Sanchez Jarrett 20109664

2023-10-08

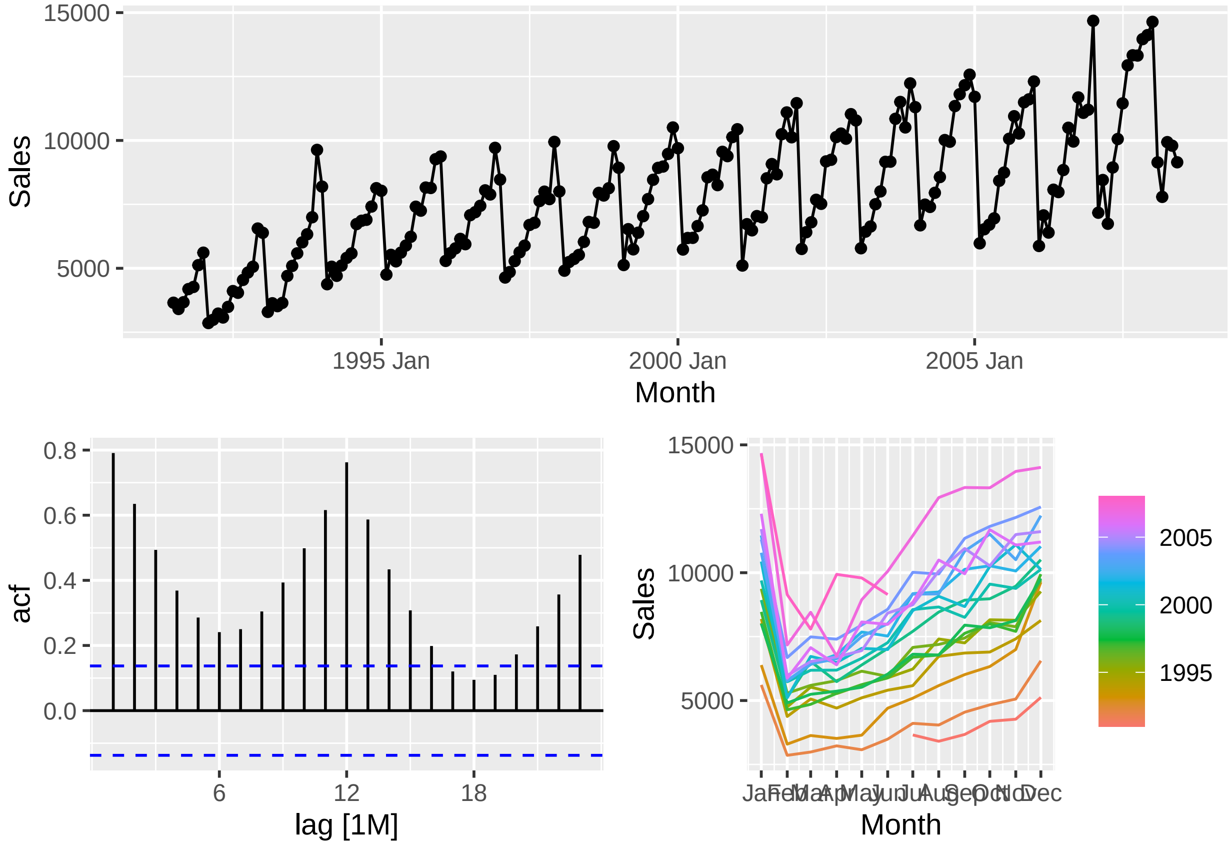
Question 1 - ETS (20 marks)

0) Scale the data (e.g., divide by 100). From now on, you'll work with the scaled series (0 marks).

```
# scale data, convert to time series
steroids <- aus_steroids |>
  mutate(Month = yearmonth(Period), Sales = Sales/100) |>
  select(Month, Sales) |>
  as_tsibble(index = Month)
```

a) Plot the series and discuss the main features, including stationarity (2 marks).

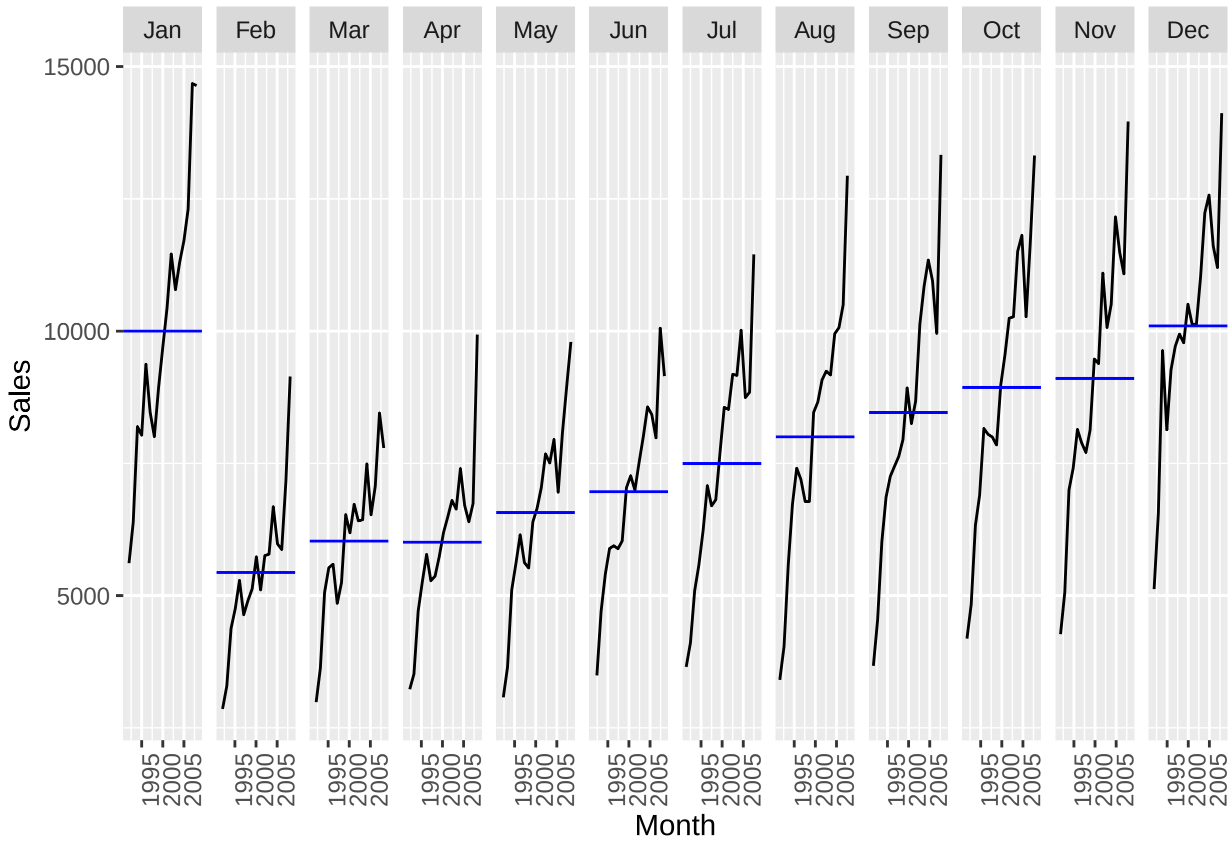
```
steroids |>
  gg_tsdisplay(Sales)
```



The series looks non-stationary. There is a clear upwards trend and a strong seasonal pattern. Variability in the data is non-constant, where the variance appears proportional to the level of series over time.

