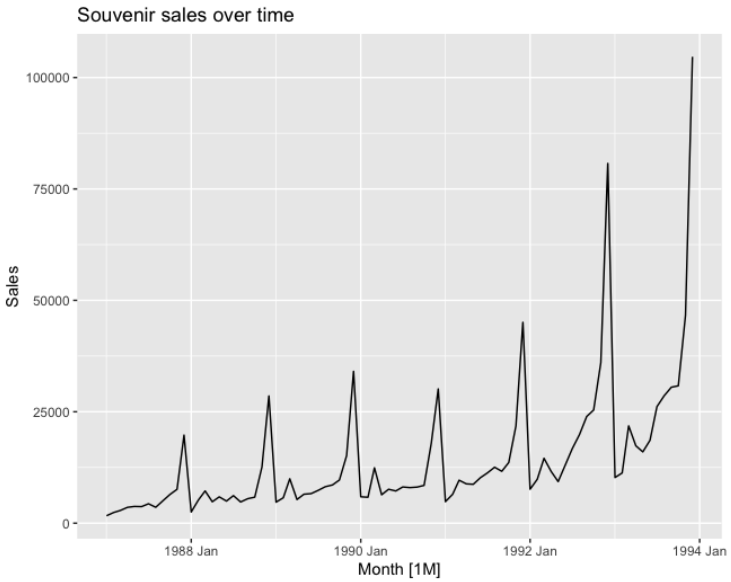
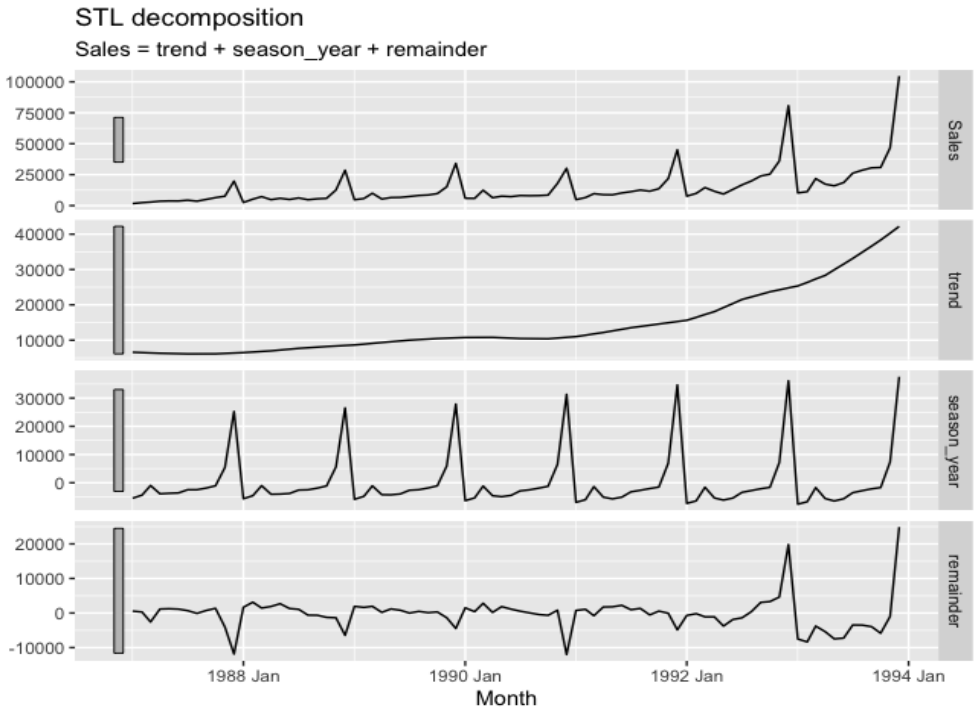
Resit – Assignment 2

Dragos Pop

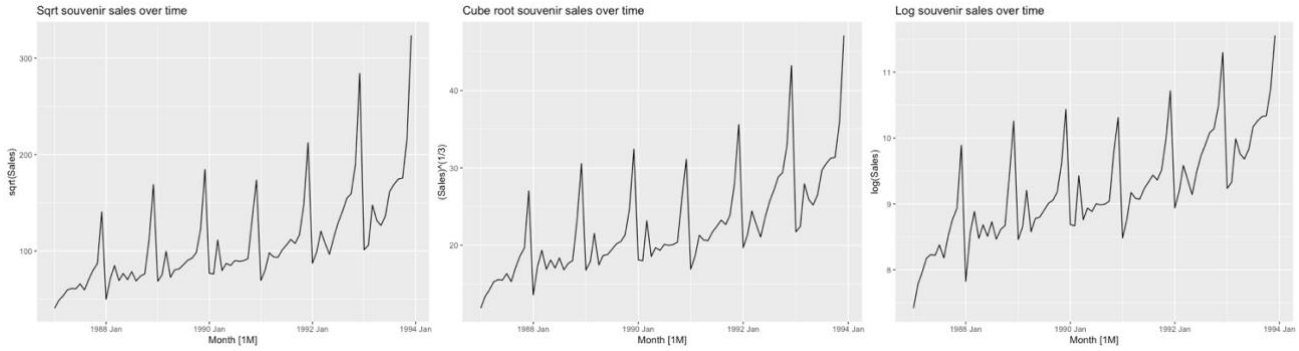
Exercise 1

1.1) In the plot on the right, one can observe the souvenir sales over time. The yearly seasonality is clear, with a high peak in December and a smaller one in March, coinciding with the Christmas and the surfing festival respectively, which are also visible in the STL decomposition picture below. Another clear pattern from the graph is the upward trend, which seems to be exponential. Although the number of sales does not increase drastically over the years in the "out-season" months, the peaks in December and March are getting higher every year, except the one from December 1990. One unexpected observation is that the sales in November are remarkably large. Considering there is no specific event, they are generally larger than in March, when the surfing competition takes place. Lastly, from 1992 onwards, the sales of souvenirs in the months before December increased visibly in contrast with the previous years.

