

Problem 1: Labour productivity

Problem 2: Monthly Average Auckland Temperatures

STATS 326

Code ▼

Assignment 04

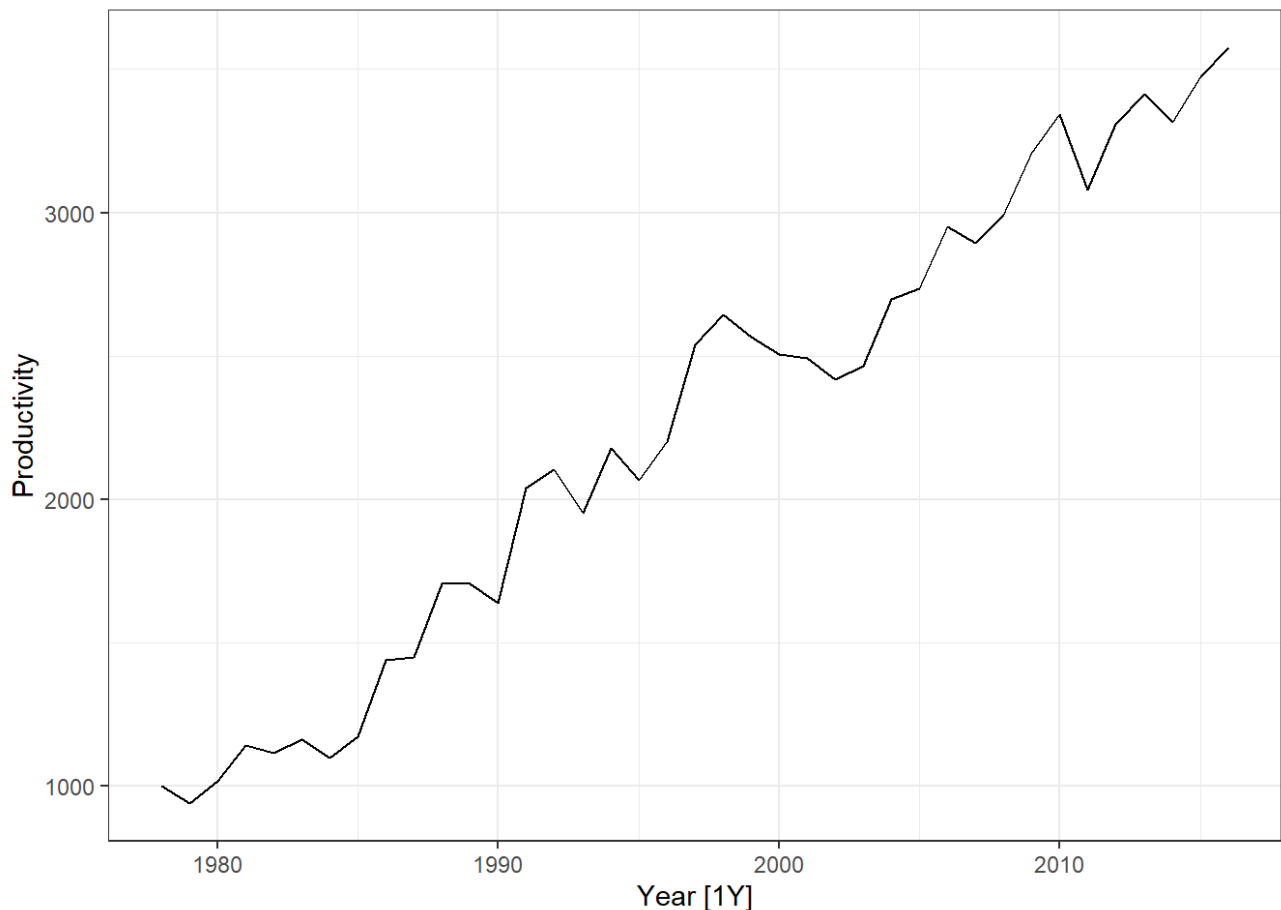
General comments:

- All the plots should be labelled appropriately (axes, legends, titles).
- Please submit both your `.Rmd`, and the generated output file `.html` or `.pdf` on Canvas before the due date/time.
- Please make sure that the `.Rmd` file compiles without any errors. The marker will not spend time fixing the bugs in your code.
- Please avoid specifying absolute paths.
- Your submission must be original, and if we recognize that you have copied answers from another student in the course, we will deduct your marks.
- You will need to use the `tidyverse` and `fpp3` libraries for this assignment.

Due: Friday 24 May 2024 at 16:00 PM (NZ time)

Problem 1: Labour productivity

In economics, productivity measures how much output has been produced per unit of input, and can therefore measure how efficient the economy is. The file `productivity.csv` contains the labour productivity (index) for primary industries in New Zealand, measured annually from 1978 to 2021. A training set called `train` has been created that withholds the years 2017–2021, and has been plotted below.



1. **4 Marks** Based on the time plot above, think of two candidate ETS models to fit. Write down the reasons you think these models will be appropriate for the labour productivity series.

Based on time plot above of the labor productivity series, we can see a positive increasing trend without any evidence of seasonality. Based on this I think the following ETS models would be appropriate: ETS(A, A, N), (Additive Error (A), Additive Trend (A), No Seasonality (N)) this is because this model is simple and seems to be doing a decent job in terms of capturing linear positive trend effectively with additive trend component, assuming constant variance with the additive error term and no seasonality as the plot does not seem to be predictable.

Another appropriate model is ETS(M, A, N), (Multiplicative Error (M), Additive Trend (A), No Seasonality (N)). This is very similar to the previous model with the only difference of multiplicative error meaning that the variability of the plot is not constant and it the model allows for the possibility of an increase in the variance as the y levels increase, however the variability does not seem to be increasing a lot but I wanted to try a model with multiplicative error to captures the values and decide on what is the best model according to levels such as AICc, as it is evident that the trend is additive and there is no seasonality in the time plot.

2. 6 Marks Fit these two candidate models as well as the automatic ETS model on your training set. Compare the models using AICc and output the parameter estimates from the model that has the best predictive ability. Name the best model and interpret the smoothing parameters.

[Hide](#)

```
fit <- train %>%
model(
  additive = ETS(Productivity ~ error("A") + trend("A") +
                  season("N")),
  multiplicative = ETS(Productivity ~ error("M") + trend("A") +
                       season("N")),
  ets_automatic = ETS(Productivity)
)

fit %>%
select(additive) %>%
report()
```

```
## Series: Productivity
## Model: ETS(A,A,N)
## Smoothing parameters:
##   alpha = 0.4889942
##   beta  = 0.0001000043
##
## Initial states:
##   l[0]   b[0]
## 880.7263 68.67191
##
## sigma^2: 19414.32
##
##      AIC      AICc      BIC
## 533.7355 535.5536 542.0533
```

[Hide](#)

```
fit %>%
  select(multiplicative) %>%
  report()
```

```
## Series: Productivity
## Model: ETS(M,A,N)
## Smoothing parameters:
##   alpha = 0.5050836
##   beta  = 0.000100004
##
## Initial states:
##   l[0]    b[0]
## 879.4898 66.35083
##
## sigma^2: 0.005
##
##      AIC      AICc      BIC
## 538.1444 539.9626 546.4622
```

[Hide](#)

```
fit %>%
  select(ets_automatic) %>%
  report()
```

```
## Series: Productivity
## Model: ETS(A,A,N)
## Smoothing parameters:
##   alpha = 0.4889942
##   beta  = 0.0001000043
##
## Initial states:
##   l[0]    b[0]
## 880.7263 68.67191
##
## sigma^2: 19414.32
##
##      AIC      AICc      BIC
## 533.7355 535.5536 542.0533
```

The code chunk above plots and compares the three ETS models. The first model (A,A,N) has a lower AICc value compared to the second model (547.4036) and seems to be considered as the best model as the ETS automatic model suggests as well. Consistent to the AICc value, the alpha in A,A,N model (alpha = 0.4858404) suggests that the weight given to the most recent observation, the alpha is also the lowest suggesting that there is less variability in the data making it more predictable and a better model, The beta parameter in an ETS model measures the smoothing for the trend component and how much weight is given to the most recent observations, both my A,A,N and M,A,N models have a very similar beta value (A,A,N = 0.000100113), (M, A, N = 0.0001000401) with A,A,N being slightly higher but the difference is so small that is not significant. (M,A,N) model accounts for the proportional variability in the error term but the is not the best model based on the AICc

(554.6037) and other values in the data all being higher than the (A,A,N) model. Likewise the results in the ETS automatic model are identical to the (A,A,N) giving us further evidence to believe that our first model is the best fit as the values suggest that the model places moderate weight on recent observations for the level component, while the trend component is updated very slowly.

