ETF3231/5231 Assignment 2

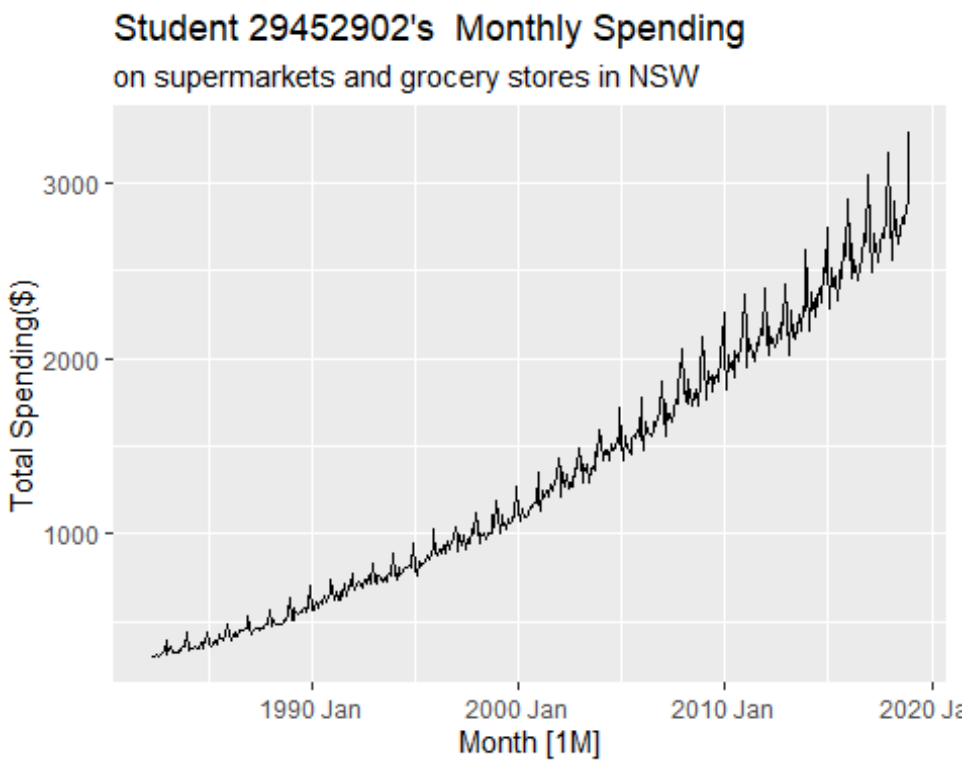
Darren Luwi (ID 29051754)

```
# Read in and tidy up your data
# Make sure you select the column with your student ID

# First three rows contain metadata, read them in separately
meta <- read_csv("Undergrad_Data.csv", col_names = TRUE, n_max = 3)

##
## -- Column specification --------------------------------------------------
------
## cols(
##   .default = col_character()
## )
## i Use `spec()` for the full column specifications.

# meta
# The data follows after the third row, we skip the metadata and read the
data.
# Note: Skipping the first row skips the column names, we add them back from
the
#       metadata.
dat <- read_csv("Undergrad_Data.csv",
                # use column names from the metadata
                col_names = colnames(meta),
                # skip 4 rows as we also skip column names, specified above
                skip = 4,
                # The automatic column types correctly guess all columns but
the
                # date, we specify the date format manually here to correctly
                # get dates.
                col_types = cols("Student ID" = col_date("%b-%y")))

my_series <- dat %>%
  # feel free to rename your series appropriately
  rename(Month = "Student ID", y ="29452902") %>%
  select(Month, y) %>%
  mutate(Month=yearmonth(Month)) %>%
  as_tsibble(index = Month)
```