The seasonal plot on the bottom right confirms seasonality. There appears to be a drop in H03 sales around February annually. This is likely due to Australia's Pharmaceutical Benefits Scheme which subsidises medicine, making it easier for patients to stockpile them at the end of the year. The Scheme's co-payment amount changes on 1 January each year, which explains the increase in sales towards December as patients would like to stockpile medicine before any potential price increase occurs.

```
steroids |>
  gg_subseries(Sales)
```



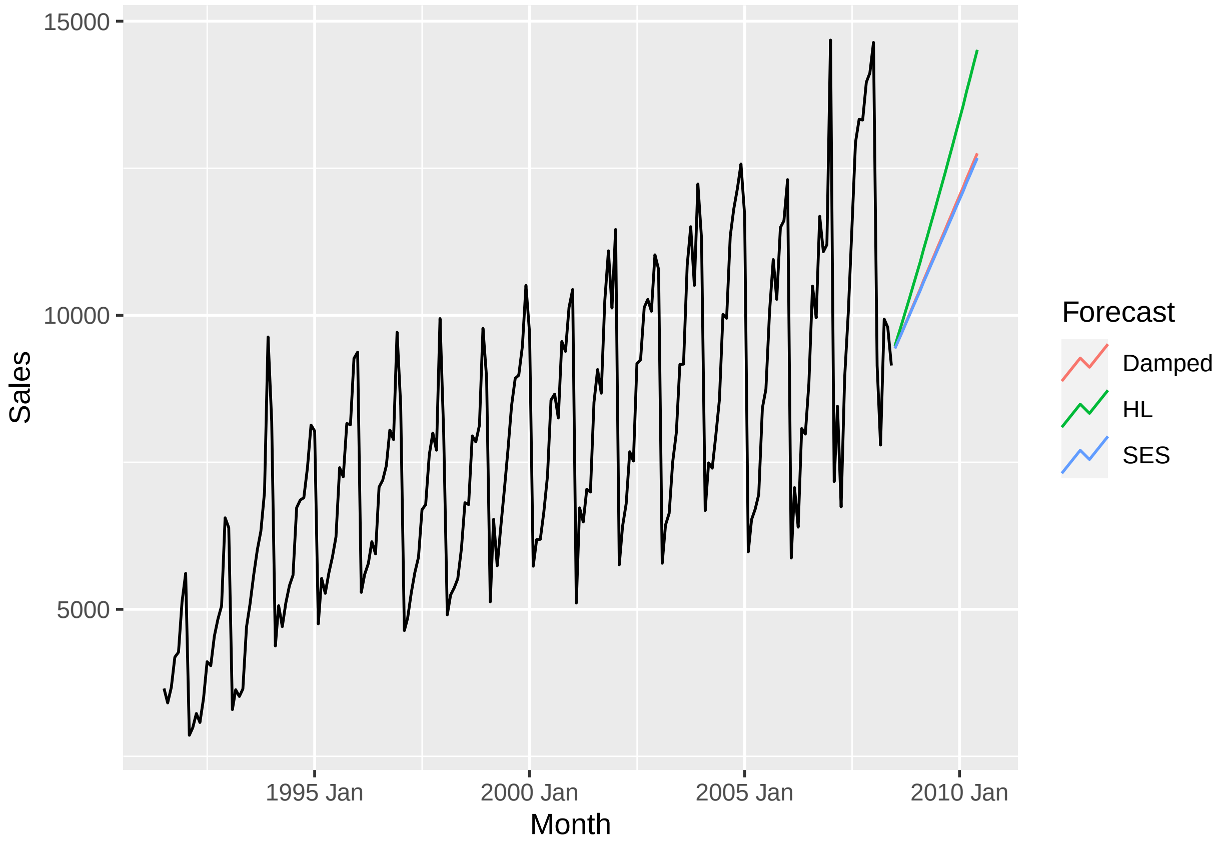
Strong trend in all months, largest trend in January, larger increase in second half of the year (July to December) compared to first half (excluding January).

b) Forecast next two years using SES, Holt's linear and Holt's damped trend. Plot the series and the forecasts. Merely based on this plot, discuss the adequacy of these methodologies to forecast from this series. Explain your answer (5 marks total).

```
# fit models, use log(Sales) to stabilise variance
fit_ses_holt <- steroids |>
  model(
    SES = ETS(log(Sales) ~ error("A") + trend("N") + season("N")),
    HL = ETS(log(Sales) ~ error("A") + trend("A") + season("N")),
    Damped = ETS(log(Sales) ~ error("A") + trend("Ad") + season("N"))
  )
```

```
fc_steroids <- fit_ses_holt |>
  forecast(h="2 years")

fc_steroids |>
  autoplot(steroids, level=NULL) +
  guides(colour=guide_legend(title="Forecast"))
```



As evident from the plot shown above, the SES, Holt's Linear and Damped methods captured the overall upwards trend, but did not seem to capture the seasonal aspect of the series. This occurs since the parameter for seasonality in every model is passed the value "N". More specifically, these methodologies are not adequate at forecasting data with seasonality as: SES is suitable for forecasting data with no clear trend or seasonal pattern, while the Holt's Linear + Damped trend methods are suitable for forecasting data with only trend patterns.

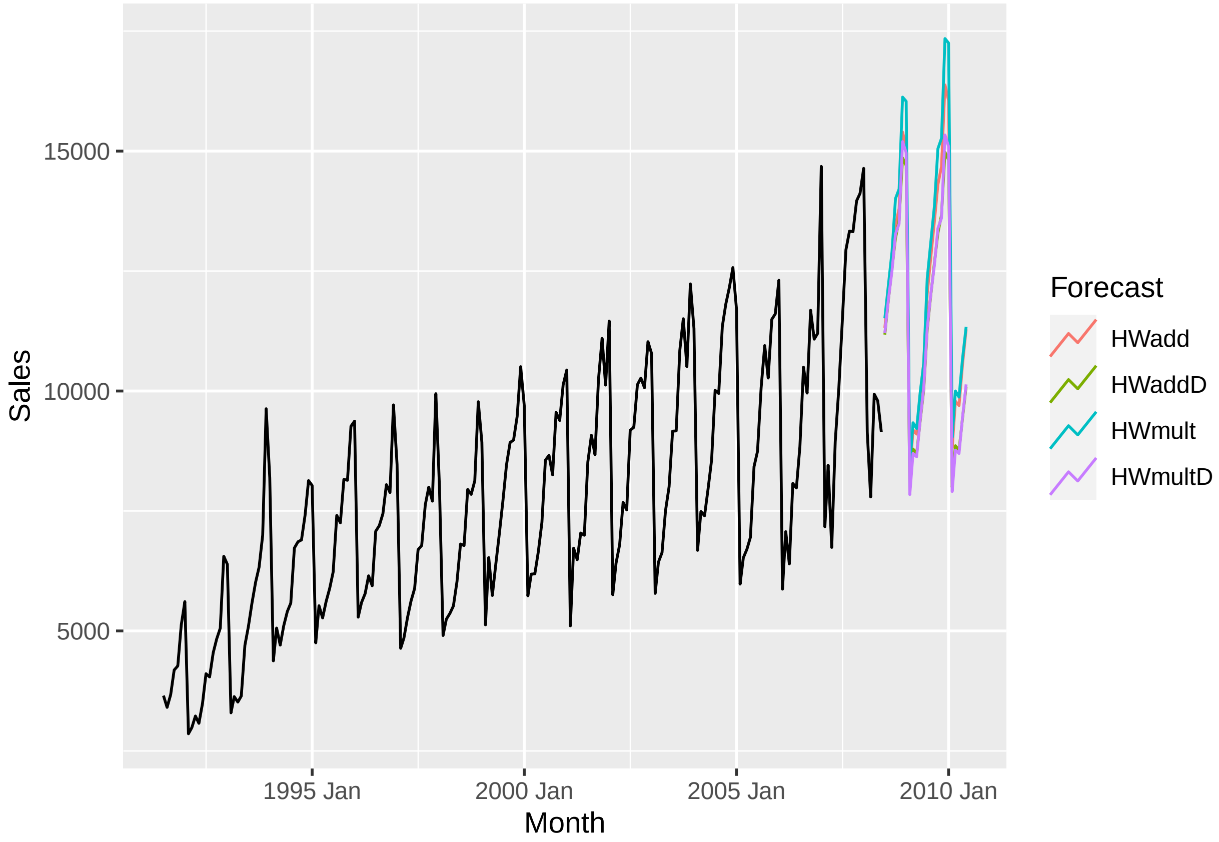
c) Repeat b) with Holt-Winters' seasonal methods. Discuss whether additive or multiplicative seasonality is necessary. Explain your answer (5 marks total).

```
fit_hw <- steroids |>
  model(
    HWadd = ETS(log(Sales) ~ error("A") + trend("A") + season("A")),
    HWmult = ETS(log(Sales) ~ error("M") + trend("A") + season("M")),
    HWaddD = ETS(log(Sales) ~ error("A") + trend("Ad") + season("A")),
    HWmultD = ETS(log(Sales) ~ error("M") + trend("Ad") + season("M"))
  )

fc_steroids <- fit_hw |>
```

```r
  forecast(h="2 years")

fc_steroids |>
  autoplot(steroids, level=NULL) +
  guides(colour=guide_legend(title="Forecast"))
```