3. 5 Marks Plot the differenced productivity series from the training set and comment on whether you believe this series is stationary. Verify this using a KPSS test on the differenced productivity series, making sure you interpret the p -value.

[Hide](#)

```
diff_productivity <- train %>%
  mutate(Differenced = difference(Productivity))

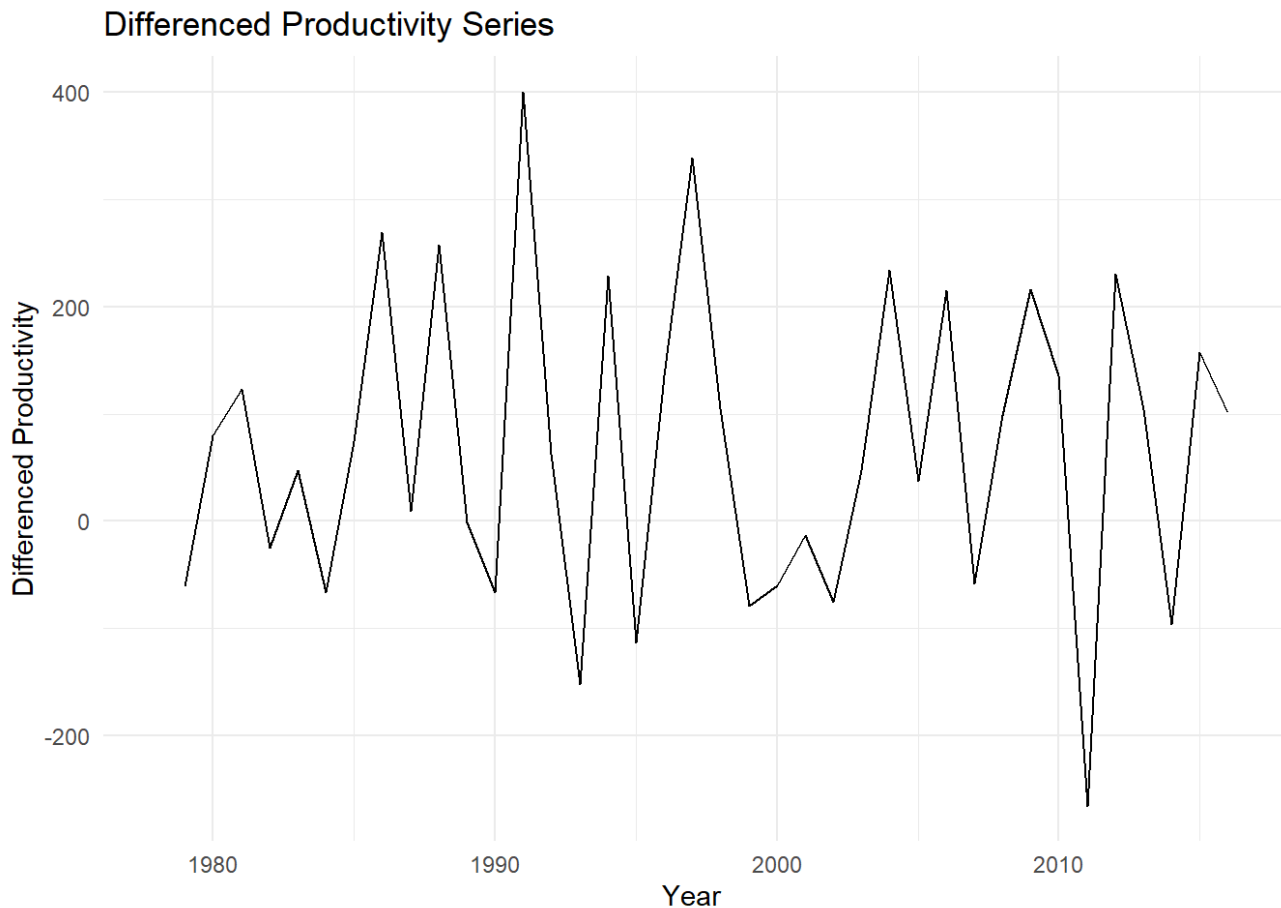
kpss <- diff_productivity %>%
  features(Differenced, unitroot_kpss)
print(kpss)
```

```
## # A tibble: 1 × 2
##   kpss_stat kpss_pvalue
##   <dbl>      <dbl>
## 1    0.0520        0.1
```

[Hide](#)

```
diff_productivity %>%
  ggplot(aes(x = Year, y = Differenced)) +
  geom_line() +
  labs(title = "Differenced Productivity Series", y = "Differenced Producti
vity", x = "Year") +
  theme_minimal()
```

```
## Warning: Removed 1 row containing missing values or values outside the s
cale range
## (`geom_line()`).
```



Based on the KPSS test results, with a test statistic of 0.05198573 and a p-value of 0.1, we do not reject the null hypothesis (that the time series is stationary) as we no evidence against the null.

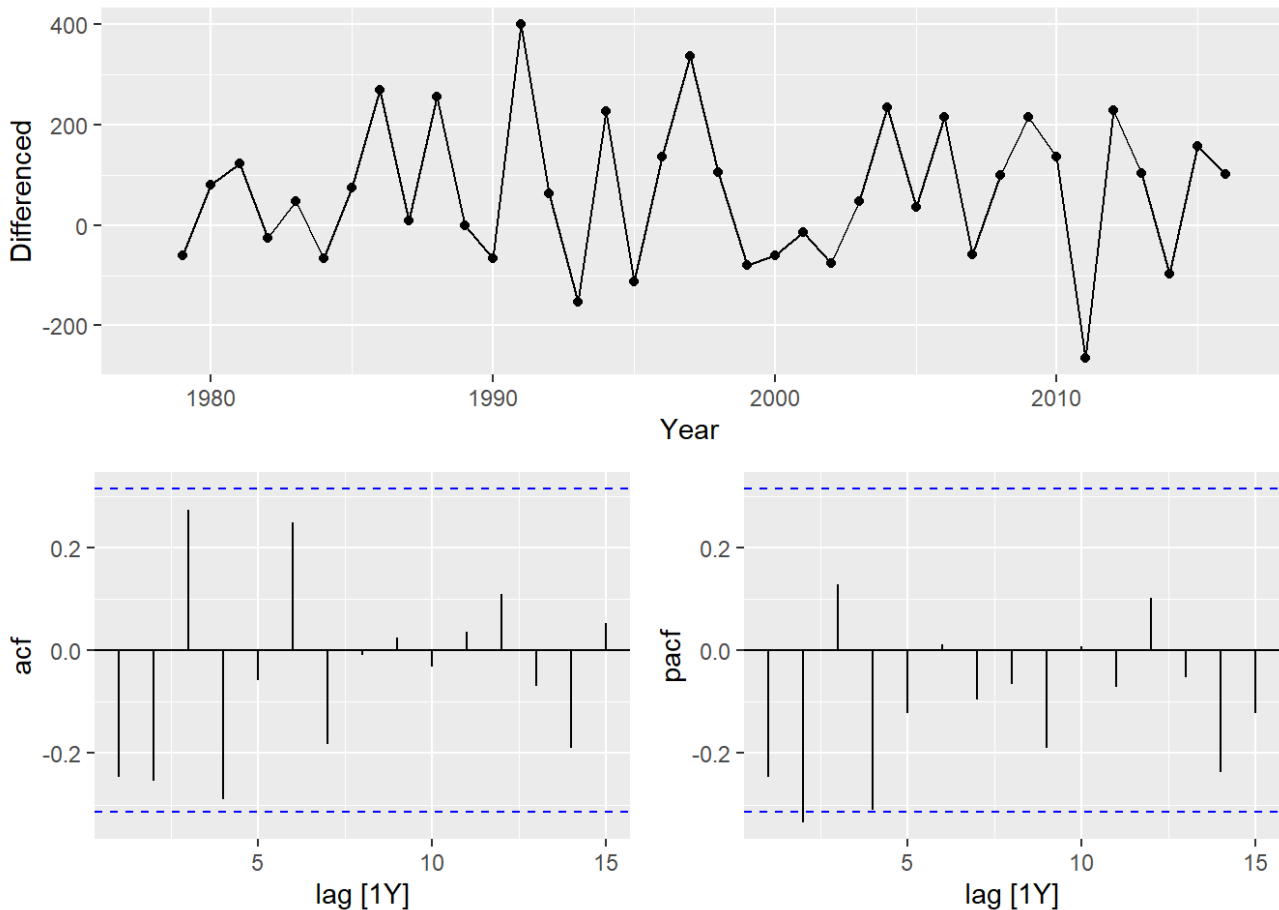
4. **7 Marks** Plot the ACF and PACF plots for your differenced training data (you may use the `gg_tsdisplay` function). Analyse these plots to come up with two candidate ARIMA models. Explain why you have chosen these ARIMA models.

