Josh Copeland - ECOM90024 - Assignment 1

```
# Libraries used for this assignment:

# library(tidyverse)
# library(readxl)
# library(lubridate)
# library(janitor)
```

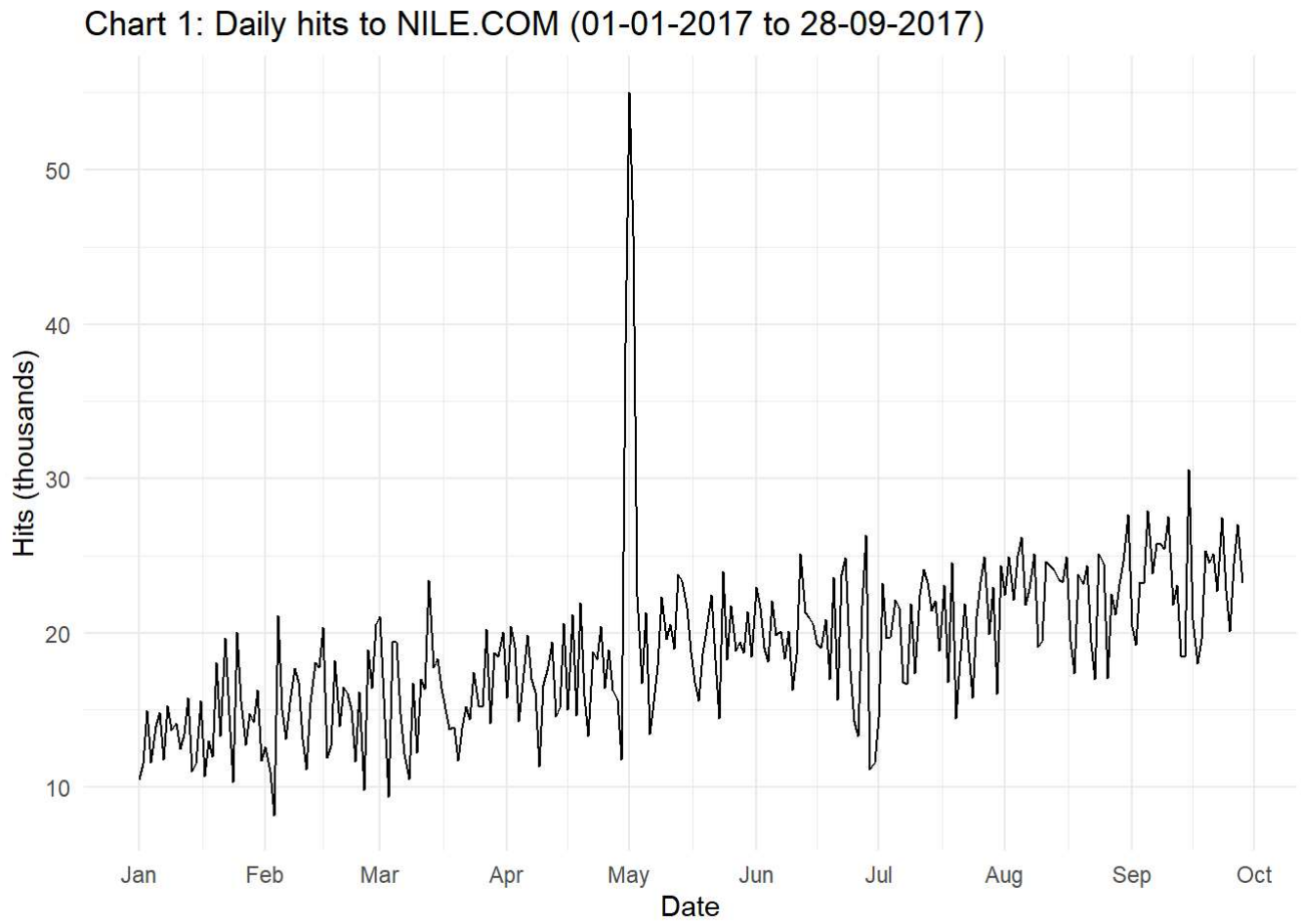
Question 1

You are an analyst working for NILE.COM, a successful online bookseller. You have been tasked with monitoring and forecasting the number of hits (i.e., visits) per day to its website. The file nile.csv on the LMS contains daily hits data for the period 1/1/2017 through to 28/9/2017. You are required to compute all your estimations and plots in R.

a.) Generate an appropriate plot of data and provide a brief description of the observed time series. Make sure to point out any notable visual features/characteristics of the data. (1 Mark)

The graph below (Chart 1) shows daily hits (visits) to the website NILE.COM from 1 January 2017 to 28 September 2017 in thousands. Over this time period, hits are trending upwards. There are no clear cycles in this data, but its possible there may be some high frequency seasonal patterns (i.e. weeks or months). However, more analysis is required to confirm this seasonality. There was also a very large spike in website traffic in late April and early May, potentially due to a sale on the website.

```
nile <- read_csv("nile.csv", show_col_types = FALSE) %>%
  mutate(date = seq(as.Date("2017-01-01"),as.Date("2017-09-28"), by = "day")) %>%
  select(date, hits = Hits) %>%
  mutate(hits = hits / 1000)

ggplot(nile, aes(date, hits)) + geom_line() + scale_x_date(date_breaks = "1 month", date_label = "%b") + labs(x = "Date", y = "Hits (thousands)") + ggtitle("Chart 1: Daily hits to NILE.COM (01-01-2017 to 28-09-2017)")
```



b.) Fit and assess the linear, quadratic and exponential trend models to the data. In your analysis, make sure to include the estimation results as well as appropriate plots of the fitted trends that you have generated. Given your results, which trend model would you choose as your forecasting model? Make sure to provide your reasoning. (1 Mark)