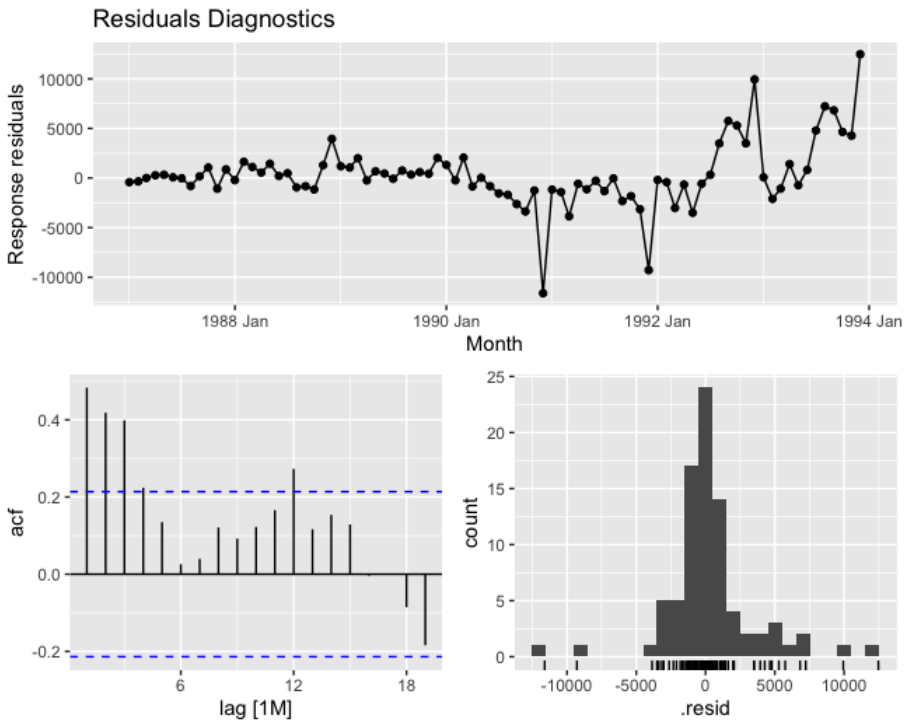




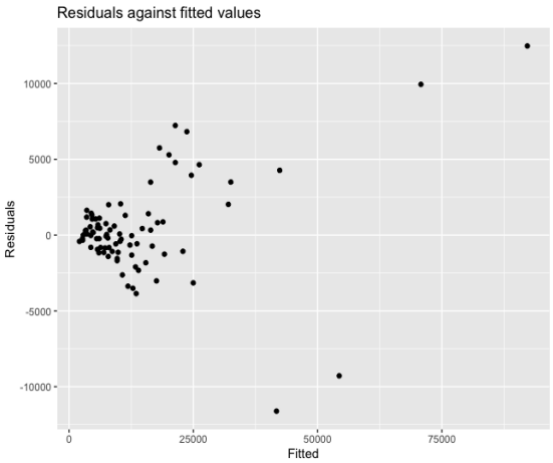
1.2) As expressed previously, the peaks are getting higher every year and the trend seems exponential, meaning the variation of the data increases with the level of series. Hence, a transformation is useful [1]. The reason why taking logarithms of the data instead of other transformation is first that they stabilize the variance better than the square or cube root, as shown below, and that they are interpretable, hence, the changes in a log value are relative changes on the original scale [1].

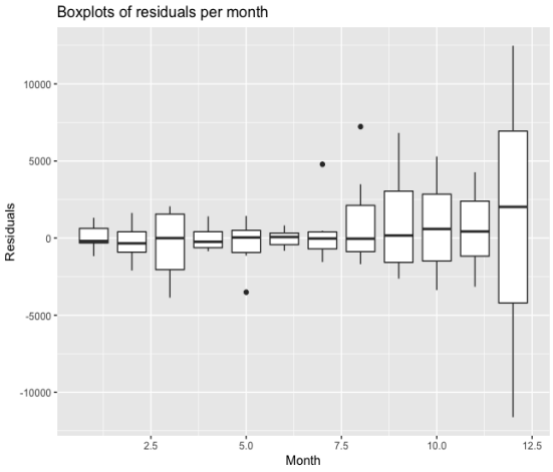


1.3) Initially, the "surf" dummy variable was attached to the data, taking a value of 1 for the rows corresponding to dates in March starting from 1988, and a value of 0 everywhere else. Then, the regression model was generated with the help of *TSLM()* function and the formula *log(Sales) ~ trend() + season() + surf*. The formula starts with the target variable, the logarithms of sales, followed by the trend variable. Next comes the seasonal dummy variables generated through the *season()* function. This creates 11 dummy features, avoiding the "dummy variable trap" [2]. Finally, there is "surf" dummy feature created earlier. Accordingly, after the model was fitted to the logarithms of the sales, the report was represented, which suggested approximately 95% of the variance is explained, since R^2 and Adjusted R^2 are around 0,95. Using the function *gg_tsresiduals()* with type parameter "response" ensured the residuals diagnostics are computed on the back-transformed data.