Hide

```
diff_productivity %>%  
  gg_tsdisplay(Differenced, plot_type = "partial")
```

```
## Warning: Removed 1 row containing missing values or values outside the s  
cale range  
## (`geom_line()`).
```

```
## Warning: Removed 1 row containing missing values or values outside the s  
cale range  
## (`geom_point()`).
```



The PACF plot shows significant spike at around lag 1 and gradually tails off indicating a MA(1) process. The PACF plot shows significant spikes at lag 1 and then it gradually drops off as well showing an AR(1) process. The first candidate model could be ARIMA(1,1,0) includes an auto regressive term, a first difference component for the stationary to achieve stationary as well as no moving averages (MA0) as there ACF plot does not suggest the need of additional error smoothing beyond lag 1. Another model could be ARIMA(0,1,1) with no auto regressive component as the PACF pattern may not need auto regressive terms after lag 1, there is a first difference indicating that the original series has been difference to achieve stationary and (MA 1) moving average term suggesting that the current value of differences series is influenced by the lagged term. To determine which model is the best I will fit my candidate models as well as automatic ARIMA model on my training set to determine which has the best predictive ability.

5. 6 Marks Fit your candidate model as well as the automatic ARIMA model on your training set. Compare the models using AICc and output the parameter estimates from the model that has the best predictive ability. Name the best model and write the fitted model equation using backshift notation.

Hide

```
fit_aimaa <- train %>%
  model(
    arima110 = ARIMA(Productivity ~ pdq(1, 1, 0)),
    arima011 = ARIMA(Productivity ~ pdq(0, 1, 1)),
    auto = ARIMA(Productivity)
  )

aic_values <- glance(fit_aimaa) %>%
  arrange(AICc) %>%
  select(.model:BIC)
print(aic_values)
```

```
## # A tibble: 3 × 6
##   .model    sigma2 log_lik   AIC   AICc   BIC
##   <chr>      <dbl>   <dbl> <dbl> <dbl> <dbl>
## 1 arima011 18873.   -240.  486.  487.  491.
## 2 auto      18873.   -240.  486.  487.  491.
## 3 arima110 20280.   -241.  489.  489.  494.
```

[Hide](#)

```
fit_aimaa %>%
  select(arima110) %>%
  report()
```

```
## Series: Productivity
## Model: ARIMA(1,1,0) w/ drift
##
## Coefficients:
##          ar1  constant
##        -0.2455   85.0252
## s.e.    0.1566   22.6052
##
## sigma^2 estimated as 20280:  log likelihood=-241.35
## AIC=488.71   AICc=489.41   BIC=493.62
```

Based on the results from the output above ARIMA(0,1,1) model has the lowest AICc with the value of 487 making it the best model.

The fitted model equation: $(1-B)Y_t = (1-\theta_1 B)\epsilon_t$

estimate: θ_1 is $\theta_1 = \theta$

the fitted equation will be: $(1+0.2455B)(1-B)Y_t = 85.0252 + \epsilon_t$

- B is the back shift operator
- ϵ_t is the white noise error term at time t

6. 6 Marks Fit your best ETS and ARIMA model on the training data in the same mable . Forecast $h = 5$ periods into the future. Plot your forecasts. Compare your forecasts against the truth by computing RMSE, MAE, MAPE, and MASE. State which model has the best forecast accuracy.

Hide

```
fit <- train %>%
  model(
    ets = ETS(Productivity ~ error("A") + trend("A") + season("N")),
    arima = ARIMA(Productivity ~ pdq(1, 1, 0))
  )

forecasts <- fit %>%
  forecast(h = 5)

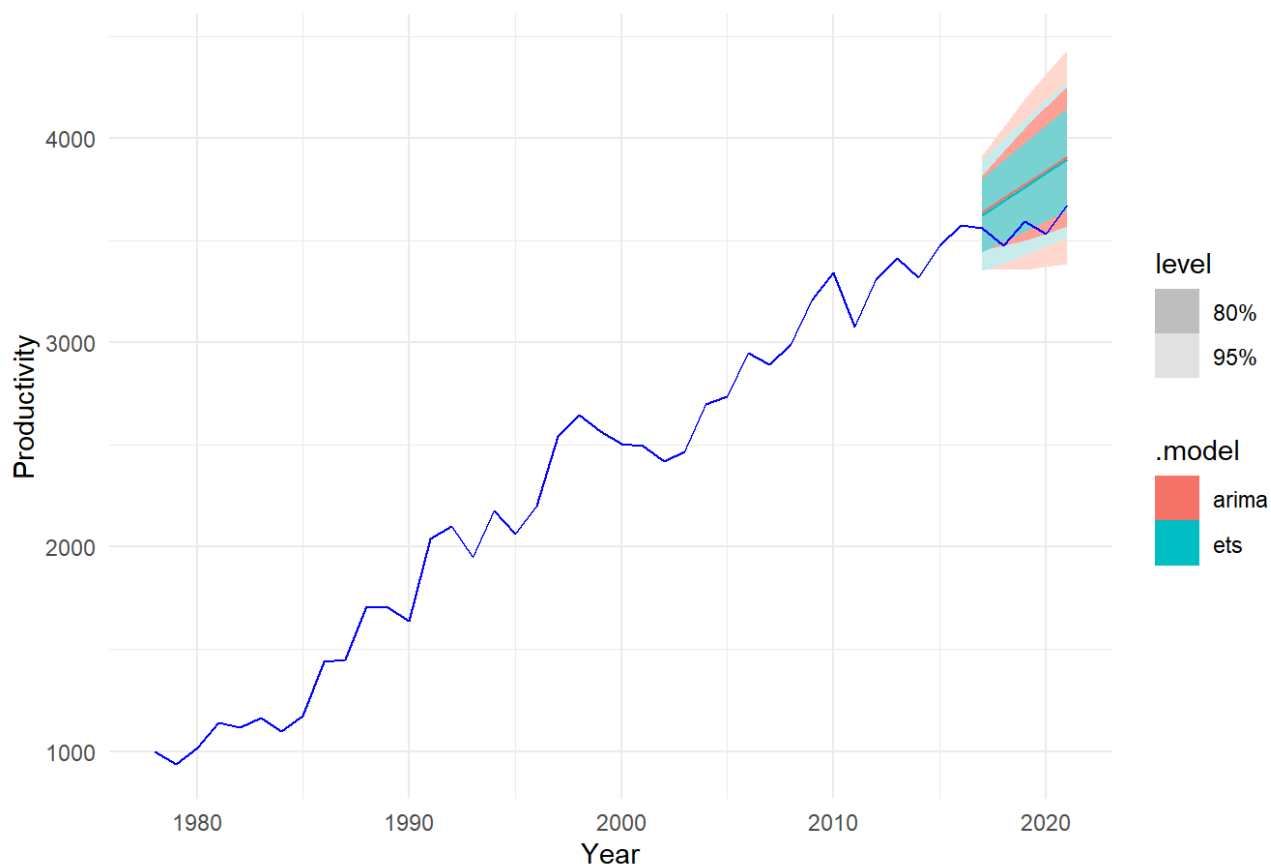
# Plot the forecasts
plot_forecast <- forecasts %>%
  autoplot(train) +
  autolayer(data, color = "blue") +
  labs(title = "Forecasts from Best ETS and ARIMA Models",
       y = "Productivity",
       x = "Year") +
  theme_minimal()
```

```
## Plot variable not specified, automatically selected `.vars = Productivit
y`
```