For my log-transformed series, I would say that additive seasonality is necessary since the seasonal variations appear roughly constant through the series post-transformation. However, if I was to forecast using the original series which had a non-constant variance where the seasonal variations changed proportional to the level of the series, then I would need to use multiplicative seasonality. This can be confirmed through the accuracy report shown below:

```r
# Note: reverting log(Sales) to Sales in fit_hw will return HW's multiplicative damped
# method with the lowest RMSE and MAE scores, followed by HW's multiplicative method.
fit_hw |>
  fabletools::accuracy() |>
  select(.model, RMSE, MAE) |>
  arrange(RMSE)
```

```
## # A tibble: 4 x 3
##   .model   RMSE   MAE
##   <chr>   <dbl> <dbl>
## 1 HWaddD   574.  413.
## 2 HWmultD  586.  422.
## 3 HWadd    587.  422.
## 4 HWmult   605.  434.
```

d) Compare MSE and MAE of one-step-ahead, four-step-ahead, and six-step-ahead forecasts from methods discussed in b and c above. Report your results neatly and clearly. You can use a Table. Which method has the highest accuracy? Does this selection depend on the number of pre-specified (steps-ahead) forecasts? Explain your answer (5 marks total).

Since RMSE is just a scaled down (square root) MSE, in this report, RMSE will be used for convenience's sake.

One-step-ahead cross-validation:

```r
stretch <- steroids |>
  select(Sales) |>
  stretch_tsibble(.init = 24, .step = 1) |>
  relocate(Month, Sales, .id)

fit_cv <- stretch |>
  model(
    SES = ETS(log(Sales) ~ error("A") + trend("N") + season("N")),
    HL = ETS(log(Sales) ~ error("A") + trend("A") + season("N")),
    Damped = ETS(log(Sales) ~ error("A") + trend("Ad") + season("N")),
    HWadd = ETS(log(Sales) ~ error("A") + trend("A") + season("A")),
    HWmult = ETS(log(Sales) ~ error("M") + trend("A") + season("M")),
    HWaddD = ETS(log(Sales) ~ error("A") + trend("Ad") + season("A")),
    HWmultD = ETS(log(Sales) ~ error("M") + trend("Ad") + season("M"))
  )

# forecast up to 6 steps ahead, data needs to have 6 observations per fold.
test <- new_data(stretch, n=6) |>
  left_join(steroids, by="Month")

fc <- forecast(fit_cv, new_data=test) |>
  group_by(.id) |>
  mutate(h=row_number() %% 6 + 1) |>
  ungroup() |>
  as_fable(response="Sales", distribution = Sales)

fc |>
  accuracy(steroids, by=c("h", ".model")) |>
  select(.model, h, RMSE, MAE) |>
  filter(h %in% c(1,4,6)) |> # only show one/four/six-step-ahead forecasts
  group_by(h) |>
  top_n(-3, wt=RMSE) |>
  ungroup() |>
  arrange(h, RMSE)
```

```
## Warning: The future dataset is incomplete, incomplete out-of-sample data will be treated as missing.
## 6 observations are missing between 2008 Jul and 2008 Dec
```

```
## # A tibble: 9 x 4
##   .model      h  RMSE   MAE
##   <chr>   <dbl> <dbl> <dbl>
## 1 HWaddD      1  862.  659.
## 2 HWmultD     1  896.  688.
## 3 HWadd       1  956.  739.
## 4 HWaddD      4  732.  566.
## 5 HWmultD     4  759.  588.
## 6 HWadd       4  796.  619.
## 7 HWaddD      6  816.  644.
## 8 HWmultD     6  838.  661.
## 9 HWadd       6  886.  702.
```

From the one/four/six-step-ahead forecast accuracy results, the Holt-Winters' additive damped method performed with the best accuracy for my log-transformed data set on all three occasions. Specifically for this data set, the selection does not seem to depend on the number of pre-specified steps-ahead forecasts. For other data sets however, we may observe bigger fluctuations depending on the errors observed, meaning other models may outperform with different forecasting horizons. For example, MSE is more sensitive to outliers, therefore one or few extreme predictions from a model can be the difference for whether it is selected or not.

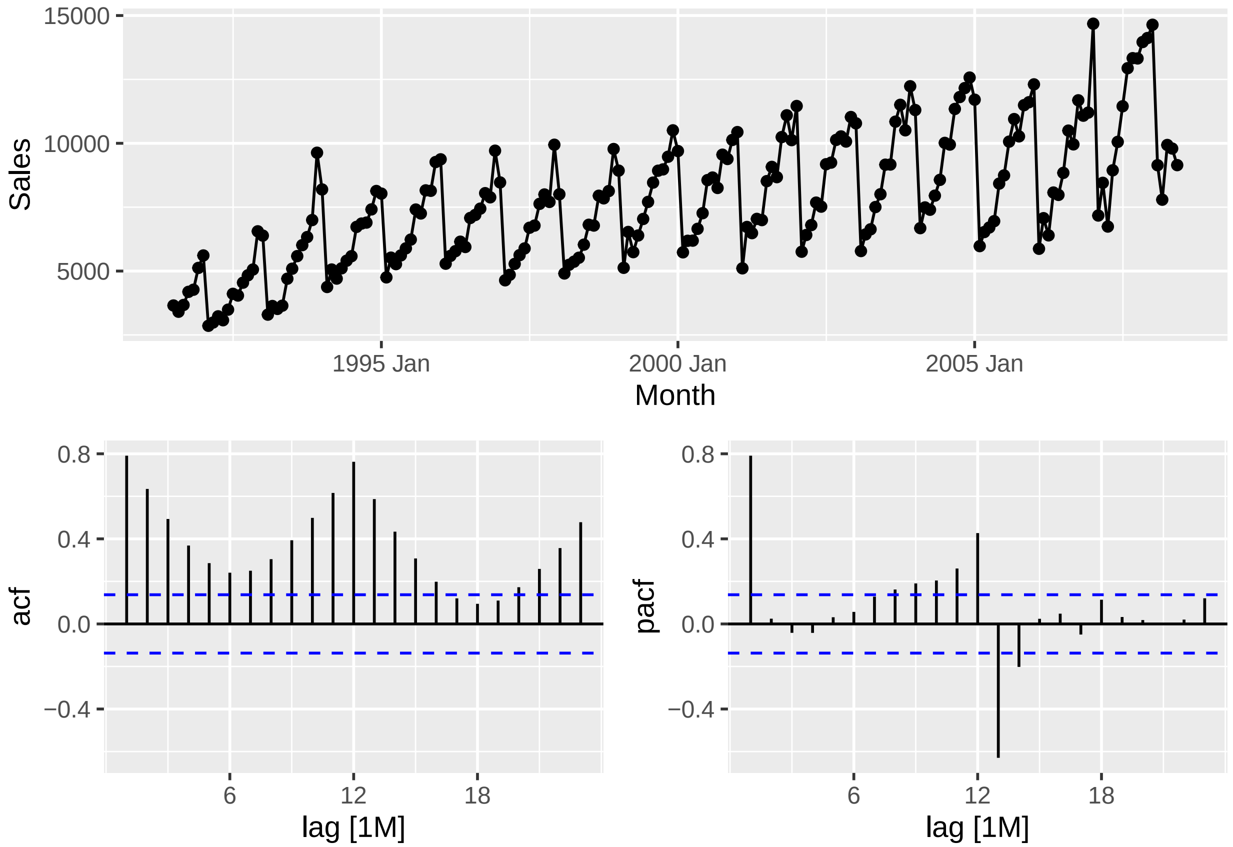
e) Briefly discuss the potential mistake/error we may unintentionally introduce in the discussion when comparing models from b) and c) using the MSE and MAE. (3 marks)

A potential mistake that may be unintentionally introduced is the judgement of selecting the best method of forecasting based on the lowest MSE/MAE scores alone. Reasons for this are: the predictive-ability measures from the training data can only be compared when methods have the same number of parameters to estimate, and selecting a model based on lowest MSE/MAE scores may favour complex models that could lead to overfitting.