From the plot of residuals over time and the histogram above, one can see that the residuals have mean zero, thus the forecasts are not systematically biased, and that their variation changes drastically over time, making the prediction internal inaccurate. Additionally, their distribution seems to be right-skewed, hence, not normal. Moreover, the autocorrelation plot shows multiple significant spikes, at lags 1, 2, 3, 4, and 12, and the first three are particularly large. Subsequently, the model violates the assumption of no autocorrelation in the errors, and our forecasts may be inefficient — there is some information left over which should be accounted for in the model in order to obtain better forecasts [3]. In the picture from the right hand-side, the residuals are plotted against the fitted values. This displays fitted values converging towards 0, with 4 outliers, suggesting heteroscedasticity of the residuals. From the boxplots of residuals per month, it is clear that the residuals vary significantly in December, meaning, the model does not learn properly and makes erroneous predictions regarding the impact of the Christmas on the sales of souvenirs.

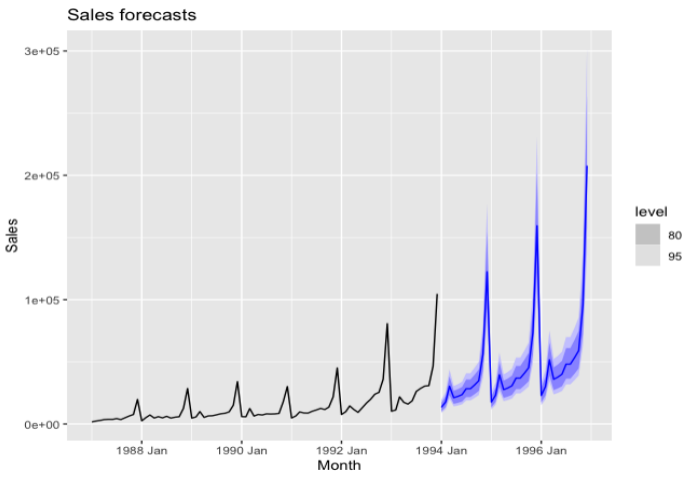




1.4) The Ljung Box test is a formal test for autocorrelation of the innovation residuals. In order to apply the test, one need to select k, the number of parameters, in our case 13, and l as two times the period of seasonality, 24 respectively [4]. The output of the test, as shown on the right, indicates a 0 p-value, significantly proving that the residuals are not white noise.

```
# A tibble: 1 x 3
  .model lb_stat lb_pvalue
  <chr>     <dbl>     <dbl>
1 reg       112.          0
```

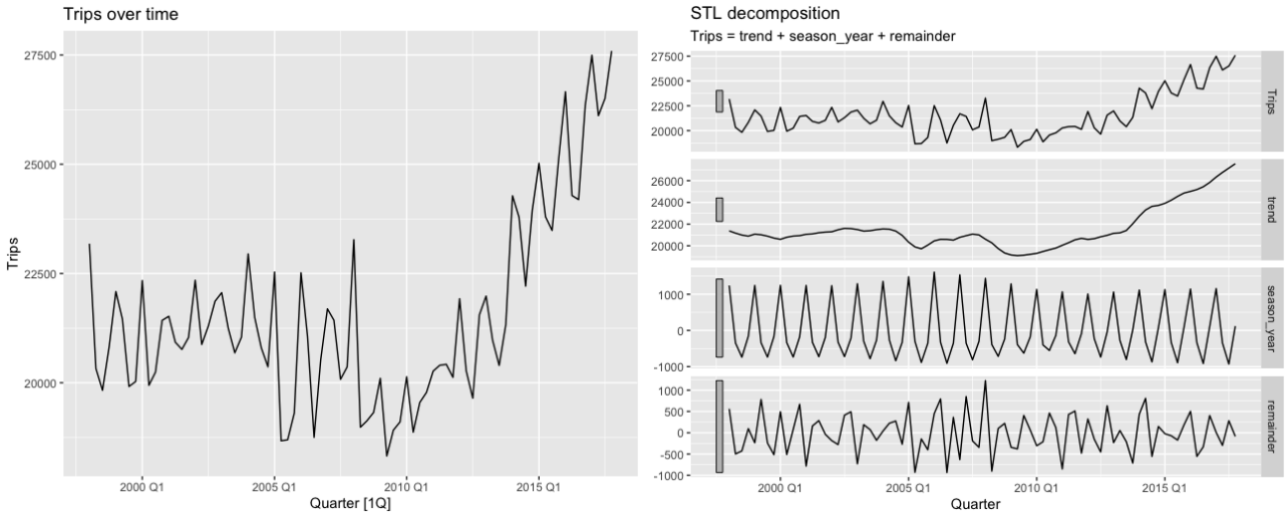
1.5) In order to make a prediction in the future, the dummy variable “surf” has to be manually imputed for the duration of the desired forecast, in this case three years, because the case at hand is an ex-post forecast, using later information on the predictors [5]. Then, the predictions can be generated. As shown in the plot, the forecasts respect the yearly seasonality of the data, however, the prediction intervals are large, which is a consequence of the fact that the errors of the models are autocorrelated.



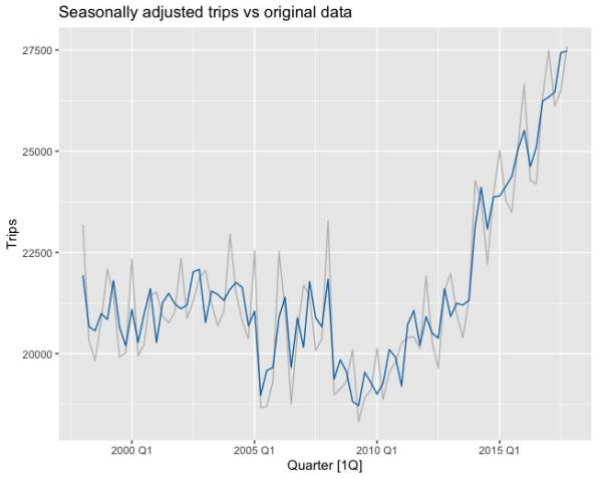
One possible solution that may lead to better predictions may be selecting predictors using backwards stepwise regression.

