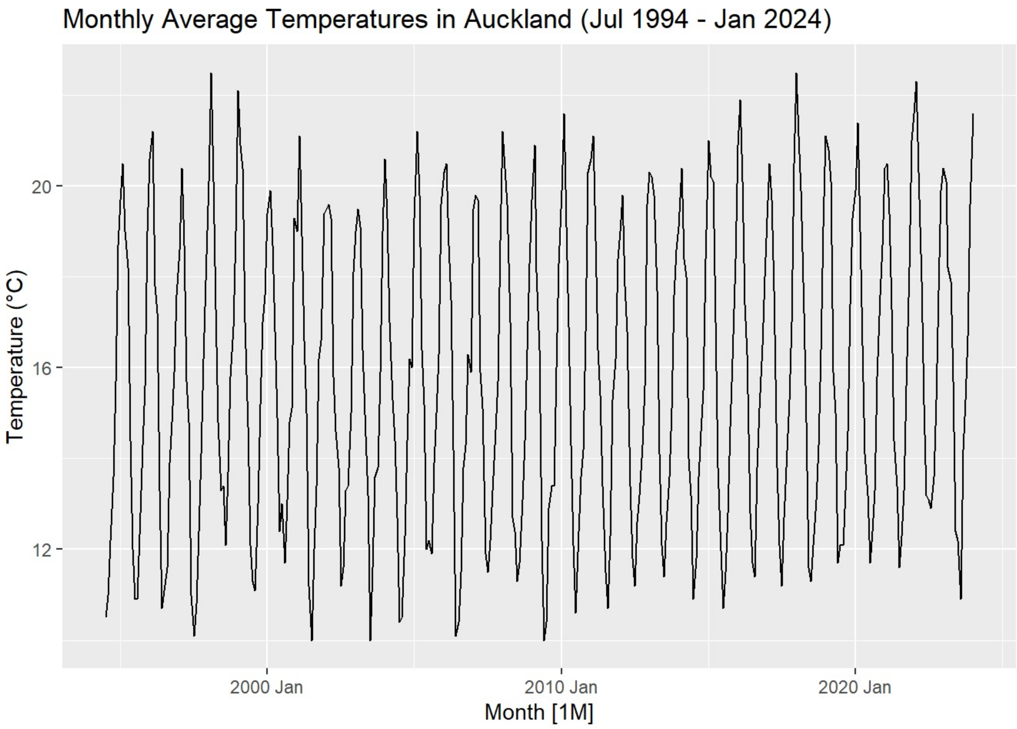
Code ▾

STATS 326/786

Assignment 02

Problem 1: Monthly Average Auckland Temperatures

In Assignment 1, 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. The time series plot is given below.

