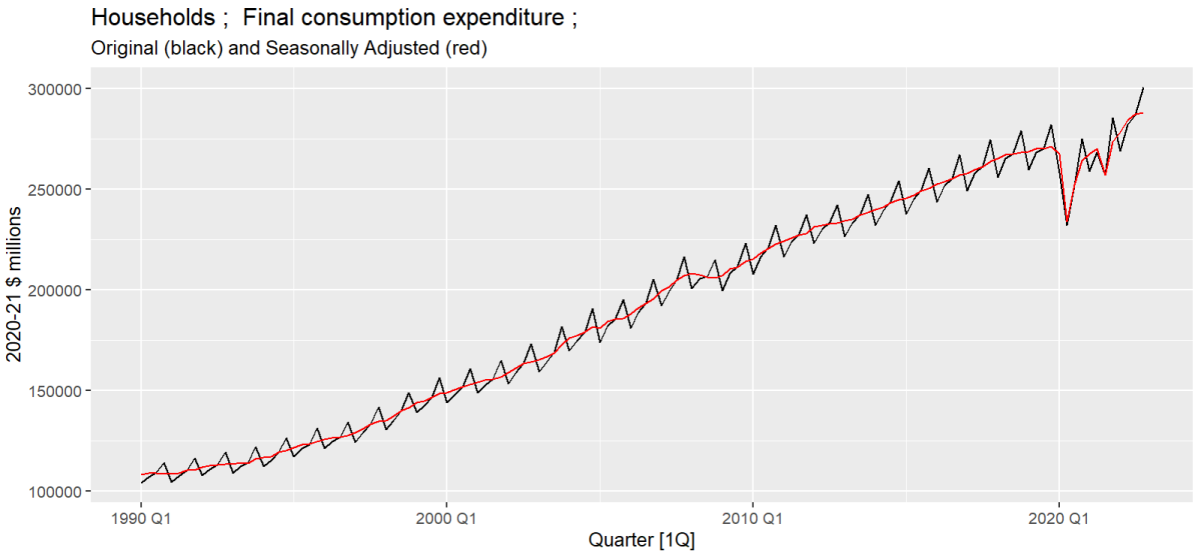
ECON2209 Assignment

Kah Hong Wan z5417138

Part 1: Data Exploration and Transformation

a. Based on the plot, discuss characteristics of each series.



The original series appears to feature a linear upwards trend in household consumption expenditure. The main exception to this trend occurs after around 2020 Q1 where there is a sharp drop in consumption expenditure. This drop can be attributed to a downward cycle caused by the Covid-19 pandemic which resulted in restrictions on household activity that reduced consumption opportunities. [1]

The original series also features a seasonal pattern which seems to be growing over time. The peaks look like they occur every year in Q4 while the troughs occur in Q1. The reason for this may be because household consumption expenditure denotes spending on household needs including food, clothing, transport, and leisure, all of which may be consumed more at the end of the year during the holiday period, hence the spike in expenditure in Q4. [2]

The seasonally adjusted series shares a similar upward trend as the original series but does not showcase the presence of a seasonal pattern in household consumption expenditure. Additionally, since the seasonally adjusted series does not feature seasonal patterns, the presence of a similar downward spike at around 2020 Q1 supports the argument that the spike is a cyclic occurrence influenced by the economic cycle.

[1] Bishop R, Boulter J and Rosewall T, *Tracking Consumption during the COVID-19 Pandemic*, RBA, March 2020.

https://www.rba.gov.au/publications/bulletin/2022/mar/tracking-consumption-during-the-covid-19-

pandemic.html#:~:text=Changes%20in%20economic%20activity%20during,household%20consumption%20(Graph%201).

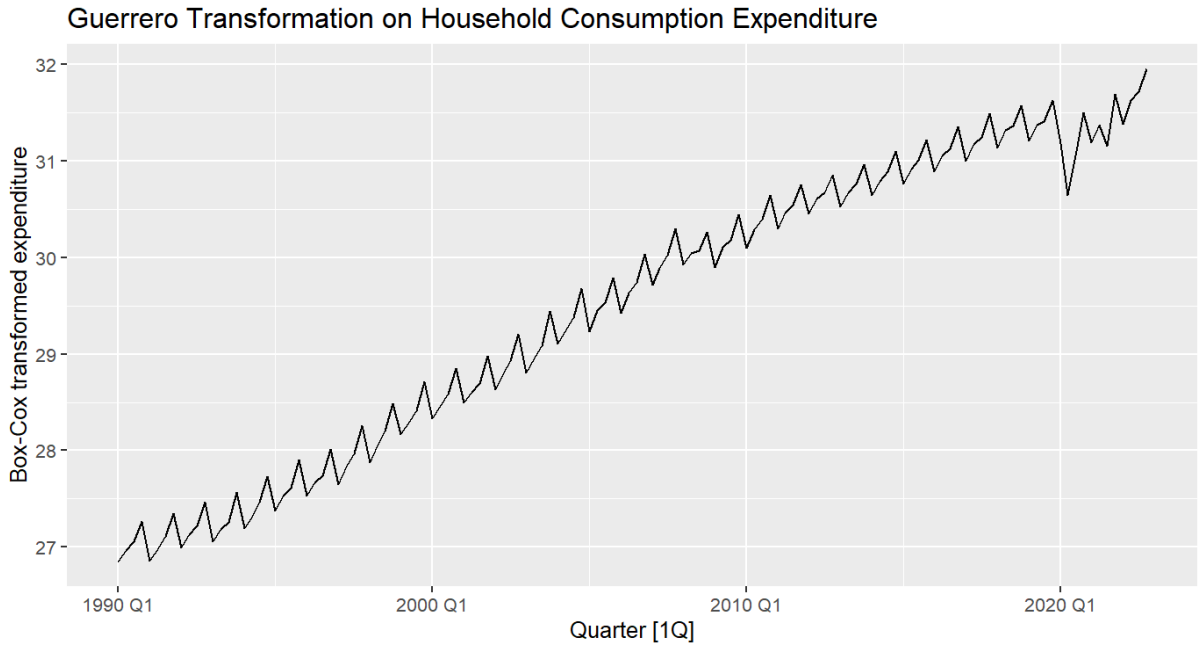
[2] *Household spending*, OECD.

https://data.oecd.org/hha/household-spending.htm#:~:text=Household%20spending%20is%20the%20amount,%2C%20leisure%2C%20and%20miscellaneous%20services.

b. What Box-Cox transformation, if any, would you select for your (non-seasonally adjusted) data? Explain.