Exercise 2

2.1) In the picture below, one can see the evolution of overnight trips over time. One can observe the level of the data is quite constant up to 2010, after which it surges drastically. Another noticed feature of the graph is the seasonal pattern of the data with most trips in the first quarter and least in the third one. This is better visible in the third panel of the picture from the right, which was generated following an STL decomposition of the data.



Next, the components resulted from the STL decomposition, namely trend and reminder, were used to derive the seasonally adjusted data, and this was plotted with blue against the original data in grey. In other words, the seasonally adjusted data is composed by eliminating the seasonal component from the original data. This allows the following modeling of the data to address the overall trend of the data as opposed to placing interest on the seasonal variation. In the given case, seasonally adjusted data removes the effect that each season has on the number of trips.



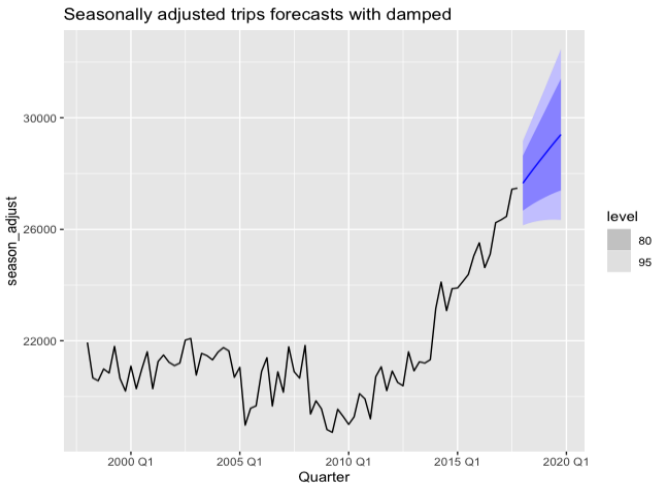
2.2) To fit the seasonally adjusted data to an additive damped trend model, the formula *damped = ETS(season_adjust ~ error("A") + trend("Ad") + season("N"))* was used. The damped part was considered in the "Ad" argument of trend, while the "N" for season suggests the time series has no seasonality, which is true because the data is seasonally adjusted. The report of the model, as well as the forecasts generated for the next two years can be seen below. When it comes to the predictions, these look good.

```
Series: season_adjust
Model: ETS(A,Ad,N)
  Smoothing parameters:
    alpha = 0.4785236
    beta  = 0.04836097
    phi   = 0.9799998

  Initial states:
     l[0]       b[0]
 21365.67 -9.032045

  sigma^2:  593579.7

     AIC     AICc      BIC
1420.913 1422.064 1435.205
```



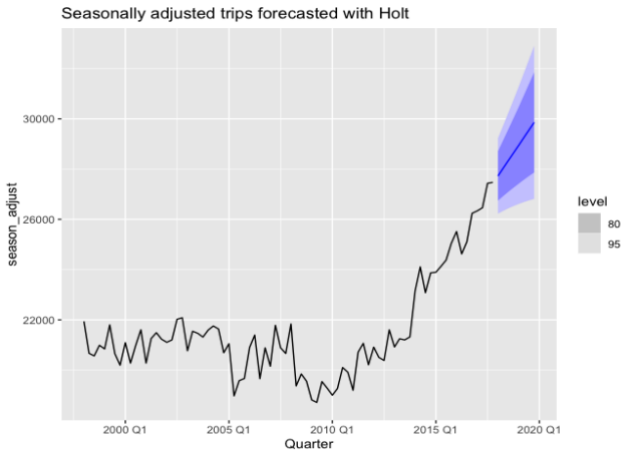
2.3) Applying the function *holt = ETS(season_adjust ~ error("A") + trend("A") + season("N"))* fits the seasonally adjusted data to a Holt's linear model with additive errors. The difference with the previous method is the parameter "A" in the trend, that signifies additive in this case and damped in the previous one. Similarly, the report and the forecasts using the new model are represented below and they look well.

```
Series: season_adjust
Model: ETS(A,A,N)
  Smoothing parameters:
    alpha = 0.4806411
    beta  = 0.04529462

  Initial states:
     l[0]     b[0]
 21366.07 1.279692

  sigma^2:  585061

     AIC     AICc      BIC
1418.816 1419.627 1430.727
```



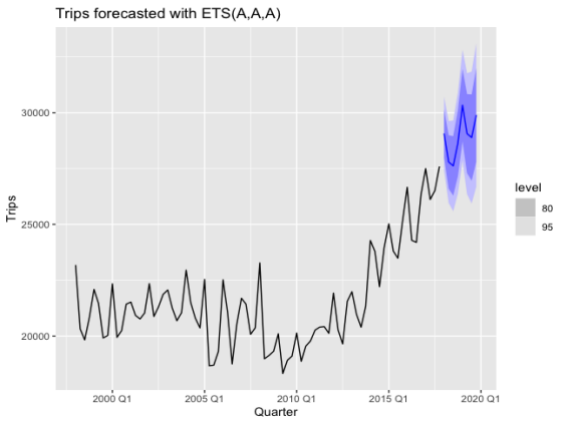
Next, the *ETS()* function is used to automatically select the best model from the original data. As one can expect given the seasonality of the data that does not change over time, the seasonal component is picked additive. The other two are selected additive as well, resulting in a ETS(A,A,A) model. It is visible in the graph that the model was able to learn the seasonality pattern well and make accurate predictions.

```
Series: Trips
Model: ETS(A,A,A)
  Smoothing parameters:
    alpha = 0.4495675
    beta  = 0.04450178
    gamma = 0.0001000075

  Initial states:
     l[0]      b[0]      s[0]     s[-1]     s[-2]    s[-3]
 21689.64 -58.46946 -125.8548 -816.3416 -324.5553 1266.752

  sigma^2:  699901.4

     AIC     AICc      BIC
1436.829 1439.400 1458.267
```

