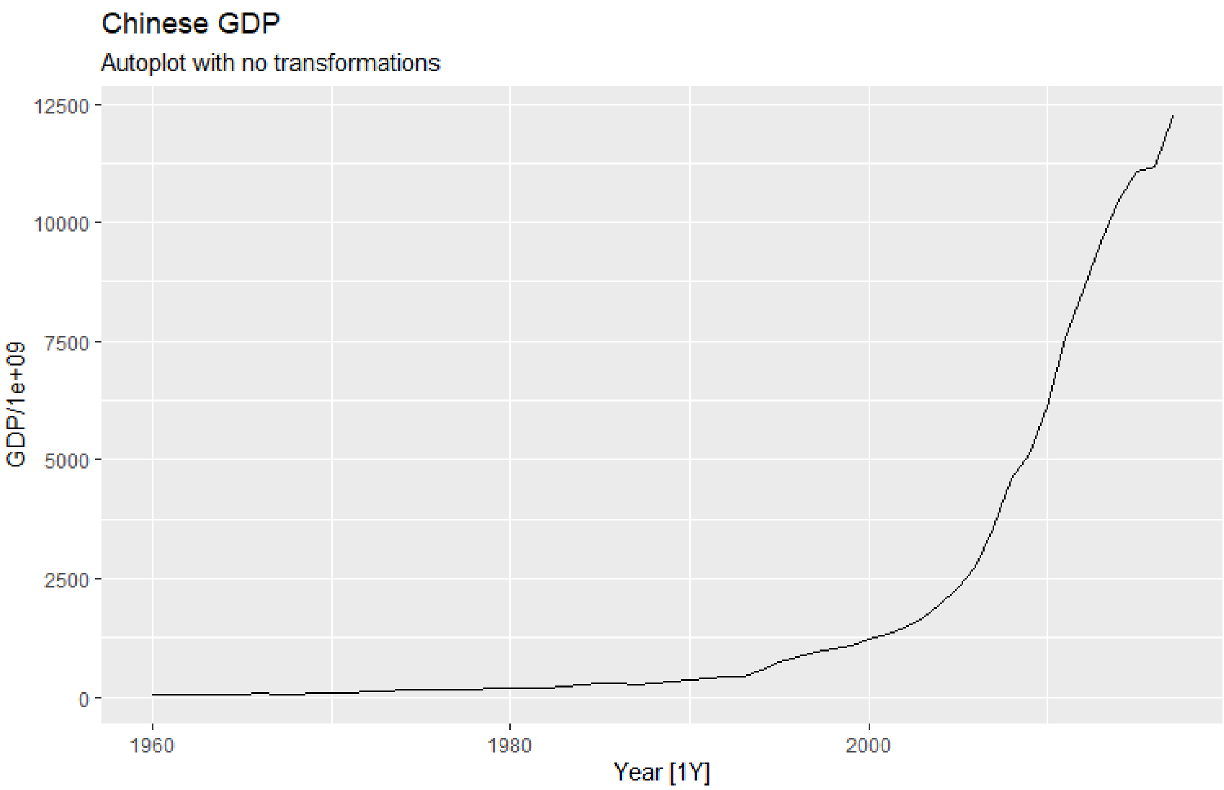


Figure 1 has been produced using autoplot to interpret the data for Chinese GDP on the global economy. A clear upward trend throughout the years is apparent, in addition the plot has an exponential shape which indicates that a logarithmic transformation could assist in forecasting. We can see that there is no upward trend until the 1990s, where the GDP increases considerably. There is no seasonality in the dataset, but a clear increasing trend. We can forecast the model using ETS with log transformations (because there are no values at 0 or below) and box-cox transformation. A dampened (A, Ad, N) trend method can also be successful in Chinese GDP because figure 1 shows an upward trend with no seasonality. Therefore, following plots will contain no seasonality.

```
# Question 1

# Figure 1: Autoplot with no transformations
View(global_economy)
china_gdp <- global_economy %>%
  filter(Country == "China")
View(china_gdp)
china_auto <- china_gdp %>%
  autoplot(GDP/1e9) +
  labs(title = "Chinese GDP", subtitle = "Autoplot with no transformations")
china_auto
```

Figure 2: Autoplot with log transformation

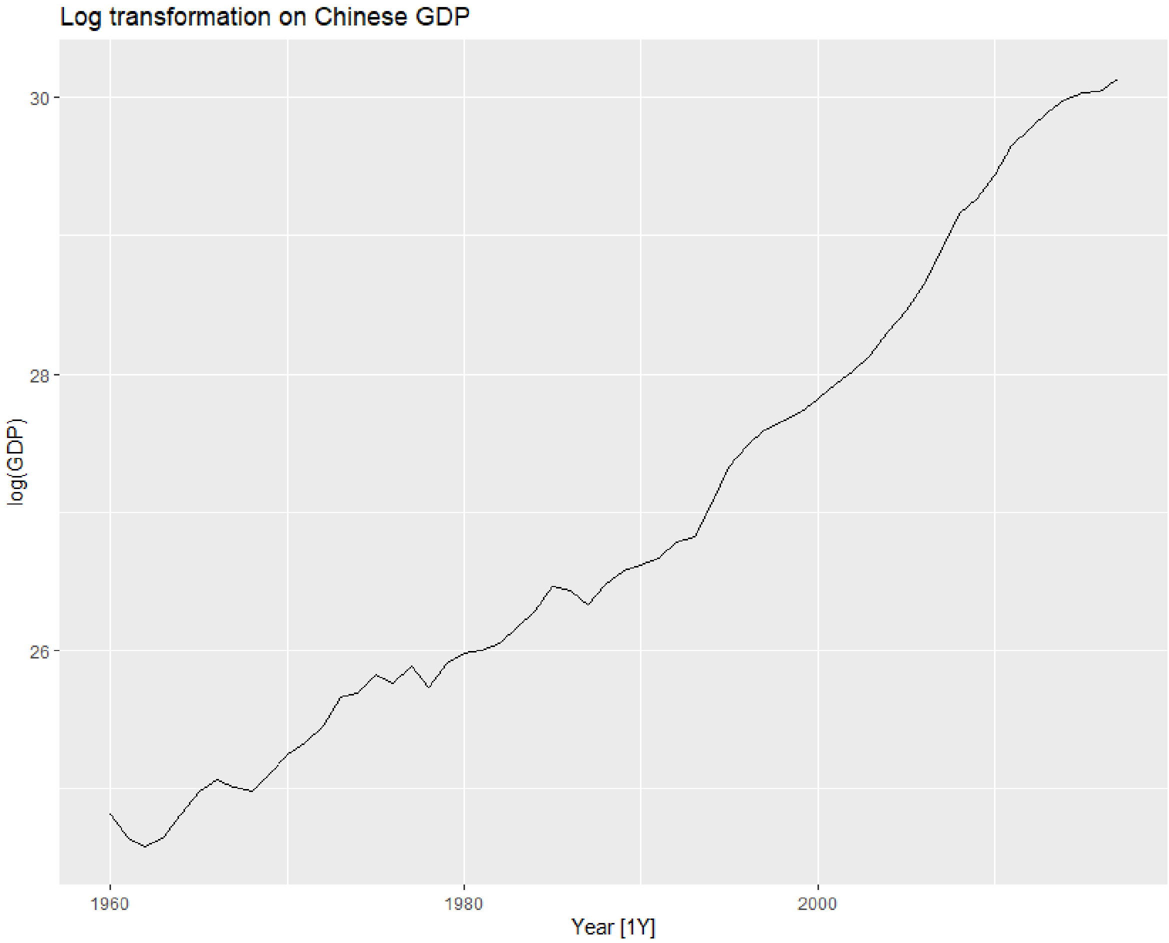


Figure 2 contains an autoplot of Chinese GDP transformed using a log transformation. The log transformation has clearly smoothed out the variation within the data. We can see a clear increasing trend, whereas the normal autoplot generated in figure 1 showcased a clear increasing trend from around 1980.

```
# Figure 2: Autoplot with log transformation
china_gdp %>%
  autoplot(log(GDP)) +
  labs(title = "Log transformation on Chinese GDP")
```

Figure 3: Autoplot with log transformation

Figure 3 showcases a box-cox transformed data of Chinese GDP using autoplot. The box-cox transformation has significantly transformed the data with a lambda of -0.0345 which is an inverse transformation. Using a lambda value of 0.2 would seem to be appropriate as the Guerrero method suggesting a stronger transformation. The variation has decreased much like in figure 2, which is very similar to figure 3.

```
# Figure 3: Autoplot with box-cox transformation
china_lambda <- china_gdp %>%
  features(GDP, guerrero)
china_gdp %>%
  autoplot(box_cox(GDP, -0.0345)) +
  labs(title = "Box-cox transformation on Chinese GDP")
```