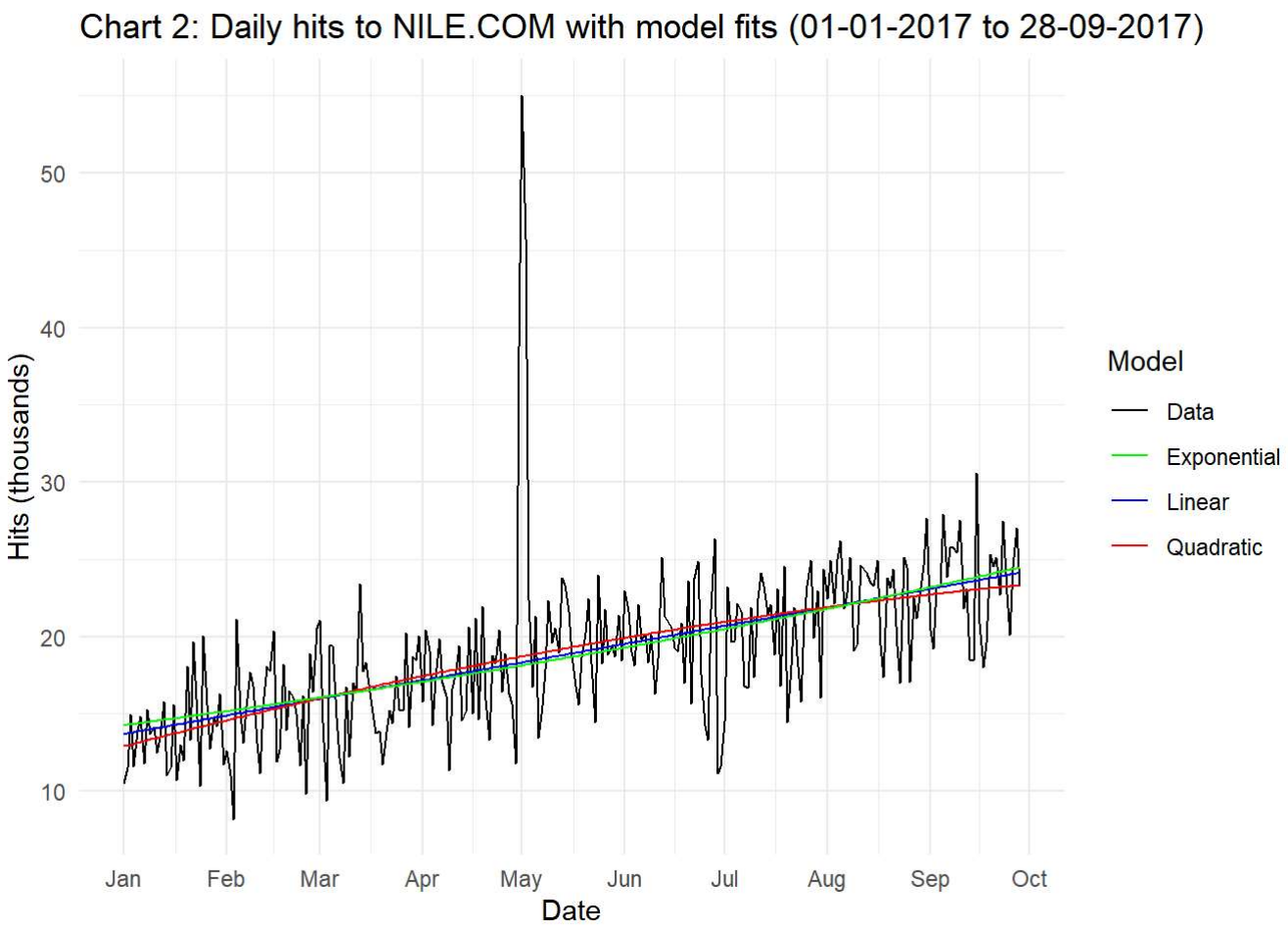
The analysis below estimates and fits linear, quadratic and exponential trend models to the hits data for NILE.COM. Chart 2 shows how well the fitted trends accord with the underlying data. Without further analysis, it seems that they all provide roughly the same fit.

To determine which model provides the best fit, it is important to use the relevant model selection criteria generated by each model: Akaike information criterion (AIC) and Bayesian information criterion (BIC). The function used to generate the exponential model uses a nonlinear least squares methodology, which makes the use/interpretation of any r-squared values complex. the nls() function does not generate this automatically and I have not attempted to create my own measure to make comparisons as simple as possible.

A comparison of the relevant model selection criteria, as summarised in the model_selection_criteria table below, shows us the linear model is the most appropriate for forecasting. This is because it has the lowest AIC and BIC values relative to the other two.

```
#First we need to create some numeric time variables for the linear, quadratic  and exponential
models.

nile <- nile %>%
  mutate(
    time = seq(1,length(nile$hits)),
    time_square = time^2,
    hits_log= log(hits))



#ESTIMATING MODELS#

#Linear model

hits_model_linear <- lm(hits ~ time, data = nile)

#Quadratic model

hits_model_quadratic <- lm(hits ~ time + time_square, data = nile)

# Exponential model (need to define exponential trend first)

hits_model_exponential_descaled <- nls(hits ~ a*exp(b*time), start=list(a=8000, b = 0.04), data
= nile)


#EXTRACTING MODEL FITS#

nile <- nile %>%
  mutate(

    linear_model = predict(hits_model_linear),

    quadratic_model = predict(hits_model_quadratic),

    exponential_model = predict(hits_model_exponential_descaled)

  )

# PLOTTING MODEL FITS #

ggplot(nile, aes(x = date)) +
  geom_line(aes(y = hits, colour = "Data")) +
  geom_line(aes(y = linear_model, colour = "Linear")) +
  geom_line(aes(y = quadratic_model, colour = "Quadratic")) +
  geom_line(aes(y = exponential_model, colour = "Exponential")) +
  scale_color_manual(values = c("Data"= "black" ,"Linear" = "blue", "Quadratic" = "red", "Expone
ntial" = "green")) +
  labs(colour = "Model") +
  scale_x_date(date_breaks = "1 month", date_label = "%b") +
```

```
  labs(x = "Date", y = "Hits (thousands)") +
  ggtitle("Chart 2: Daily hits to NILE.COM with model fits (01-01-2017 to 28-09-2017)")
```




```
# DISPLAYING MODEL OUTPUTS #

summary(hits_model_linear)
```

```
##
## Call:
## lm(formula = hits ~ time, data = nile)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.480 -2.465  0.105  1.853 36.714
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.66346    0.52720   25.92   <2e-16 ***
## time         0.03862    0.00336   11.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.327 on 269 degrees of freedom
## Multiple R-squared:  0.3293, Adjusted R-squared:  0.3268
## F-statistic: 132.1 on 1 and 269 DF,  p-value: < 2.2e-16
```

```
summary(hits_model_quadratic)
```

```
##
## Call:
## lm(formula = hits ~ time + time_square, data = nile)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.756 -2.359  0.061  1.963 36.325
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.285e+01  7.932e-01  16.198  < 2e-16 ***
## time         5.655e-02  1.347e-02   4.199 3.65e-05 ***
## time_square -6.592e-05  4.794e-05  -1.375     0.17
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.32 on 268 degrees of freedom
## Multiple R-squared:  0.334,  Adjusted R-squared:  0.329
## F-statistic: 67.21 on 2 and 268 DF,  p-value: < 2.2e-16
```

```
summary(hits_model_exponential_descaled)
```

```
##
## Formula: hits ~ a * exp(b * time)
##
## Parameters:
##    Estimate Std. Error t value Pr(>|t|)
## a 1.424e+01  4.580e-01   31.10   <2e-16 ***
## b 1.999e-03  1.815e-04   11.01   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.35 on 269 degrees of freedom
##
## Number of iterations to convergence: 16
## Achieved convergence tolerance: 2.59e-07
```

```
# COMPARING INFORMATION CRITERIA #

model_list <- list(hits_model_linear, hits_model_quadratic, hits_model_exponential_descaled)

model_names <- c("hits_model_linear", "hits_model_quadratic", "hits_model_exponential_descaled")

extract_criteria <- function(model, model_name) {
  summary_data <- summary(model)
  aic <- AIC(model)
  bic <- BIC(model)
  return(data.frame(Model = model_name, AIC = aic, BIC = bic))
}

model_selection_criteria <- map2_df(model_list, model_names, extract_criteria)

print(model_selection_criteria)
```

```
##                             Model      AIC      BIC
## 1               hits_model_linear 1567.074 1577.880
## 2            hits_model_quadratic 1567.169 1581.578
## 3 hits_model_exponential_descaled 1569.858 1580.664
```