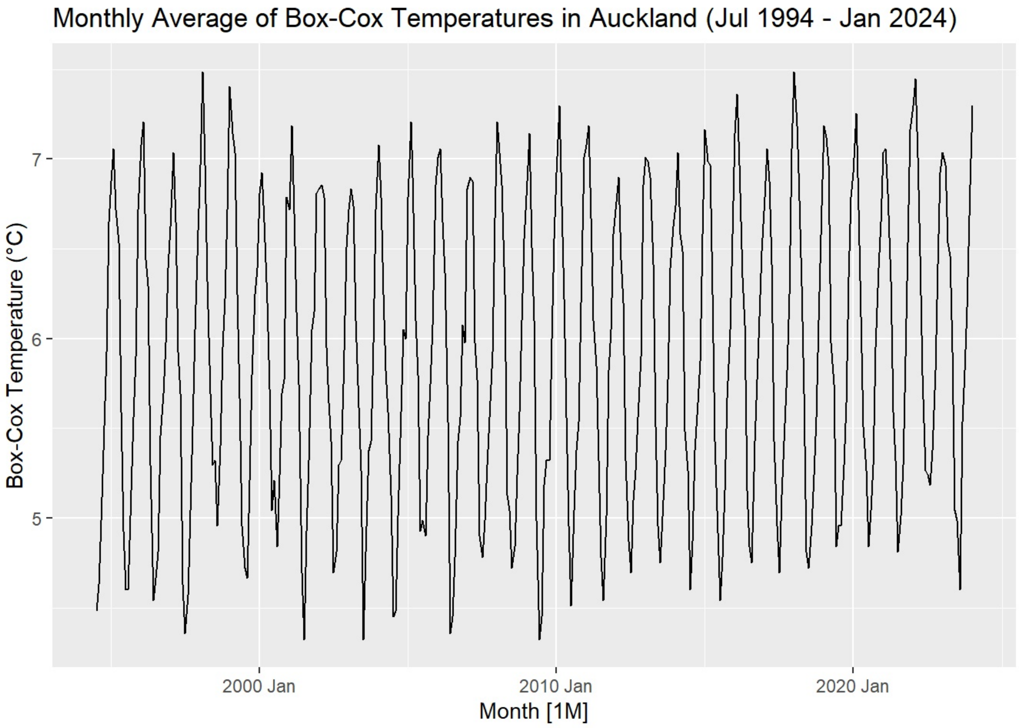


1. 7 Marks

- Without using the `box_cox` function in the `feasts` package, create a new variable by manually performing a Box-Cox transformation on `Temperature` with $\lambda = 0.5$.
- Plot the Box-Cox transformed `Temperature` time series and correlogram (setting `lag_max = 36`).
- Comment on the patterns you observe in the correlogram. Hint: Think about your answer to the lag plot question in Assignment 1.

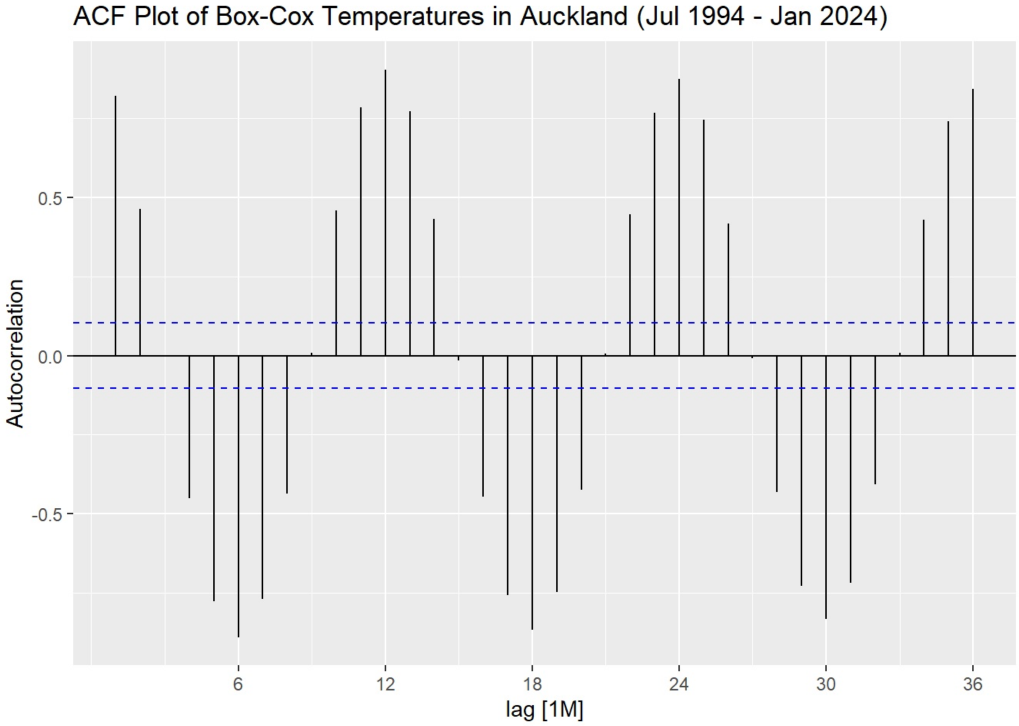
Hide

```
#Manual Box-Cox transformation
lambda <- 0.5

data$Temperature.T <- (data$Temperature^lambda-1)/lambda

#Plot time series and correlogram
data %>%
  autoplot(Temperature.T) +
  labs(y = "Box-Cox Temperature (\u00B0C)",
       title = "Monthly Average of Box-Cox Temperatures in Auckland (Jul 1994 - Jan 2024)")
```