Figure 4: ETS model (A, A, N) of Chinese GDP

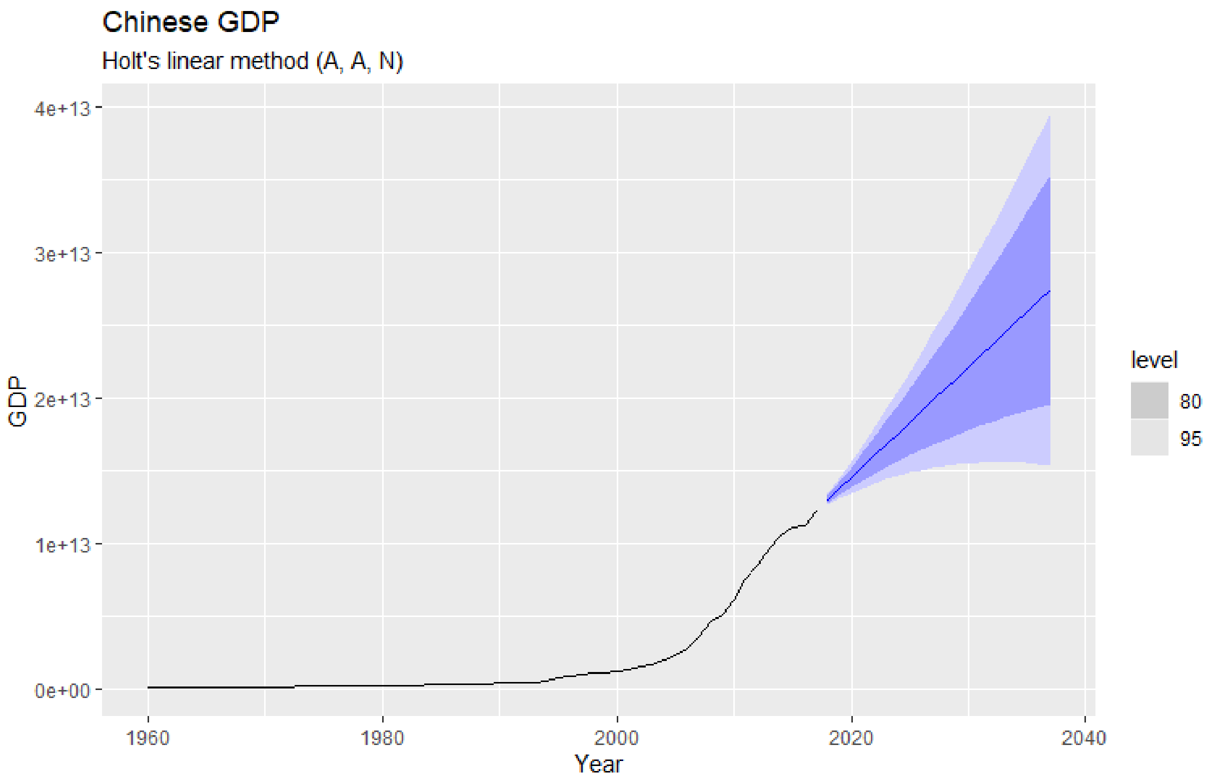


Figure 4 contains a forecast of an ETS model with additive errors and trend and no seasonality. The forecast after 20 years seems linear and increasing, which is accurate. The prediction intervals are asymmetrical and on the positive side.

```
Series: GDP
Model: ETS(A,A,N)
  Smoothing parameters:
    alpha = 0.9998964
    beta  = 0.5518569

  Initial states:
           l          b
 50284778074 3288256684

  sigma^2:  3.87701e+22

     AIC     AICc      BIC
3258.053 3259.207 3268.356
```

```
# Figure 4: ETS model (A, A, N) of Chinese GDP
chinese_gdp <- global_economy %>%
  filter(Country == "China") %>%
  summarise(GDP = sum(GDP))
fit <- chinese_gdp %>%
  model(ETS(GDP ~ error("A") + trend("A") + season("N")))
fc <- fit %>% forecast(h=20)
fc %>%
  autoplot(chinese_gdp) +
  labs(y = "GDP", title = "Chinese GDP", subtitle = "Holt's linear method (A, A, N)")

# ETS model (A, A, N) report
fit <- china_gdp %>%
  model(ETS(GDP ~ error("A") + trend("A") + season("N")))
report(fit)
```

Figure 5: ETS model (A, Ad, N) of Chinese GDP

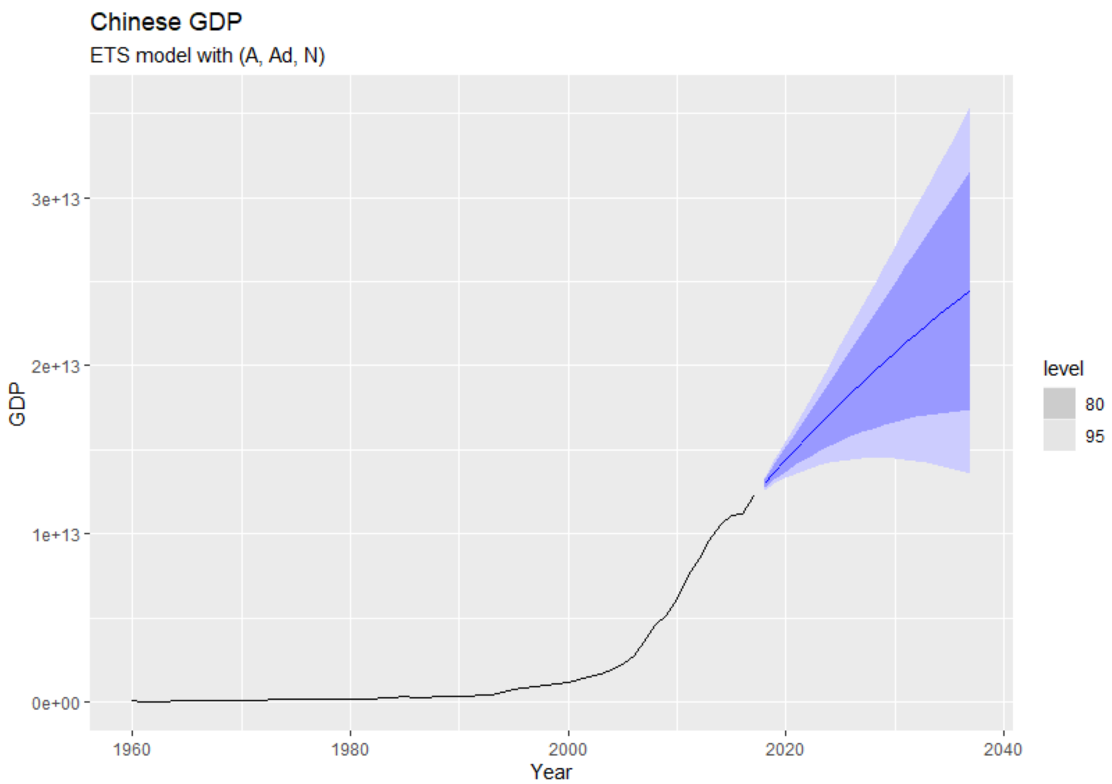


Figure 5 displays a forecasting ETS model containing additive errors, damped additive trend and no seasonality. Despite change from additive trend to damped trend in figure 4 and 5, there seems to be no difference in terms of the trend and shape of the plot. The forecast is positive and increasing with relatively large prediction intervals.

```
Series: GDP
Model: ETS(A,Ad,N)
  Smoothing parameters:
    alpha = 0.9998747
    beta  = 0.5634078
    phi   = 0.9799999

  Initial states:
           l          b
 50284778075 3288256683

  sigma^2:  3.959258e+22

     AIC     AICc      BIC
3260.187 3261.834 3272.549
```

Using the report() command on R produces the smoothing parameter estimates: alpha = 0.9998747 and beta = 0.5634078. These parameters are required for exponential smoothing method and all combinations of forecasting through the ETS model can be utilised. Since we are dealing with point forecasts, if we have the same smoothing parameters for some of the models then their ETS(A, *, *) models and ETS(M, *, *) are the same. The estimated smoothing coefficient for the level is alpha = 1.00. This value indicates that the level changes quickly to capture the highly trended series. The estimated smoothing coefficient of the slope is beta = 0.5634078 which suggests that the trend changes very slightly.

```
# Figure 5: ETS model (A, Ad, N) of Chinese GDP
chinese_gdp <- global_economy %>%
  filter(Country == "China") %>%
  summarise(GDP = sum(GDP))
fit <- chinese_gdp %>%
  model(ETS(GDP ~ error("A") + trend("Ad") + season("N")))
fc <- fit %>% forecast(h=20)
fc %>%
  autoplot(chinese_gdp) +
  labs(y = "GDP", title = "Chinese GDP", subtitle = "ETS model with (A, Ad, N)")

# ETS model (A, Ad, N) report
fit <- china_gdp %>%
  model(ETS(GDP ~ error("A") + trend("Ad") + season("N")))
report(fit)
```

