Sanchez Jarrett 20109664

2023-10-08

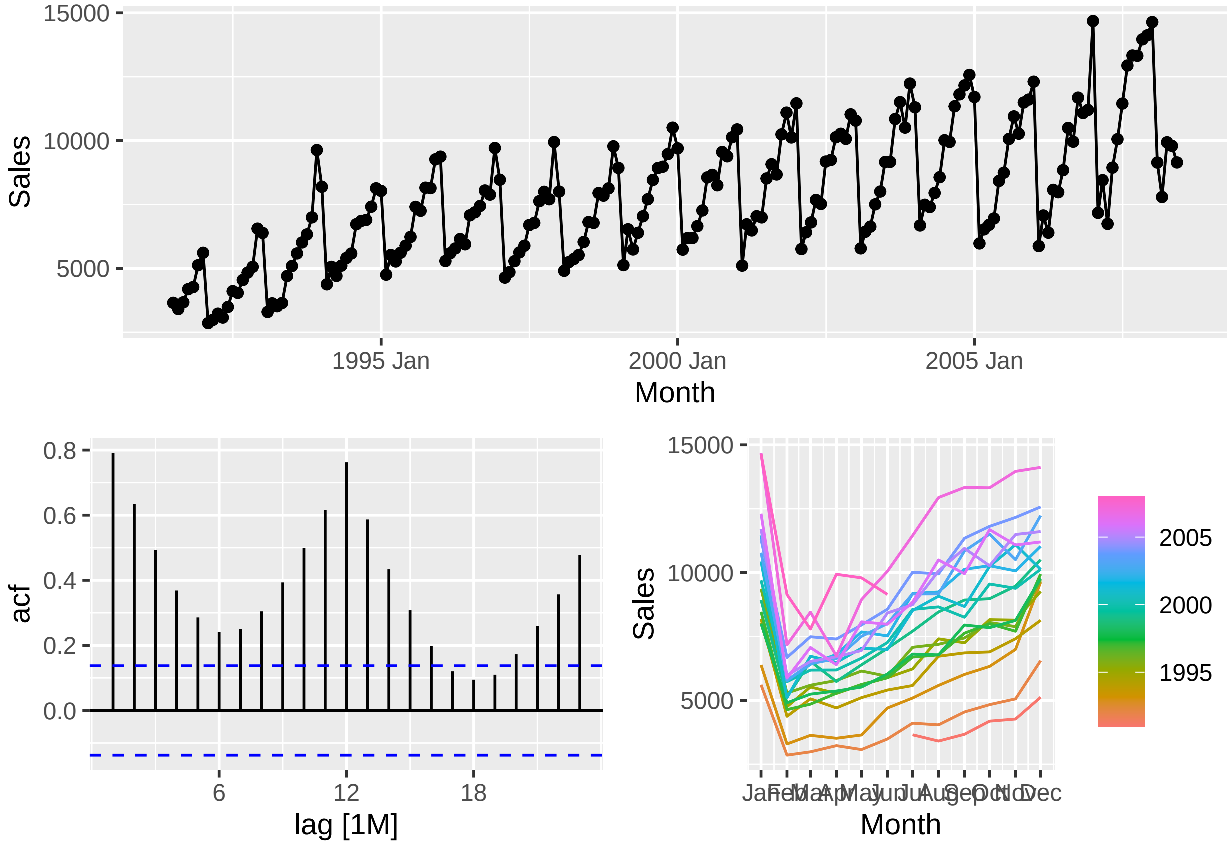
Question 1 - ETS (20 marks)