Hide

```
data %>%
  ACF(Temperature.T, lag_max = 36) %>%
  autoplot() +
  labs(y = "Autocorrelation",
       title = "ACF Plot of Box-Cox Temperatures in Auckland (Jul 1994 - Jan 2024)")
```



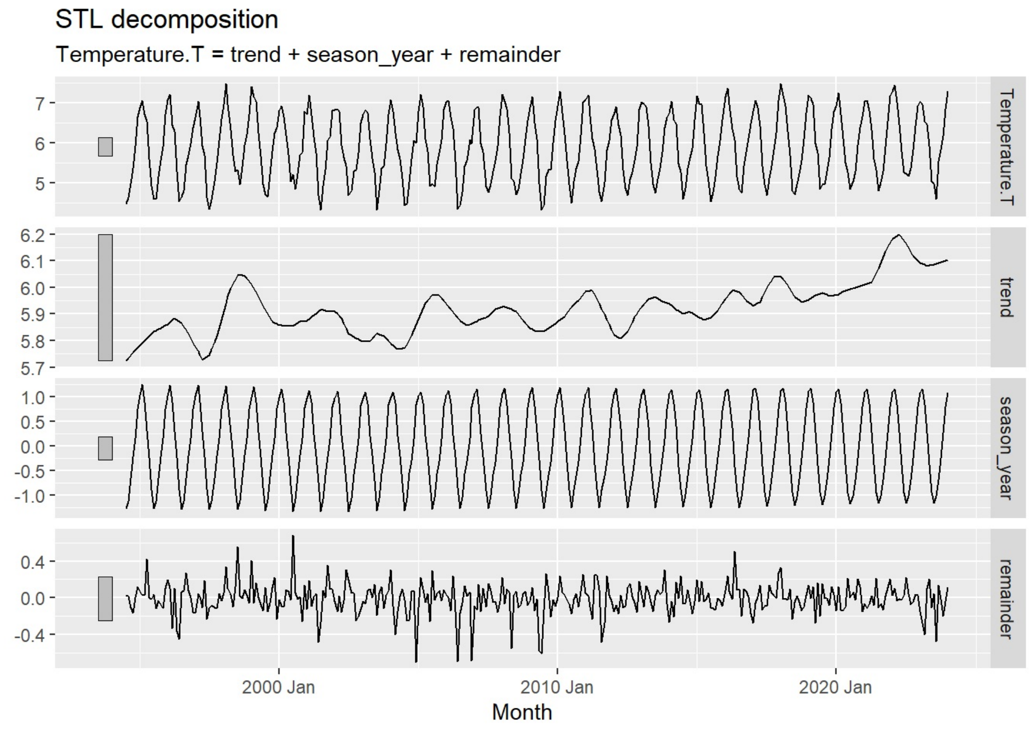
It appears there are patterns of positive and negative autocorrelation coefficients associated with different lag values. Lag 1 has a highly positive ACF value because when the monthly temperatures are shifted by 1, the similarities in their temperature values results in a positive correlation. For example, the temperature of January is similar to February. As we move along the x-axis, lag 6 has a highly negative ACF value. This is to be expected because when the monthly temperatures are lagged by 6, the temperature of January (summer) would be paired with July (winter) which creates negative autocorrelation. Moving further right of the x-axis, it's also evident that there is a highly positive ACF value for lag 12. This is also expected because as we shift the temperatures by 12 months, we have shifted the lagged values by a complete cycle. As a result, the temperature of January would be paired with January from the following year, which is expected to have a highly positive correlation. Therefore, it makes sense that the ACF plot shows repeated cycles of positive and negative autocorrelation dependent on lag.

2. 7 Marks

- Perform an STL decomposition on the Box-Cox transformed temperatures. Keep the defaults for the trend and seasonal components, but specify `robust = TRUE` for the remainder component.
- Plot the time series decomposition and comment on features you observe in the three components of the time series. Do you believe average temperatures in Auckland are rising? (Yes or no).

Hide

```
#Perform STL decomposition
temp.dcmp <- data %>%
  model(STL(Temperature.T, robust = TRUE)) %>%
  components()

#Plot time series decomposition
temp.dcmp %>%
  autoplot()
```



In observing our trend plot, there appears to be an average temperature increase over the years, and considering how much larger the scales are in comparison to the original time series plot, there's only a slight positive trend.