<u>Figure 6: ETS model (A, Md, N) of Chinese GDP</u>

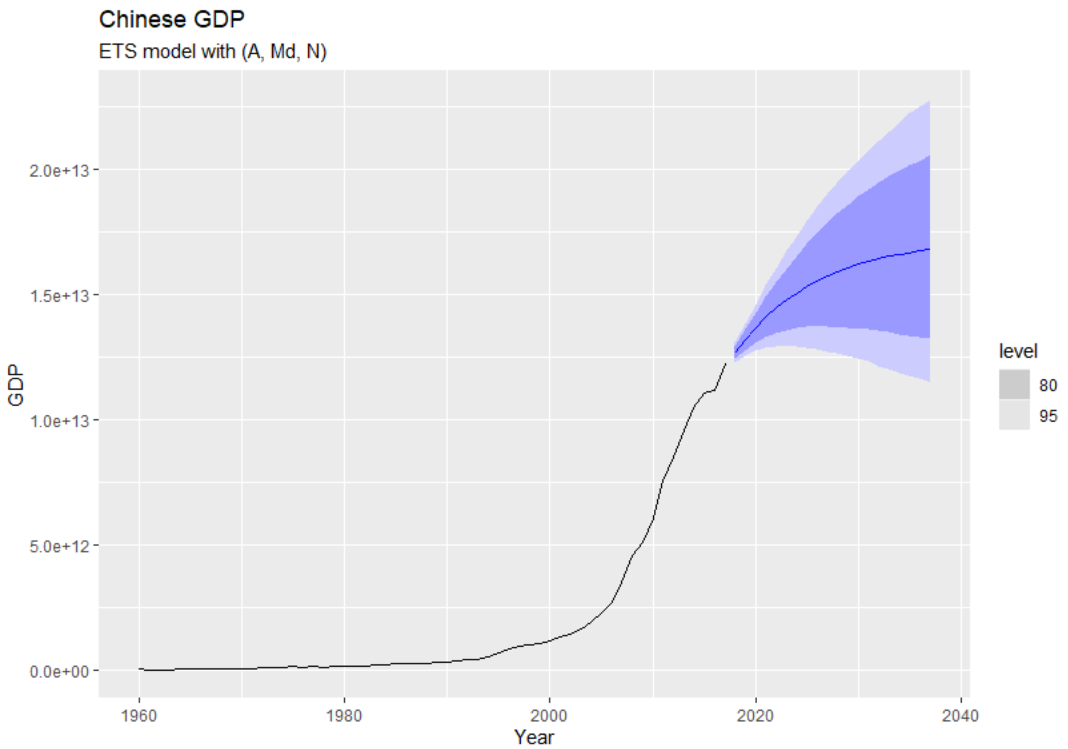
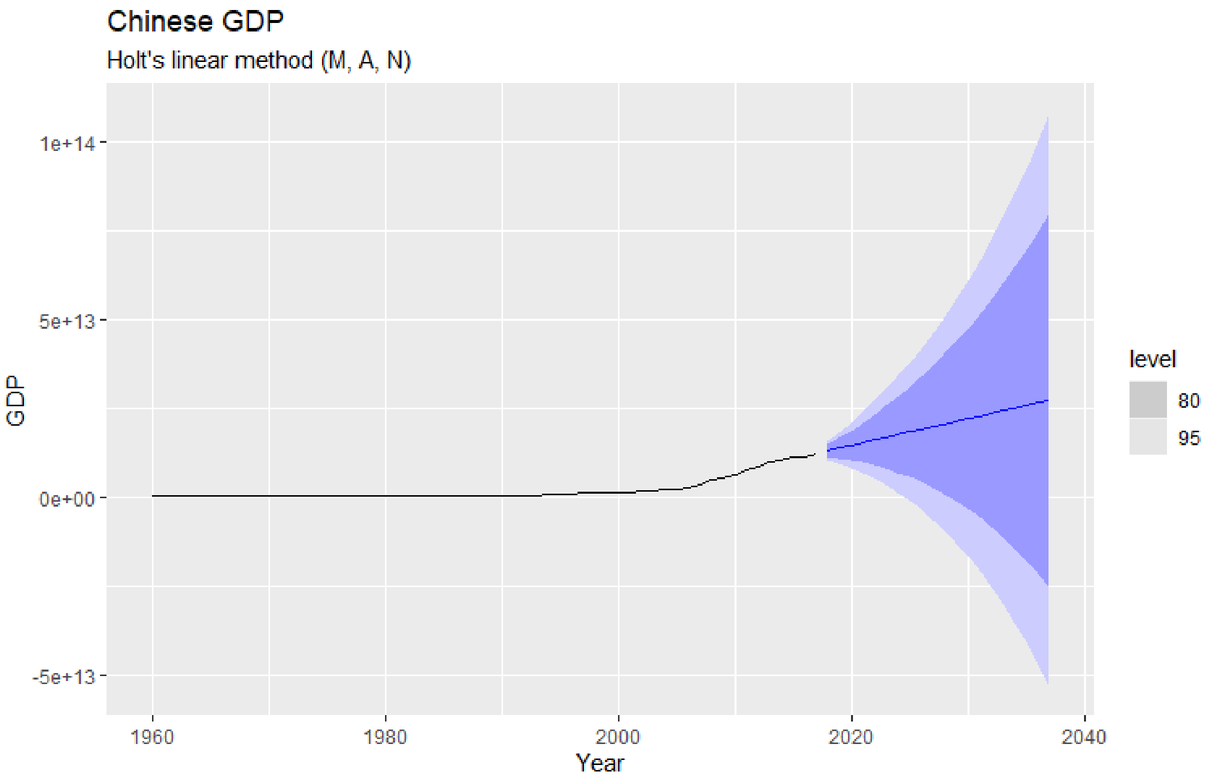


Figure 6 contains a forecasting ETS plot of additive errors, multiplicative damped trend and no seasonality. The difference between figure 5 and 6 is that figure 6 contains multiplicative damped trend and figure 3 contains additive damped trend. From this output, we can see that the forecast of 20 years is increasing but only gradually, compared to the exponential trend from the 1990s to 2010s. This sudden change in trend may be due to COVID-19. The overall Chinese GDP have certainly been affected in certain ways which may cause the nation's GDP to increase, but in a slower rate.

```
Series: GDP
Model: ETS(A,Md,N)
  Smoothing parameters:
    alpha = 0.8316326
    beta  = 0.4963642
    phi   = 0.8584629

  Initial states:
          l        b
 50512238526 2.035976

  sigma^2:  3.864352e+22

     AIC     AICc      BIC
3258.780 3260.427 3271.142
```

```
# Figure 6: ETS model (A, Md, N) of Chinese GDP
chinese_gdp <- global_economy %>%
  filter(Country == "China") %>%
  summarise(GDP = sum(GDP))
fit <- chinese_gdp %>%
  model(ETS(GDP ~ error("A") + trend("Md") + season("N")))
fc <- fit %>% forecast(h=20)
fc %>%
  autoplot(chinese_gdp) +
  labs(y = "GDP", title = "Chinese GDP", subtitle = "ETS model with (A, Md, N)")

# ETS model (A, Md, N) report
fit <- china_gdp %>%
  model(ETS(GDP ~ error("A") + trend("Md") + season("N")))
report(fit)
```