Hide

```
print(plot_forecast)
```

Forecasts from Best ETS and ARIMA Models


[Hide](#)

```
accuracy_measures <- accuracy(forecasts, data)
print(accuracy_measures)
```

```
## # A tibble: 2 × 10
##   .model .type    ME  RMSE  MAE  MPE  MAPE  MASE  RMSSE  ACF1
##   <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 arima Test  -207.  220.  207. -5.80  5.80  1.62  1.39 -0.122
## 2 ets   Test  -194.  208.  194. -5.43  5.43  1.52  1.31 -0.115
```

The ETS model has lower RMSE, MAE, MAPE, and MASE values compared to the ARIMA model making it a better model in terms of forecast accuracy out of the two.

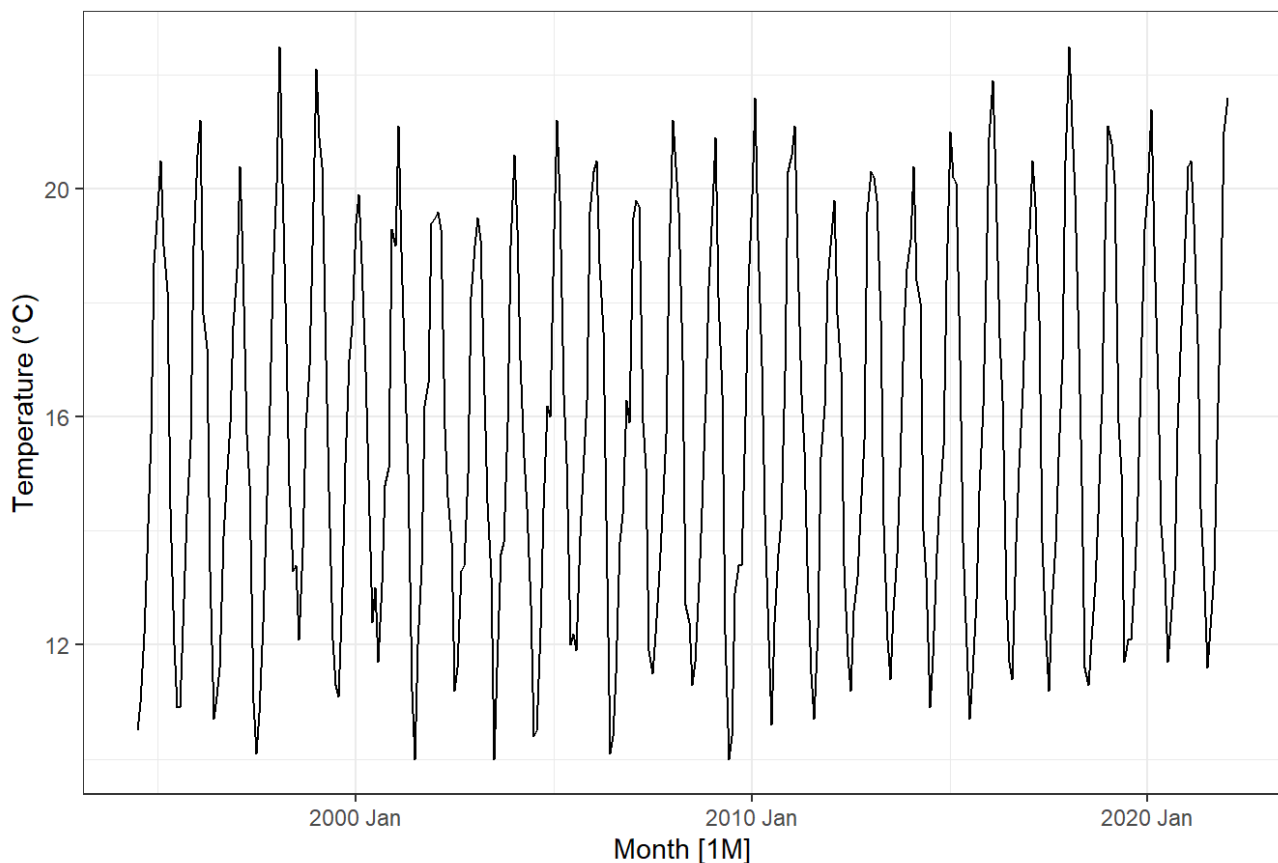
Possible marks for **Problem 1:** 34 Marks

Problem 2: Monthly Average Auckland Temperatures

In Assignments 1, 2, and 3, you investigated monthly average temperatures in Auckland. In this problem, you will do some further analysis. The data set `auckland_temps.csv` contains the monthly average temperatures in Auckland from July 1994 until January 2024. For this

question do not Box-Cox transform the data. A training set called `train` has been created that withholds the most recent two years of data, and this has been plotted below.

Monthly Average Temperatures in Auckland (Jul 1994 - Jan 2022 (training set))



1. **4 Marks** Based on the time plot above, think of two candidate ETS models to fit. Write down why you think these models will be appropriate for the time series given.

The time plot above is a representation of Auckland temperature in different months, The characteristics of the data can be used to predict candidate models. The plot looks to have a fluctuating trend, with no clear increase or decrease trend in the entire plot, Unlike the last question this plot is indicating a seasonal pattern that is indicating seasonal patterns that may changes slightly over time, and lastly the time plot above has semi constant variance.

Based on the reasons mentioned before the first ETS candidate model would be (A,A,A), (Additive Error, Additive Trend, Additive Seasonality) this is because the fluctuations around the trend and seasonal components are not increasing and decreasing in overtime and with a constant variance making it an additive error, There is no specific trend but in case of any trend I can assume that there would be an additive increase or decrease, and the seasonality is additive as the variance in the plot is constant.

The other candidate model is (M, A, M), this is because the model assume that the size of the errors depends on the level of the series making it an appropriate component for the future variability in the data, The trend is additive as similar to the last model there an additive trend would be adequate to capture any ongoing trends in the plot, Lastly I picked

a multiplicative seasonality mainly as a to compare it against the additive seasonality and maybe to account for possible uncertainty and as the fluctuation on the plot may be growing or shrinking however that is highly not the case in this plot.

2. 6 Marks Fit these two candidate models as well as the automatic ETS model on your training set. Compare the models using AICc and output the parameter estimates from the model that has the best predictive ability. Interpret the smoothing parameters.