c.) For a few days in late April and early May, website hits were much higher than usual during a big sale. Do you find evidence of a corresponding group of outliers in the residuals from your trend models? Do they influence your trend estimates much? How should you treat them? (1 Mark)

Chart 3 shows the residuals from each of the three trend models estimated in part (b). It is obvious that in each model the residuals increase significantly for a few days in late April and early May - it is very likely this sale has impacted our results in part (b).

To determine the magnitude of their influence on model estimates, we should dummy out the relevant period of high website traffic in a new set of models and then compare the outputs between them to gauge its impact on the original models.

After investigating the data more closely, it seems the sale was likely active between 30-04-17 and 02-05-17 as this is the only period of time where residuals are unusually high. Therefore, we will re-estimate these models with these periods dummied out and then compare to the original models.

The result of this exercise are evident by comparing the model_selection_criteria and model_selection_criteria_dummy tables print below. By dummying out this period, we significantly improve the fits of our models:

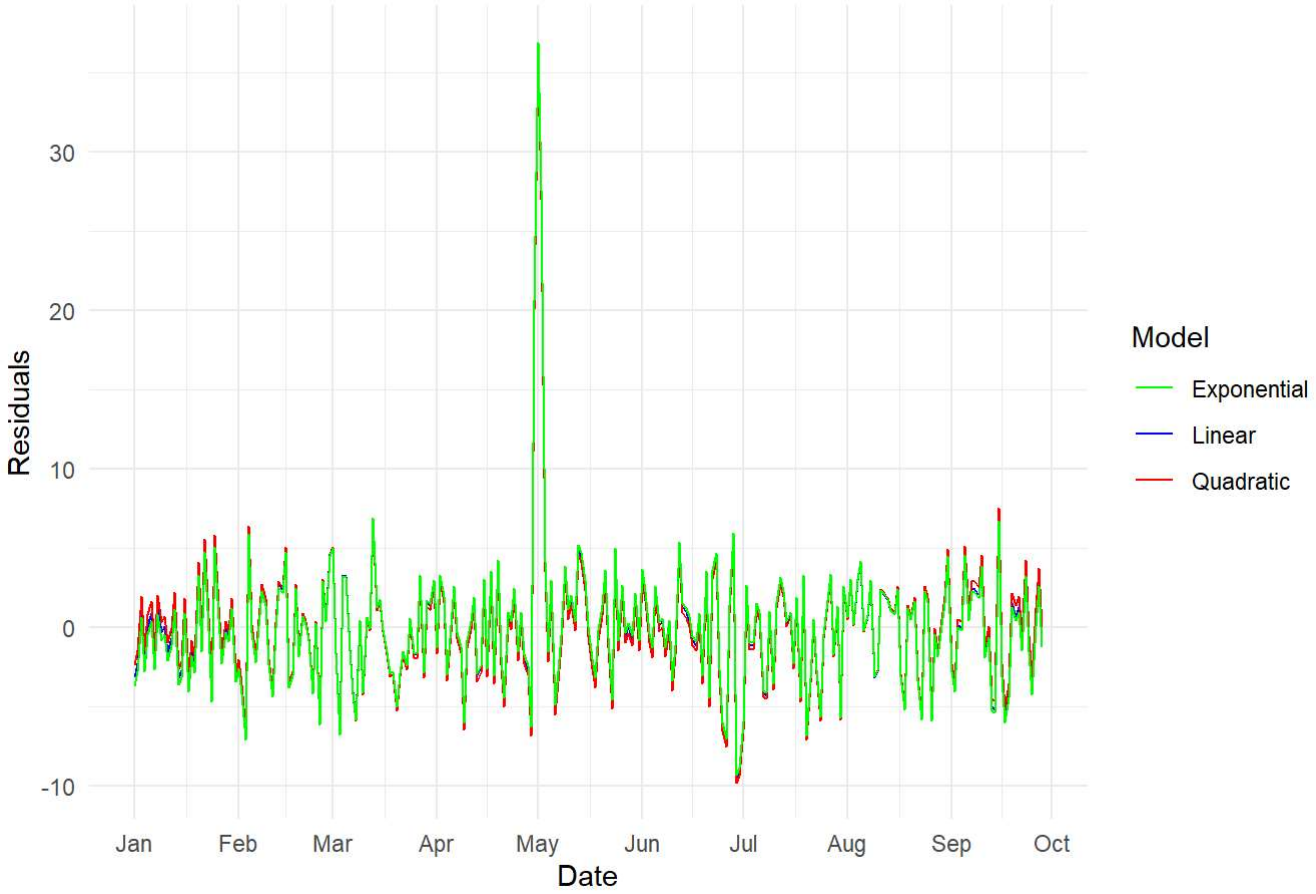
The AIC and BIC of the:

- Linear model reduces from 1567.074 and 1577.880 to 1392.585 and 1406.993 respectively.
- Quadratic model reduces from 1567.169 and 1581.578 to 1394.572 and 1412.583 respectively.
- Exponential model reduces from 1569.858 and 1580.664 to 1560.798 and 1571.604 respectively.