Our seasonal plot clearly displays a periodic cycle of warmer and colder temperature values, which is expected as a result of seasonal changes.

There appears to be a mean of zero in our remainder plot, this suggests that the variations in our time series plot is due to randomness which are not explained by the underlying trend. In other words, the remainder plot captures most of the variation in the original plot.

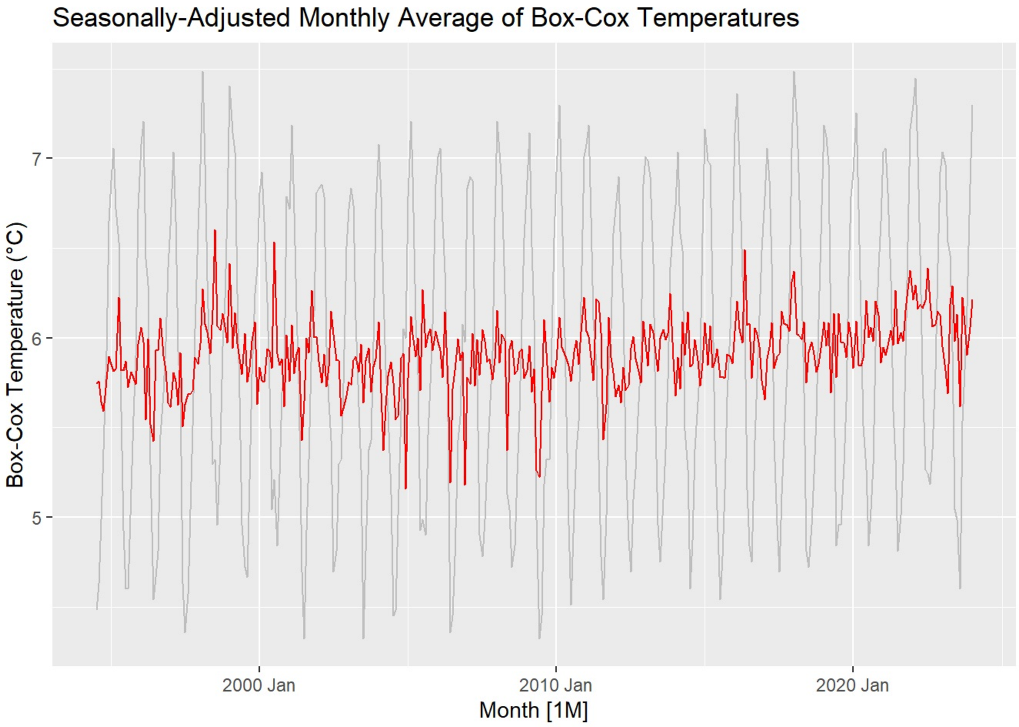
Although there appears to an average increasing trend in the trend plot, it has a much larger scale than the time series plot and thus has only a slight effect on the temperature increase. Therefore, I believe that the average temperatures in Auckland increases slightly.

3. 8 Marks

- Explain why you would seasonally adjust the Box-Cox transformed temperature series you created in (1).
- Comment on how the seasonally-adjusted series for the Box-Cox transformed temperature series is calculated when performing the STL decomposition in (2).
- Plot the seasonally-adjusted series for the Box-Cox transformed temperatures by extracting it from your STL decomposition model, and plot the subseries plot of the seasonally-adjusted series. Comment on whether there is seasonality still present.
- Conclude whether you believe STL decomposition is doing a good job of performing seasonal adjustment on the Box-Cox adjusted temperatures.