[Hide](#)

```
fit_2 <- data_2 %>%
  model(
    additive = ETS(Temperature ~ error("A") + trend("A") + season("A")),
    multiplicative = ETS(Temperature ~ error("M") + trend("A") + season
("M")),
    ets_automatic = ETS(Temperature)
  )

fit_2 %>%
  select(additive) %>%
  report()
```

```
## Series: Temperature
## Model: ETS(A,A,A)
## Smoothing parameters:
##   alpha = 0.1300257
##   beta  = 0.0001006388
##   gamma = 0.0001012808
##
## Initial states:
##      l[0]      b[0]      s[0]      s[-1]      s[-2]      s[-3]      s[-4]
s[-5]
## 15.50789 0.002851272 -3.608317 -1.423928 1.105249 3.389499 4.872682 4.3
26804
##      s[-6]      s[-7]      s[-8]      s[-9]      s[-10]      s[-11]
## 2.952823 0.6142062 -1.077091 -2.547704 -4.030632 -4.573591
##
## sigma^2: 0.6191
##
##      AIC      AICc      BIC
## 1931.989 1933.805 1997.815
```

[Hide](#)

```
fit_2 %>%
  select(multiplicative) %>%
  report()
```

```
## Series: Temperature
## Model: ETS(M,A,M)
## Smoothing parameters:
##   alpha = 0.0307359
##   beta  = 0.0001950088
##   gamma = 0.000110681
##
## Initial states:
##   l[0]      b[0]      s[0]      s[-1]      s[-2]      s[-3]      s[-4]      s
[-5]
## 15.45971 0.001105608 0.7762216 0.9144982 1.067805 1.22437 1.30217 1.277
844
##   s[-6]      s[-7]      s[-8]      s[-9]      s[-10]      s[-11]
## 1.18284 1.037086 0.9231145 0.8421315 0.7390945 0.7128236
##
## sigma^2: 0.0028
##
##      AIC      AICc      BIC
## 1960.403 1962.219 2026.229
```

[Hide](#)

```
fit_2 %>%
  select(ets_automatic) %>%
  report()
```

```
## Series: Temperature
## Model: ETS(A,N,A)
## Smoothing parameters:
##   alpha = 0.1412324
##   gamma = 0.0001008507
##
## Initial states:
##   l[0]      s[0]      s[-1]      s[-2]      s[-3]      s[-4]      s[-5]      s[-6]
## 15.14005 -3.570715 -1.367942 1.081023 3.353203 4.864854 4.380036 2.916462
##   s[-7]      s[-8]      s[-9]      s[-10]      s[-11]
## 0.6049708 -1.149266 -2.523008 -4.059954 -4.529664
##
## sigma^2: 0.6137
##
##   AIC      AICc      BIC
## 1926.989 1928.405 1985.071
```

In the code chunks above the candidate models are plotted and compared against each other. Building up on what was said in the last part, as expected the (A,A,A) model does a better job indicating a better model fit according to the lower AICc (A,A,A = 1933.805), (M,A,M = 1962.219) and other tests, the two models both have very small gamma values indicating stable seasonal components, both have a very small beta values, indicating a very smooth trend component however the alpha levels are higher in (A,A,A) model meaning that they are moderately responsive to recent changes ($0.1300257 = \alpha$) and the alpha value is smaller indicating a less responsive and higher levels of smoothing of the level component ($0.0307359 = \alpha$), in other words the additive model is more responsive to the recent changes, while the multiplicative model smooths the level component more heavily as it is reacting slower to the changes in within the data.

based on ETS automatic function model (A,N,A) is the best model with the lowest AICc value (1928.405) and the predictive ability among the three models. The alpha level of 0.1412324 initiates that the level is responsive to the recent changes but still relies in the information and values from the past there is a very low gamma value 0.0001008507 indicating that the seasonal component is quite stable and changes very little over time making it consonant over years, in other words the smoothing parameters in this model indicate that the level component is moderately smoothed and the seasonal component is stable.

3. **11 Marks** Explore how many orders of seasonal differencing, D , and first differencing, d , are required to make the training data stationary by doing the following:

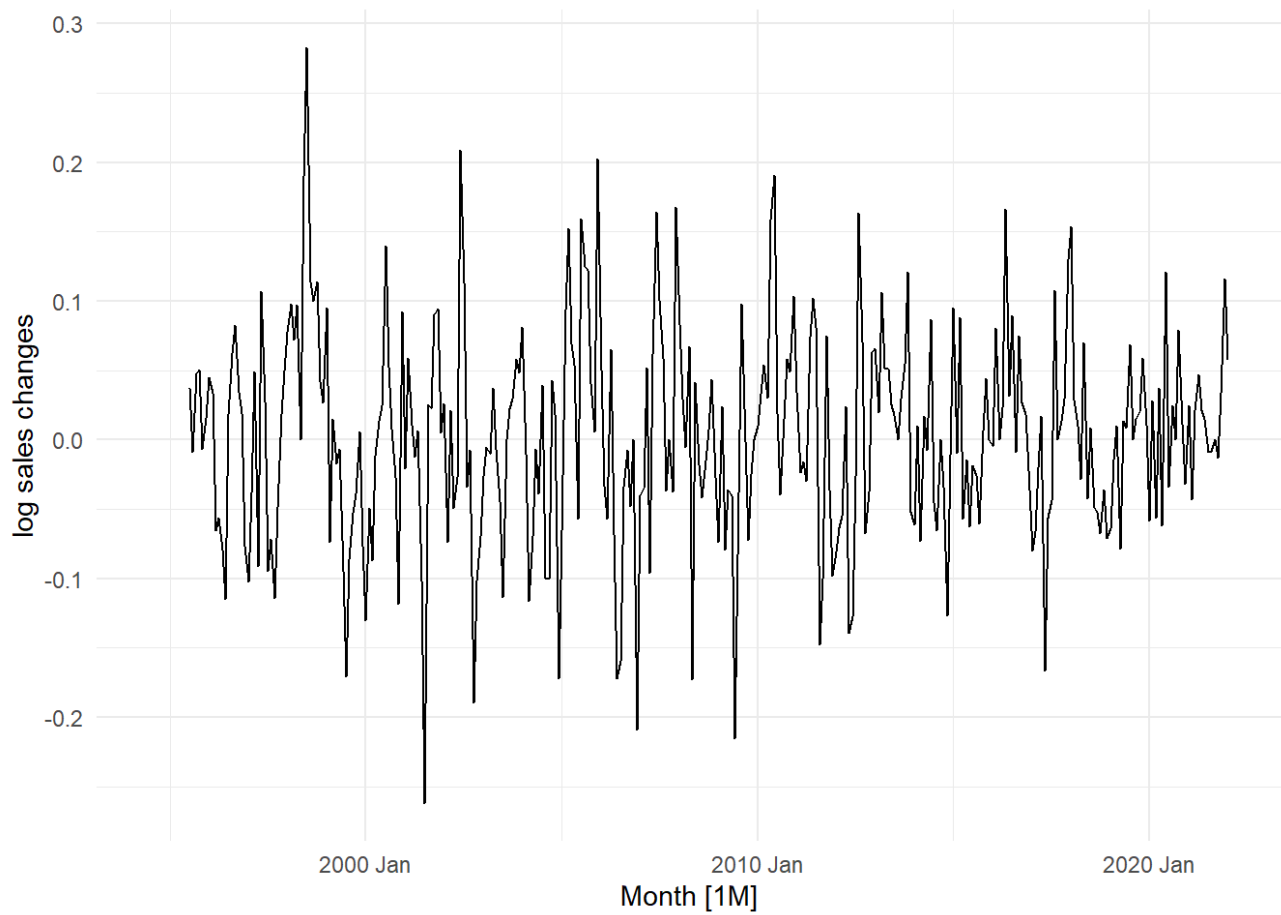
- Calculate the seasonal strength, F_s , on the training data. Report the value and interpret it in plain English.
- If $F_s > 0.64$, seasonally difference (using $\text{lag} = 12$) the training data and calculate F_s again. Determine if you need to seasonally difference again. If necessary, continue this process until you have found an appropriate order of seasonal differencing.
- Verify the order of seasonal differencing using the `unitroot_nsdiffs` feature on your training data. Report D .
- Plot your seasonally differenced training data and comment on whether you believe the time series has constant mean over time.
- Perform a KPSS unit root test on your seasonally differenced training data to determine if this series is stationary. Report your p -value and interpret it in plain English. (If your time series requires first differencing, difference it and repeat the KPSS test on the differenced series until stationarity is satisfied).
- Verify the order of first differencing using the `unitroot_ndiffs` feature on your seasonally differenced training data. Report d .