Figure 7: ETS model (M, A, N) of Chinese GDP

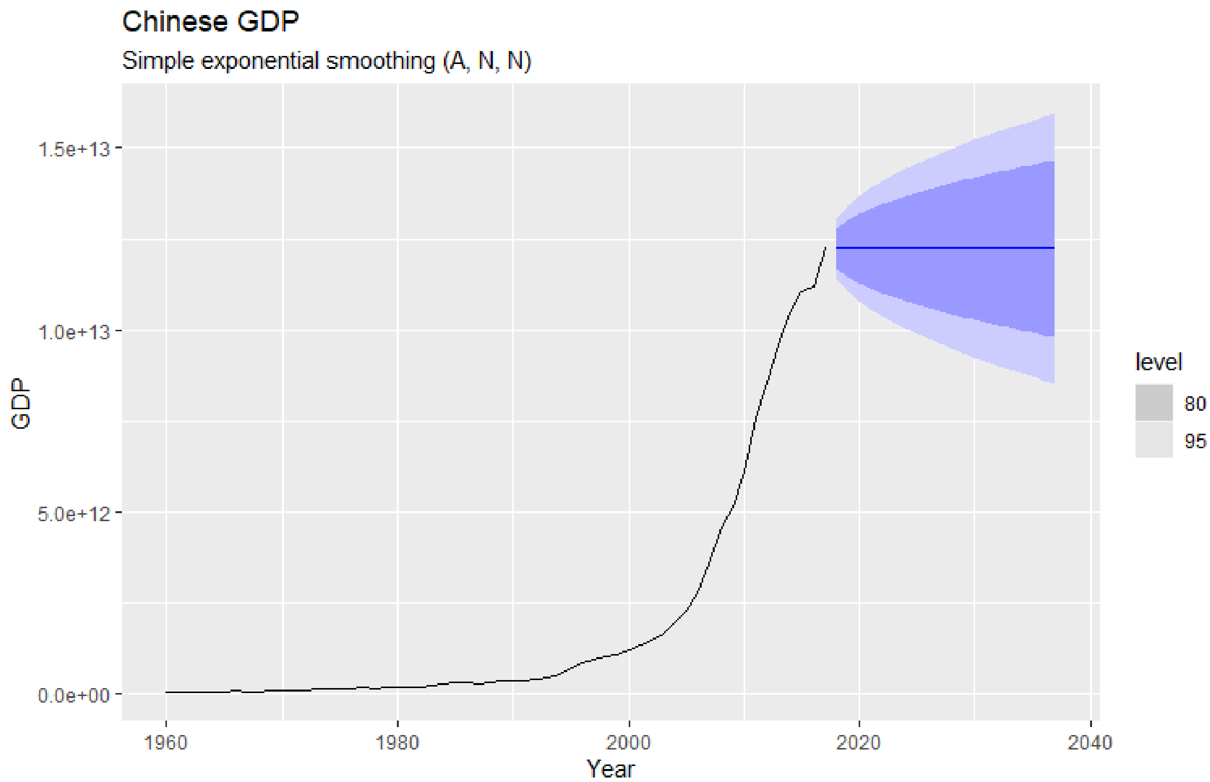


In figure 7, the plot displays Holt's linear model with multiplicative errors, additive trend, and no seasonality. From interpreting the plot, we can conclude that it appears to provide appropriate point forecasts for the next 20 years with an overall increasing trend. However, due to the large and symmetrical prediction intervals which extend to negative values of GDP, the forecast seems inaccurate. The default ETS model gives an output of multiplicative errors, additive trend and no seasonality, which equates to the lowest AICc value compared to all the other plots. However, a transformed forecasting model can have a lower AICc value, which will be explored in the models below.

```
Series: GDP
Model: ETS(M,A,N)
  Smoothing parameters:
    alpha = 0.9998998
    beta  = 0.3119984

  Initial states:
           l          b
 45713434615 3288256682

  sigma^2:  0.0108

     AIC     AICc      BIC
3102.064 3103.218 3112.366
```

```
# Figure 7: ETS model (M, A, N) of Chinese GDP
chinese_gdp <- global_economy %>%
  filter(Country == "China") %>%
  summarise(GDP = sum(GDP))
fit <- chinese_gdp %>%
  model(ETS(GDP ~ error("M") + trend("A") + season("N")))
fc <- fit %>% forecast(h=20)
fc %>%
  autoplot(chinese_gdp) +
  labs(y = "GDP", title = "Chinese GDP", subtitle = "Holt's linear method (M, A, N)")

# ETS model (M, A, N) report
fit <- china_gdp %>%
  model(ETS(GDP ~ error("M") + trend("A") + season("N")))
report(fit)
```

Figure 8: ETS model (A, N, N) of Chinese GDP



In figure 8, the ETS model contains additive errors, no trend and seasonality. From the plot we can see that the forecast for 20 years contains a symmetrical prediction interval and a point forecast which is a straight line. This seems quite inaccurate given that the GDP of China has vastly increased from the 1990s. As a result of COVID-19, we expect China's GDP to fluctuate some sort, whether increasing or decreasing, but not creating a horizontal trend. Thus, by interpreting this model we can conclude that there are better ETS models for forecasting Chinese GDP.

```
Series: GDP
Model: ETS(M,N,N)
  Smoothing parameters:
    alpha = 0.9998997

  Initial states:
           l
 63798846367

  sigma^2:  0.0208

     AIC     AICc      BIC
3130.708 3131.152 3136.889
```

```
# Figure 8: ETS model (A, N, N) of Chinese GDP
chinese_gdp <- global_economy %>%
  filter(Country == "China") %>%
  summarise(GDP = sum(GDP))
fit <- chinese_gdp %>%
  model(ETS(GDP ~ error("A") + trend("N") + season("N")))
fc <- fit %>% forecast(h=20)
fc %>%
  autoplot(chinese_gdp) +
  labs(y = "GDP", title = "Chinese GDP", subtitle = "Simple exponential smoothing (A, N, N)")

# ETS model (A, N, N) report
fit <- china_gdp %>%
  model(ETS(GDP ~ error("A") + trend("N") + season("N")))
report(fit)
```

Figure 9: Log transformation with Holt's Linear Method

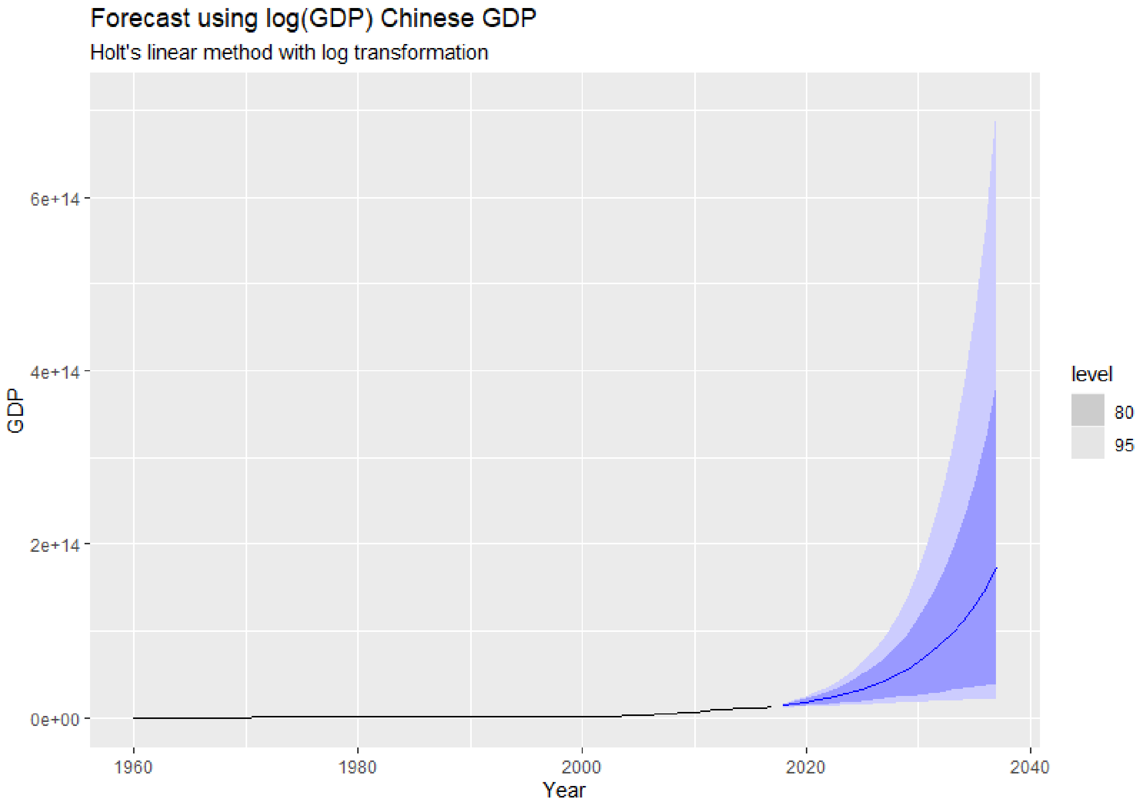


Figure 9 consists of a plot with Holt's Linear method with log transformation on Chinese GDP. This logarithm transformation of GDP produces an appropriate forecast for the next 20 years. An exponential increase in Chinese GDP from 2010 is apparent. Compared to multiplicative errors, the prediction intervals in figure 9 are more appropriate as multiplicative errors produce errors which are disproportionate on the positive side of the point forecasts. However, the overall forecast seems to be appropriate when using this model including the logarithmic transformation.

```
Series: GDP
Model: ETS(A,A,N)
Transformation: box_cox(GDP, -0.0345)
  Smoothing parameters:
    alpha = 0.9968112
    beta  = 0.07819836

  Initial states:
        l          b
 16.64718 0.02235156

  sigma^2:  0.0014

      AIC      AICc       BIC
-138.6232 -137.4694 -128.3210
```

```
# Figure 9: Log transformation with Holt's Linear Method
china_log <- china_gdp %>%
  model(ETS(log(GDP)))
china_log
china_log %>%
  forecast(h=20) %>%
  autoplot(china_gdp) +
  labs(title = "Forecast using log(GDP) Chinese GDP",
       subtitle = "Holt's linear method with log transformation")

# ETS model of log transformation on Holt' Linear Method
fit <- china_gdp %>%
  model(ETS(log(GDP)))
report(fit)
```