As stated in Part 1 a), the presence of a seasonal pattern that seems to be growing over time indicates increasing variance. A Box-Cox transformation would be useful to stabilize the variance to ensure further statistical tests performed in the future are not influenced by variability.

```
> myseries %>%
+     features(value, features = guerrero)
# A tibble: 1 x 2
  series_id lambda_guerrero
  <chr>               <dbl>
1 A2302484C           0.130
> myseries %>% autoplot(box_cox(value, 0.13)) +
+     ggtitle("Guerrero Transformation on Household Consumption Expenditure") +
+     ylab("Box-Cox transformed expenditure")
```

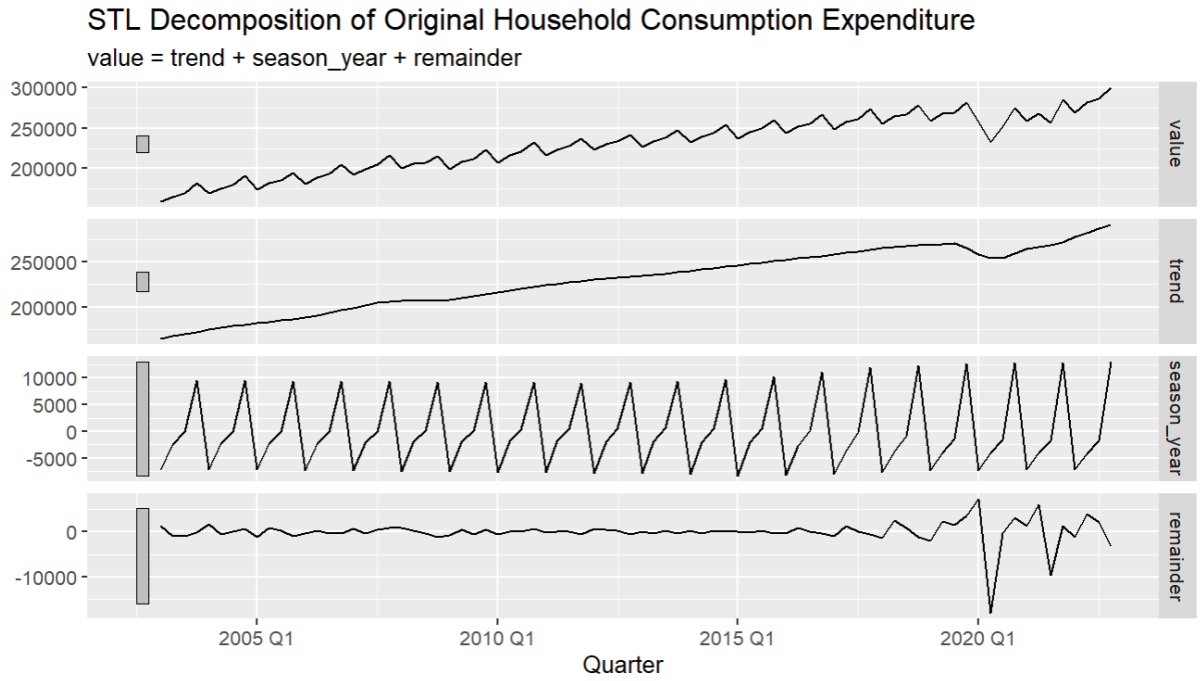


The Guerrero transformation method is used because R selects a lambda value that attempts to balance the seasonal fluctuations and random variation throughout the series. As can be seen in the transformed plot, the level of variation in the latter years of data appear to be at a much lower level in comparison to the original plot in Part 1 a).

Part 2: Time Series Decomposition

a) Consider the last twenty years of your untransformed data. Use an STL decomposition and produce a standard decomposition plot showing the trend-cycle, seasonal and remainder components. Discuss what you find from the decomposition plot.

```
> myseries_stl <- myseries %>%
+     filter(year(Quarter) > max(year(Quarter)) - 20) %>%
+     model(stl = STL(value))
> components(myseries_stl) %>% autoplot() + ggtitle("STL Decomposition
 of Original Household Consumption Expenditure")
```



The trend plot supports the observations made in Part 1 a) where throughout the series, there is a continuous upward trend in household consumption expenditure apart from around 2020 Q1 where there is a concave up curve due to an economic cycle.

The impact of the Covid-19 recession is further highlighted in the remainder plot, which showcases a relatively stationary and unvarying level of spending up until 2020 Q1 where there is a significant fall in consumption expenditure.

The remainder plot also allows us to notice that the Covid-19 recession had a relatively significant negative influence on the level of household consumption expenditure on two occasions, which is not easily observable when looking at the value plot. The first occurrence was between 2020 Q1-Q2 which coincides with the first positive case of Covid-19's L strain happening in January and the triggering of a nationwide lockdown in March. [3] The second occurrence in 2021 Q2-Q3 coincides with the first positive case of the Delta variant of Covid-19 in June which resulted in various lockdowns across all states and territories. [3]

Additionally, the trend plot also makes it easier to notice an instance where consumption expenditure seems to stagnate and flatten roughly around 2008-2009 which coincides with the Global Financial Crisis. However, the remainder plot shows no significant change during the period, indicating that the GFC was likely to have not had a substantial impact on Australian household's

consumption spending, an argument supported by Australia not experiencing a large economic downturn during the GFC. [4]

The seasonal plot shows that before 2015 Q1, there appears to be low levels of variance between each year's seasonal component. However, after 2015 Q1, the variance appears to increase over time. This is caused by a general rise in the level of Australian consumer spending. [5]

[3] Australian Bureau of Statistics 2022, *Effects of COVID-19 strains on the Australian economy*, ABS, June 2022.

https://www.abs.gov.au/articles/effects-covid-19-strains-australian-economy.

[4] *The Global Financial Crisis*, RBA.