Question 2 - Stationarity (20 marks)

a) Plot ACF and partial ACF. (8 marks total for both parts)

```
steroids |>
  gg_tsdisplay(Sales, plot_type = 'partial')
```



a.1) Briefly discuss the stationarity of the series based on the ACF. Does your answer here conform with your answer to Question 1a)?

Based on the ACF, it is evident that the series is non-stationary. The seasonal component is captured through a sinusoidal pattern, where the ACF of the data decreases per season slowly as the time goes by, signifying a non-stationary data. There is also a large and positive autocorrelation value (approximately 0.8) at lag 1, further indicating that the series is non-stationary. This conforms to my answer to Question 1a) where I discussed non-stationarity through trend, seasonal pattern, and non-constant variance.

a.2) Should series be differenced in order to obtain a stationary series? Explain your answer.

```
steroids |>
  features(Sales, unitroot_kpss)
```

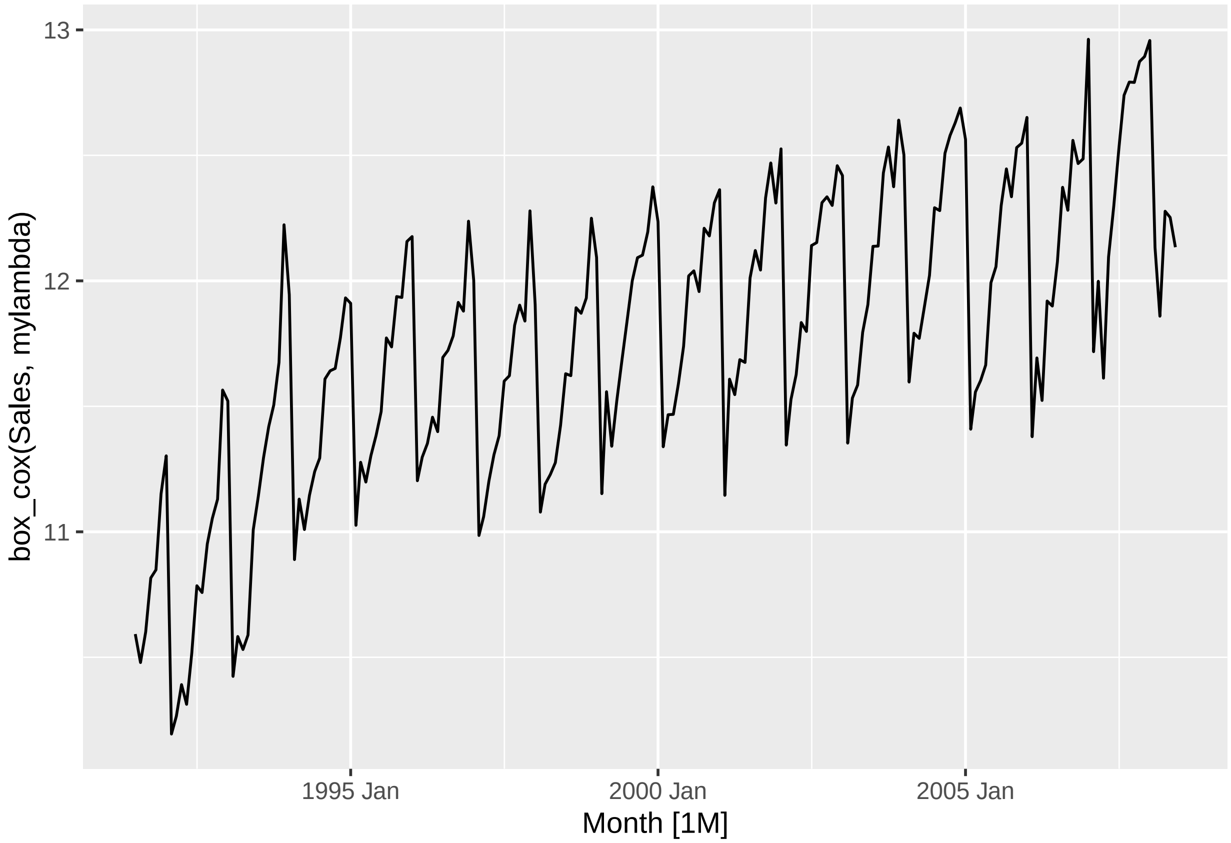
```
## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##       <dbl>       <dbl>
## 1      2.85        0.01
```

Null states that the series is stationary. Since we have a significant p-value 0.01 from the KPSS test, then the evidence does not support null, suggesting that differencing is required.

b) Find appropriate Box-Cox transformation and order of differencing to obtain stationary data. Justify answer even if no Box-Cox transformation required. (12 marks total)

```
mylambda <- steroids |>
  features(Sales, features=guerrero) |>
  pull(lambda_guerrero)

steroids |>
  autoplot(box_cox(Sales, mylambda))
```



```
steroids |>
  mutate(bc_sales = box_cox(Sales, mylambda)) |>
  features(bc_sales, unitroot_nsdiffs)
```

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

Return value 1 indicates one seasonal difference is required.

```
steroids |>
  mutate(bc_sales = difference(box_cox(Sales, mylambda), lag=12)) |>
  features(bc_sales, unitroot_ndiffs)
```

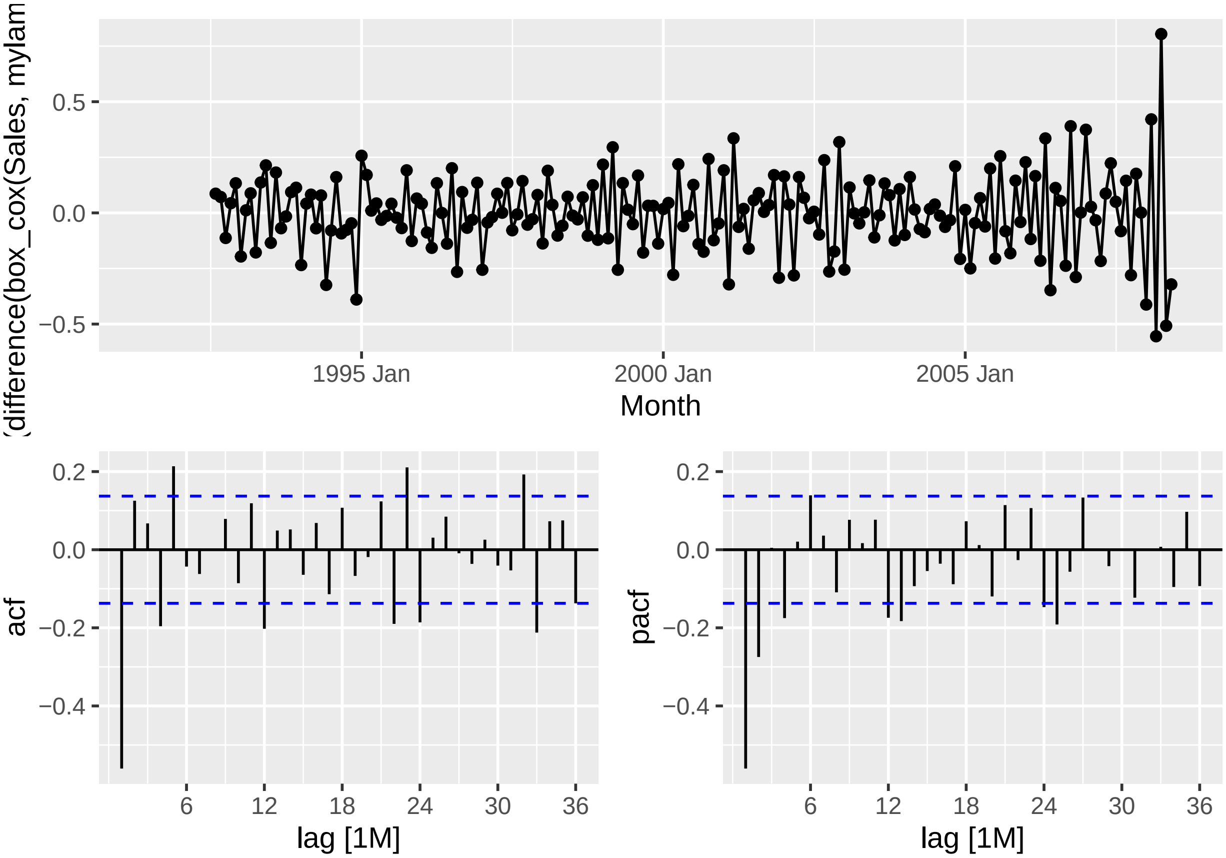
```
## # A tibble: 1 x 1
##   ndiffs
##     <int>
## 1       1
```