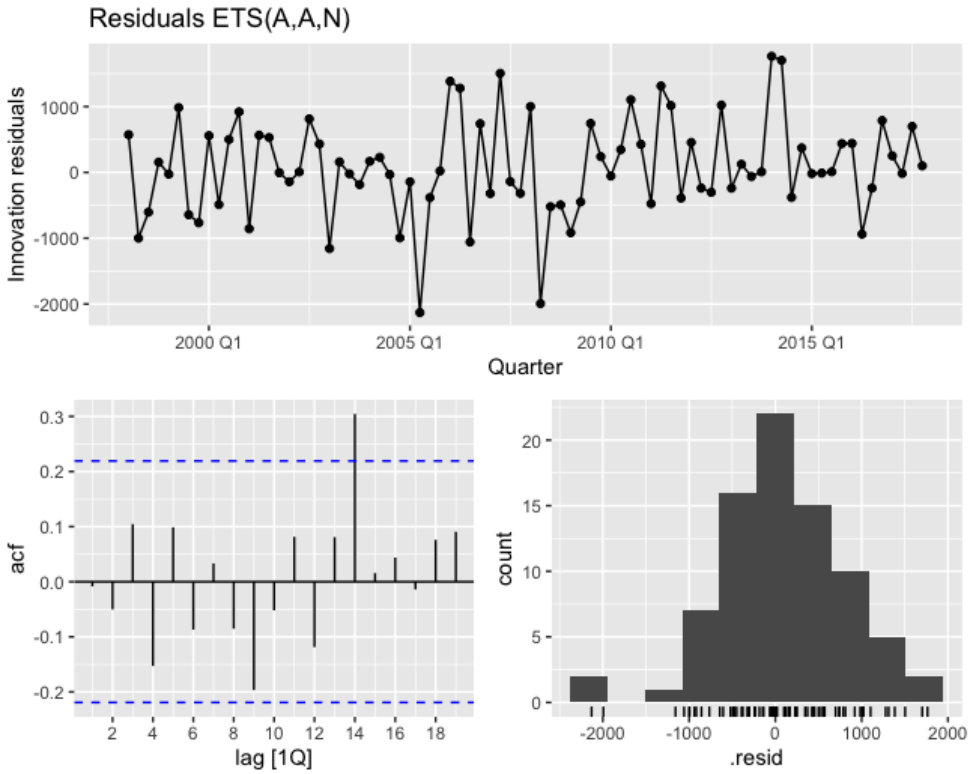


Visually, it is difficult to compare the three models, especially because the data is different in the last one. However, among the first two, the Holt's linear model without damped leads to a smaller prediction interval, hence, it is preferred.

To better assess the performance of the models, evaluation metrics were computed for each through the *accuracy()* function. Considering this, the best model is the second one, because it has the smallest errors, except the RMSE, which is the same for the damped model. Therefore, one can state that the second model makes the most reasonable predictions. On the other hand, it makes sense for the seasonal model to perform the worse as it has an extra task to learn the seasonal evolution of the data.

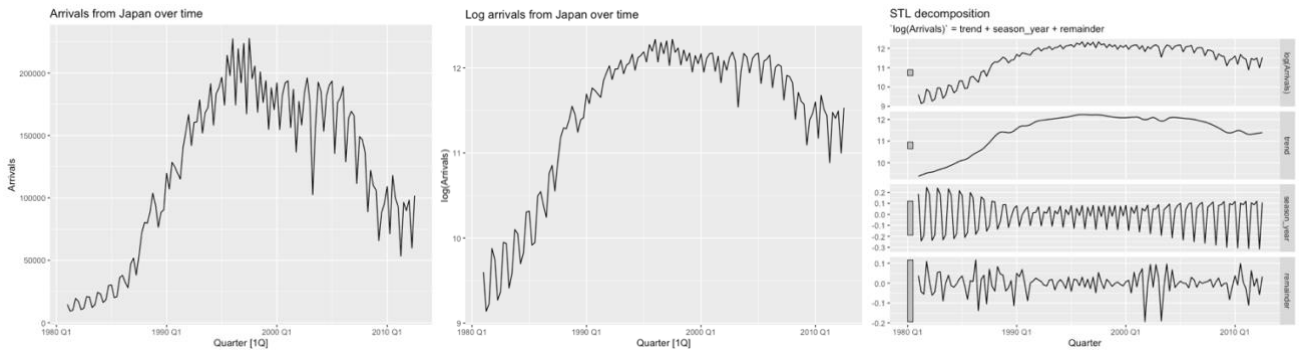
```
> fit %>% accuracy ()
# A tibble: 1 × 10
  .model .type       ME  RMSE   MAE   MPE  MAPE  MASE RMSSE     ACF1
  <chr>  <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>
1 damped Training  87.6  746.  566. 0.301  2.67 0.594 0.618 -0.00831
> fit2 %>% accuracy ()
# A tibble: 1 × 10
  .model .type       ME  RMSE   MAE   MPE  MAPE  MASE RMSSE     ACF1
  <chr>  <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>
1 holt   Training  83.7  746.  563. 0.289  2.66 0.590 0.617 -0.00833
> fit3 %>% accuracy ()
# A tibble: 1 × 10
  .model   .type       ME  RMSE   MAE   MPE  MAPE  MASE RMSSE     ACF1
  <chr>    <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>
1 seasonal Training  105.  794.  604. 0.379  2.86 0.636 0.653 -0.00151
```

2.4) The picture below displays the residuals of the Holt model with additive errors. These look roughly normally distributed, slightly right-skewed, with a single significant spike at lag 14. To make sure the residuals are indeed white noise, the Ljung Box test was applied, selecting k equals 0 (number of parameters) and l equals 8 (2 * period of seasonality) [4]. The result is not significant (i.e., the p-value = 0,7 is relatively large). Thus, one can conclude that the residuals are not distinguishable from a white noise series.