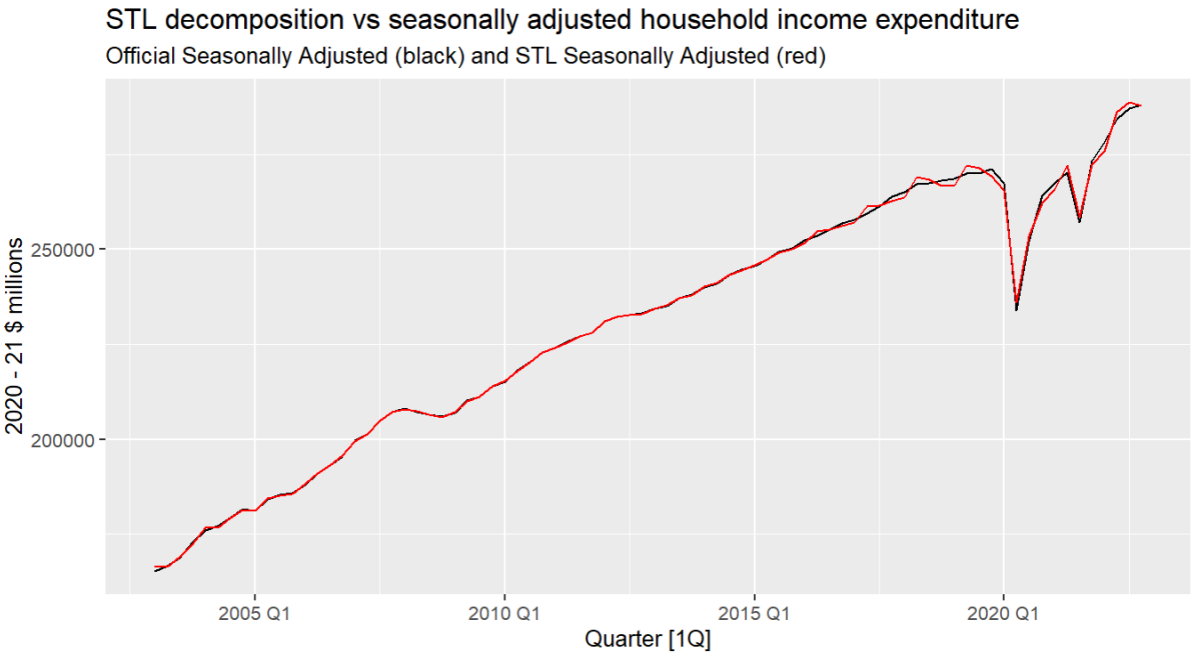
https://www.rba.gov.au/education/resources/explainers/the-global-financial-crisis.html.

[5] *Australian Consumer Spending 1960-2023*, Macrotrends.

https://www.macrotrends.net/countries/AUS/australia/consumer-spending.

b. Then plot your seasonally adjusted data from the STL decomposition together with the official seasonally adjusted data for the last twenty years. What observations can you make about the respective series?

```
> myseries_sa_20 <- myseries_sa %>%
+     filter(year(Quarter) > max(year(Quarter)) - 20)
> myseries_sa_20 %>%
+     autoplot(value) +
+     autolayer(components(myseries_stl), season_adjust, color='re
d') +
+     labs(y = "2020 - 21 $ millions",
+          title = "STL decomposition vs seasonally adjusted househ
old income expenditure",
+          subtitle = "Official Seasonally Adjusted (black) and STL
 Seasonally Adjusted (red)")
```



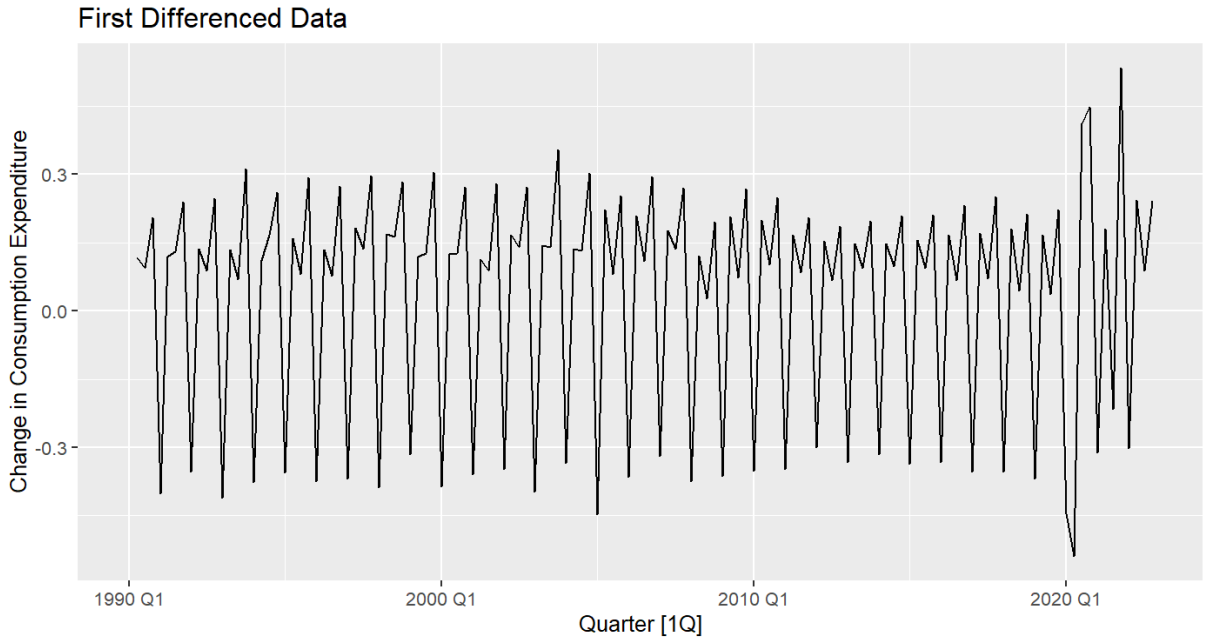
The STL decomposition appears to perform extremely well to adjust the seasonality of the actual data. This is especially the case prior to 2015 Q1. After this point, the STL decomposition weakens in accuracy very slightly but still manages to produce a graph that follows the actual seasonally adjusted data closely. The STL decomposition also manages to account for cyclic factors, evident by the STL plot following the actual seasonally adjusted data closely when the latter drops sharply in 2020 Q2.

Part 3: ARIMA Modelling

a. Using the visual inspection of plots, find the appropriate order of differencing to obtain stationary data. Explain your choices, step-by-step.

Firstly, we can investigate whether seasonal differencing is required to make the data stationary by observing the plots when taking the first difference of the data.