Return value 1 indicates one ordinary difference is required after seasonal differencing.

```
# remove seasonality and trend through seasonal differencing, then ordinary differencing
steroids |>
  gg_tsdisplay(difference(difference(box_cox(Sales, mylambda), lag=12)), plot_type = 'partial', lag_max=
```

```
## Warning: Removed 13 rows containing missing values (‘geom_line()‘).

## Warning: Removed 13 rows containing missing values (‘geom_point()‘).
```



Given the differenced plot and the unit root tests conducted above, I would say that the data is now sufficiently stationary. This can be confirmed by conducting another KPSS test:

```
steroids |>
  features(difference(difference(box_cox(Sales, mylambda), lag=12)), unitroot_kpss)
```

```
## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##       <dbl>       <dbl>
## 1    0.0348         0.1
```

A p-value of 0.1 is greater than 0.05, therefore there is not enough evidence to reject null that the series is stationary after seasonal and ordinary differencing.

Question 3 - Seasonal & non-seasonal ARIMA modelling (30 marks)

a) By studying the appropriate graphs of the series in R, propose an appropriate ARIMA(p, d, q) or ARIMA(p, d, q)(P, D, Q) structure to model the series. Explain your answer. Plot/figures can be included as part of your answer (8 marks total).

Using ACF:

Non-seasonal (q): significant spike at lag 1 in ACF suggests non-seasonal MA(1) component.

Seasonal (Q): significant spikes at lag 12 and 24 in ACF suggests seasonal MA(2) component.

Possible model: ARIMA(0,1,1)(0,1,2)

Using PACF:

Non-seasonal(p): significant spikes at at lag 1 and 2 in PACF suggests non-seasonal AR(1) and AR(2) components.

Seasonal (P): significant spikes at lag 12 and 24 in PACF suggests seasonal AR(2) component.

Possible models: ARIMA(1,1,0)(2,1,0) and ARIMA(2,1,0)(2,1,0)

Variants to consider: ARIMA(1,1,1)(0,1,2), ARIMA(1,1,1)(2,1,2), ARIMA(1,1,0)(2,1,2), and ARIMA(2,1,0)(2,1,2)

```
bc_steroids <- steroids |>
  mutate(bc_sales = box_cox(Sales, mylambda)) |>
  select(Month, bc_sales)

fit_arima <- bc_steroids |>
  model(
    arima011012 = ARIMA(bc_sales ~ pdq(0,1,1) + PDQ(0,1,2)),
    arima111012 = ARIMA(bc_sales ~ pdq(1,1,1) + PDQ(0,1,2)),
    arima111212 = ARIMA(bc_sales ~ pdq(1,1,1) + PDQ(2,1,2)),
    arima110210 = ARIMA(bc_sales ~ pdq(1,1,0) + PDQ(2,1,0)),
    arima210210 = ARIMA(bc_sales ~ pdq(2,1,0) + PDQ(2,1,0)),
    arima110212 = ARIMA(bc_sales ~ pdq(1,1,0) + PDQ(2,1,2)),
    arima210212 = ARIMA(bc_sales ~ pdq(2,1,0) + PDQ(2,1,2))
  )

glance(fit_arima) |>
  arrange(AICc) |>
  select(.model:BIC)
```

```
## # A tibble: 7 x 6
##   .model      sigma2 log_lik   AIC  AICc   BIC
##   <chr>        <dbl>   <dbl> <dbl> <dbl> <dbl>
## 1 arima210212 0.0136    136. -258. -258. -236.
## 2 arima111212 0.0138    135. -255. -254. -232.
## 3 arima111012 0.0144    131. -252. -251. -235.
## 4 arima011012 0.0148    127. -247. -247. -234.
## 5 arima210210 0.0153    127. -244. -244. -228.
## 6 arima110212 0.0149    126. -240. -240. -221.
## 7 arima110210 0.0181    112. -215. -215. -202.
```

I propose ARIMA(2,1,0)(2,1,2) as the most appropriate ARIMA model to structure the series seeing that it has the lowest AICc value. Seasonal components are necessary since the data contains seasonality. Further justification for these values were discussed above the code.

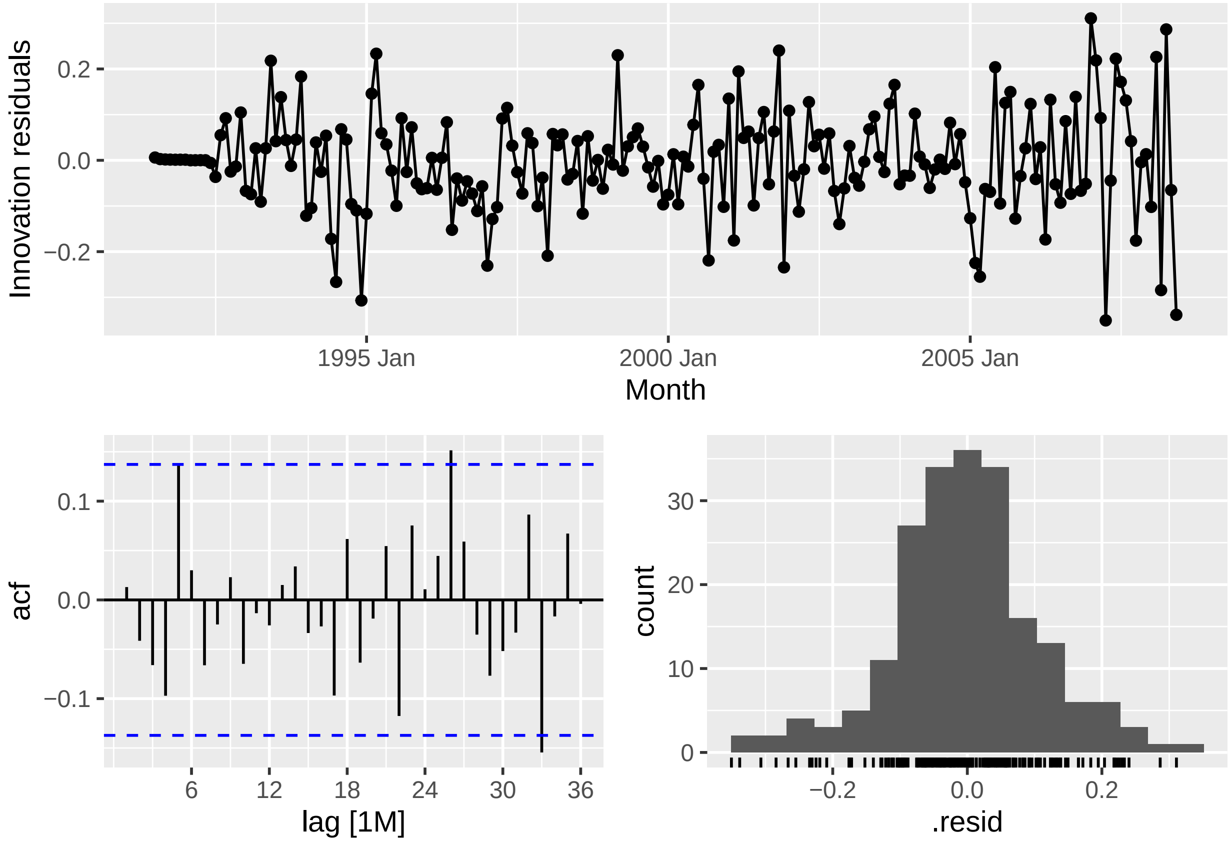
b) Should a constant be included in the model? Justify your answer. (2 marks)