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

```r
# 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).

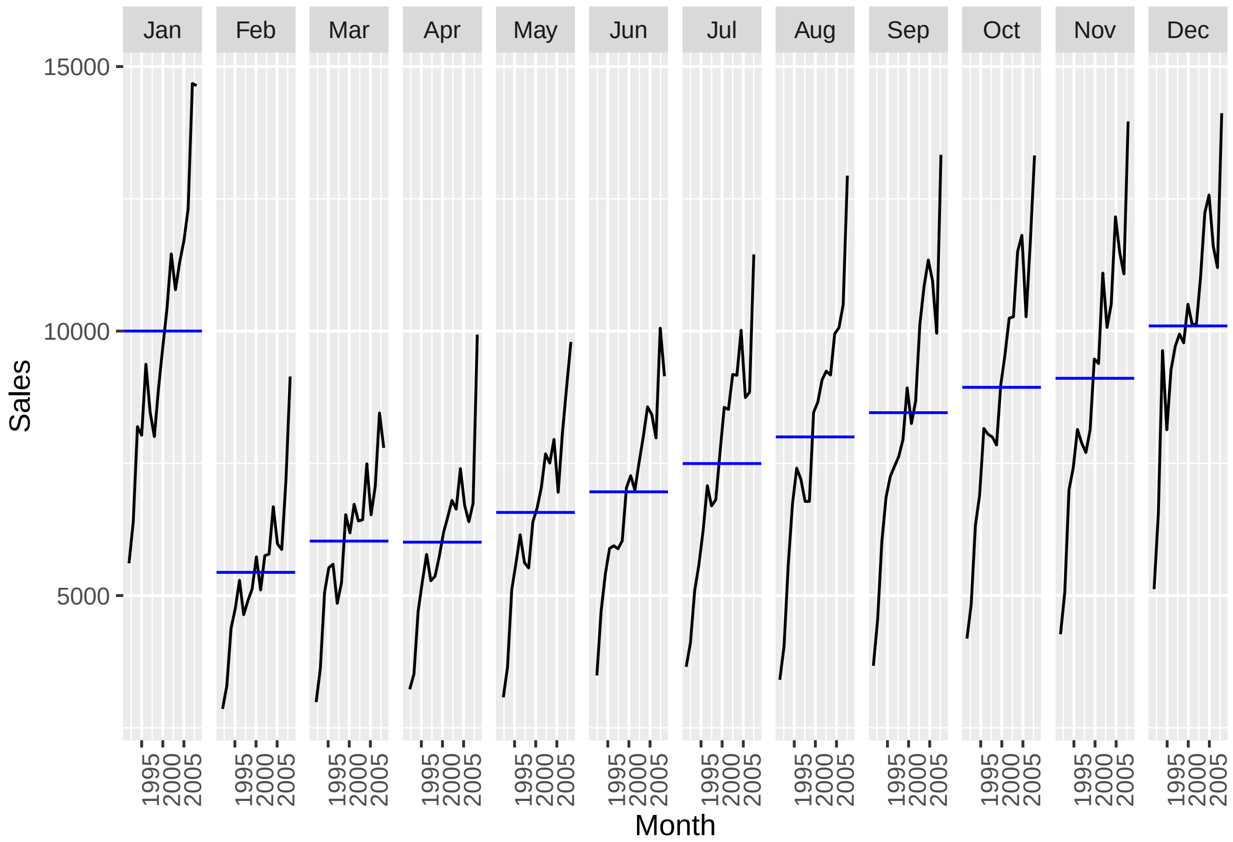
```r
steroids |>
  gg_tsdisplay(Sales)
```



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)
```