Question 1

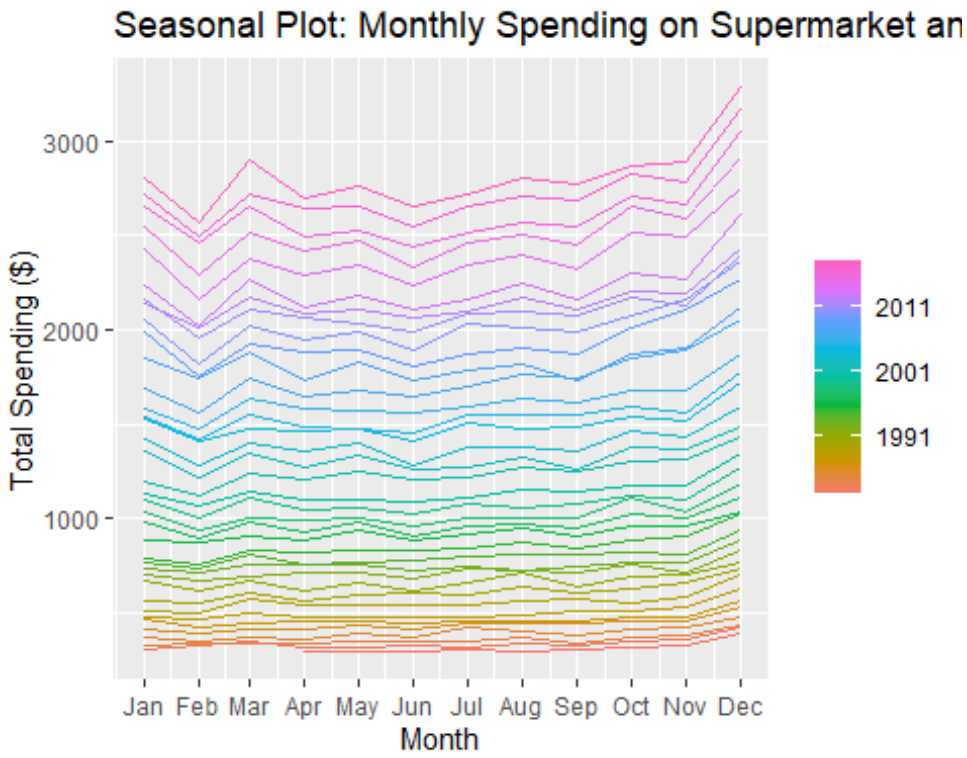
What is your data about (no more than 50 words)? Produce appropriate plots in order to become familiar with your data. Make sure you label your axes and plots appropriately. Comment on these. What do you see? (no more than 50 words per plot). (14 marks)

```r
view(meta)
view(dat)
view(my_series)
##After running the three "view" functions above & observing how these data changes,

##I came into conclusion that the data (ie my_series) shows the total monthly spending of student 29452902

##on supermarkets and grocery stores in New South Wales from April 1982 to December 2018.


autoplot(my_series) + labs(title = "Student 29452902's  Monthly Spending",

subtitle = "on supermarkets and grocery stores in NSW",

y= "Total Spending($)")

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

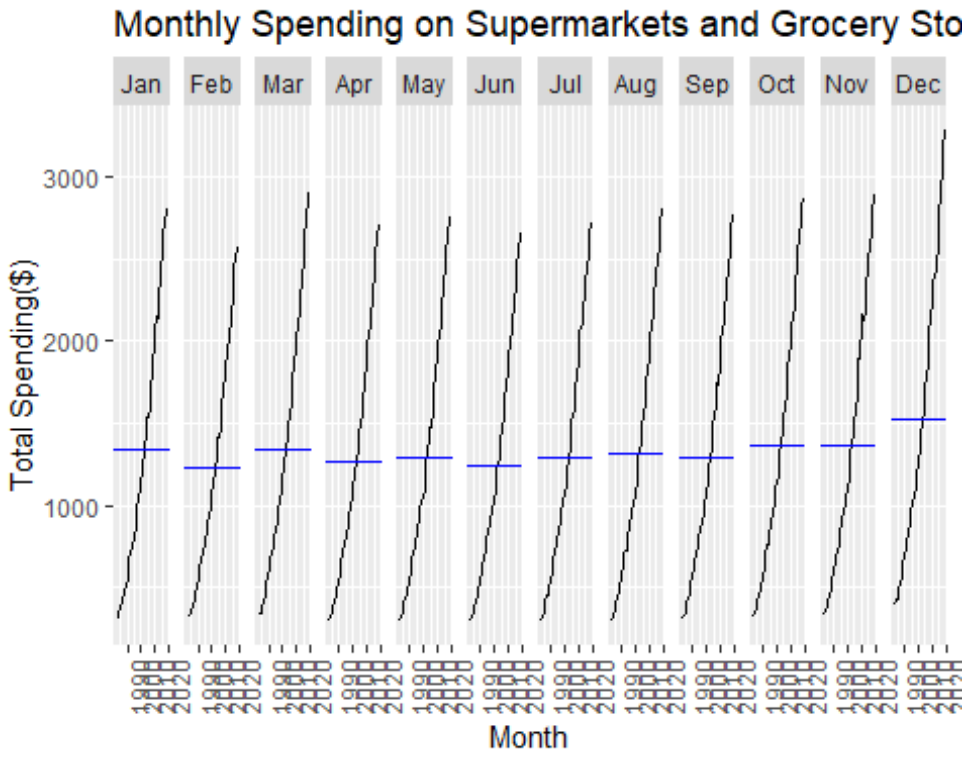