A constant can be included in the model IF it improves the AICc value. Since the proposed model has d=1, then a constant can be included. If d was greater than 1 however, the constant cannot be used, as a higher order trend is dangerous when forecasting.

c) Fit the ARIMA model proposed in 3(a) using R and examine the residuals. Is the proposed model satisfactory? Explain your answer (8 marks total).

```
fit_final <- bc_steroids |>
  model(arima210212 = ARIMA(bc_sales ~ pdq(2,1,0) + PDQ(2,1,2)))

fit_final |>
  gg_tsresiduals(lag=36)
```



From the plot above, we can see that the residuals hover around zero and is approximately normally distributed. There are also only two significant spikes out of 36 lags, therefore the data is consistent with white noise.

This can be confirmed through a Ljung-Box test:

```
augment(fit_final) |>
  features(.innov, ljung_box, lag=36, dof=6)
```

```
## # A tibble: 1 x 3
##   .model      lb_stat lb_pvalue
##   <chr>         <dbl>     <dbl>
## 1 arima210212    37.8     0.155
```

Large p-value 0.15 > 0.05 confirms that residuals are similar to white noise, therefore confirming that the proposed model is satisfactory.

d) Now, let ARIMA() choose an ARIMA model for this data. Does ARIMA() return the same model as the one chosen in 3(a)? If not, which model do you think suits best? Explain your answer. (8 marks total)

```r
fit_auto <- bc_steroids |>
  model(
    auto = ARIMA(bc_sales, stepwise=FALSE, approx=FALSE)  # make R work for better model
  )

glance(fit_auto) |>
  arrange(AICc) |>
  select(.model:BIC)
```

```
## # A tibble: 1 x 6
##   .model sigma2 log_lik   AIC  AICc   BIC
##   <chr>   <dbl>   <dbl> <dbl> <dbl> <dbl>
## 1 auto   0.0136    136. -258. -258. -236.
```

```r
fit_auto
```

```
## # A mable: 1 x 1
##                         auto
##                      <model>
## 1 <ARIMA(2,1,0)(2,1,2)[12]>
```

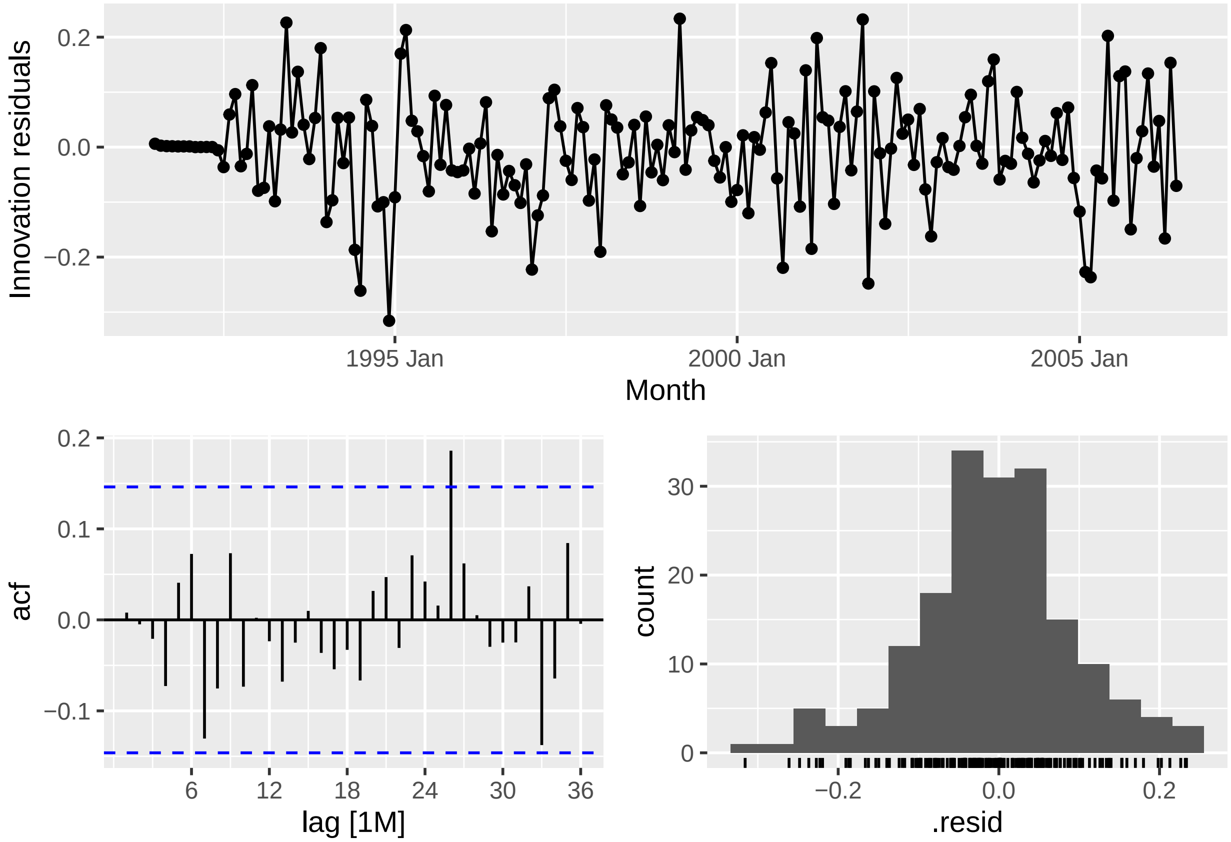
ARIMA returns the same model chosen in 3(a). I believe that this was the case because I found it as one of the variants to consider just from manual selection. This confirms that the model ARIMA(2,1,0)(2,1,2) suits best.

e) Which method do you think is the best between ETS and ARIMA to forecast from this series (compare Q3 and Q1 results). Justify your answer. (4 marks)

```r
train_compare <- bc_steroids |>
  select(bc_sales) |>
  slice(1:180)

fit_arima2 <- train_compare |>
  model(auto = ARIMA(bc_sales, stepwise=FALSE, approx=FALSE))

fit_arima2 |>
  gg_tsresiduals(lag_max=36)
```