It is worth pointing out that controlling for this large sale in our modelling had the largest impact on the linear model, followed closely by the quadratic. The impact on the exponential model is notably smaller. Therefore, following this exercise, the linear model with a dummy variable for the large sale becomes our preferred model for forecasting.

```
model_residuals <- data.frame(date = nile$date) %>%
  mutate(

    linear_model_resids = resid(hits_model_linear),

    quadratic_model_resids = resid(hits_model_quadratic),

    exponential_model_resids = resid(hits_model_exponential_descaled)

    )


ggplot(model_residuals, aes(x = date)) +
  geom_line(aes(y = linear_model_resids, colour = "Linear")) +
  geom_line(aes(y = quadratic_model_resids, colour = "Quadratic")) +
  geom_line(aes(y = exponential_model_resids, colour = "Exponential")) +
    scale_color_manual(values = c("Linear" = "blue", "Quadratic" = "red", "Exponential" = "gree
n")) +
  labs(colour = "Model") +
    scale_x_date(date_breaks = "1 month", date_label = "%b") +
  labs(x = "Date", y = "Residuals") +
  ggtitle("Chart 3: Residuals of NILE.COM daily traffic models")
```

Chart 3: Residuals of NILE.COM daily traffic models



```
# DEFINING DUMMY VARIABLES #

sale_dummy <- nile %>%
  select(date) %>%
  mutate(sale_dummy = case_when(

    date %in% c(as.Date("2017-04-30"), as.Date("2017-05-01"), as.Date("2017-05-02")) ~ 1,
    TRUE ~ 0

  ))


nile <- nile %>%
  inner_join(sale_dummy, by ="date")

# RE-ESTIMATING MODELS #

#Linear model

hits_model_linear_dummy <- lm(hits ~ time + sale_dummy, data = nile)

#Quadratic model

hits_model_quadratic_dummy <- lm(hits ~ time + time_square + sale_dummy, data = nile)

# Exponential model

hits_model_exponential_descaled_dummy <- nls(hits ~ a*exp(b*time) + sale_dummy, start=list(a=800
0, b = 0.04), data = nile)


# COMPARE MODEL FITS #

model_list_dummy <- list(hits_model_linear_dummy, hits_model_quadratic_dummy, hits_model_exponen
tial_descaled_dummy)

model_names_dummy <- c("hits_model_linear_dummy", "hits_model_quadratic_dummy", "hits_model_expo
nential_descaled_dummy")

model_selection_criteria_dummy <- map2_df(model_list_dummy, model_names_dummy, extract_criteria)

print(model_selection_criteria)
```

```
##                             Model      AIC      BIC
## 1              hits_model_linear 1567.074 1577.880
## 2           hits_model_quadratic 1567.169 1581.578
## 3 hits_model_exponential_descaled 1569.858 1580.664
```

```
print(model_selection_criteria_dummy)
```

```
##                                   Model      AIC      BIC
## 1               hits_model_linear_dummy 1392.585 1406.993
## 2            hits_model_quadratic_dummy 1394.572 1412.583
## 3 hits_model_exponential_descaled_dummy 1560.798 1571.604
```

```
# These dummy variables improve the model fit, therefore incorporate their predictions into the
main data frame.

nile <- nile %>%
  mutate(

    linear_model_dummy = predict(hits_model_linear_dummy),

    quadratic_model_dummy = predict(hits_model_quadratic_dummy),

    exponential_model_dummy = predict(hits_model_exponential_descaled_dummy)

  )
```

d.) Using your preferred model, determine whether there are statistically significant day-of-week effects in the hits to the NILE.COM website. Make sure to provide your reasoning and outline any statistical tests you have utilised in making this determination. (1 Mark)

As noted in part (c), the linear model with a dummy variable controlling for the early April/late May sale is our preferred model for forecasting.

The output of the linear_model_dummy model demonstrates there is evidence is day-of-week effects in this model. To come to this conclusion, I have created a linear regression model where the independent variables are dummy variables denoting each day of the week in the data. In this model, each day of the week has a different impact on website visits on average, as can be see through very small p-values for each of my estimated variables in the output.

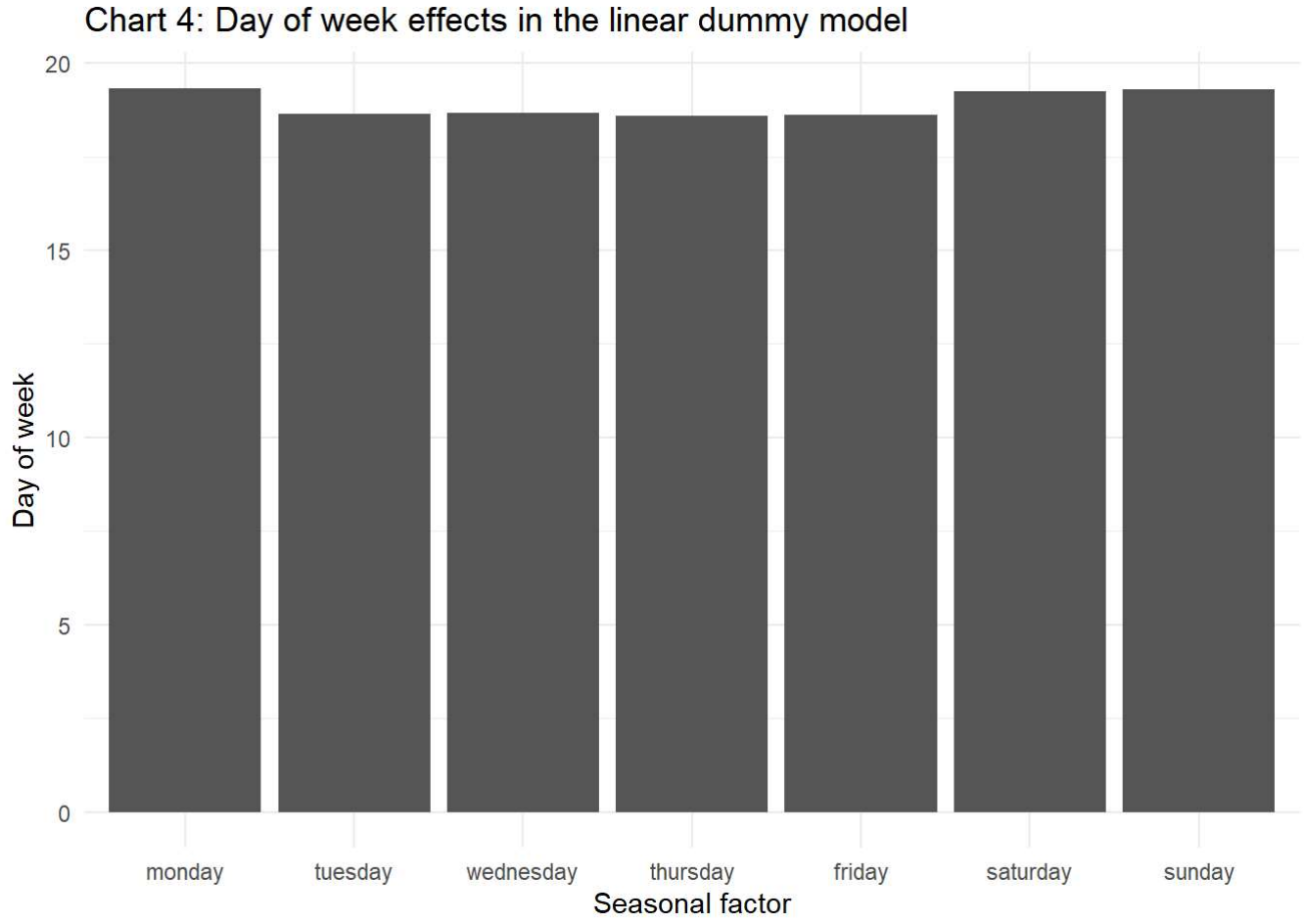
It shows that each day, on average, visits range between 18.64 thousand to 19.33 thousand on average, where the busiest and quietest days are Tuesday and Monday respectively.