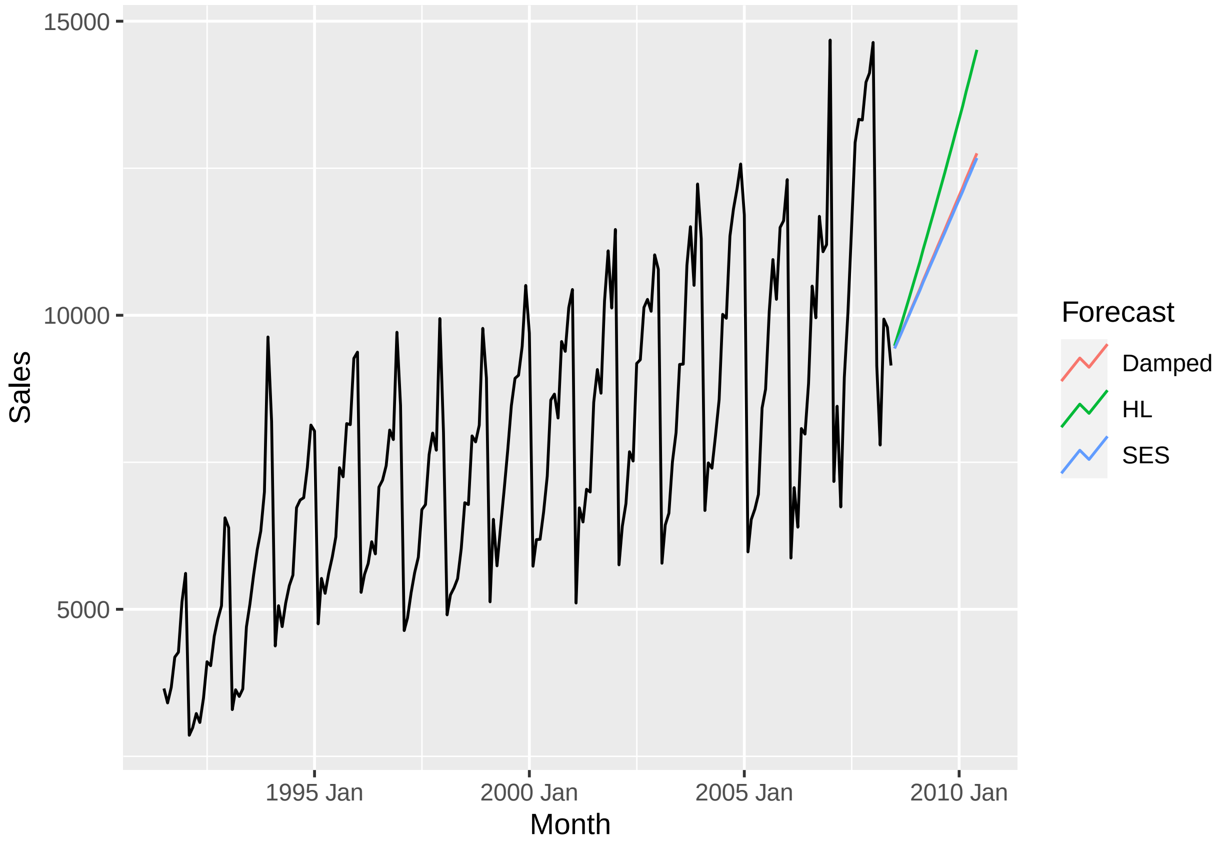
Strong trend in all months, largest trend in January, larger increase in second half of the year (July to December) compared to first half (excluding January).

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.

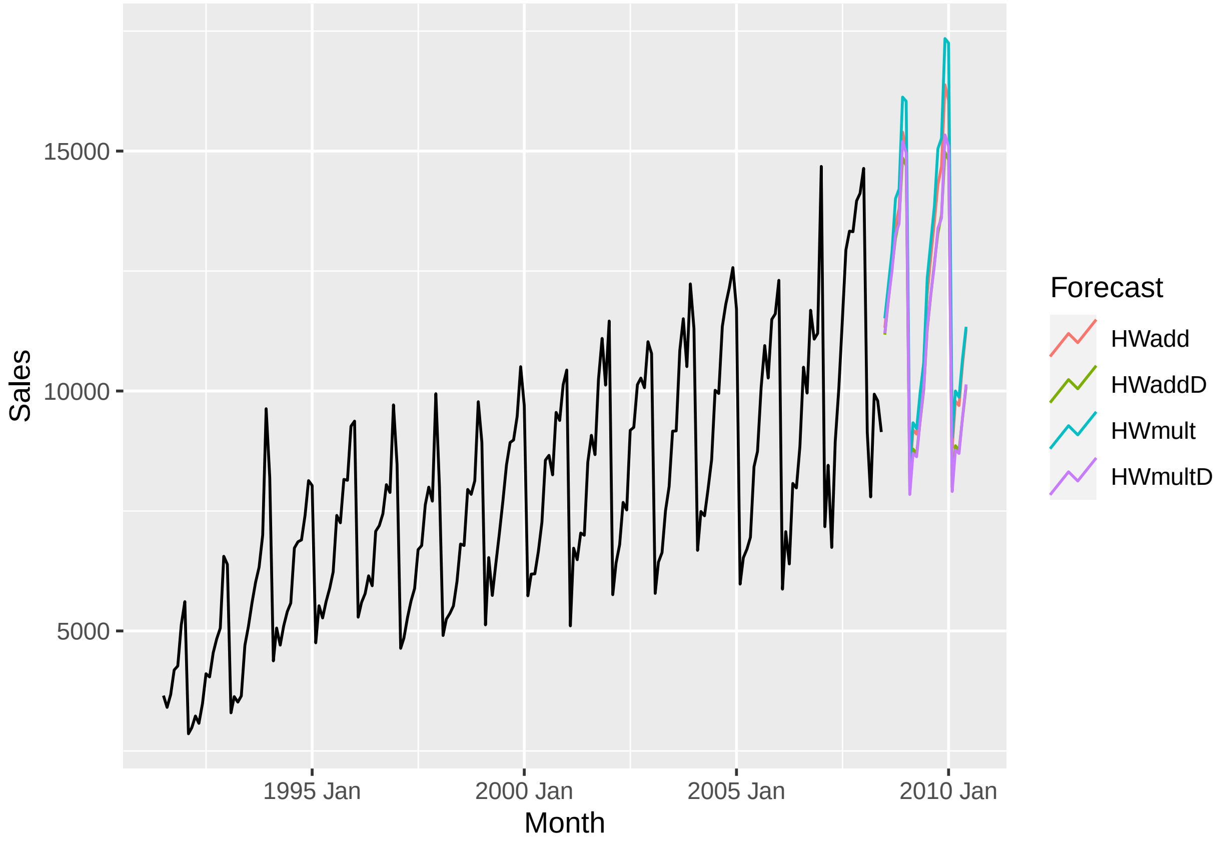
c) Repeat b) with Holt-Winters' seasonal methods. Discuss whether additive or multiplicative seasonality is necessary. Explain your answer (5 marks total).

```
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 |>
```

```
  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:

```
# 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) version of the MSE, in this report, RMSE will be used for convenience's sake.

```
train1 <- steroids |>
  slice(1:(n()-1))

test1 <- steroids |>
  slice((n()-1):n())

fit_all1 <- train1 |>
  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"))
  )

fc_all1 <- fit_all1 |>
  forecast(h=1)

suppressWarnings(
  fc_all1 |>
    fabletools::accuracy(test1) |>
    select(.model, RMSE, MAE) |>
    arrange(RMSE)
)
```

```
## # A tibble: 7 x 3
##   .model   RMSE   MAE
##   <chr>   <dbl> <dbl>
## 1 SES      832.  832.
## 2 Damped   836.  836.
## 3 HL       903.  903.
## 4 HWaddD  1349. 1349.
## 5 HWmultD 1361. 1361.
## 6 HWmult  1543. 1543.
## 7 HWadd   1561. 1561.
```

For a one-step-ahead forecast where only one month is predicted into the future, it would make sense that the SES, Holt's linear and Holt's linear damped methods have the best accuracy scores (respectively) since these models tend to capture trends well (especially when dealing with a straight line where seasonality is insignificant).

```
train4 <- steroids |>
  slice(1:(n()-4))
```

```r
test4 <- steroids |>
  slice((n()-4):n())

fit_all4 <- train4 |>
  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"))
  )

fc_all4 <- fit_all4 |>
  forecast(h=4)

fc_all4 |>
  fabletools::accuracy(test4) |>
  select(.model, RMSE, MAE) |>
  arrange(RMSE)
```

```
## # A tibble: 7 x 3
##   .model   RMSE   MAE
##   <chr>   <dbl> <dbl>
## 1 HWmultD 1143.  995.
## 2 HWaddD  1170. 1015.
## 3 HWadd   1340. 1164.
## 4 HWmult  1345. 1172.
## 5 Damped  1353. 1111.
## 6 SES     1357. 1115.
## 7 HL      1389. 1168.
```

When dealing with a four-step-ahead forecast, seasonality becomes more important, giving Holt-Winters' multiplicative damped method the best accuracy score in this situation.

```r
train6 <- steroids |>
  slice(1:(n()-6))

test6 <- steroids |>
  slice((n()-6):n())

fit_all6 <- train6 |>
  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"))
  )
```

```
fc_all6 <- fit_all6 |>
  forecast(h=6)

fc_all6 |>
  fabletools::accuracy(test6) |>
  select(.model, RMSE, MAE) |>
  arrange(RMSE)
```

```
## # A tibble: 7 x 3
##   .model   RMSE   MAE
##   <chr>   <dbl> <dbl>
## 1 HWaddD   964.  841.
## 2 HWadd    981.  841.
## 3 HWmultD  991.  858.
## 4 HWmult   992.  885.
## 5 SES     5376. 4916.
## 6 Damped  5401. 4937.
## 7 HL      5759. 5247.
```