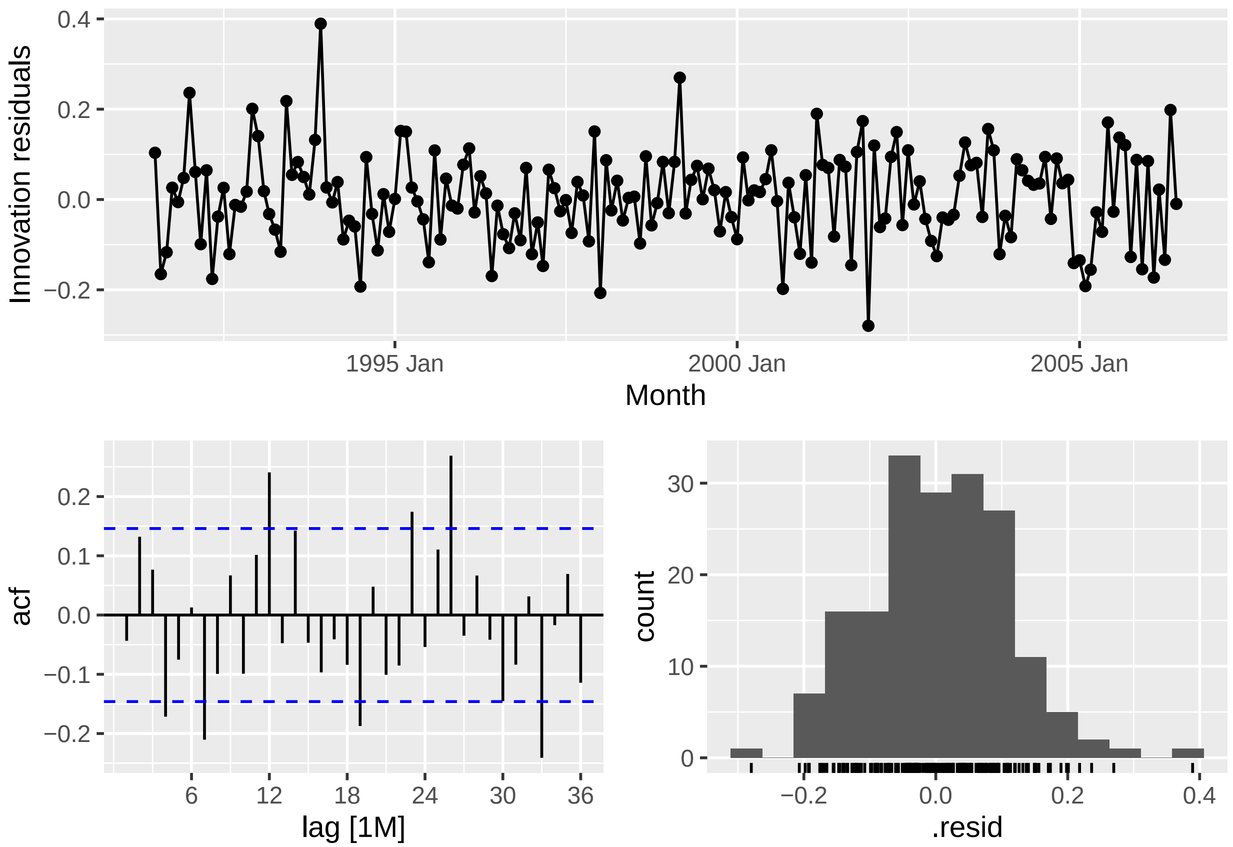


```
augment(fit_arima2) |>
  features(.innov, ljung_box, lag=16, dof=6)
```

```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 auto      10.2     0.426
```

```
fit_ets <- train_compare |>
  model(HWaddD = ETS(bc_sales ~ error("A") + trend("Ad") + season("A")))

fit_ets |>
  gg_tsresiduals(lag_max=36)
```



```r
augment(fit_ets) |>
  features(.innov, ljung_box, lag=16)
```

```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 HWaddD    44.3  0.000179
```

```r
bind_rows(
  fit_arima2 |> accuracy(),
  fit_ets |> accuracy(),
  fit_arima2 |> forecast(h=24) |> accuracy(bc_steroids),
  fit_ets |> forecast(h=24) |> accuracy(bc_steroids)
) |>
  select(.model:MASE)
```

```
## # A tibble: 4 x 8
##   .model .type          ME   RMSE    MAE     MPE  MAPE  MASE
##   <chr>  <chr>       <dbl>  <dbl>  <dbl>   <dbl> <dbl> <dbl>
## 1 auto   Training -0.00559 0.0976 0.0737 -0.0487 0.628 0.455
## 2 HWaddD Training  0.00454 0.102  0.0806  0.0347 0.689 0.498
## 3 auto   Test      0.197   0.270  0.228   1.58   1.84  1.41
## 4 HWaddD Test      0.273   0.344  0.297   2.19   2.38  1.83
```

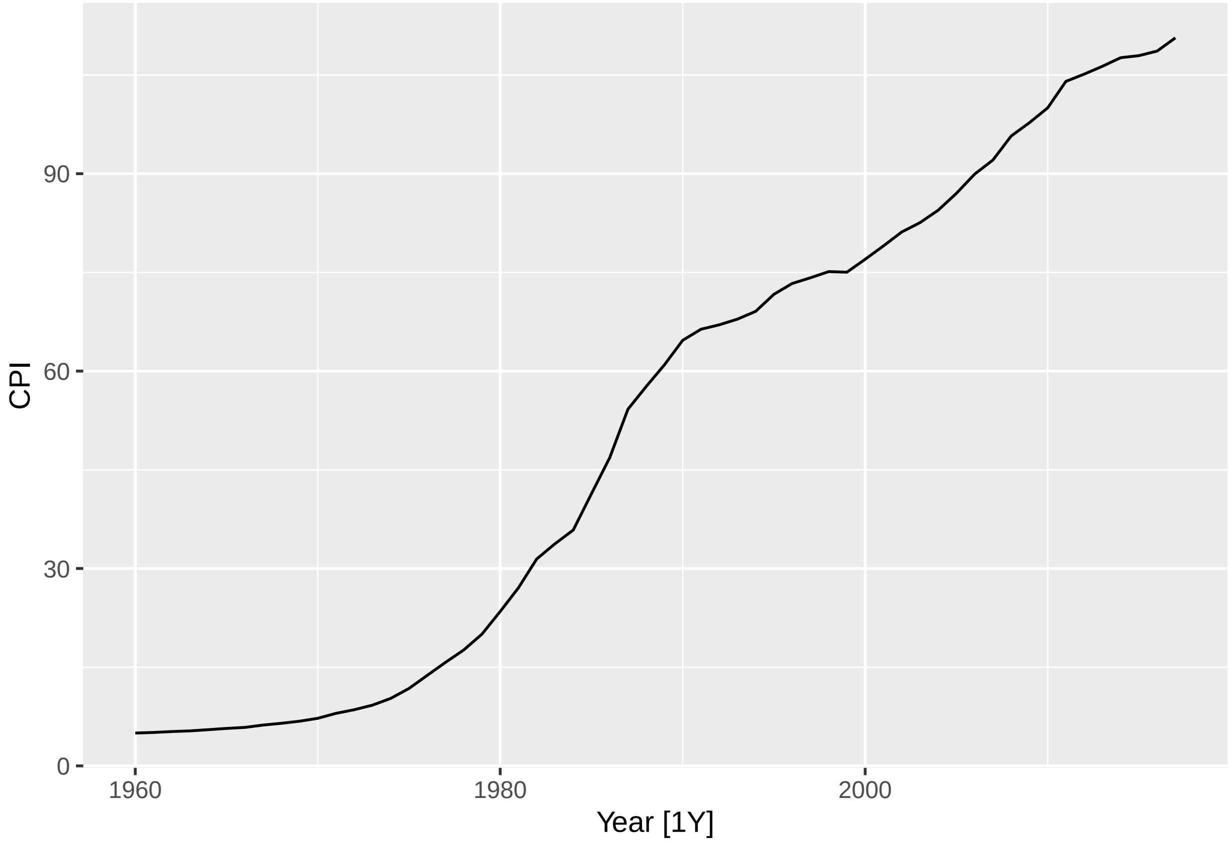
The plot shows that the ARIMA model is slightly more accurate based on the test set RMSE, MAPE, and MASE. We also observe from the Ljung-Box test that the ETS model's residuals do not resemble white noise with a low p-value of 0.00018. Therefore, the ARIMA model performs best between the two models when forecasting this series.