Figure 10: Box-Cox transformation with Holt's Linear Method

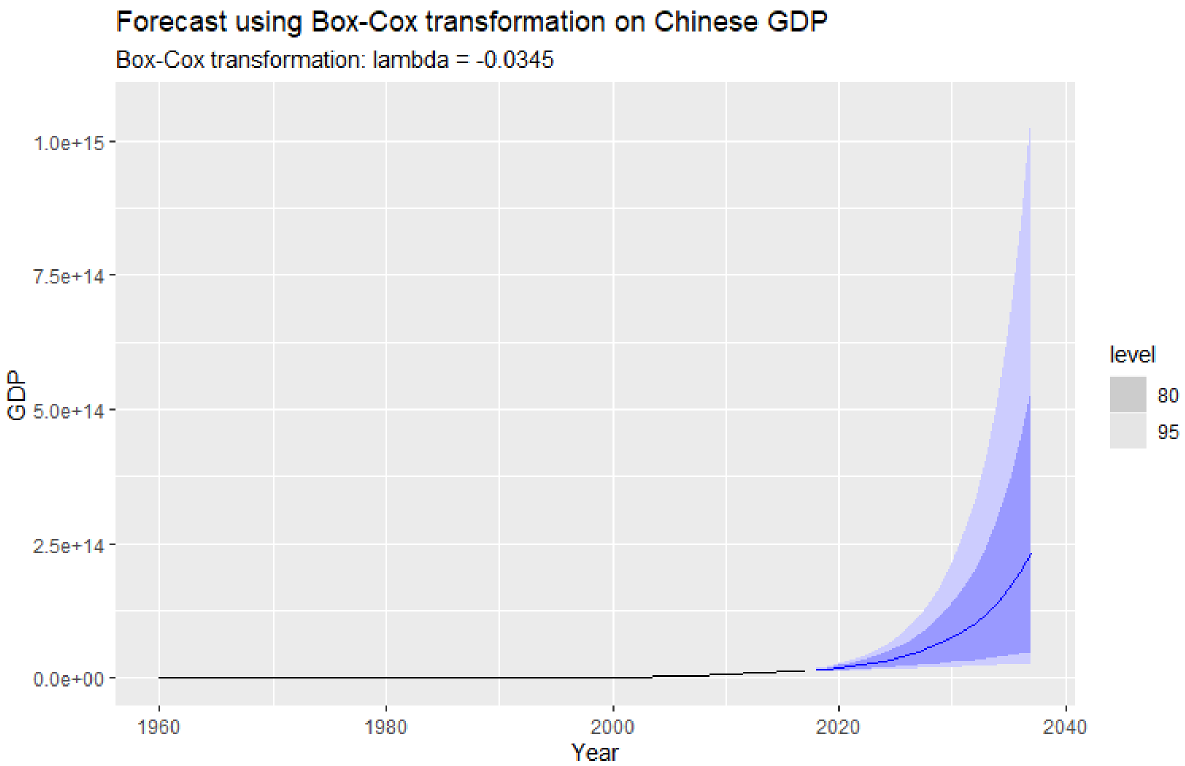


Figure 10 indicates Holt's Linear Method with additive errors, no trend and seasonality on a Box-Cox transformation of Chinese GDP which seems to produce an appropriate forecast. Similar to the log(GDP) forecast, this method indicates an exponential and increasing trend from 2010 with an accurate increasing point forecast and prediction intervals which seem appropriate. This plot is similar to the log transformed plot in figure 9 because lambda is close to 0, indicating that a Box-Cox transformation will produce similar values to the log transformed data.

```
Series: GDP
Model: ETS(A,A,N)
Transformation: log(GDP)
  Smoothing parameters:
    alpha = 0.9999
    beta  = 0.1079782

  Initial states:
        l          b
 24.77007 0.04320226

  sigma^2:  0.0088

      AIC      AICc       BIC
-33.07985 -31.92600 -22.77763
```

```
# Figure 10: Box-Cox transformation with Holt' Linear Method
china_lambda <- china_gdp %>%
  features(GDP, guerrero)
china_lambda
auto_box_fit <- china_gdp %>%
  model(ETS(box_cox(GDP, -0.0345)))
auto_box_fit
auto_box_fit %>%
  forecast(h=20) %>%
  autoplot(china_gdp) +
  labs(title = "Forecast using Box-Cox transformation on Chinese GDP",
       subtitle = "Box-Cox transformation: lambda = -0.0345")

# ETS model of Box-Cox transformation on Holt's Linear Method
fit <- china_gdp %>%
  model(ETS(box_cox(GDP, -0.0345)))
report(fit)
```

Figure 11: All ETS models together

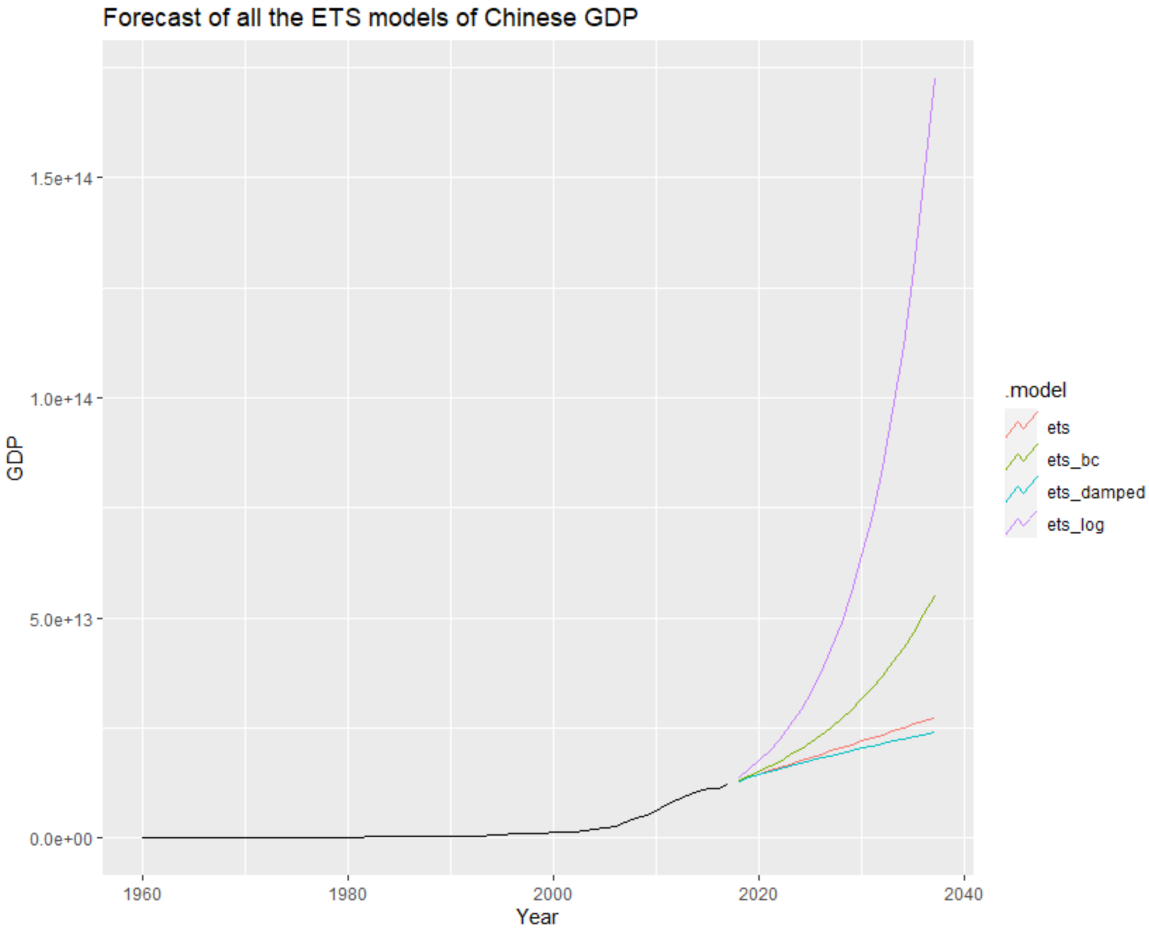


Figure 11 showcases all the ETS models including the transformed data into one plot. The transformations seem to have a large effect, with small lambda values causing a big increase in the forecast. We observe that the damped trend on the ETS model has resulted in a small effect compared to the regular ETS model. All the forecasts are increasing, with the logarithmic transformation showing an obvious exponential increase which is much greater than the other models. The box-cox transformation also forecasts accurately however is much less exaggerated compared to the logarithmic transformed ETS model. After comparing these ETS models in their respective forecasts, we can compare the models using model selection, specifically looking at their Akaike's Information Criterion.

```
# Figure 10: Box-Cox transformation with Holt' Linear Method
china_lambda <- china_gdp %>%
  features(GDP, guerrero)
china_lambda
auto_box_fit <- china_gdp %>%
  model(ETS(box_cox(GDP, -0.0345)))
auto_box_fit
auto_box_fit %>%
  forecast(h=20) %>%
  autoplot(china_gdp) +
  labs(title = "Forecast using Box-Cox transformation on Chinese GDP",
       subtitle = "Box-Cox transformation: lambda = -0.0345")

# ETS model of Box-Cox transformation on Holt's Linear Method
fit <- china_gdp %>%
  model(ETS(box_cox(GDP, -0.0345)))
report(fit)
```

Figure 12: Model Selection on ETS models of Chinese GDP

```
# A tibble: 11 x 10
   Country .model     sigma2 log_lik    AIC   AICc    BIC      MSE     AMSE      MAE
   <fct>   <chr>       <dbl>   <dbl>  <dbl>  <dbl>  <dbl>    <dbl>    <dbl>    <dbl>
 1 China   given    1.08e- 2 -1546.   3102.  3103.  3112.  4.00e+22 1.61e+23 7.93e- 2
 2 China   ann      1.79e+23 -1669.   3345.  3345.  3351.  1.73e+23 7.22e+23 2.13e+11
 3 China   aadn     3.96e+22 -1624.   3260.  3262.  3273.  3.62e+22 1.30e+23 9.49e+10
 4 China   amdm     3.86e+22 -1623.   3259.  3260.  3271.  3.53e+22 1.24e+23 9.77e+10
 5 China   holtsALM 3.88e+22 -1624.   3258.  3259.  3268.  3.61e+22 1.31e+23 9.59e+10
 6 China   holtsMLM 1.08e- 2 -1546.   3102.  3103.  3112.  4.00e+22 1.61e+23 7.93e- 2
 7 China   mnn      2.08e- 2 -1562.   3131.  3131.  3137.  1.73e+23 7.22e+23 1.23e- 1
 8 China   madn     1.13e- 2 -1547.   3105.  3107.  3117.  3.98e+22 1.59e+23 8.03e- 2
 9 China   mmdn     9.85e- 3 -1544.   3100.  3102.  3112.  6.50e+22 3.19e+23 7.41e- 2
10 China   log      8.81e- 3    21.5  -33.1  -31.9  -22.8  8.20e- 3 2.30e- 2 7.22e- 2
11 China   boxcox   1.43e- 3    74.3 -139.  -137.  -128.   1.33e- 3 3.55e- 3 2.87e- 2
```