For a six-step-ahead forecast where seasonality is more significant, Holt-Winters' additive damped has the best accuracy for this series.

```
fc_all_sum <- bind_rows(
  ## compute accuracy of forecasts
  suppressWarnings(
    fabletools::accuracy(fc_all1, test1)
  ),
  fabletools::accuracy(fc_all4, test4),
  fabletools::accuracy(fc_all6, test6)
  ) |>
  ## compute mean of RMSE and MAE for all forecasts
  group_by(.model) |>
  summarise(RMSE = mean(RMSE), MAE = mean(MAE))

fc_all_sum |>
  arrange(RMSE)
```

```
## # A tibble: 7 x 3
##   .model   RMSE   MAE
##   <chr>   <dbl> <dbl>
## 1 HWaddD  1161. 1068.
## 2 HWmultD 1165. 1072.
## 3 HWmult  1293. 1200.
## 4 HWadd   1294. 1189.
## 5 SES     2522. 2287.
## 6 Damped  2530. 2295.
## 7 HL      2684. 2439.
```

After taking the mean of every model's RMSE and MAE scores for one/four/six-step-ahead forecasts, the Holt-Winters' additive damped method has the highest accuracy.

However from the previous analysis of the one/multi-step-ahead forecasts, it is evident that the selection of which model is "best" depends on the number of pre-specified steps-ahead forecasts. In order to optimise forecast accuracy, different models should be used for varying forecasting horizons. For example, a short-term forecast may have the best accuracy with SES (as exhibited before with one-step-ahead forecast), whereas a longer-term forecast may require a more complex model that captures the overall trend and seasonality of the data.

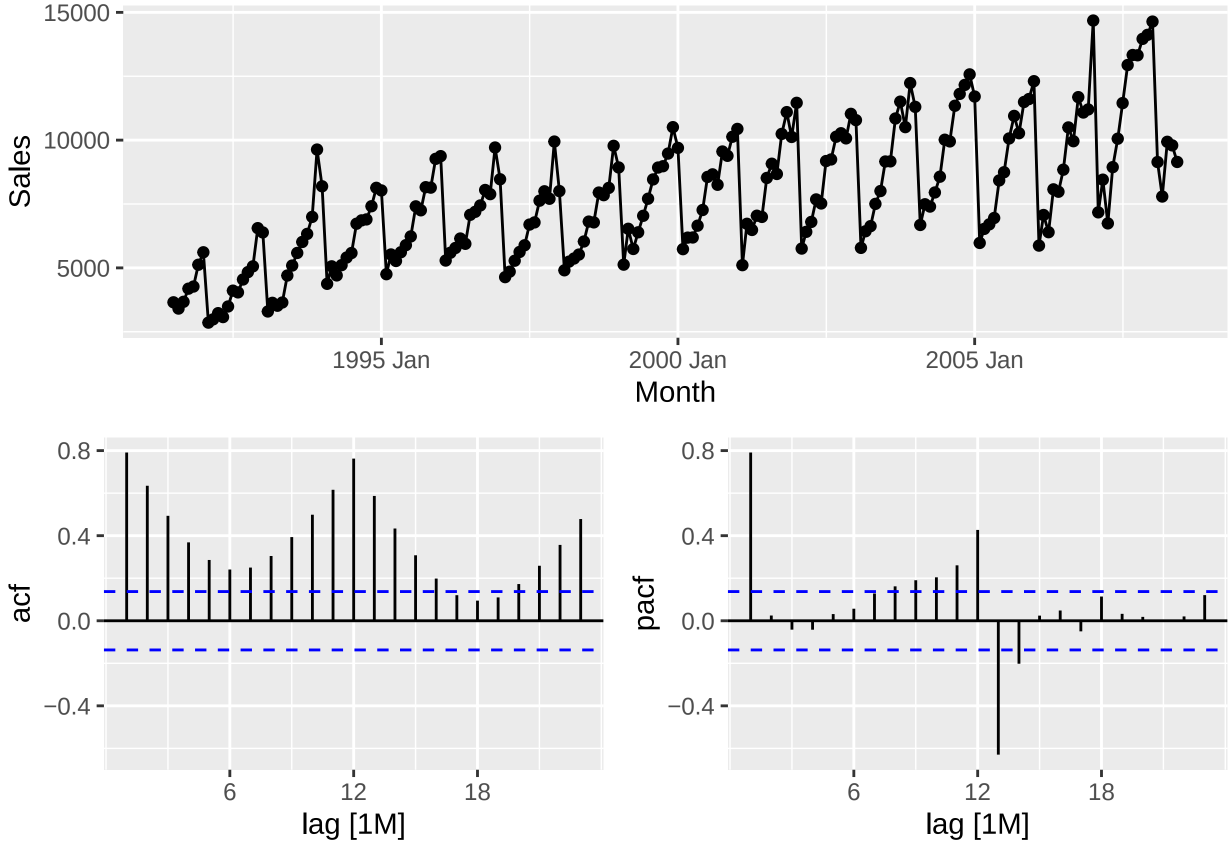
e) Briefly discuss the potential mistake/error we may unintentionally introduce in the discussion when comparing models from b) and c) using the MSE and MAE. (3 marks)