[Hide](#)

```
temp_diff <- train_2 %>%
  mutate(Differenced = difference(Temperature))

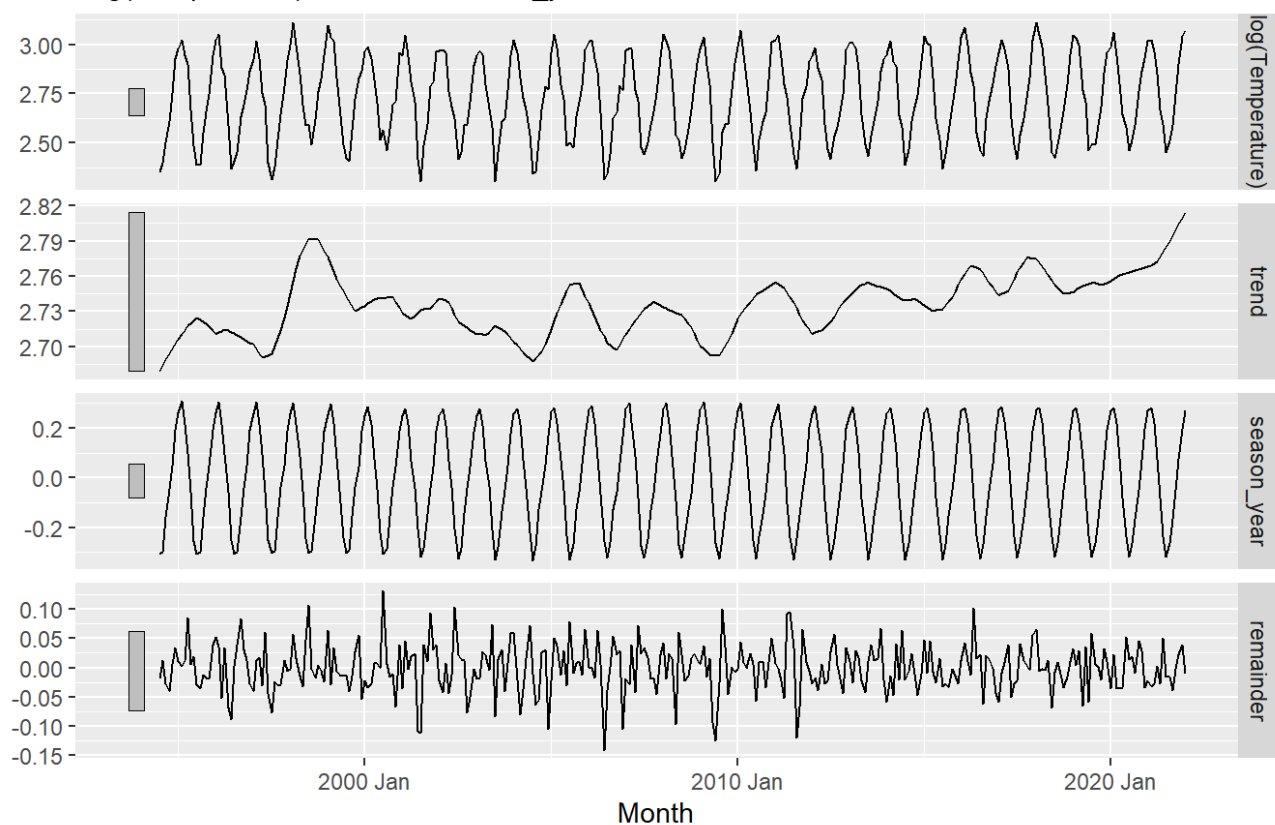
temp_diff %>%
  autoplot(log(Temperature) %>% difference(12)) + ylab("log sales c
hanges") +
  theme_minimal()
```

```
## Warning: Removed 12 rows containing missing values or values outside the
scale range
## (`geom_line()`).
```

[Hide](#)

```
temp_diff %>%  
  model(STL(log(Temperature))) %>% components() %>%  
  autoplot()
```


STL decomposition

 $\text{'log(Temperature)' = trend + season_year + remainder}$


Hide

```
temp_diff %>%
  features(Temperature, unitroot_nsdiffs)
```

```
## # A tibble: 1 × 1
##   nsdiffs
##   <int>
## 1      1
```

Hide

```
temp_diff %>%
  features(log(Temperature), feat_stl)
```

```
## # A tibble: 1 × 9
##   trend_strength seasonal_strength_year seasonal_peak_year seasonal_trough_year
##   <dbl> <dbl> <dbl> <dbl>
## 1 0.332 0.962 8
## # i 5 more variables: spikiness <dbl>, linearity <dbl>, curvature <dbl>,
## #   stl_e_acf1 <dbl>, stl_e_acf10 <dbl>
```

Hide

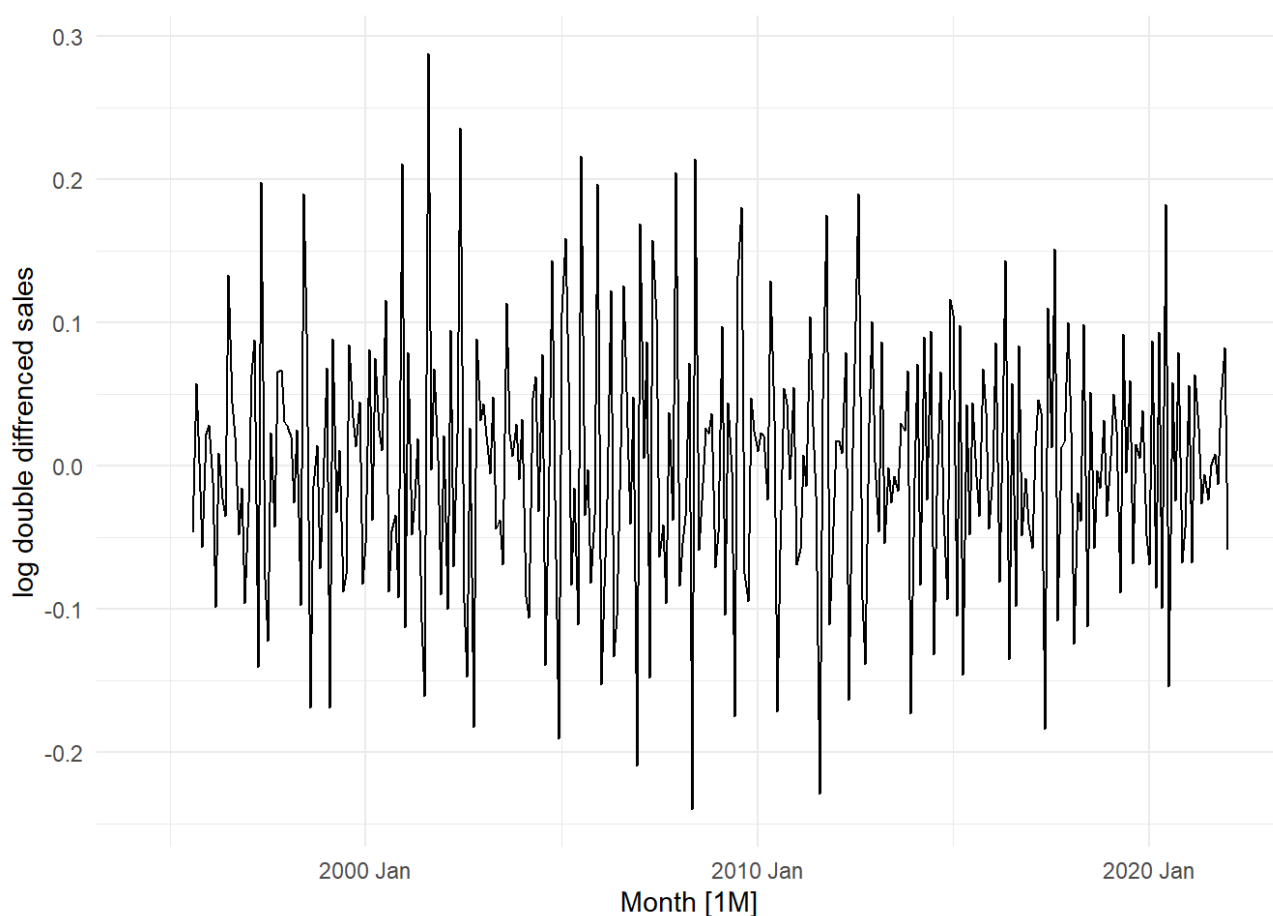
```
temp_diff %>%
  features(difference(Temperature), unitroot_kpss)
```

```
## # A tibble: 1 × 2
##   kpss_stat kpss_pvalue
##   <dbl>      <dbl>
## 1    0.00867        0.1
```