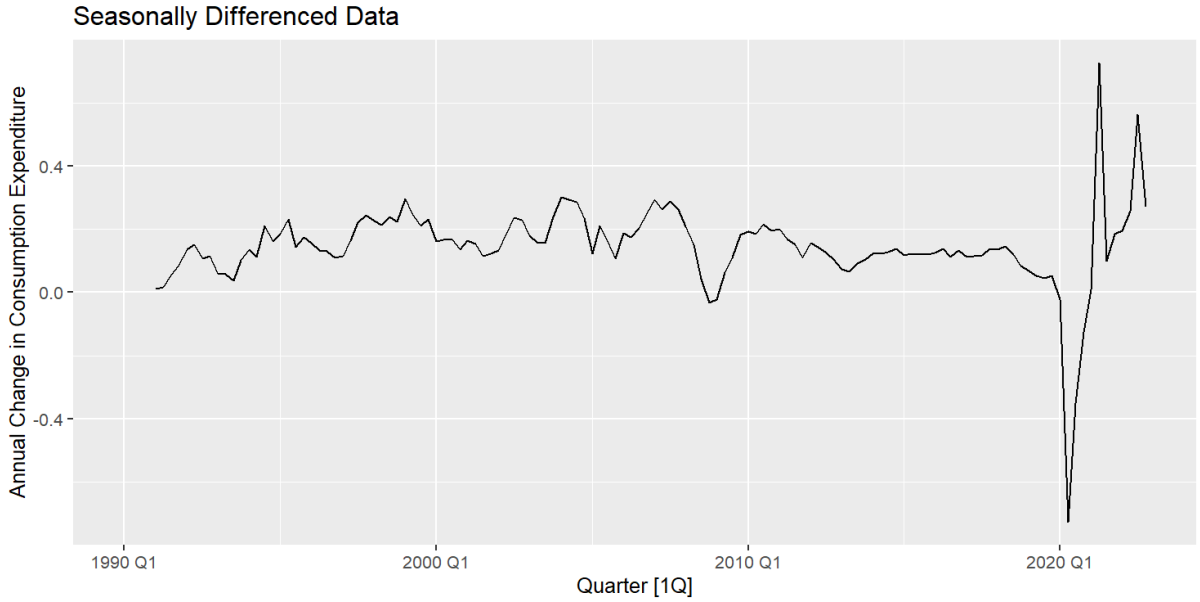
```
> myseries %>% autoplot(
+     box_cox(value, 0.13) %>% difference()
+ ) + labs(
+     title = "First Differenced Data",
+     y = "Change in Consumption Expenditure"
+ )
```



The result of the first difference plot is a graph that features obvious seasonality. Up until after 2020 Q1, there is an observable pattern where three consecutive quarters will have a change greater than zero before the final vertex falls to a level less than zero. Since stationary data cannot have predictable patterns, we can determine that a seasonal difference is likely to be required to handle the seasonality of the data.

Instead of adding a seasonal difference onto the first difference, we first try applying only a seasonal difference to the Box-Cox transformed series to determine whether we only need a seasonal difference.

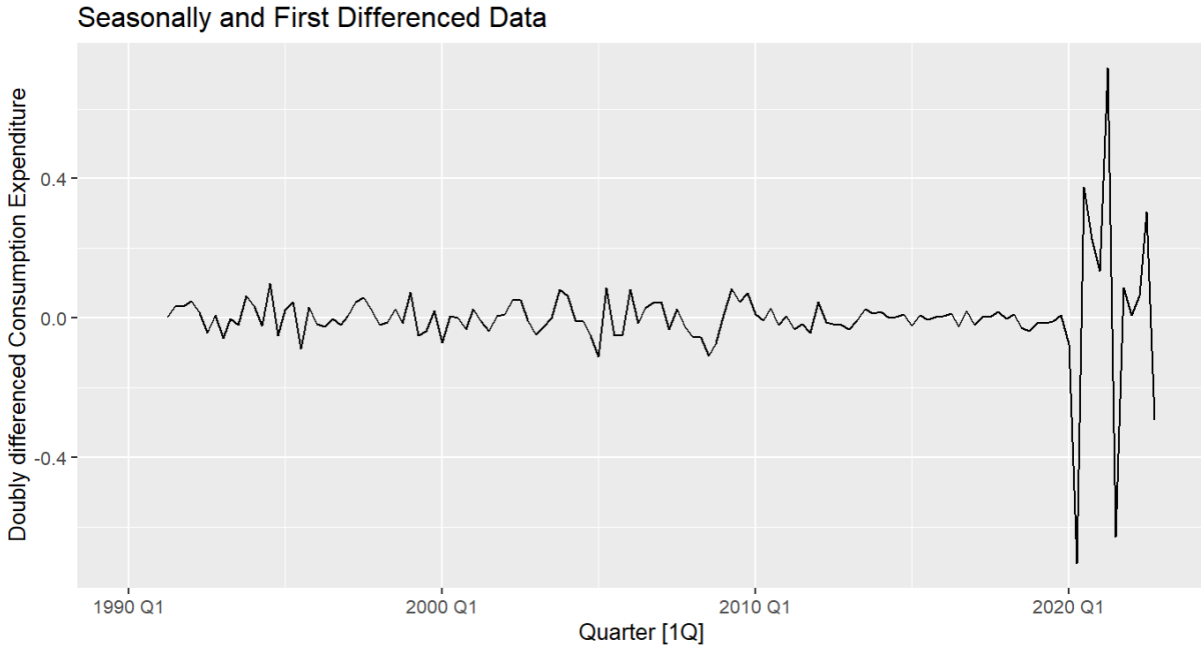
```
> myseries %>% autoplot(
+     box_cox(value, 0.13) %>% difference(4)
+ ) + labs(
+     title = "Seasonally Differenced Data",
+     y = "Annual Change in Consumption Expenditure"
+ )
```



The seasonally differenced data seems to have successfully eliminated the predictability of the data. Although the plot at this point does appear to be relatively horizontal, based on visual inspection alone, we cannot confirm this.

Adding a first difference may help in checking whether the plot is horizontal.

```
> myseries %>% autoplot(
+     box_cox(value, 0.13) %>% difference(4) %>% difference()
+ ) + labs(
+     title = "Seasonally and First Differenced Data",
+     y = "Doubly differenced Consumption Expenditure"
+ )
```



The doubly differenced data appears to lack any form of predictable patterns. Additionally, having the vertices “hover” around 0 allows us to observe that the data looks horizontal. Note that when both seasonal and first differencing are applied, the order of differencing does not matter. Therefore, by visual inspection, a seasonal difference followed by a first difference seems to be an appropriate order of differencing to make the data stationary.

b. Use statistical tests to check your choices in part a.