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)

```r
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)
```

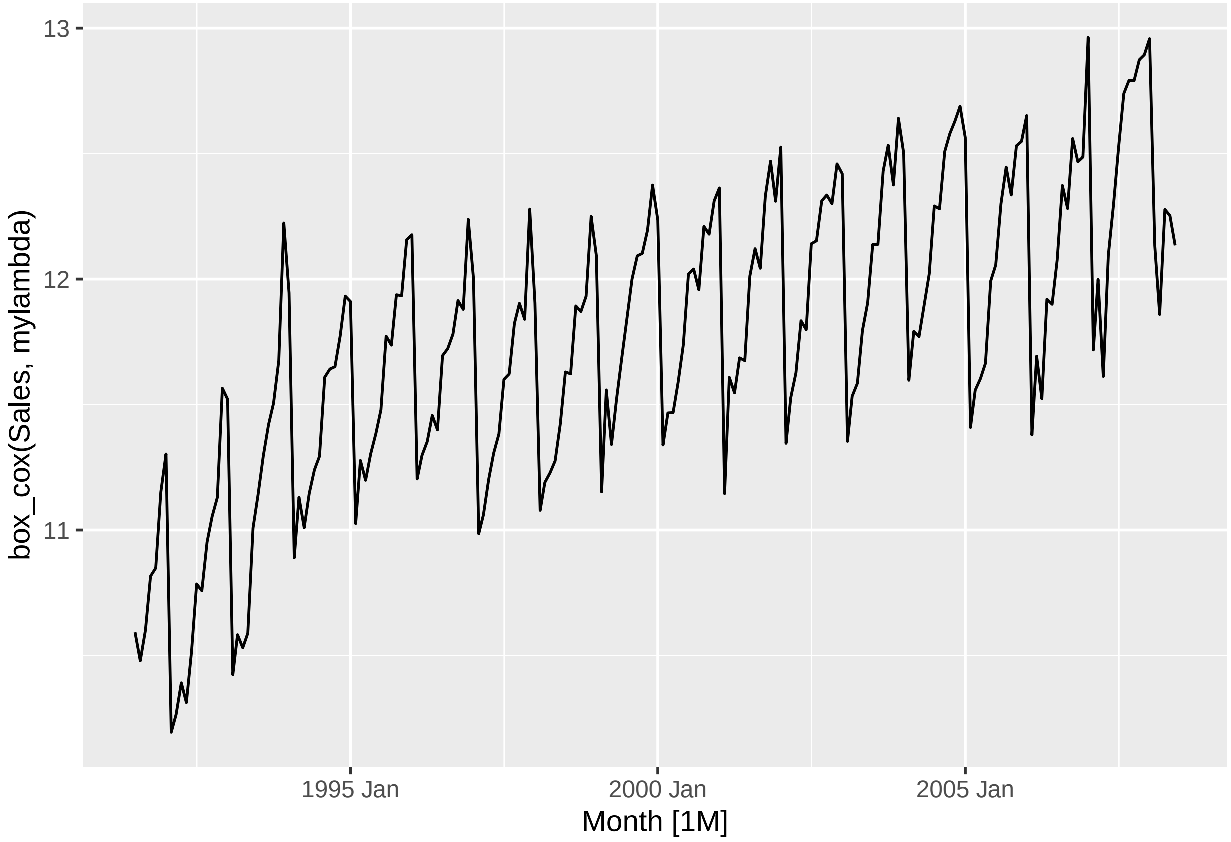
```
## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##       <dbl>       <dbl>
## 1      2.85        0.01
```