```r
##The autoplot() function shows that there is a strong upward trend and strong seasonality in the student's total monthly spending.

##Meanwhile, it does not suggest any cyclicities.

gg_season(my_series) + labs(
```

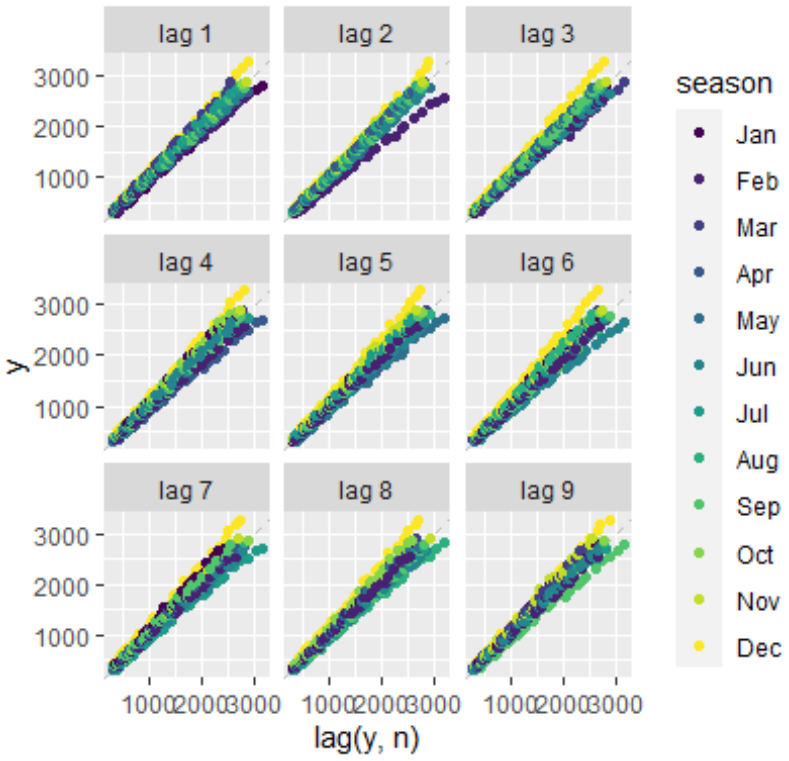
```
title = "Seasonal Plot: Monthly Spending on Supermarket and Grocery Stores in
NSW",

y = "Total Spending ($)")