Hide

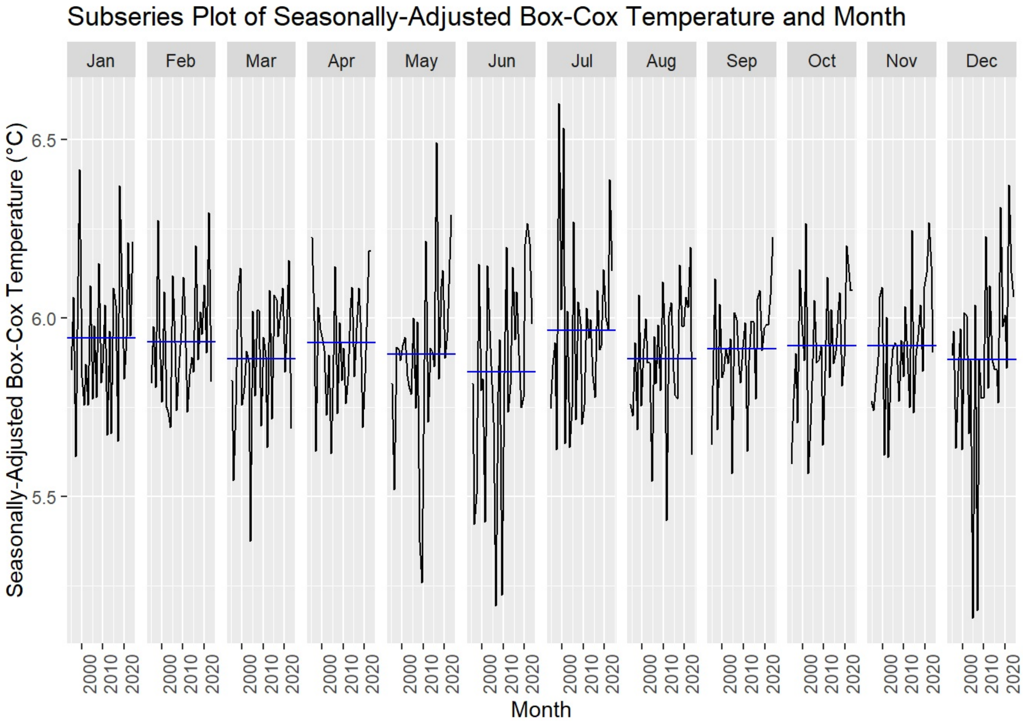
```
#Seasonally-adjusted series
data %>%
  autoplot(Temperature.T, colour = "grey") +
  autolayer(temp.dcmp, season_adjust, colour = "red") +
  labs(y = "Box-Cox Temperature (\u00B0C)",
       title = "Seasonally-Adjusted Monthly Average of Box-Cox Temperatures")
```


Seasonally-Adjusted Monthly Average of Box-Cox Temperatures
Box-Cox Temperature (°C)
7
6
5
2000 Jan
2010 Jan
2020 Jan
Month [1M]

Hide

Hide

```
#plot subseries plot
data %>%
  gg_subseries(temp.dcmp$season_adjust) +
  ggtitle("Subseries Plot of Seasonally-Adjusted Box-Cox Temperature and Month") +
  ylab("Seasonally-Adjusted Box-Cox Temperature (\u00B0C)")
```



Previously, we applied the Box-Cox transformation to reduce the variance in our data which makes our plots more consistent. In our decomposition plot, there appears to be a seasonal pattern which influences the trend in our data. As the seasonal effect obscures the trend, it is necessary to seasonal adjust our Box-Cox transformed temperature series to remove any influences from the seasonal cycles. This allows us to create more accurate analysis of the underlying trend in the temperature series.

It appears that our time series plot has additive effects, thus the seasonally-adjusted series are calculated by subtracting the seasonal component from our Box-Cox transformed temperature series.