From the R output shown in figure 12 showcases information regarding model selection on the various ETS models. The correlated Akaike's Information Criterion (AICc) will indicate which model is the most appropriate (the lower the better). From figure 9 we can see that the box-cox transformed data for Holt's Linear Method contains the lowest AICc of -137. Whereas a logarithmic transformation gives -31.9 AICc. Therefore, through model selection of AICc we can conclude that the best and most accurate model for forecasting Chinese GDP is through box-cox transformation on Holt's Linear Method.

```
# Figure 12: Model Selection on ETS models of Chinese GDP
fit <- chinese_gdp <- global_economy%>%
  filter(Country =="China") %>%
  model(
    given = ETS(GDP),
    ann = ETS(GDP ~ error("A") + trend("N") + season("N")),
    aadn = ETS(GDP ~ error("A") + trend("Ad") + season("N")),
    amdm = ETS(GDP ~ error("A") + trend("Md") + season("N")),
    holtsALM = ETS(GDP ~ error("A") + trend("A") + season("N")),
    holtsMLM = ETS(GDP ~ error("M") + trend("A") + season("N")),
    mnn = ETS(GDP ~ error("M") + trend("N") + season("N")),
    madn = ETS(GDP ~ error("M") + trend("Ad") + season("N")),
    mmdn = ETS(GDP ~ error("M") + trend("Md") + season("N")),
    log = ETS(log(GDP)),
    boxcox = ETS(box_cox(GDP, 0.2))
  )
fit %>%
  glance()
```

Problem 2:

a)

Figure 13: Australian Arrivals from New Zealand

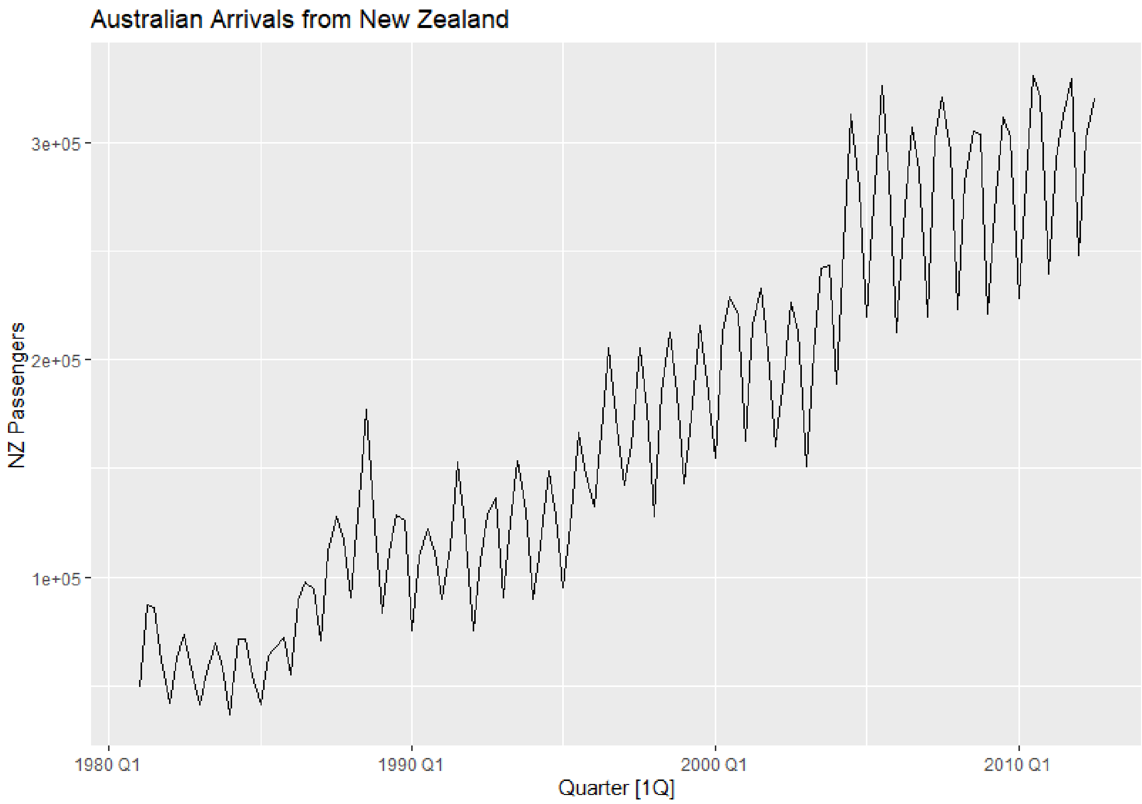
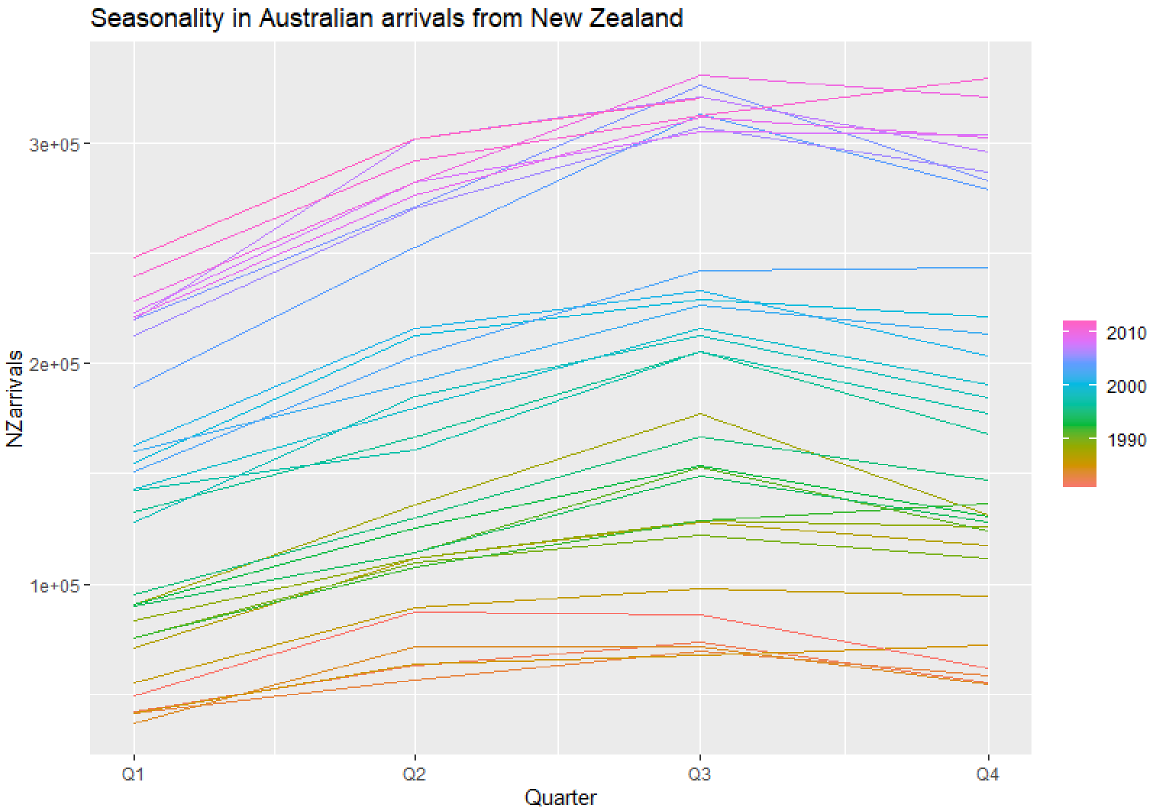


Figure 13 is a plot containing all the New Zealand passengers arriving to Australia from quarter 1 1981 to quarter 3 2012. There is an obvious increasing trend along the years, suggesting a positive relationship between the years (in quarters) and number of NZ passengers. The plot also reflects cyclicity due to the apparent troughs and crests between the years in quarters. We can further analyse the seasonality in this data set using gg_season().

```
# Problem 2
# Part a)
NZ_arrivals <- aus_arrivals %>%
  filter(Origin == "NZ") %>%
  summarise(NZarrivals = sum(Arrivals))
autoplot(NZ_arrivals) + labs(y="NZ Passengers", title = "Australian Arrivals from New Zealand")
```