## Plot variable not specified, automatically selected `y = y`
```



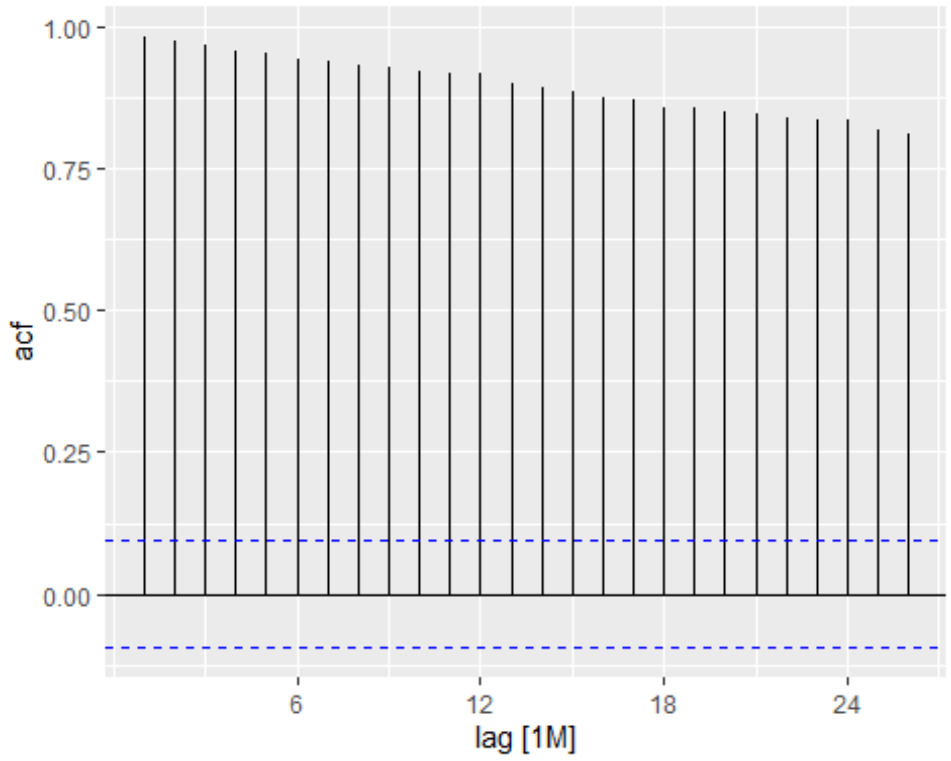

```
##As we can see, the student spent most on December annually.

##This is perhaps due to Christmas and New Year celebrations.

##Simultaneously, spending was the lowest on February (although some years
experienced lower spending on other months),

##before rising again on March (perhaps due to Easter).

gg_subseries(my_series) + labs(

title = "Monthly Spending on Supermarkets and Grocery Stores in NSW",

y= "Total Spending($)")

## Plot variable not specified, automatically selected `y = y`
```




```
##The plot proves my hypothesis of December being the month with the highest spending.

##However, there is a tie between February and June in terms of lowest spending.

gg_lag(my_series, geom = 'point')

## Plot variable not specified, automatically selected `y = y`
```




```
##Since the plots form a 45 degree line, it suggests a strong correlation.

##This is no surprise, considering the strong seasonality of the data.

ACF(my_series) %>%
  autoplot()

## Response variable not specified, automatically selected `var = y`
```

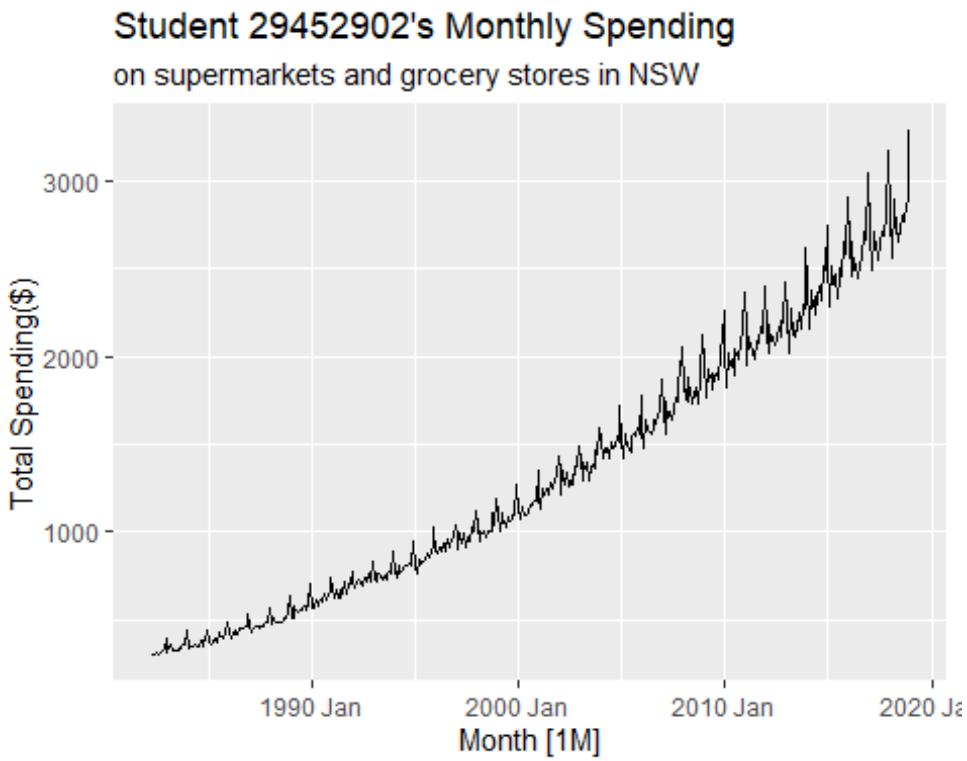



```
##As the "spikes" lie beyond the blue line, we conclude that the data is not
a White Noise,

##showing that there is an autocorrelation in the time series.