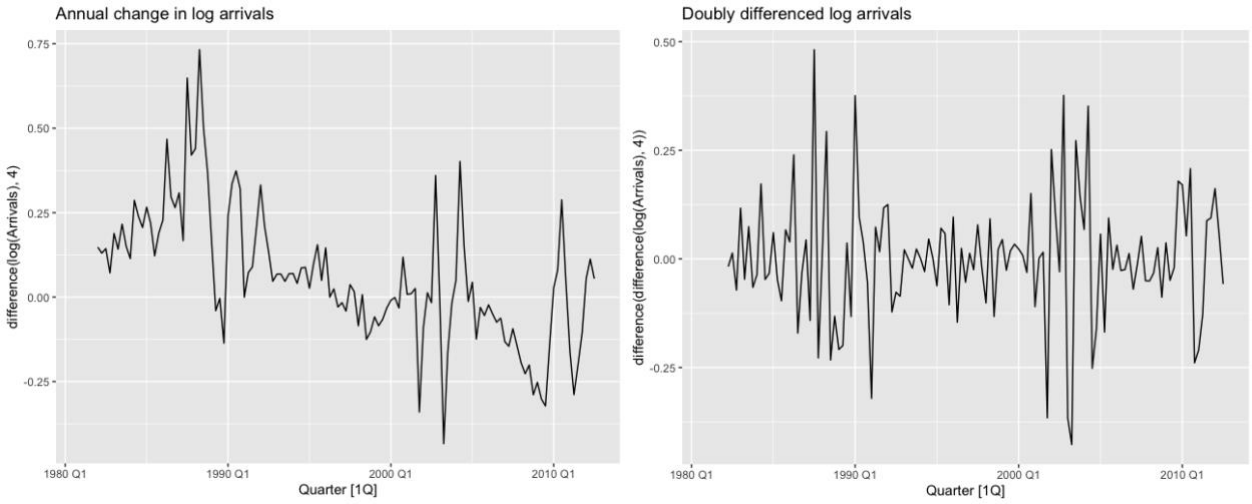
Exercise 3

3.1) At the beginning, the data is plotted. The first picture shows that the variation of the time series increases over time, therefore, a log transformation was applied. The data then shows changing levels, thus, using differencing is the next step to stabilize the mean. However, it is unclear whether the data is seasonal. For this reason, an STL decomposition was performed next, which indicates, as seen in the third panel of the third picture, that the elements nearby are similar even though the ones far apart are not, implying seasonality.

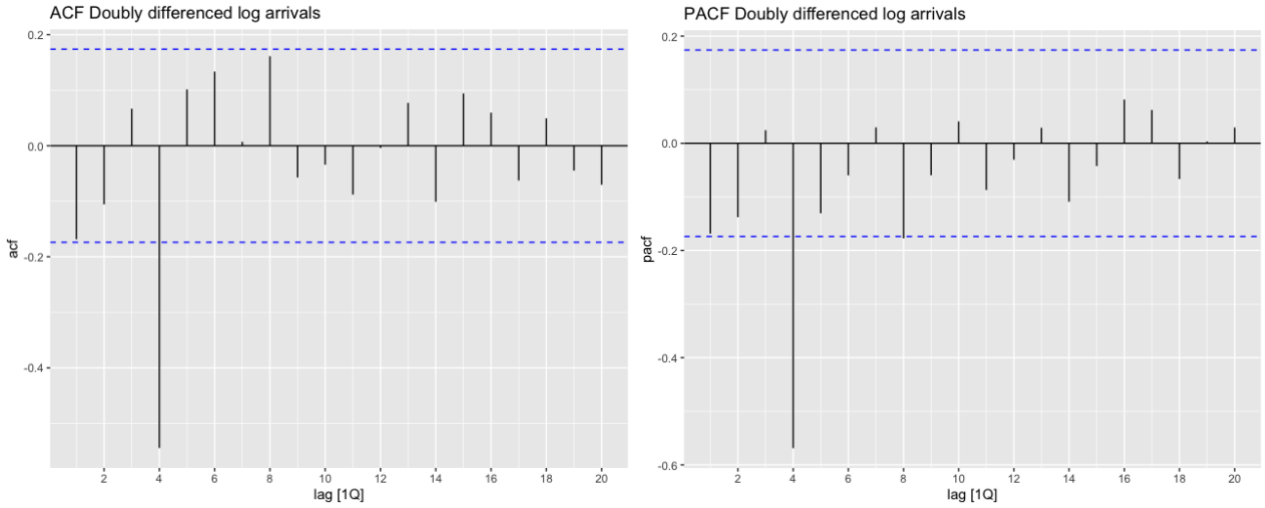


Additionally, the *unitroot_nsdiffs()* function was applied to the log of the arrivals. This returned one, suggesting one seasonal differencing is required. Subsequently, seasonal difference was applied to the log of arrivals through the *difference()* function with the lag parameter set to four (since data is in quarters) and the resulted data can be visualized below. As it is subjective from this picture whether the data is stationary, the *unitroot_ndiffs()* function was applied this time to check if the data requires another first difference. The output of the function is one, suggesting one first

difference is necessary. After this is applied, the data is represented another time, as seen in the left-hand side, and another unit root tests points out there are 0 differencing operations needed.