Question 4 - Seasonality and the function accuracy() (10 marks)

a) Fit Holt's Linear with no damping parameter, a Holt-Winters additive and Holt-Winters multiplicative model to NZ CPI from global_economy data set. Compare in-sample accuracy with function accuracy. The output will return NaN for the Holt-Winters models. Why is this happening? Write down a short paragraph discussing this question (5 marks total).

```
nz_cpi <- global_economy |>
  filter(Country == "New Zealand") |>
  select(CPI)

nz_cpi |>
  autoplot(CPI)
```



```
fit_cpi <- nz_cpi |>
  model(
    HL = ETS(CPI ~ error("A") + trend("A") + season("N")),
    HWadd = ETS(CPI ~ error("A") + trend("A") + season("A")),
    HWmult = ETS(CPI ~ error("M") + trend("A") + season("M"))
  )
```

```
## Warning: 1 error encountered for HWadd
## [1] A seasonal ETS model cannot be used for this data.
```

```
## Warning: 1 error encountered for HWmult
## [1] A seasonal ETS model cannot be used for this data.
```

```
fit_cpi |>
  accuracy()
```

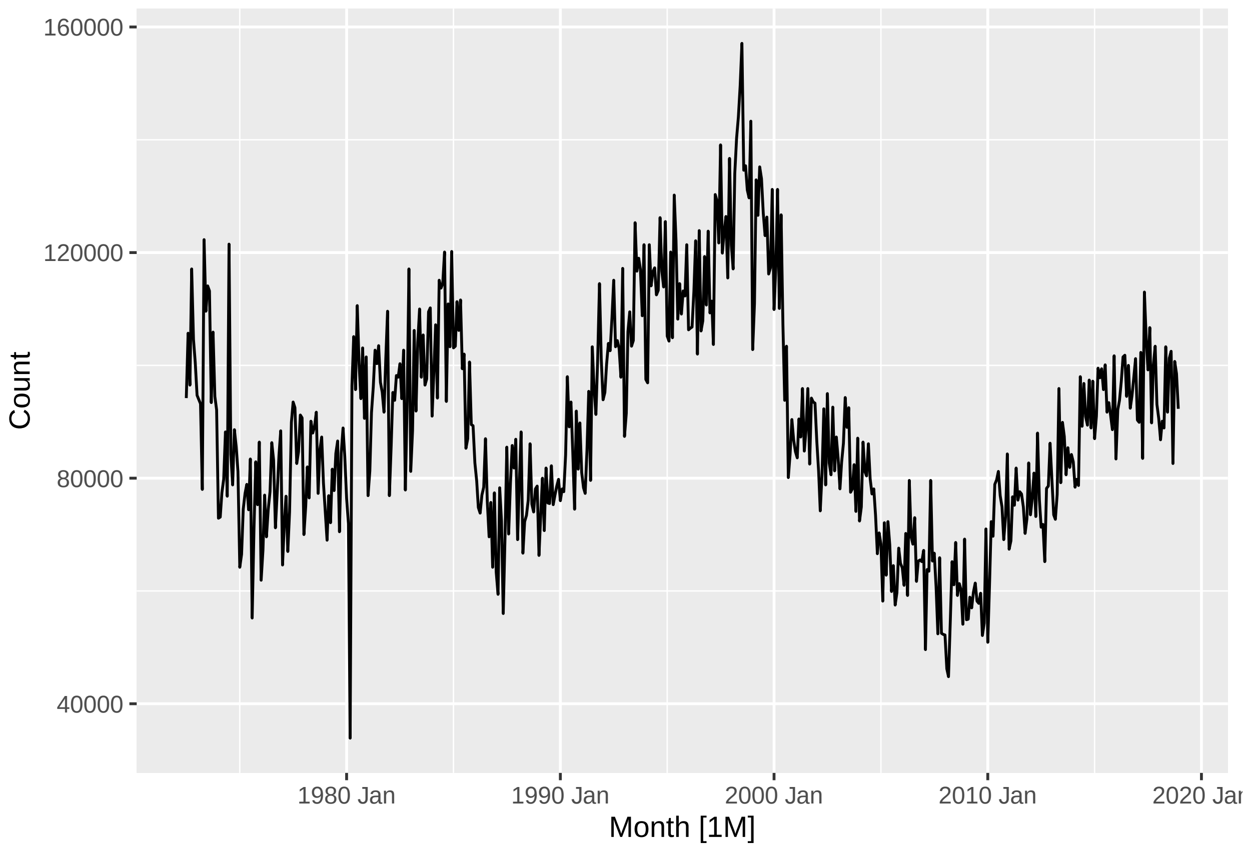
```
## # A tibble: 3 x 10
##   .model .type         ME  RMSE   MAE   MPE  MAPE   MASE  RMSSE   ACF1
##   <chr>  <chr>      <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>  <dbl>  <dbl>
## 1 HL     Training  0.0447  1.08 0.759 0.721  1.86  0.409  0.448 0.0319
## 2 HWadd  Training NaN     NaN   NaN   NaN   NaN   NaN    NaN    NA
## 3 HWmult Training NaN     NaN   NaN   NaN   NaN   NaN    NaN    NA
```

Holt-Winters models return NaN as there is very little to no seasonality observed for the NZ CPI series (see the plot above). This means that the seasonal component is not necessary since there is no seasonality to capture. The model attempts to estimate a seasonal component, even though no meaningful seasonality exists in the data.

b) Now, repeat this analysis with the number of pigs slaughtered in Victoria, available in the data set aus_livestock. Did you observe any warnings from Holt-Winters? Why did you get no errors as opposed to (a)? Briefly explain your answer - compare to 4(a). (5 marks total)

```
vic_pigs <- aus_livestock |>
  filter(Animal == "Pigs", State == "Victoria")

vic_pigs |>
  autoplot(Count)
```



```
fit_pigs <- vic_pigs |>
  model(
    HL = ETS(Count ~ error("A") + trend("A") + season("N")),
    HWadd = ETS(Count ~ error("A") + trend("A") + season("A")),
    HWmult = ETS(Count ~ error("M") + trend("A") + season("M"))
  )

fit_pigs |>
  accuracy()
```

```
## # A tibble: 3 x 12
##   Animal State  .model .type    ME  RMSE   MAE    MPE  MAPE  MASE RMSSE     ACF1
##   <fct>  <fct>  <chr>  <chr> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>    <dbl>
## 1 Pigs   Victo~ HL     Trai~  417. 9386. 7244. -0.430  8.36 0.782 0.755  0.00804
## 2 Pigs   Victo~ HWadd  Trai~  222. 7755. 5747. -0.274  6.74 0.620 0.624 -0.0386
## 3 Pigs   Victo~ HWmult Trai~  355. 7794. 5849. -0.144  6.84 0.631 0.627  0.0151
```

No warnings from Holt-Winters method this time around. This is because there is evident seasonality observed for this series (see the plot above). This allows for the model to estimate a seasonal component as opposed to the time series from 4(a).

Question 5 - Select the correct answer and explain as requested (20 marks)

a) In general, prediction intervals from the ARIMA models increase as the forecast horizon increases. Explain your answer.

True.

Predictions become less certain and accurate as the model makes forecasts for farther time points. This could be due to errors adding up over time, or the fact that there is limited historical data to work with, therefore the model may struggle to capture long-term trends, thus increasing the prediction intervals/uncertainty.

b) The AICc cannot be used to compare between ARIMA and ETS models. Explain your answer.

True.

AICc cannot be used to compare between ETS and ARIMA models because they are in different model classes, and the likelihood is computed in different ways.

c) Time series cross-validation can be used to compare between ARIMA and ETS models. Explain your answer.

True.

Cross-validation can be used to compare between ARIMA and ETS models. This is because cross-validation splits the data into training and test sets, where both models are trained on one segment, and then tests both the models' predictions on the other. Since this method evaluates how well the models can forecast unseen data, we can then determine which of the models work better for the time series.

d) After a deep analysis, the ETS model was selected for forecasting based on its forecasting performance from the test set. Explain your answer.

False.

The ETS model was not selected. It was the ARIMA model that was selected based on the test set RMSE, MAPE, and MASE since this model had less errors, therefore a better forecasting performance compared to the ETS model.