Hide

```
temp_diff %>%
  autoplot(log(Temperature) %>% difference(12) %>% difference(1)) +
  ylab("log double differenced sales") +
  theme_minimal()
```

```
## Warning: Removed 13 rows containing missing values or values outside the
scale range
## (`geom_line()`).
```



Hide

```
temp_diff %>%
  features(difference(log(Temperature), 12), unitroot_ndiffs)
```

```
## # A tibble: 1 × 1
##   ndiffs
##   <int>
## 1      0
```

Hide

```
temp_diff %>%
  features(Temperature, unitroot_kpss)
```

```
## # A tibble: 1 × 2
##   kpss_stat kpss_pvalue
##   <dbl>     <dbl>
## 1  0.0661      0.1
```

I first plotted seasonal difference and initial steps to evaluate my seasonal strength value, once I got my F value 0.962 which is fairly close to 1 indicating that seasonality causes the data variability. The data is non stationary at the start as the first plot is showing as well, indicating a high seasonality paired with the F value test therefore further steps are done to make the data stationary. As stated in the question uniroot function with an output of 1 was used for the seasonal difference. The data is then plotted again and it seems more normal as there is a more constant variance and mean showing stationary difference. To make sure this is correct a KPSS test is done and the p value in the kpss test indicated that there is stationary as the p value was more than 0.05 and the uniroot is showing d=1 in the end meaning there is no more adjustments and differencing are needed for the seasonality.

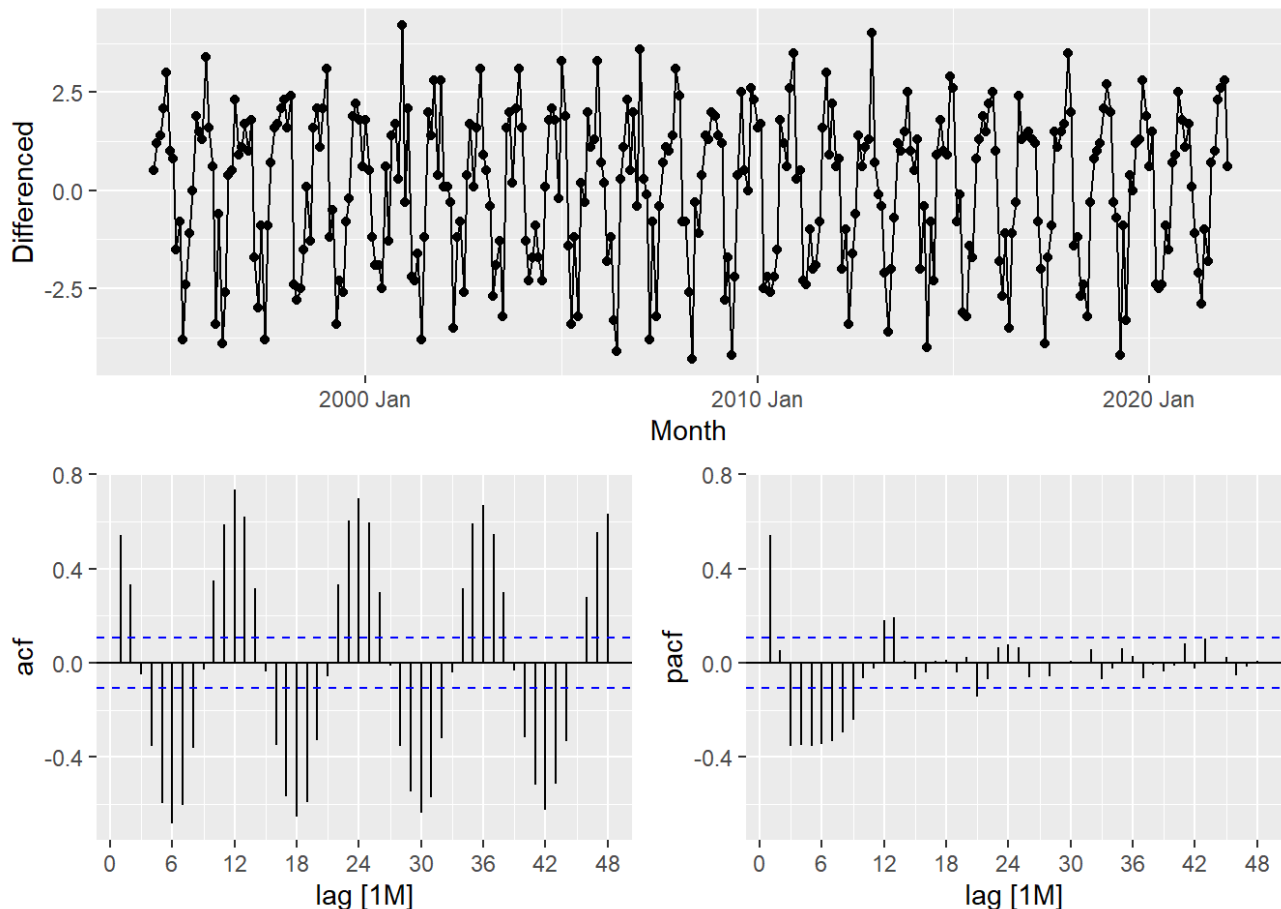
4. **11 Marks** Plot the ACF and PACF plots on your differenced training data (you may use the `gg_tsdisplay` function). Set `lag_max = 48`. Analyse these plots to come up with four candidate ARIMA models. Explain why you have chosen these ARIMA models.

Hide

```
temp_diff %>%
  gg_tsdisplay(Differenced,
    plot_type = "partial", lag_max = 48)
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



The residual plot on the top is showing a constant variance with a zero mean indicating that the model is adequate for future forecasting.

The ACF plot shows spikes at 12, 24, 36 etc showing a seasonal/annual seasonality with the 12 month period and the increases and decreases in the plot also approve this.

The first candidate model is $ARIMA(1,1,1)(1,1,1)$, this is because the non seasonal part includes an autoregressive term of 1 because of the lag 1 spike, there is a difference needed to make it stationary and the decay in the ACF means there is one moving average term, the seasonal part shows a significant spike showing an AR term, with a need of seasonal difference and an moving average term by looking at the ACF plot.

The second candidate model is $ARIMA(2,1,0)(1,1,0)$, There is an additional spike in the PACF plot therefore I decided to do an $AR(2)$ to capture more complex possibilities, difference is still needed like the last model but I removed the moving average this time. The seasonal plot is same as the last model but the seasonal MA is removed as well to see how the model does without it.

The third candidate model is $ARIMA(0,1,2)(0,1,1)$, The model is like the first model as well but this time I removed autoregressive term, this is because the PACF does not show significant lags beyond the first spike and it seems like the MA term is sufficient to capture the dependencies.

The last candidate model $ARIMA(1,1,2)(1,1,1)$ is also very similar to my first model with the only difference of two MA terms being used I did this to test whether or not using multiple MA terms will adequately capture the time series short term dependencies and the decays in the ACF plot and to account for the auto correlations in a different way.

5. 8 Marks Fit your four candidate ARIMA models, as well as the automatic stepwise search model (noting that `stepwise = FALSE` will take too long to run) on the training data. Compare the models using AICc and output the parameter estimates from the model that has the best predictive ability. Write the fitted model equation using backshift notation. Note: If you get warnings in your code, you may have to include a constant in your model. You can do this by starting your model fit with `Temperature ~ 1 + .`

[Hide](#)