Although all the variables are significant, the differences are most material between two groups of days: Saturday to Monday and Tuesday to Friday. Chart 4 clearly shows the difference between these two tranches of days, as hits are materially different between Saturday to Monday and Tuesday to Friday. Beyond general day-of-the-week effects which are significant, it looks like there is some kind of long-weekend impact in the data.

```
# CREATING WEEKDAY DUMMY VARIABLES #

nile <- nile %>%
  mutate(weekday = wday(date, week_start = 2)) %>%
  mutate(
    monday_dummy = case_when(weekday == 1 ~ 1, TRUE ~ 0),
    tuesday_dummy = case_when(weekday == 2 ~ 1, TRUE ~ 0),
    wednesday_dummy = case_when(weekday == 3 ~ 1, TRUE ~ 0),
    thursday_dummy = case_when(weekday == 4 ~ 1, TRUE ~ 0),
    friday_dummy = case_when(weekday == 5 ~ 1, TRUE ~ 0),
    saturday_dummy = case_when(weekday == 6 ~ 1, TRUE ~ 0),
    sunday_dummy = case_when(weekday == 7 ~ 1, TRUE ~ 0)
  )

# CREATING SEASONAL MODEL FOR PREFERRED MODEL (LINEAR) #

hits_model_linear_dummy_seasonal <- lm(linear_model_dummy ~ 0 + monday_dummy + tuesday_dummy + w
ednesday_dummy + thursday_dummy + friday_dummy + saturday_dummy + sunday_dummy, data = nile)

summary(hits_model_linear_dummy_seasonal)
```

```
##
## Call:
## lm(formula = linear_model_dummy ~ 0 + monday_dummy + tuesday_dummy +
##     wednesday_dummy + thursday_dummy + friday_dummy + saturday_dummy +
##     sunday_dummy, data = nile)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9702 -2.9371 -0.1796  2.4817 27.2304
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## monday_dummy     19.3310     0.6867   28.15   <2e-16 ***
## tuesday_dummy    18.6393     0.6867   27.14   <2e-16 ***
## wednesday_dummy  18.6787     0.6867   27.20   <2e-16 ***
## thursday_dummy   18.5802     0.6957   26.71   <2e-16 ***
## friday_dummy     18.6196     0.6957   26.77   <2e-16 ***
## saturday_dummy   19.2522     0.6867   28.04   <2e-16 ***
## sunday_dummy     19.2916     0.6867   28.09   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.288 on 264 degrees of freedom
## Multiple R-squared:  0.9523, Adjusted R-squared:  0.9511
## F-statistic: 753.4 on 7 and 264 DF,  p-value: < 2.2e-16
```

```
hits_model_linear_dummy_seasonal_coef <- data.frame(coef(hits_model_linear_dummy_seasonal)) %>%
  rownames_to_column() %>%
  select(weekday = rowname, seasonal_factor = coef.hits_model_linear_dummy_seasonal.) %>%
  mutate(weekday = str_replace(weekday, "_dummy","")) %>%
  mutate(weekday = factor(weekday)) %>%
  mutate(weekday = fct_relevel(weekday,
                               "monday", "tuesday", "wednesday", "thursday", "friday", "saturda
y", "sunday"))

ggplot(hits_model_linear_dummy_seasonal_coef, aes(weekday, seasonal_factor)) +
  geom_col() +
  labs(x = "Seasonal factor" , y = "Day of week") +
  ggtitle("Chart 4: Day of week effects in the linear dummy model")
```



e.) Your manager would like a forecast of the number of visits on the website for the remaining year. Using your results from parts a.) to d.), select a final model to use for forecasting. Use your model to produce appropriate interval forecasts through the end of 2017. Make sure to provide an interpretation of your forecast. (1 Mark)

Although there is evidence of significant day-of-week effects in the linear model, its not clear if this operation creates a prefferrable model for forecasting. By comparing the AIC and BIC outputs of the various linear models, we can be clear about which model we should use to forecast out to the end of the year. From the model_selection_criteria_linear_models table printed below, was can see that the pure dummy model is preferable from an in-sample fit perspective as is generates the lowest AIC/BIC. Interestingly, the day-of-week model actually generates the worst fit of the linear models.

Given we are seeking to forecast future values of hits to NILE.COM, rather than the conditional mean of hits to NILE.COM, we will be using prediction intervals for this forecast. According to this linear trend model, it tells us that we can expect daily hits to NILE.COM to reach approximately 28 thousand by the end of 2017. We can also be 95% confident daily hits to NILE.COM will fall between 34 thousand and 21 thousand.

```
# DETERMINING IF THE SEASONAL MODEL IMPROVES ON THE OTHER LINEAR MODELS #

model_list_dummy <- list(hits_model_linear, hits_model_linear_dummy,hits_model_linear_dummy_seas
onal)

model_names_dummy <- c("hits_model_linear", "hits_model_linear_dummy","hits_model_linear_dummy_s
easonal")

model_selection_criteria_linear_models <- map2_df(model_list_dummy, model_names_dummy, extract_c
riteria)

print(model_selection_criteria_linear_models)
```

```
##                              Model      AIC      BIC
## 1                hits_model_linear 1567.074 1577.880
## 2          hits_model_linear_dummy 1392.585 1406.993
## 3 hits_model_linear_dummy_seasonal 1567.076 1595.893
```