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)
```

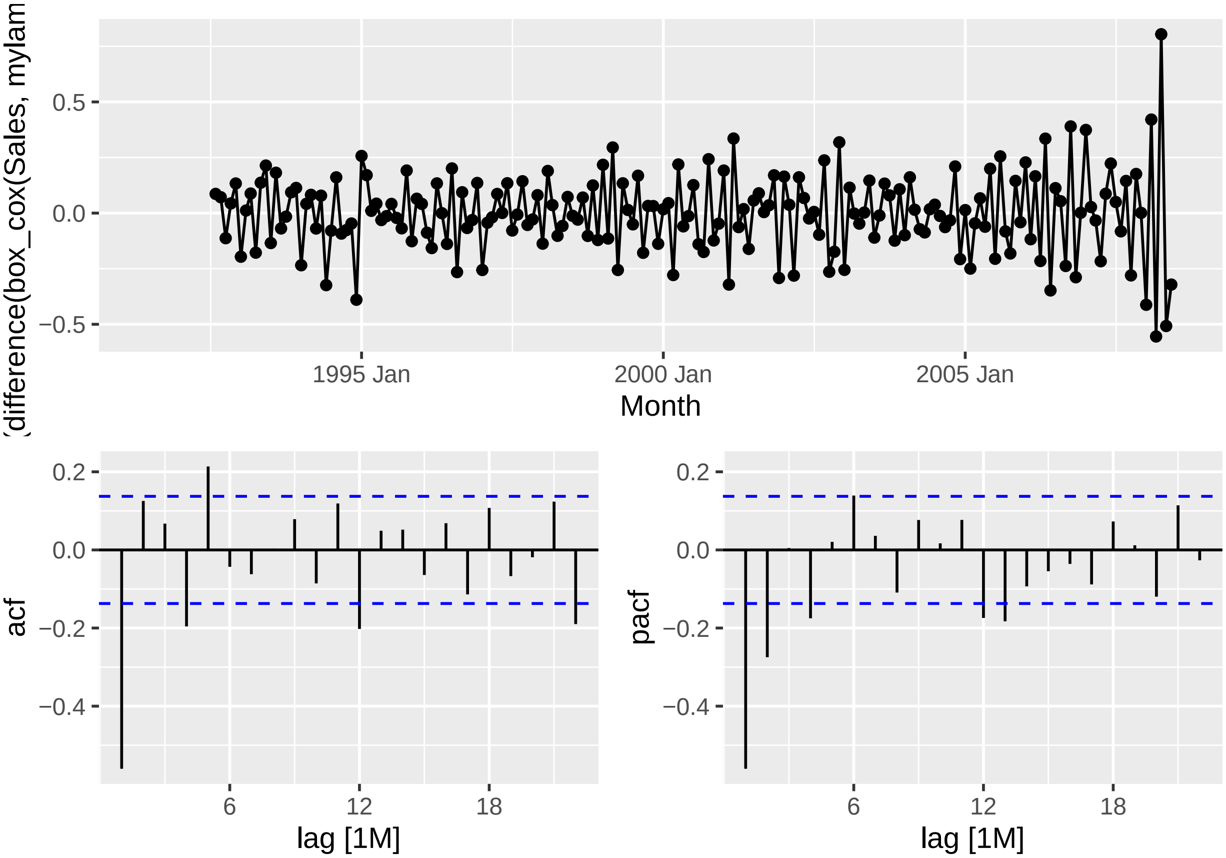
```
## # A tibble: 1 x 1
##   ndiffs
##    <int>
## 1      1
```

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')
```

```
## 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)

```
fit_arima <- steroids |>
  model(ARIMA(box_cox(Sales, mylambda), stepwise = FALSE))

report(fit_arima)
```

```
## Series: Sales
## Model: ARIMA(2,1,3)(0,1,1)[12]
## Transformation: box_cox(Sales, mylambda)
##
## Coefficients:
##           ar1      ar2     ma1     ma2      ma3     sma1
##       -1.0289  -0.8496  0.2638  0.3167  -0.3988  -0.7061
## s.e.   0.1877   0.1410  0.2374  0.1049   0.1052   0.0660
##
## sigma^2 estimated as 0.0139:  log likelihood=135.88
## AIC=-257.77   AICc=-257.15   BIC=-235
```