The first test we can use is the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test.

```
> myseries %>%
+     mutate(diff_value = difference(difference(box_cox
(value, 0.13)), 4)) %>%
+     features(diff_value, unitroot_kpss)
# A tibble: 1 × 3
  series_id kpss_stat kpss_pvalue
  <chr>          <dbl>       <dbl>
1 A2302484C     0.0350         0.1
```

This test involves a null hypothesis that the data is stationary. Since the kpss_stat is smaller than the 1% critical value and the kpss_pvalue is 0.1, which is greater than 0.05, we can determine that the differenced data created by the order of differencing in Part 3 a) is indeed stationary.

We can also allow KPSS to inform us about the appropriate number of first differences and seasonal differences required to make the data stationary using nsdiffs and ndiffs.

```
> myseries %>%
+     mutate(boxc_value = (box_cox(value, 0.13))) %>%
+     features(boxc_value, unitroot_nsdiffs)
# A tibble: 1 × 2
  series_id nsdiffs
  <chr>       <int>
1 A2302484C       1
> myseries %>%
+     mutate(d_boxc_value = difference(box_cox(value, 0.13)), 4) %>%
+     features(d_boxc_value, unitroot_ndiffs)
# A tibble: 1 × 2
  series_id ndiffs
  <chr>       <int>
1 A2302484C       0
```

Checking unitroot_nsdiffs determined that 1 seasonal difference was required. Applying unitroot_ndiffs to the seasonally differenced data resulted in an output of 0. This indicates that we were not required to take the first difference to make the data stationary, only a single seasonal difference was needed.

Therefore, although a potentially more appropriate order of differencing involving only a single seasonal difference could have been used. The order of differencing selected in Part 3 a) was still successful in making the data stationary.

c. Select an appropriate ARIMA model. Explain your choice and report the results.

We can allow R to select an appropriate ARIMA model for us.

```
> fit <- myseries %>%
+     model(ARIMA(box_cox(value, 0.13)))
> report(fit)
Series: value
Model: ARIMA(1,0,1)(2,1,1)[4] w/ drift
Transformation: box_cox(value, 0.13)

Coefficients:
         ar1      ma1     sar1     sar2     sma1  constant
      0.9421  -0.2292  -0.3792  -0.4394  -0.5358    0.0143
s.e.  0.0387   0.1191   0.1898   0.1709   0.1881    0.0026

sigma^2 estimated as 0.007537:  log likelihood=131.78
AIC=-249.56   AICc=-248.63   BIC=-229.6
```

The selection made by R seems to be an appropriate model since it features a non-seasonal difference of 0 and a seasonal difference of 1 which were the two characteristics determined in Part 3 b) to be needed to ensure the stationarity. We can try some other potential models by experimenting with the orders.

```
> fit <- myseries %>% model(
+     arima100211 = ARIMA(box_cox(value, 0.13) ~ 1 + pdq(1,0,0) + PDQ(2,1,1)),
+     arima102211 = ARIMA(box_cox(value, 0.13) ~ 1 + pdq(1,0,2) + PDQ(2,1,1)),
+     arima201211 = ARIMA(box_cox(value, 0.13) ~ 1 + pdq(2,0,1) + PDQ(2,1,1)),
+     arima101211 = ARIMA(box_cox(value, 0.13) ~ 1 + pdq(1,0,1) + PDQ(2,1,1)),
+     arima202211 = ARIMA(box_cox(value, 0.13) ~ 1 + pdq(2,0,2) + PDQ(2,1,1)),
+     arima101111 = ARIMA(box_cox(value, 0.13) ~ 1 + pdq(1,0,1) + PDQ(1,1,1)),
+     arima101212 = ARIMA(box_cox(value, 0.13) ~ 1 + pdq(1,0,1) + PDQ(2,1,2)),
+     arima101111 = ARIMA(box_cox(value, 0.13) ~ 1 + pdq(1,0,1) + PDQ(1,1,1)),
+     arima201213 = ARIMA(box_cox(value, 0.13) ~ 1 + pdq(2,0,1) + PDQ(2,1,3)),
+ )
> fit %>%
+     glance() %>%
+     arrange(AICc) %>%
+     select(.model, AICc)
# A tibble: 8 × 2
  .model       AICc
  <chr>       <dbl>
1 arima202211 -252.
2 arima101211 -249.
3 arima102211 -249.
4 arima101111 -248.
5 arima201211 -248.
6 arima100211 -247.
7 arima101212 -247.
8 arima201213 -244.
```

Based on these results, there appear to be better models with lower AICc values. Although there can potentially be even better models, we will select the arima202211 model which appears to have the lowest AICc in the sample.