##Furthermore, we can see a slow decrease alongside a weak "scalloping"
pattern in the ACF.

##These are due to the trend and seasonality of the data.
```

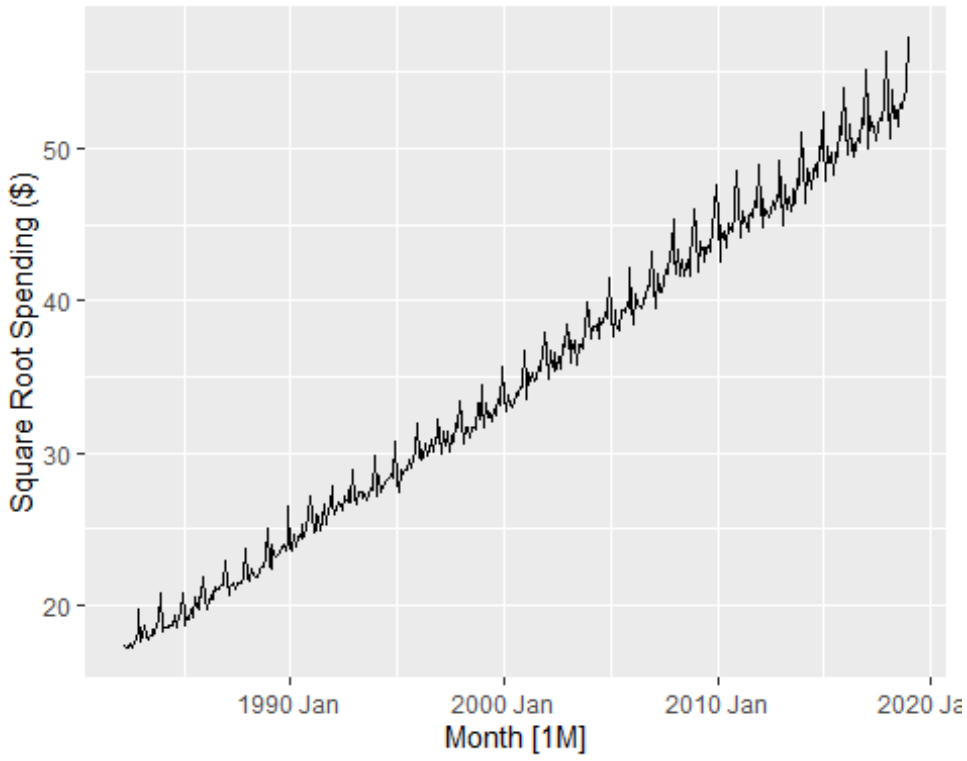
Question 2

Would transforming your data be useful (no more than 50 words)? Compare different transformations graphically. Choose the best transformation if you think a transformation is needed. Justify your choice (no more than 50 words). (5 marks)

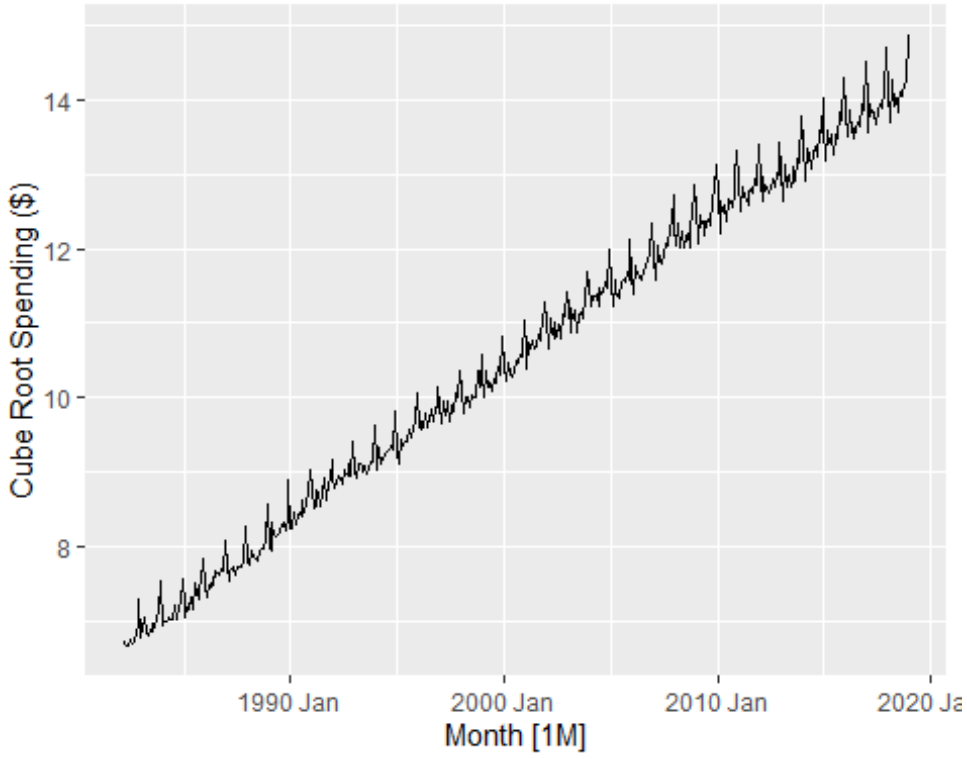
```
autoplot(my_series) + labs(