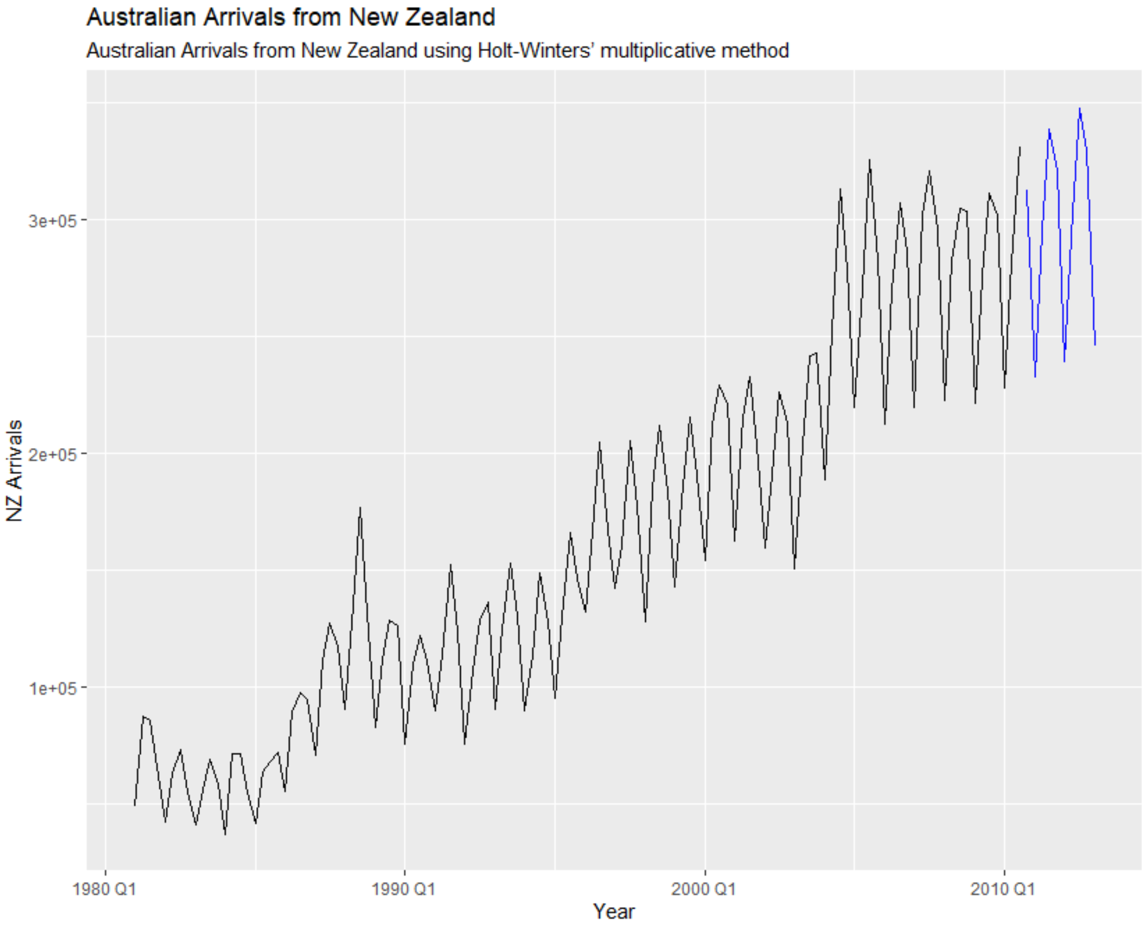
Figure 14: Seasonality in Australian arrivals from New Zealand



From figure 14 and interpreting the seasonality of New Zealand arrivals, we can confirm that seasonality exists. The gg seasonal plot showcases peaks around quarter 3 of the year, which is possible due to spring sales in plane tickets and decreases from quarter 4 to 1, due to winter which may have caused passengers to not visit Australia.

```
gg_season(NZ_arrivals) + labs(title = "Seasonality in Australian arrivals from New Zealand")
```

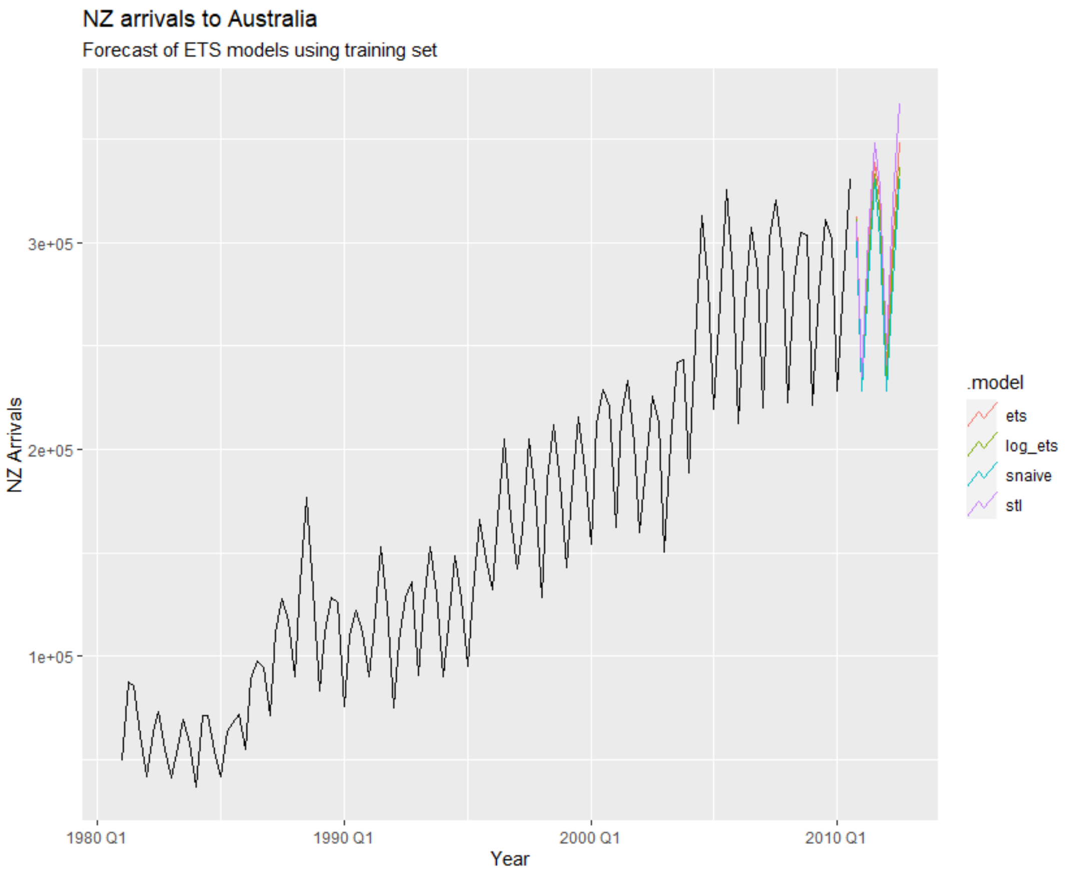
b)

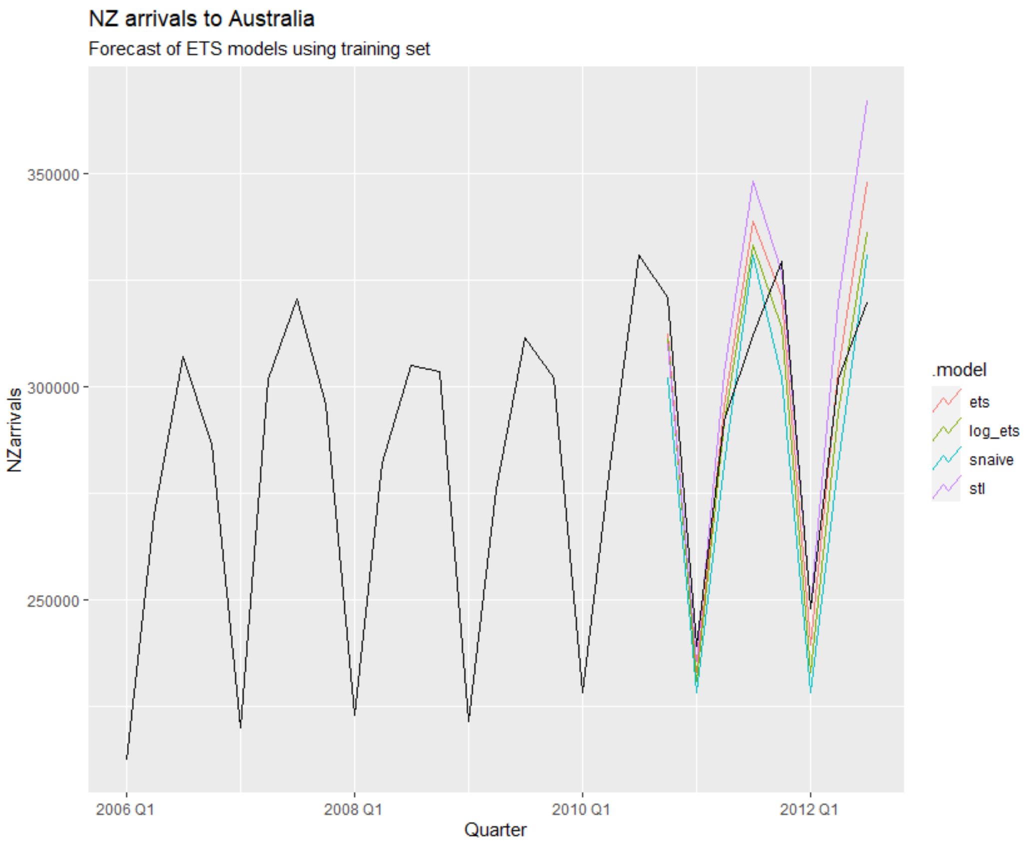



```
# Part b)
train_NewZealandPassengers <- NZ_arrivals %>%
  filter_index("1981 Q1" ~ "2010 Q3")
fit <- train_NewZealandPassengers %>% model(multiplicative  = ETS(NZarrivals ~  error("M")  + trend("A")  + season("M")))
fit <- train_NewZealandPassengers %>%
  model(multiplicative = ETS(NZarrivals ~ error("M") + trend("A") + season("M")))
fc <- fit %>% forecast(h=10)
fc %>% autoplot(train_NewZealandPassengers, level = NULL) + xlab("Year") +
  ylab("NZ Arrivals") +   scale_color_brewer(type = "qual", palette = "Dark2") +
  labs(title = "Australian Arrivals from New Zealand",
       subtitle = 'Australian Arrivals from New Zealand using Holt-Winters' multiplicative method')
```

c) Multiplicative seasonality is necessary here because the seasonal pattern increases in size, which is proportional to the level of the trend, which can be seen in figure 10. We can explain the behaviour of the seasonal pattern to be projected and captured in a model with multiplicative seasonality.