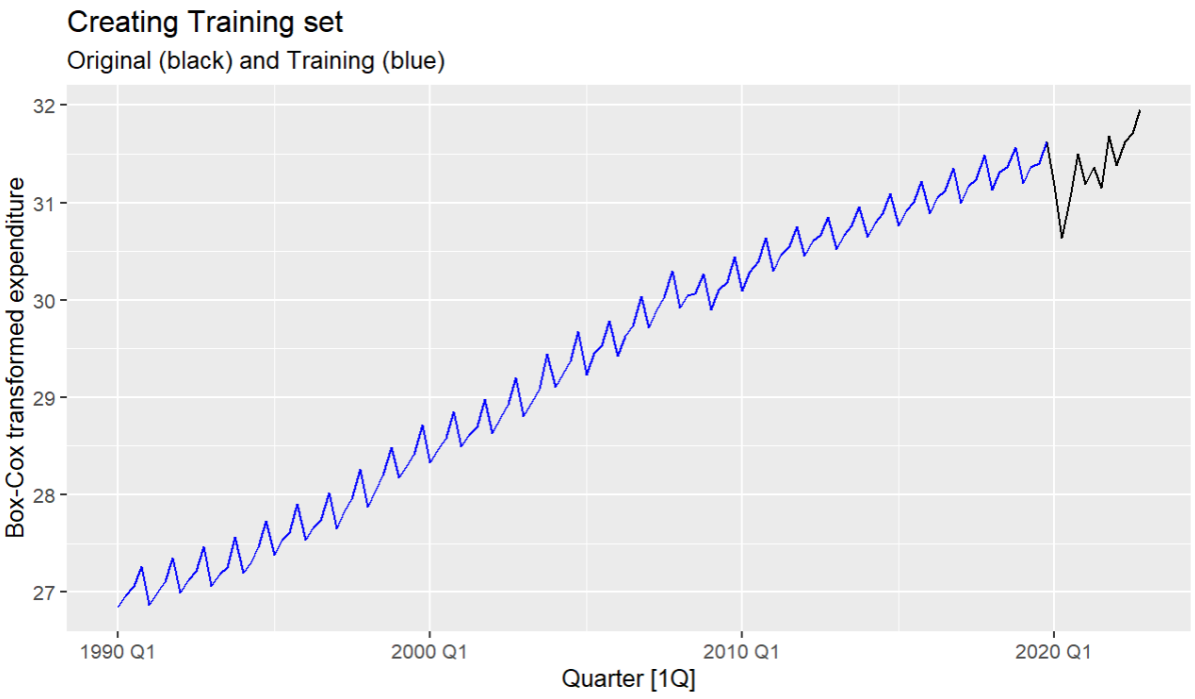
```
> fit <- myseries %>% model(
+     arima202211 = ARIMA(box_cox(value, 0.13) ~ 1 + pdq(2,0,2) + PDQ(2,1,1))
+ )
> report(fit)
Series: value
Model: ARIMA(2,0,2)(2,1,1)[4] w/ drift
Transformation: box_cox(value, 0.13)

Coefficients:
         ar1     ar2     ma1      ma2     sar1     sar2     sma1  constant
      0.1734  0.7262  0.6162  -0.3285  -0.3845  -0.4726  -0.4643    0.0252
s.e.  0.1238  0.1011  0.1524   0.1150   0.1921   0.1596   0.1963    0.0048

sigma^2 estimated as 0.007111:  log likelihood=135.72
AIC=-253.44   AICc=-251.92   BIC=-227.78
```

d. Create a training dataset (myseries_train) consisting of observations before 2020. Check that that your data have been split appropriately by producing a plot of myseries_train and myseries in one figure.

```
> myseries_train <- myseries %>%
+     mutate(boxd_value = box_cox(value, 0.13)) %>%
+     filter(year(Quarter) < "2020")
> myseries %>% autoplot(box_cox(value, 0.13)) +
+     autolayer(myseries_train, boxd_value, color='blue') +
+     labs(
+         y = "Box-Cox transformed expenditure",
+         title = "Creating Training set",
+         subtitle = "Original (black) and Training (blue)"
+     )
```



As can be observed from the plot, the training set only includes data up until 2019 Q4. This means that data between 2019 Q4 to 2020 Q1 which includes January, February and March of 2020 are not included. Additionally note that from now on, the myseries_train dataset has a new column titled 'boxd_value' which will be used for the following questions to avoid needing to repetitively transform the 'value' dataset with 'box_cox(value, 0.13)'.

e. Using the training data set, consider the following models:
   • The ARIMA model you selected in part c.
   • An STL decomposition, followed by an ARIMA model on the seasonally adjusted data; that is, an STL-ARIMA model.
   • An ETS model chosen automatically.
   Using the test data set, plot the forecasts from all three models on the same figure along with the actual data from 2005 onwards. Include the prediction intervals and discuss the relative performance of the models based on the figure, and on the RMSE and MAPE.

First, we create the three models.

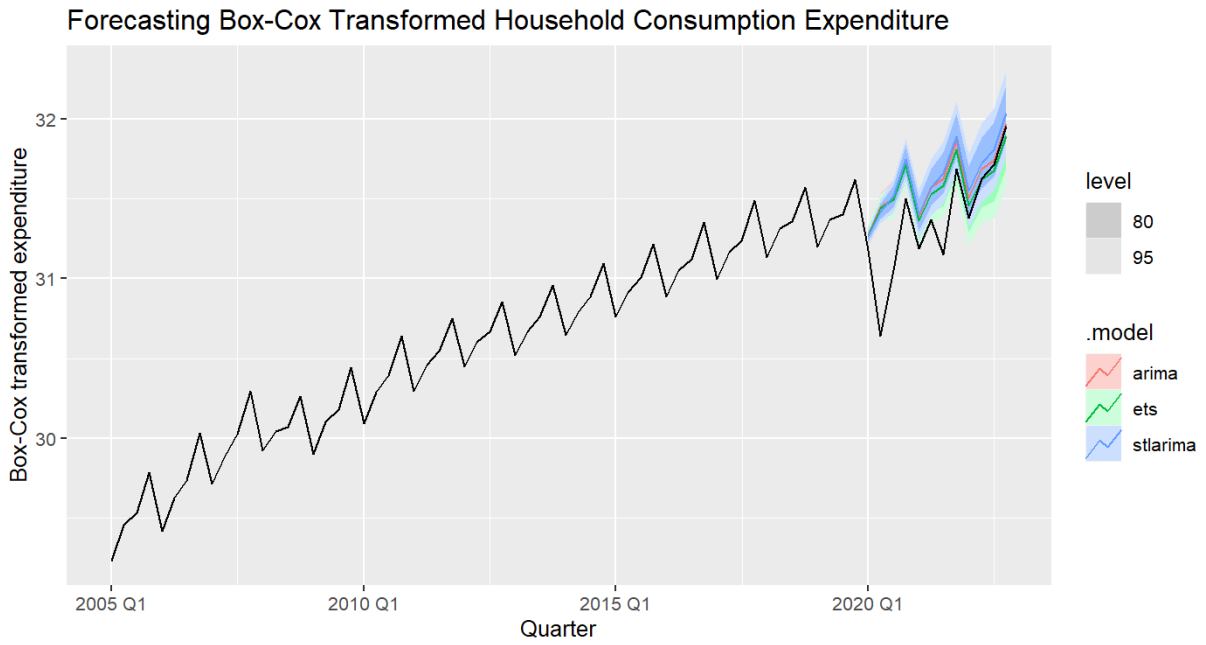
```
> three_models <- myseries_train %>%  model(
+     arima = ARIMA(boxd_value ~ 1 + pdq(2,0,2) + PDQ(2,1,1)),
+     stlarima = decomposition_model(
+         STL(boxd_value),
+         ARIMA(season_adjust)
+     ),
+     ets = ETS(boxd_value)
+ )
```