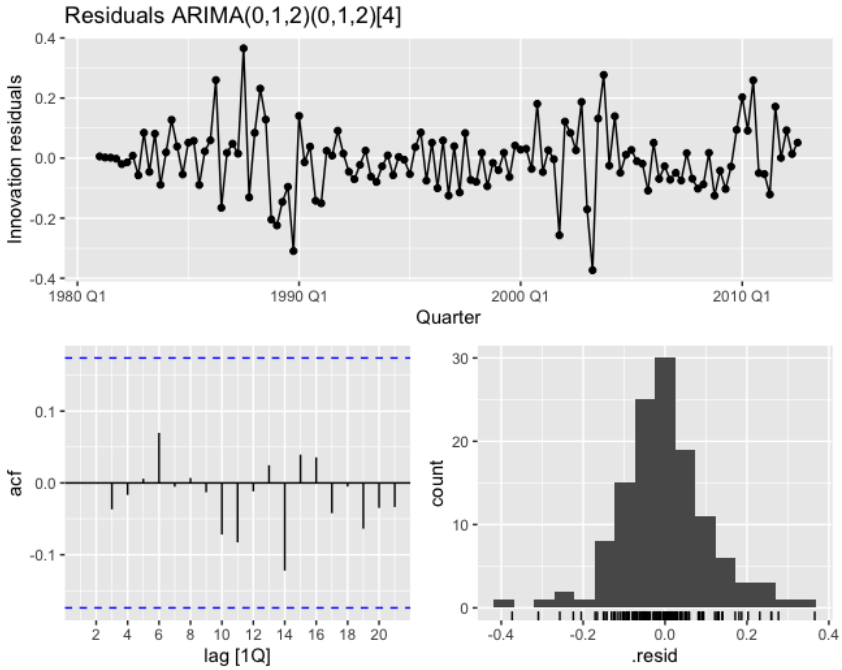
Below, one can see the ACF and PACF graphs of the doubly differenced log arrivals data. The ACF exposes a single autocorrelation outside the 95% limits, r1 is small and negative, and ACF drops to 0 quickly, implying white noise, thus, stationarity of the data.



As there is a roughly decaying sinusoidal pattern in the ACF and there is a single significant spike in the PACF at lag 4, the non-seasonal AR (i.e. p) is 0 and MA (i.e q) are 0 and 4 respectively. Since only one first differencing was applied to the data, d is 1, implying a non-seasonal ARIMA(0,1,4). As for the non-seasonal part, the degree of seasonal differencing (i.e. D) is 1. Then, the ACF shows a single significant lag (4) and exponential decays in seasonal lags of the PACF, suggesting a seasonal MA(1) (i.e. Q=1). Therefore, the ACF and PACF graphs suggest an ARIMA(0,1,4)(0,1,1)$_4$, however, the auto function ARIMA() gives a different model, namely ARIMA(0,1,2)(0,1,2)$_4$. In order to check which one is performing better, the log arrivals were fit to both as follows: *arima1 = ARIMA(log(Arrivals) ~ pdq(0,1,4) + PDQ(0,1,1))* and *auto = ARIMA(log(Arrivals)).* As for the evaluation of the models, the auto version has a better AICc, as it can be seen in the output below.

```
> glance(fit) %>% arrange(AICc) %>% select(.model, AIC, AICc, BIC)
# A tibble: 2 x 4
  .model   AIC  AICc   BIC
  <chr>  <dbl> <dbl> <dbl>
1 auto   -176. -176. -162.
2 arima1 -170. -169. -153.
```

Lastly, the residuals of the best model (auto arima) were analyzed and they look like white noise, with no significant spike and normally distributed, suggesting clearly the correctness of the model.



3.2) Using the backshift operator, the equation of ARIMA(0,1,2)(0,1,2)$_4$ is:

$$(1 - B)\,(1 - B^4)\,y_t = (1 + \theta_1 B + \theta_2 B^2)\,(1 + \Theta_1 B^4 + \Theta_2 B^8)\varepsilon_t$$

Without the backshift operator, the equation is:

$$(1 - B - B^4 + B^5)\,y_t = (1 + \Theta_1 B^4 + \Theta_2 B^8 + \theta_1 B + \theta_1\Theta_1 B^5 + \theta_1\Theta_2 B^9 + \theta_2 B^2 + \theta_2 B^2\Theta_1 B^6 + \theta_2 B^2\Theta_2 B^{10})\,\varepsilon_t$$

$$y_t - y_{t-1} - y_{t-4} + y_{t-5} = (1 + \Theta_1 B^4 + \Theta_2 B^8 + \theta_1 B + \theta_1\Theta_1 B^5 + \theta_1\Theta_2 B^9 + \theta_2 B^2 + \theta_2 B^2\Theta_1 B^6 + \theta_2 B^2\Theta_2 B^{10})\,\varepsilon_t$$

References

[1] https://otexts.com/fpp3/transformations.html#mathematical-transformations
[2] https://otexts.com/fpp3/useful-predictors.html#seasonal-dummy-variables
[3] https://otexts.com/fpp3/regression-evaluation.html#acf-plot-of-residuals
[4] https://otexts.com/fpp3/diagnostics.html#portmanteau-tests-for-autocorrelation
[5] https://otexts.com/fpp3/forecasting-regression.html#ex-ante-versus-ex-post-forecasts