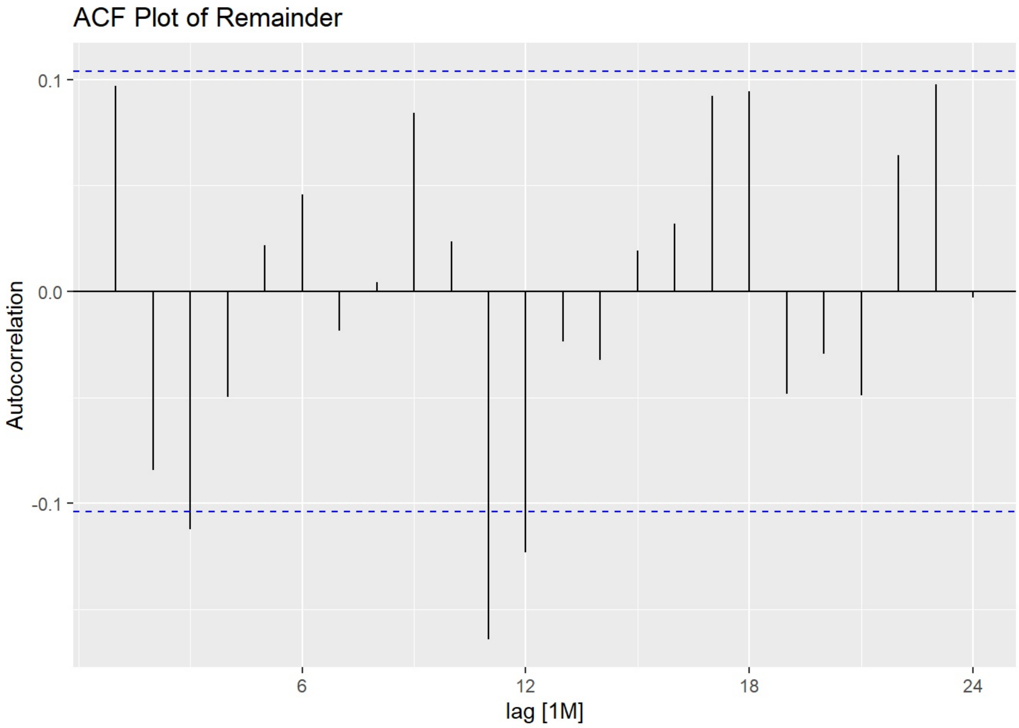
In observing our subseries plot, it seems that the average of the seasonally adjusted values appear to be roughly similar, and seasonality is no longer present. This means the STL decomposition has effectively removed the seasonal component, which allows for clearer observations of the underlying trend. Therefore, we conclude that the STL decomposition has done an adequate job on performing seasonal adjustment on the data.

4. 5 Marks

- Plot the correlogram of the `remainder` term from the STL decomposition.
- Comment on any interesting features you observe in the correlogram and explain whether the remainder term is consistent with white noise.
- Verify your conclusion by performing a Ljung-Box test (using `lag = 24` and `dof = 0`).

Hide

```
#create correlogram
temp.dcmp %>%
  ACF(remainder, lag_max = 24) %>%
  autoplot() +
  labs(y = "Autocorrelation",
       title = "ACF Plot of Remainder")
```



Hide

```
#perform Ljung-Box test
temp.dcmp %>%
  features(remainder, features = ljung_box, lag = 24, dof = 0)
```

```
## # A tibble: 1 × 3
##   .model                           lb_stat lb_pvalue
##   <chr>                              <dbl>     <dbl>
## 1 STL(Temperature.T, robust = TRUE)    45.8   0.00470
```

In observation of our correlogram, it appears that a few ACF values exceed the -0.1 threshold for specific lags of 3,11 and 12 out of 24 lags. This means there is a 12.5% chance (3/24 = 0.125) that the lagged values exceed the 95% confidence interval, which is larger than the 5% significance level. This suggests that the remainder term is not consistent with white noise as there is autocorrelation, non-zero mean and non-constant variance in our remainder.

As we obtained a p-value of 0.0047 in our Ljung-Box test, we have extremely strong evidence against the null hypothesis that the remainder term is consistent with white noise, in favor of the alternative hypothesis that there is autocorrelation, non-zero mean and non-constant variance present in our remainder.