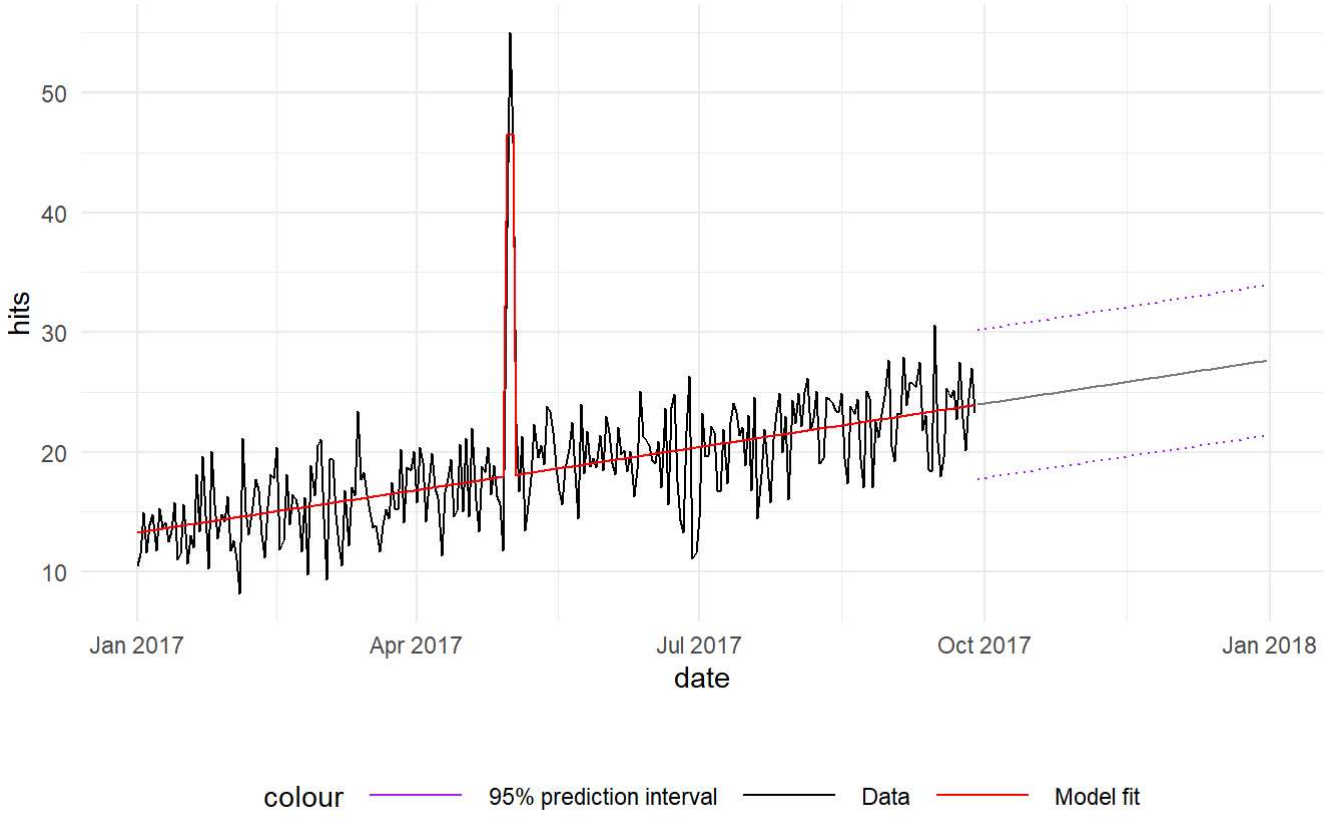
```
# FORECASTING PREFFERRED MODEL (LINEAR DUMMY MODEL) #

nile_f <- data.frame(date = seq(tail(nile$date,1) + days(1), as.Date("2017-12-31"), by = "day"))
%>%
  mutate(

    time = seq(length.out = n(), from = tail(nile$time, 1) + 1, by = 1),
    sale_dummy = 0

  )

pi_forecast <- data.frame(predict(hits_model_linear_dummy, newdata = nile_f, interval = "predict
ion")) %>%
  mutate(date = seq(tail(nile$date,1) + days(1), as.Date("2017-12-31"), by = "day"))


nile_final <- pi_forecast %>%
  select(date, point_estimate = fit)


nile_final <- pi_forecast %>%
  select(date, point_estimate = fit, upper = upr, lower = lwr) %>%
  left_join(nile_f, by = "date") %>%
  bind_rows(nile) %>%
  select(date, hits, linear_model_dummy, point_estimate, upper, lower) %>%
  arrange(date)


ggplot(nile_final, aes(x = date)) + geom_line(aes(y = hits, colour = "Data")) +
  geom_line(aes(y = linear_model_dummy, colour = "Model fit")) +
  geom_line(aes(y = point_estimate, colour = "Point estimate forecast")) +
  geom_line(aes(y = upper, colour = "95% prediction interval"), linetype = "dotted", size = 0.5)
+
  geom_line(aes(y = lower, colour = "95% prediction interval"),linetype = "dotted", size = 0.5)
+
  scale_colour_manual(values = c("Data" = "black", "Model fit" = "red", "Prediction interval for
ecast" = "blue", "95% prediction interval" = "purple")) +
  theme(legend.position = "bottom", legend.key.size = unit(3, "lines")) +
  ggtitle("Chart 5: Daily hits to NILE.COM (01-01-2017 to 28-09-2017), preferred model (linear d
ummy model) fit, \nforecast and prediction intervals") +
  theme(plot.title = element_text(size = 10))
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: Removed 94 rows containing missing values (`geom_line()`).
## Removed 94 rows containing missing values (`geom_line()`).
```

```
## Warning: Removed 271 rows containing missing values (`geom_line()`).
## Removed 271 rows containing missing values (`geom_line()`).
## Removed 271 rows containing missing values (`geom_line()`).
```

Chart 5: Daily hits to NILE.COM (01-01-2017 to 28-09-2017), preferred model (linear dummy model) fit, forecast and prediction intervals