Appendix

```
 ## Resit Assignment 2
library(fpp3)

# 1.1
souvenirs %>%
  autoplot(Sales)+
  labs(title = "Souvenir sales over time")

souvenirs %>% model(STL(Sales))%>% components() %>% autoplot()

# 1.2
souvenirs %>%
  autoplot(sqrt(Sales))+
  labs(title = "Sqrt souvenir sales over time")

souvenirs %>%
  autoplot((Sales)^(1/3))+
  labs(title = "Cube root souvenir sales over time")

souvenirs %>%
  autoplot(log(Sales))+
  labs(title = "Log souvenir sales over time")

# 1.3 regression model
souvenirs_ <- souvenirs %>%
  mutate(surf = as.integer(month(Month)==3 & year(Month)>1987))

fit <- souvenirs_ %>%
  model(reg = TSLM(log(Sales) ~ trend() + season() + surf))

report(fit)

fit %>% gg_tsresiduals(type = "response")+
  labs(title = "Residuals Diagnostics")

augment(fit) %>%
  ggplot(aes(x = .fitted, y = .resid)) +
  geom_point() + labs(x = "Fitted", y = "Residuals", title = "Residuals against fitted values")

ggplot(augment(fit),aes( x = month(Month), y = .resid, group = month(Month))) +  geom_boxplot() + labs(x = "Month", y
= "Residuals", title = "Boxplots of residuals per month")

# 1.4
augment(fit) %>% features(.innov, ljung_box, lag = 24, dof = 13)

# 1.5
test <- tsibble(
  Month = yearmonth("1994 Jan") + 0:35,
  Sales = rep(0.01),
  key = Sales
) %>% mutate(surf = as.integer(month(Month)==3 & year(Month)>1987))

test <- bind_rows(souvenirs_,test) %>% filter(year(Month)>1993)
```

```
fit %>% forecast(new_data=test) %>% autoplot(souvenirs_) +labs(title = "Sales forecasts")

glance(fit) %>%
  select(.model, r_squared, adj_r_squared, AICc, CV)
##
fit2 <- souvenirs_ %>%
  model(reg2 = TSLM(log(Sales) ~ trend + season() + surf))

report(fit2)

glance(fit2) %>%
  select(.model, r_squared, adj_r_squared, AICc, CV)

fit2 %>% forecast(new_data=test) %>% autoplot(souvenirs_) +labs(title = "Sales forecasts 2")


#### 2.1
data <- tourism %>% summarise(Trips = sum(Trips))
data %>% autoplot(Trips) + labs(title = "Trips over time")

dcmp <- data %>%
  model(stl = STL(Trips))
components(dcmp) %>% autoplot()

components(dcmp) %>%
  as_tsibble() %>%
  autoplot(Trips, colour = "gray") +
  geom_line(aes(y=season_adjust), colour = "#0072B2") +
  labs(title = "Seasonally adjusted trips vs original data")

# 2.2
season_adj <- components(dcmp) %>% select(season_adjust)

fit <-season_adj  %>%
  model(damped = ETS(season_adjust ~ error("A") + trend("Ad") + season("N")))

report(fit)

fit %>% forecast(h = 8) %>% autoplot(season_adj) + labs(title = "Seasonally adjusted trips forecasts with damped")

fit %>% accuracy ()

# 2.3
fit2 <-season_adj  %>%
  model(holt = ETS(season_adjust ~ error("A") + trend("A") + season("N")))

report(fit2)

fit2 %>% forecast(h = 8) %>% autoplot(season_adj) + labs(title = "Seasonally adjusted trips forecasted with Holt")

fit2 %>% accuracy ()


fit3 <-data  %>%
  model(seasonal = ETS(Trips))

report(fit3)
```

```
fit3 %>% forecast(h = 8) %>% autoplot(data) + labs(title = "Trips forecasted with ETS(A,A,A)")

fit3 %>% accuracy ()

# 2.4
fit2%>% gg_tsresiduals()+ labs(title = "Residuals ETS(A,A,N)")
augment(fit2) %>% features(.innov, ljung_box, lag = 8, dof = 0)

#### 3.1
data <- aus_arrivals %>% filter(Origin == "Japan")
data <- data[-2]

data %>%autoplot(Arrivals)+
  labs(title = "Arrivals from Japan over time")

data %>%autoplot(log(Arrivals))+
  labs(title = "Log arrivals from Japan over time")

dcmp <- data %>%
  model(stl = STL(log(Arrivals)))
components(dcmp) %>% autoplot()

data %>%
  features(log(Arrivals), unitroot_nsdiffs)

data %>%autoplot(difference(log(Arrivals), 4))+
  labs(title = "Annual change in log arrivals")
data %>%
  features(difference(log(Arrivals), 4), unitroot_ndiffs)
data %>%autoplot(difference(difference(log(Arrivals), 4)))+
  labs(title = "Doubly differenced log arrivals")

data %>%
  features(difference(difference(log(Arrivals), 4)), unitroot_ndiffs)

diff_data <-data %>%
  mutate(doubly_diff = difference(difference(log(Arrivals), 4)))

diff_data %>% ACF(doubly_diff)%>% autoplot()+
  labs(title = "ACF Doubly differenced log arrivals")

diff_data %>% PACF(doubly_diff)%>% autoplot()+
  labs(title = "PACF Doubly differenced log arrivals")

fit <- diff_data %>%  model(arima1 = ARIMA(log(Arrivals) ~ pdq(0,1,4) + PDQ(0,1,1)),
                      auto = ARIMA(log(Arrivals)))

fit %>% select(auto)
fit %>% select(arima1)

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

fit %>% select(auto)  %>% gg_tsresiduals()+
  labs(title = "Residuals ARIMA(0,1,2)(0,1,2)[4]")
```