> Total possible marks for **Problem 1**: 27 Marks

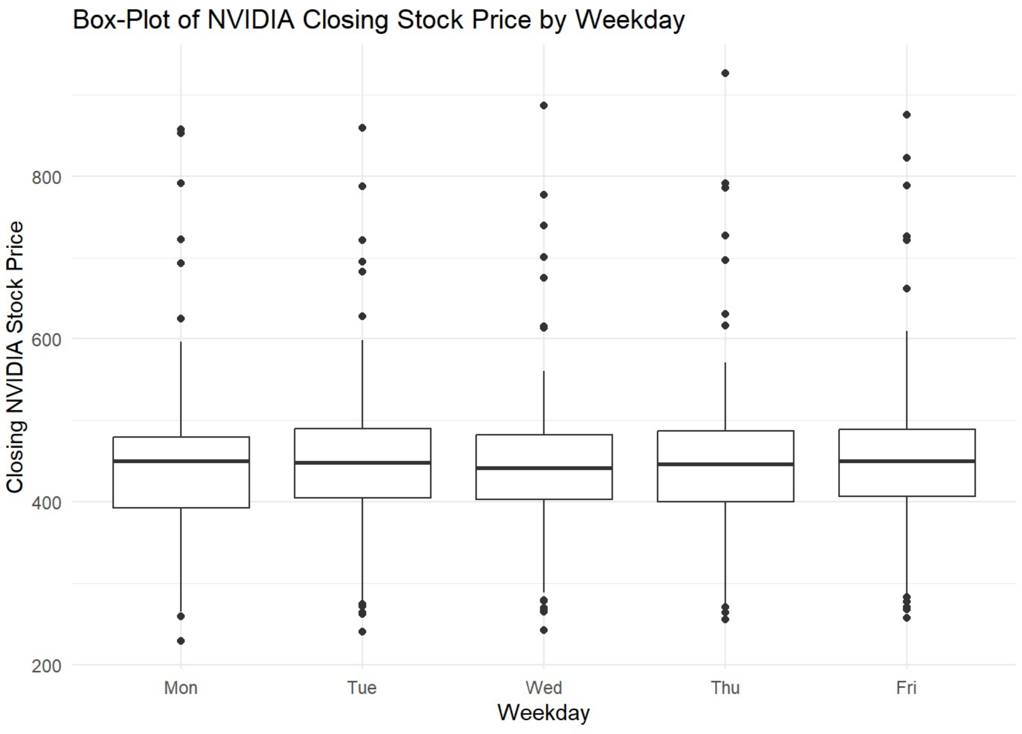
Problem 2: NVIDIA Closing Stock Prices

NVIDIA are world leaders in developing GPUs used for gaming and artificial intelligence. The data set `NVIDIA.csv` contains daily closing stock prices (in USD) from 13 March 2023 until 11 March 2024.

> 1. 6 Marks
>
> - In this question, you will explore whether there is weekly seasonality, i.e., a day-of-the-week effect. However, stock markets close in the weekends so the time series is irregular, meaning seasonal plots are not straightforward to make. We will take a different approach.
> - Read in the `NVIDIA.csv` data set, ensure `Date` is in an appropriate date format, and coerce it to a `tsibble` setting `index = Date`.
> - Within this data set, create a day-of-the-week variable (with labels). The variable should have days Mon–Fri.
> - Using `ggplot` and `geom_boxplot`, create a box-plot showing the distribution of the closing stock price for each day of the week (Monday - Friday).
> - Comment on whether there is a day-of-the-week effect.

Hide

```
#Read CSV file, convert to date object, create labels and coerce into time series data frame
NVIDIA <- read_csv("NVIDIA.csv", show_col_types = FALSE)
NVIDIA$Date <- dmy(NVIDIA$Date)
NVIDIA <- NVIDIA %>% mutate(Weekday = wday(NVIDIA$Date, label = TRUE))
NVIDIA.ts <- as_tsibble(NVIDIA, index = Date)

#Create box-plot
ggplot(NVIDIA.ts, aes(x = Weekday, y = Close)) +
  geom_boxplot() +
  labs(title = "Box-Plot of NVIDIA Closing Stock Price by Weekday",
       x = "Weekday",
       y = "Closing NVIDIA Stock Price") +
  theme_minimal()
```