Then we create the test data set which will be used as the "actual data".

```
> myseries_boxd_2005onwards <- myseries %>%
+     mutate(boxd_value = box_cox(value, 0.13)) %>%
+     filter(year(Quarter) >= "2004 Q4")
```

Finally, we can plot the forecasts.

```
> three_models %>% forecast(h = 12) %>%
+     autoplot(myseries_boxd_2005onwards) +
+     labs(
+         title = "Forecasting Box-Cox Transformed Hous
ehold Consumption Expenditure",
+         y = "Box-Cox transformed expenditure"
+     )
```



Although it is difficult to observe characteristics of each individual model without zooming in, comparing the models as a collective to the actual data helps to determine that all three models were unable to predict the cyclic fall in household consumption expenditure from 2020 Q1 to around 2021 Q4. This is largely due to the fact that each of the models make forecasts based off past data, which up until 2020 Q1 seemed to be predominantly seasonal and upward trending.

Each of the models predicted that the seasonality of the dataset and the upward trend would continue, which in reality did not occur because of Covid-19.

Nevertheless, after around 2021 Q4, all three of the models seem to become more accurate as the actual plot recovers from the downward cycle and regains its seasonal and upward trending characteristics.

Since the forecast is too small to compare each model to each other, we can use accuracy tests to determine which of the models had the smallest average difference between its own forecast and the actual plot.

```
> three_fc <- three_models %>%
+     forecast(h = 12)
> accuracy(three_fc, myseries_boxd_2005onwards)
# A tibble: 3 × 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 arima    A2302484C Test  -0.240 0.327 0.240 -0.773 0.773  1.75  2.17 0.217
2 ets      A2302484C Test  -0.201 0.310 0.218 -0.649 0.701  1.58  2.06 0.304
3 stlarima A2302484C Test  -0.263 0.334 0.263 -0.844 0.844  1.91  2.21 0.124
```

The accuracy test shows that out of the three models, the ETS model had both the lowest RMSE and MAPE. This indicates that the ETS model had the lowest average difference and hence was the most accurate model out of the three. Nevertheless, all three models generally made inaccurate forecasts because of their inability to account for cyclic changes.

f. Propose and implement your own choice of alternative model. Discuss your choice and its performance relative to the best model from part e.

Since the main issue with the three models from Part 3 e) was an inability to account for the downward cycle that occurred after 2020 Q1, which is a time period outside the training dataset meaning the models made forecasts with no knowledge of the cyclic occurrence, a potential alternative model could be a model that disregards the seasonality and trend present in most of the plot.

An example of this could be a simple naïve model, which would forecast with no seasonality and upward trend, potentially meaning it could account for the downward cycle by not forecasting an increase in household consumption expenditure.

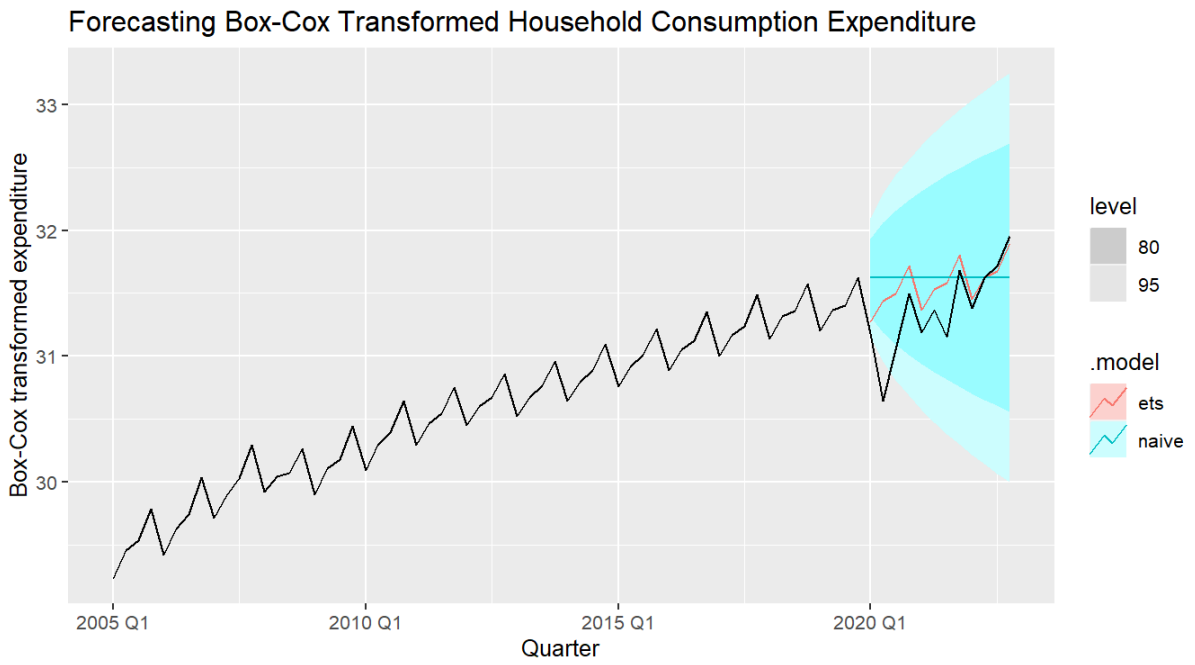
Creating a model including both the ETS model that was the best in Part 3 e) and a NAÏVE model, then checking its forecast accuracy gives:

```
> best_vs_alt_model <- myseries_train %>% model(
+     ets = ETS(boxd_value),
+     naive = NAIVE(boxd_value)
+ )

> best_vs_alt_fc <- best_vs_alt_model %>%
+     forecast(h = 12)
> accuracy(best_vs_alt_fc, myseries_boxd_2005onwards)
# A tibble: 2 × 11
  .model serie…¹ .type     ME   RMSE    MAE     MPE  MAPE  MASE
  <chr>  <chr>   <chr>  <dbl>  <dbl>  <dbl>   <dbl> <dbl> <dbl>
1 ets    A23024… Test  -0.201 0.310  0.218 -0.649  0.701  1.58
2 naive  A23024… Test  -0.254 0.425  0.335 -0.823  1.07   2.43
```

The accuracy test still deems the ETS model to be the most accurate model with the lowest RMSE and MAPE. Plotting the forecasts reveals a potential reason why this is the case.

```
> best_vs_alt_model %>% forecast(h = 12) %>%
+     autoplot(myseries_boxd_2005onwards) +
+     labs(
+         title = "Forecasting Box-Cox Transformed Household Co
nsumption Expenditure",
+         y = "Box-Cox transformed expenditure"
+     )
```



The high RMSE and MASE values for the NAÏVE model, indicating a high difference between the model's plot and the actual plot, may have been lower if the last observed value was a trough instead of a peak.