d) i) – iv)


NZ arrivals to Australia
Forecast of ETS models using training set
NZ Arrivals
3e+05
2e+05
1e+05
1980 Q1
1990 Q1
2000 Q1
2010 Q1
Year
.model
ets
log_ets
snaive
stl


NZ arrivals to Australia
Forecast of ETS models using training set
NZarrivals
350000
300000
250000
2006 Q1
2008 Q1
2010 Q1
2012 Q1
Quarter
.model
ets
log_ets
snaive
stl

```r
# Part d)
fc_new <- train_NewZealandPassengers %>% model(
  ets = ETS(NZarrivals),
  log_ets = ETS(log(NZarrivals)),
  snaive = SNAIVE(NZarrivals),
  stl = decomposition_model(STL(log(NZarrivals)), ETS(season_adjust))
) %>%
  forecast(h = "2 years")
fc_new %>% autoplot(train_NewZealandPassengers, level = NULL) +
  xlab("Year") + ylab("NZ Arrivals") + labs(title = "NZ arrivals to Australia",
                                           subtitle = "Forecast of ETS models using training set")


fc_new %>% autoplot(level = NULL) +autolayer(filter(NZ_arrivals, year(Quarter) > 2005), NZarrivals) +
  labs(title = "NZ arrivals to Australia",
       subtitle = "Forecast of ETS models using training set")
```

e)

Figure 15: Accuracy of ETS, log_ets, seasonal naïve and STL decomposition

```
# A tibble: 4 x 10
  .model  .type      ME   RMSE    MAE    MPE  MAPE  MASE RMSSE    ACF1
  <chr>   <chr>   <dbl>  <dbl>  <dbl>  <dbl> <dbl> <dbl> <dbl>   <dbl>
1 ets     Test   -3495. 14913. 11421. -0.964  3.78 0.768 0.771 -0.0260
2 log_ets Test    2467. 13342. 11904.  1.03   4.03 0.800 0.689 -0.0786
3 snaive  Test    9709. 18051. 17156.  3.44   5.80 1.15  0.933 -0.239
4 stl     Test  -11967. 22749. 16289. -3.82   5.26 1.09  1.18   0.104
```

According to the accuracy table in figure 15, the best method is the logarithmic transformed ETS model. This is because the RMSE (Root Mean Squared Error) is the lowest compared to the other models. The model specified by ets (multiplicative error, additive trend, and multiplicative seasonality) performs better compared to other accuracy measured like MAPE and MAE. This accuracy table is biased because all the models contain different number of parameters. However, based on the RMSE, the model specified by log_ets will be utilised for residual diagnostics.

Figure 16: Residual Diagnostics of NZ arrivals to Australia

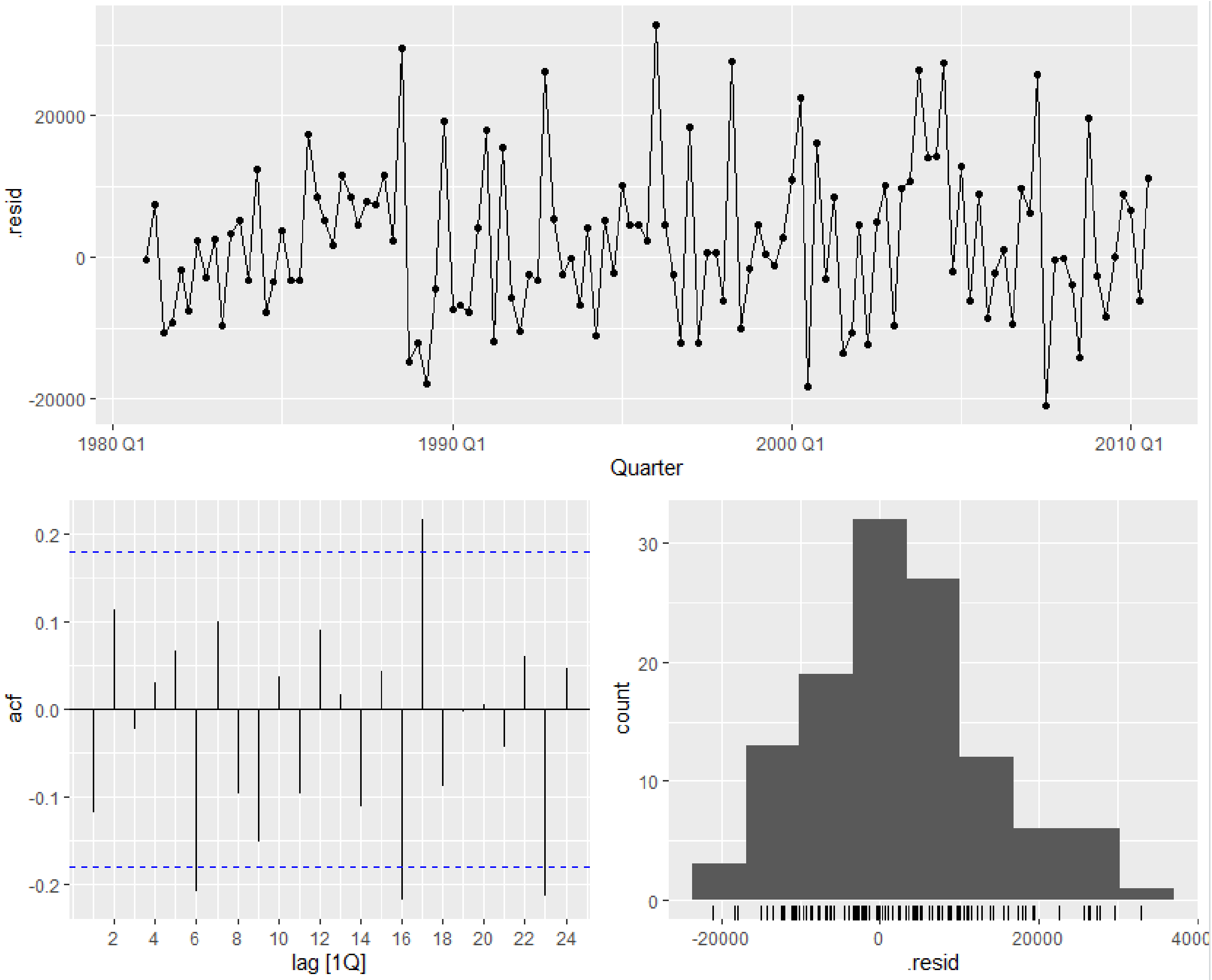


Figure 16 showcases residual diagnostic plots, including time plot, ACF and histogram for log transformation of the ETS model which is the best model to use according to the accuracy table in figure 15. The ACF residuals indicates that there is no indication of a pattern and possibility of white noise to be in the data, thus indicating no valuable information in the residuals. The histogram of the residual diagnostics in figure 16 reflects to be normally distributed with a slightly longer right tail. This slightly right trail is worth of little significance in terms of raising any concern for prediction intervals. We can then check the p-value for the log transformed data of the ETS model:

```
# A tibble: 1 x 4
  .model               lb_stat lb_pvalue .name_repair
  <chr>                  <dbl>     <dbl> <chr>
1 ETS(log(NZarrivals))    28.4    0.0400 minimal
```

The p-value for the log transformed ETS model is 0.0400 which is insignificant at the 5% significance level. The null of the Ljung-Box test is rejected. This is due to the large spike at lag 17 and the spikes at 6, 16 and 23 which are negative and go past the blue dotted line. We therefore reject the null hypothesis which is that the residuals are indistinguishable from white noise series. This indicates that there is autocorrelation between the residuals of the log transformed data of the ETS model.

```r
# Part e)
fc_new %>%
  accuracy(NZ_arrivals)

LogETS <- train_NewZealandPassengers %>% model(ETS(log(NZarrivals)))
augment(LogETS) %>% gg_tsdisplay(.resid, lag_max = 24, plot_type = "histogram")

augment(LogETS) %>%
  features(.innov, ljung_box, lag = 24, dof = 7)
```