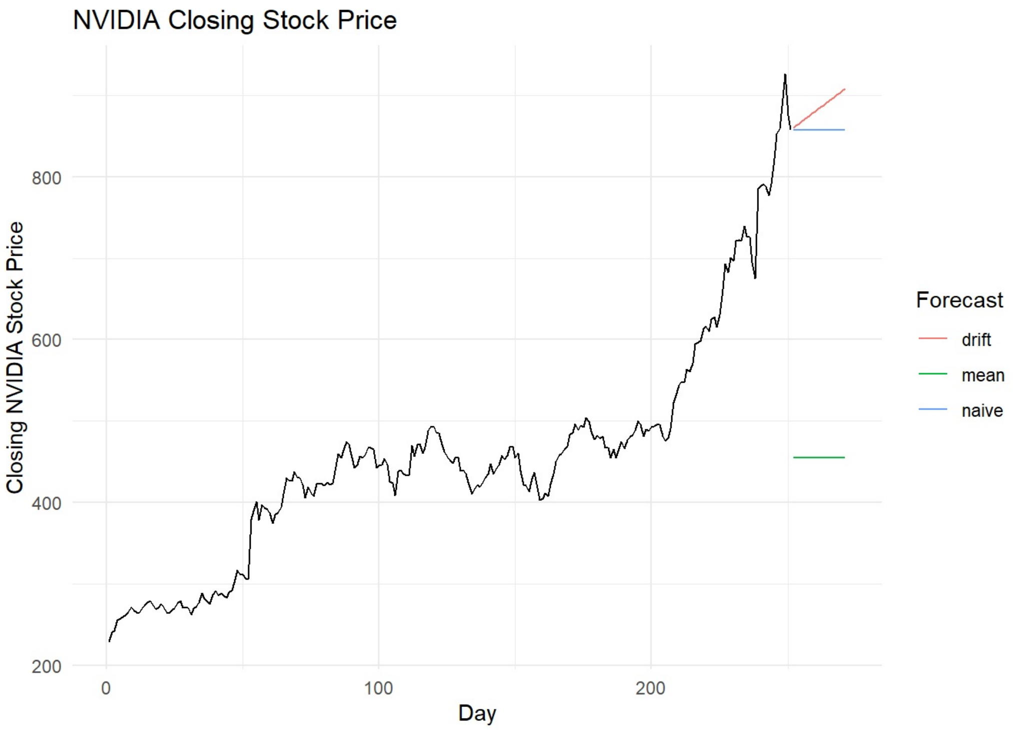
In observing our box-plots, it appears the median closing NVIDIA stock prices are similar which suggests there aren't any particular weekdays where the stock prices are significantly higher or lower on average. Furthermore, the spread distribution and interquartile range are roughly similar, which suggests the volatility of the stock are also similar. Therefore, based on our box-plots analysis, there doesn't seem to be a day-of-the-week effect.

2. 12 Marks

- There are gaps in the NVIDIA data set due to there being no observations for the weekend. Create a trading day variable using the `row_number` function, and update your `tsibble` such that the index is now the trading day.
- Fit the following three benchmark forecast models to the closing stock prices for NVIDIA: Average method, naive method, and random-walk with drift method.
- Forecast 20 trading days ahead and plot your forecasts on the same plot as the closing stock price time series. For your forecasts, only plot the point-forecasts, and not the prediction intervals.
- Of the three models, which one do you believe produces the worst forecasts? Explain your answer.
- Of the three models, which one do you believe produces the best forecasts? Explain your answer.

Hide

```
#Create trading day variable, and set to index
NVIDIA.ts <- NVIDIA.ts %>%
  mutate(Trading_Day = row_number()) %>%
  as_tsibble(index = Trading_Day)

#Create benchmark forecast models and forecast 20 trading days
NVIDIA.ts %>%
  model(mean = MEAN(Close),
        naive = NAIVE(Close),
        drift = RW(Close ~ drift())) %>%
  forecast(h = 20) %>%
  autoplot(NVIDIA.ts, level = NULL) +
  xlab("Day")+ ylab("Closing NVIDIA Stock Price") +
  ggtitle("NVIDIA Closing Stock Price") +
  guides(colour=guide_legend(title = "Forecast")) +
  theme_minimal()
```



The forecast generated by the average method seems to be the most unreliable as the predicted values appear significantly below our expectations given the NVIDIA closing stock prices are trending upwards. The average method calculates the mean of the historical stock prices, and doesn't take into an account of any trends.

As the NVIDIA stock prices appear to be trending upwards, we expect the future stock prices to remain somewhere high on the chart. The naive method assumes the stock price for the next 20 days to be equal to the last observed value, while the random-walk with drift method takes into account the average change from one-period to the next. Although the random-walk with drift method seems to be more realistic as it predicts future values based on the latest trend, however, we cannot say for certain which one of those two is more reliable. This is because the stock price may fluctuate in the next 20 tradings and it's difficult to generate a reliable short-term forecast.

Total possible marks for **Problem 2**: 18 Marks

Total possible marks for **Assignment 2**: 45 Marks for 326 55 Marks for 786

Loading [MathJax]/jax/output/HTML-CSS/jax.js