This is because the NAÏVE model forecasts equal to the last observed value which in the training data set is 2019 Q4 which is a peak in the seasonal pattern of the graph. If the last observed value were a trough, such as 2020 Q1, then the forecast would be a horizontal line that intersects the plot of the actual data many times, potentially resulting in a forecast that would account for the downward cycle caused by Covid-19.

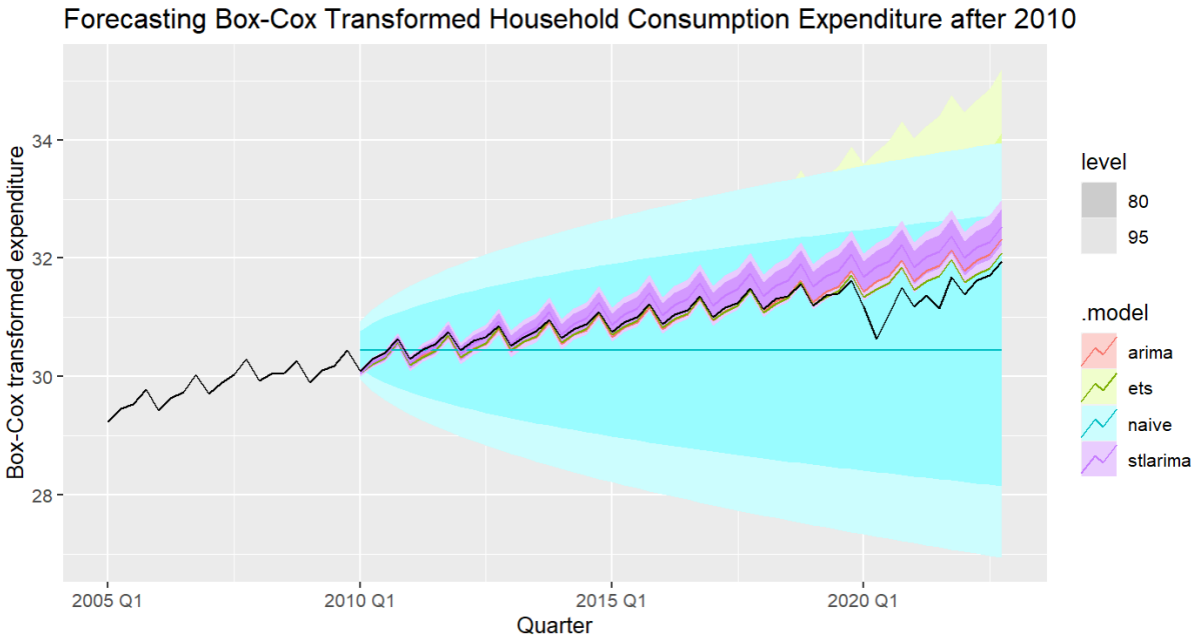
However, because the training data set does not end with a seasonal trough, meaning the horizontal forecast lies for the majority of the test dataset (after 2020 Q1) above the ETS forecast, there is no way for the NAÏVE model to be more accurate than the ETS model.

g. Now create a new training dataset (myseries_trnew) consisting of observations before 2010 and repeat the analysis in parts d. to f., plotting the forecasts from all four models on the same figure along with the actual data from 2005 onwards. Include the prediction intervals. What observations do you have about the sensitivity of the accuracy measures and forecasts to the amount of training data used?

```
> myseries_trnew <- myseries %>%
+     mutate(boxd_value = box_cox(value, 0.13)) %>%
+     filter(year(Quarter) < 2010)
```

```
> four_models <- myseries_trnew %>% model(
+     arima = ARIMA(boxd_value ~ 1 + pdq(2,0,2) + PDQ(2,1,1)),
+     stlarima = decomposition_model(
+         STL(boxd_value),
+         ARIMA(season_adjust)
+     ),
+     ets = ETS(boxd_value),
+     naive = NAIVE(boxd_value)
+ )
```

```
> four_models %>% forecast(h = 52) %>%
+     autoplot(myseries_boxd_2005onwards) +
+     labs(
+         title = "Forecasting Box-Cox Transformed Household Consumption Expen
diture after 2010",
+         y = "Box-Cox transformed expenditure"
+     )
```



```
> four_fc <- four_models %>%
+     forecast(h = 52)
> accuracy(four_fc, myseries_boxd_2005onwards)
# A tibble: 4 × 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 arima    A2302484C Test  -0.0718 0.260 0.169 -0.227   0.543 0.991  1.35 0.850
2 ets      A2302484C Test  -0.0311 0.193 0.121 -0.0976  0.390 0.710  1.00 0.769
3 naive    A2302484C Test   0.574  0.708 0.601  1.83    1.92  3.52   3.68 0.715
4 stlarima A2302484C Test  -0.262  0.393 0.279 -0.839   0.894 1.64   2.04 0.878
```

Firstly, notice that irregardless of whether the training dataset includes only 5 years or 15 years worth of data, the accuracy test results do not change in terms of determining the models with the lowest and highest RMSE and MAPE. In both the accuracy tests in this part and Part 3 e), the ETS model had the lowest RMSE and MAPE values, followed by the ARIMA model, then the STL-ARIMA model, and finally when taking into account the results from Part 3 f), the model with the largest difference between its forecast and the actual data was the NAÏVE model.

Secondly, after observing both forecast plots of this graph which had 5 years of training data to the graph in Part 3 e) which had 15 years, we can notice that because both training datasets include periods of consecutive yearly seasonality and continuous upward trends, the forecasts produced with both training datasets tend to predict the continuation of this seasonality and trend. Therefore, both forecasts irregardless of the amount of training data given, appear very similar to one another as they both forecast the continued seasonality and trend.

Hence, accuracy measures and forecasts are not sensitive to changes to the amount of training data used, as long as the actual data features seasonality or trend throughout.