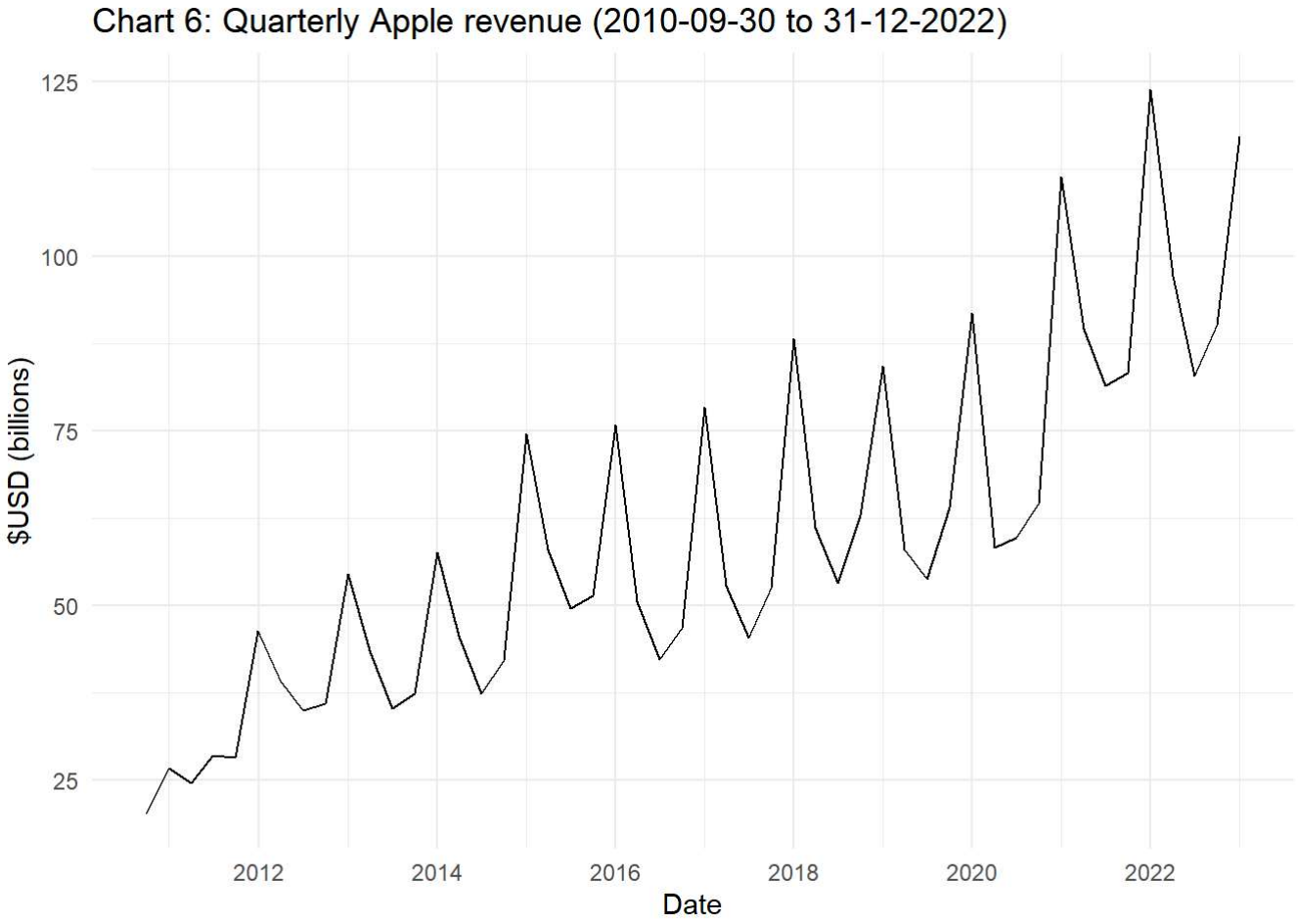
Question 2

The file applerev.csv contains Apple Inc.'s quarterly revenues (measured in billions of USD). The sample period starts at the third quarter of 2010 and ends at the fourth quarter of 2022. You are required to compute all your estimations and plots in R.

a.) Generate an appropriate plot of data and provide a brief description of the observed time series. Make sure to point out any notable visual features/characteristics of the data. (1 Mark)

Chart 6 shows a time series graph of Apple Inc.'s quarterly revenue (measures in billions of USD) from Q3 2010 to Q4 2022. The trend of this time series is upward sloping over time. There is also a very strong seasonal effect in this data, where revenue seems to peak each year in the December quarter - likely due to the holiday season. Additionally, this seasonal effect seems to be multiplicative as it increases proportionally to the level of revenue over time.

```
applerev <- read_excel("applerev.xlsx") %>%
  mutate(date_clean = parse_date_time(date, orders = "mdy")) %>%
  select(date = date_clean, revenue)


ggplot(applerev, aes(x = date, y = revenue)) +
  geom_line() +
  ggtitle("Chart 6: Quarterly Apple revenue (2010-09-30 to 31-12-2022)") +
  labs(y = "$USD (billions)", x = "Date")
```



b.) Using the steps outlined in the lecture slides, compute an appropriate decomposition of the data into its trend-cycle, seasonal and residual/remainder components. Generate appropriate plots of these components and make sure to justify any choices that you have made. (3 Marks)

Chart 7 reflects my attempt to compute an appropriate decomposition of the data. To do this I have made the following assumptions:

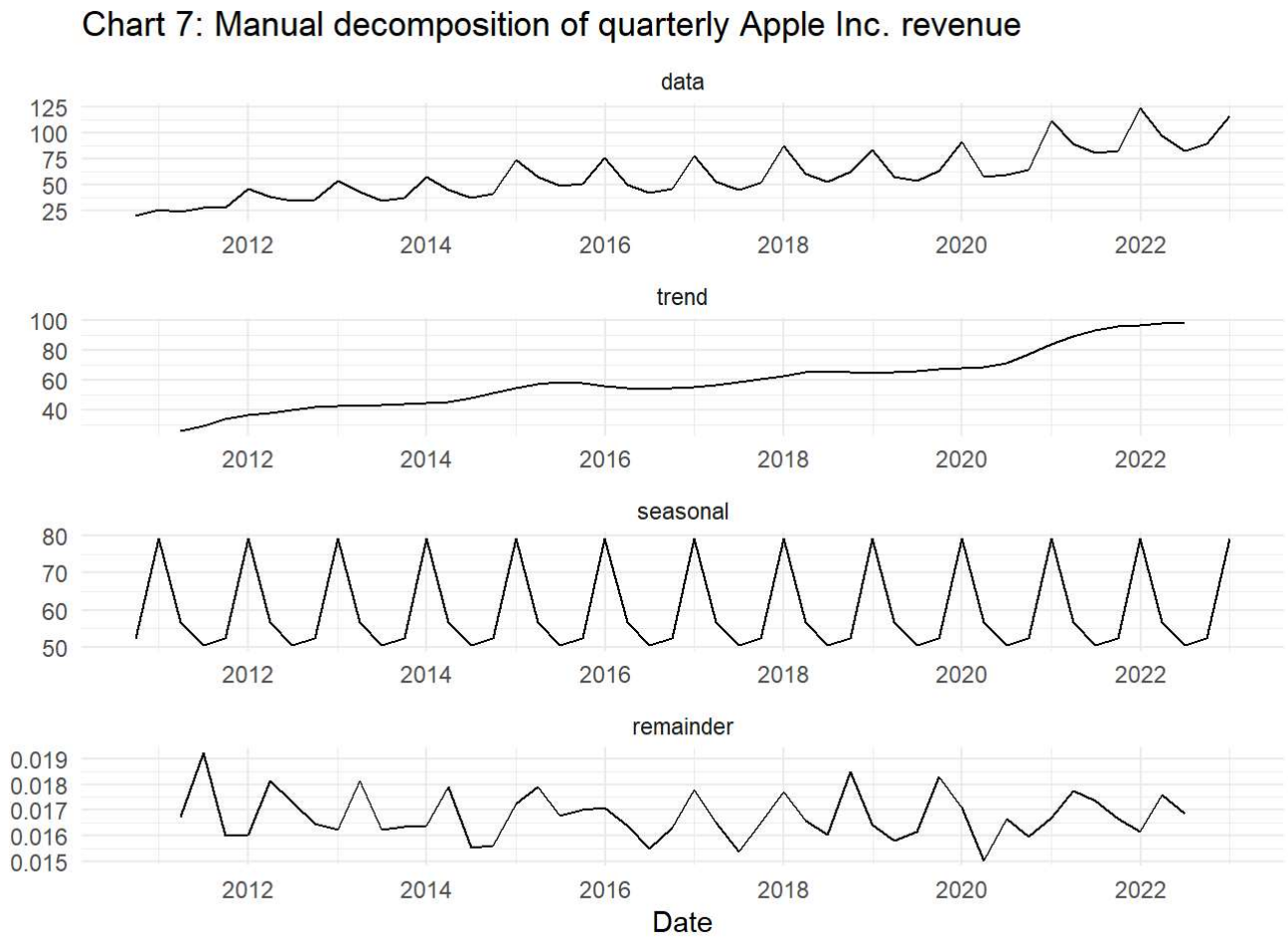
- Because the data is quarterly, I have constructed the seasonal part of the decomposition using a set of quarterly dummies.
- I have computed a multiplicative decomposition of this data because the seasonal trend is multiplicative. This can be deduced by the fact that the difference between annual peaks and troughs increases with revenue levels.
- To construct the trend-cycle part of the decomposition, I have used an MA(4) model. This is because the frequency of the data is quarterly, therefore I ought to choose an MA(m) model where the m is equal to the periodicity of the data's seasonal fluctuations: 4. Because the length of the data is even, I've have to specify a model which take the average of two different weighting periods.
- I have calculated the remainder series by dividing out the estimated seasonal and trend-cycle components form the level (R_hat = y_t / (T_t * S_t * C_t))

```
# SEASONAL #

applerev <- applerev %>%
  mutate(quarter = quarter(date)) %>% #Note: quarters are based on calendar years
  mutate(

    q1_dummy = (case_when(quarter == 1 ~ 1, TRUE ~ 0)),
    q2_dummy = (case_when(quarter == 2 ~ 1, TRUE ~ 0)),
    q3_dummy = (case_when(quarter == 3 ~ 1, TRUE ~ 0)),
    q4_dummy = (case_when(quarter == 4 ~ 1, TRUE ~ 0))

  )

model_applerev_seasonal <- lm(revenue ~ 0 + q1_dummy + q2_dummy + q3_dummy + q4_dummy, data = ap
plerev)

applerev <- applerev %>%
  mutate(seasonal = predict(model_applerev_seasonal))


# TREND-CYCLE #

print(length(applerev$date))
```

```
## [1] 50
```

```
#Note: the number of observations in our time series is 50 and the seasonal pattern is quarterl
y. Therefore, specify an MA(4) model which take the average of two weighting patterns.

applerev <- applerev %>%
  mutate(

    trend = 1/2*(1/4*(lag(revenue,2) + lag(revenue,1) + revenue + lead(revenue,1)) + 1/4*(lag(re
venue,1)  + revenue + lead(revenue,1) + lead(revenue,2)))

  )

# DETRENDED #

applerev <- applerev %>%
  mutate(

    detrended = revenue / trend

  )

# REMAINDER #

applerev <- applerev %>%
  mutate(

    remainder = revenue / (trend * seasonal)

  )


# CHARTING UP MANUAL DECOMPOSITION #

applerev_manual_decomp_data <- applerev %>%
  select(date, data = revenue, trend, seasonal, remainder) %>%
  pivot_longer(-date) %>%
  mutate(name = factor(name, levels = c("data", "trend", "seasonal", "remainder")))

ggplot(applerev_manual_decomp_data, aes(x = date, y = value)) +
  geom_line() +
  facet_wrap(~ name, ncol = 1, scale = "free") +
  ggtitle("Chart 7: Manual decomposition of quarterly Apple Inc. revenue") +
  labs(x = "Date", y = "")
```

```
## Warning: Removed 2 rows containing missing values (`geom_line()`).
```



c.) Compute the same decomposition using the decompose() and compare it with the decomposition that you have computed in part (b). (1 Mark)

There are no clear differences between these two graphs: the shape of all graphs and axes are the same. Please note, due to an error with the autplot function on my device when trying to plot the output of decompose(), I've had to extract all of the content from thsi object into a separate ggplot object via a new dataframe.

The lack of difference between outputs seems to confirm I correctly replicated the automated process for multiplicatively decomposing this timeseries in part (b).

```
applerev_auto_decomp <- applerev %>%
  pull(revenue) %>%
  ts(start = c(2010,3), end = c(2022,4), frequency = 4) %>%
  decompose(type = "multiplicative")



#Note: the autoplot function was giving me an error, so I've had to extract everything into anot
her dataframe and use ggplot2 to graph.

applerev_auto_decomp_data <- data.frame(applerev_auto_decomp$trend) %>%
  cbind(applerev_auto_decomp$seasonal) %>%
  cbind(applerev_auto_decomp$random) %>%
  cbind(applerev$date) %>%
  cbind(applerev$revenue) %>%
  clean_names() %>%
  select(date = applerev_date, data = applerev_revenue, trend = applerev_auto_decomp_trend,
         seasonal = applerev_auto_decomp_seasonal, error = applerev_auto_decomp_random) %>%
    pivot_longer(-date) %>%
  mutate(name = factor(name, levels = c("data", "trend", "seasonal", "random")))

ggplot(applerev_manual_decomp_data, aes(x = date, y = value)) +
  geom_line() +
  facet_wrap(~ name, ncol = 1, scale = "free") +
  ggtitle("Chart 8: Automated decomposition of quarterly Apple Inc. revenue") +
  labs(x = "Date", y = "")
```

```
## Warning: Removed 2 rows containing missing values (`geom_line()`).
```

Chart 8: Automated decomposition of quarterly Apple Inc. revenue