```
fit_best <- train_2 %>%
  model(
    arima_111_111 = ARIMA(Temperature ~ 1 + pdq(1, 1, 1) + PDQ(1, 1, 1)),
    arima_210_110 = ARIMA(Temperature ~ 1 + pdq(2, 1, 0) + PDQ(1, 1, 0)),
    arima_012_011 = ARIMA(Temperature ~ 1 + pdq(0, 1, 2) + PDQ(0, 1, 1)),
    arima_112_111 = ARIMA(Temperature ~ 1 + pdq(1, 1, 2) + PDQ(1, 1, 1)),
    arima_auto = ARIMA(Temperature)
  )
```

```
## Warning: Model specification induces a quadratic or higher order polynomial trend.
## This is generally discouraged, consider removing the constant or reducing the number of differences.
## Model specification induces a quadratic or higher order polynomial trend.
## This is generally discouraged, consider removing the constant or reducing the number of differences.
## Model specification induces a quadratic or higher order polynomial trend.
## This is generally discouraged, consider removing the constant or reducing the number of differences.
## Model specification induces a quadratic or higher order polynomial trend.
## This is generally discouraged, consider removing the constant or reducing the number of differences.
```

[Hide](#)

```
glance(fit_best) %>%
  arrange(AICc) %>%
  select(.model, AICc)
```

```
## # A tibble: 5 × 2
##   .model      AICc
##   <chr>      <dbl>
## 1 arima_111_111 786.
## 2 arima_112_111 787.
## 3 arima_012_011 789.
## 4 arima_auto    870.
## 5 arima_210_110 934.
```

[Hide](#)

```
fit_best %>%
  select(arima_111_111) %>%
  report()
```

```
## Series: Temperature
## Model: ARIMA(1,1,1)(1,1,1)[12] w/ poly
##
## Coefficients:
##          ar1      ma1      sar1      sma1  constant
##          0.2687 -1.0000 -0.0402 -1.0000      3e-04
## s.e.  0.0549  0.0341  0.0582  0.0664      1e-04
##
## sigma^2 estimated as 0.5796: log likelihood=-387.1
## AIC=786.19  AICc=786.46  BIC=808.76
```

Best on the output above from the best model arima_111_111 is the best model:

- AR(1) coefficient (ϕ_1): 0.2687
- MA(1) coefficient (θ_1): -1.0000
- Seasonal AR(1) coefficient (Φ_1): -0.0402
- Seasonal MA(1) coefficient (Θ_1): -1.0000

The ARIMA ARIMA(1,1,1)(1,1,1) model equation using backshift notation is:

$$(1-0.2687B)(1+0.0402B^{12})(1-B)(1-B^{12})Y_t = (1+1.0000B)(1+1.0000B^{12})\epsilon_t + 0.0003$$

- B is the backshift operator.
- 12 is the seasonal period (monthly data, assuming seasonality is yearly).

6. 6 Marks Fit your best ETS and ARIMA model on the training data in the same `mable`. Forecast $h = 5$ periods into the future. Plot your forecasts. Compare your forecasts against the truth by computing RMSE, MAE, MAPE, and MASE. State which model has the best forecast accuracy.

[Hide](#)

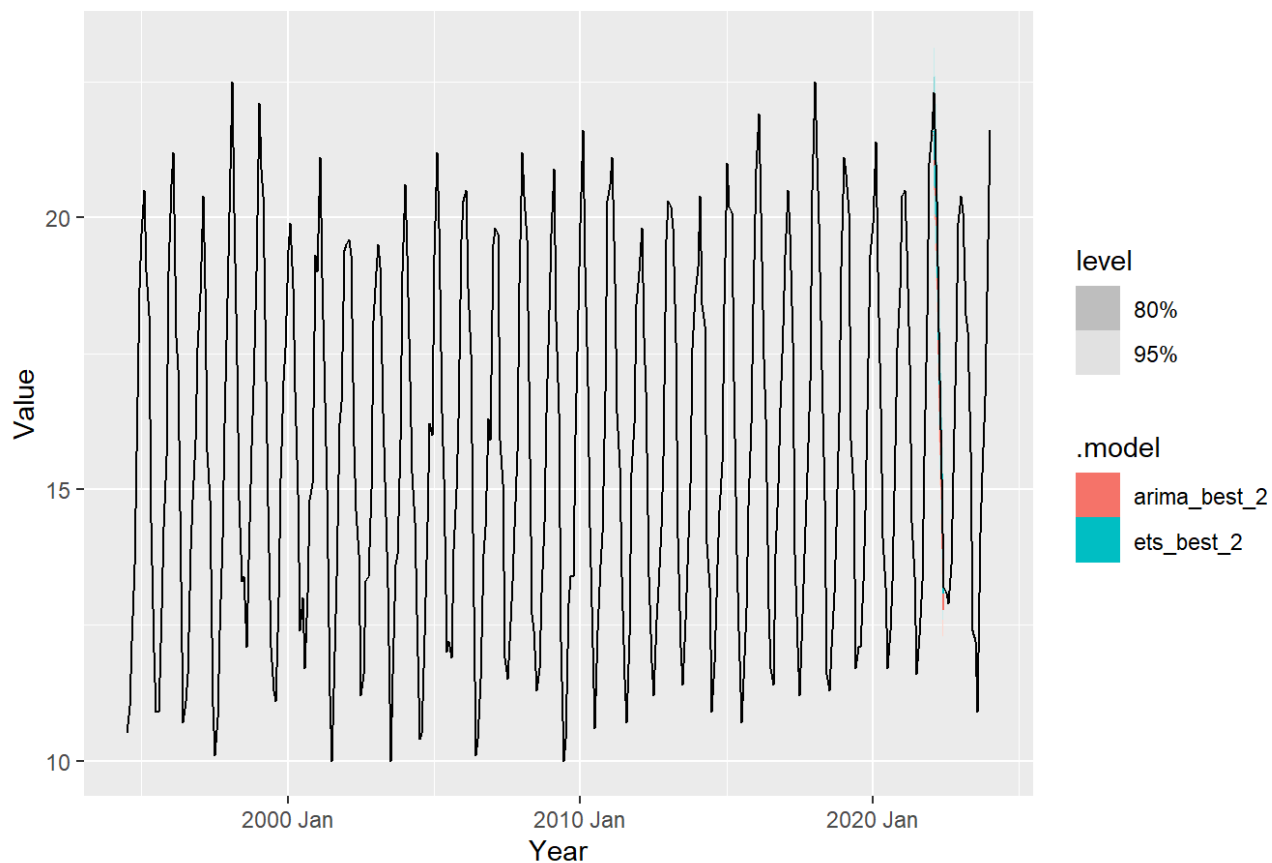
```
mable_2 <- train_2 %>%
  model(
    ets_best_2 = ETS(Temperature ~ error("A") + trend("N") + season("A")),
    arima_best_2 = ARIMA(Temperature ~ pdq(1, 1, 1) + PDQ(1, 1, 1))
  )

forecast_2 <- mable_2 %>%
  forecast(h = 5)

forecast_2 %>%
  autoplot(train_2) +
  autolayer(data_2) +
  ggtitle("Forecast Comparison: ETS vs ARIMA") +
  labs(x = "Year", y = "Value")
```

```
## Plot variable not specified, automatically selected `.vars = Temperature`
```

Forecast Comparison: ETS vs ARIMA



Hide

```
accuracy_measures_2 <- accuracy(forecast_2, data_2) %>%
  select(.model, RMSE, MAE, MAPE, MASE) %>%
  print()
```

```
## # A tibble: 2 × 5
##   .model      RMSE  MAE  MAPE  MASE
##   <chr>      <dbl> <dbl> <dbl> <dbl>
## 1 arima_best_2 0.484 0.441 2.43 0.490
## 2 ets_best_2   0.353 0.273 1.41 0.304
```

Based on the RMSE, MAE, MAPE, and MASE tests the ETS model shows better forecasting accuracy compared to the ARIMA model. Therefore ETS model is proffered forecasting model.

Possible marks for **Problem 2:** 46 Marks

Total possible marks for **Assignment 4:** 80 Marks