title = "Student 29452902's Monthly Spending",

subtitle = "on supermarkets and grocery stores in NSW",

y= "Total Spending($)")

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



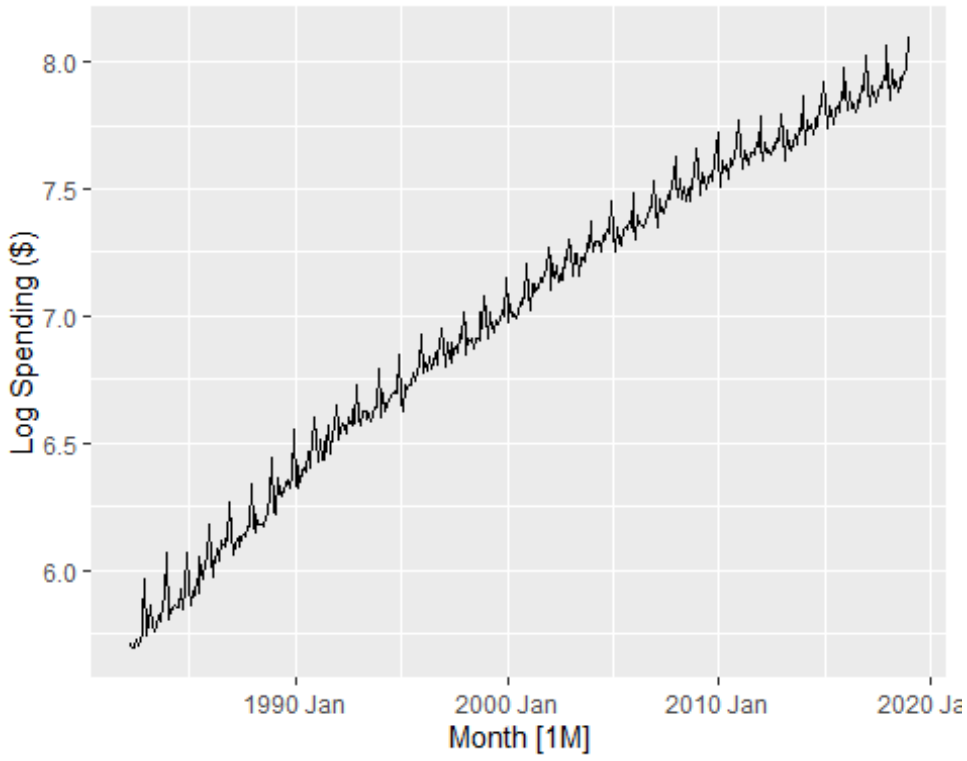

```
##Based on the plot, we can observe that the variations at different levels
differ.

##For instance, in 1995, right after the trough, total spending slowly rose;

##In contrast in 2000 and 2005, after a slight increase after the
trough,total spending nosedived once more.

##This indicates that transformation might be useful.
my_series %>%
  autoplot(sqrt(y)) + labs(y = "Square Root Spending ($)")
```




```
##Graph looks better and more linear.

##However, variations still exist at different levels,

##suggesting that perhaps there are better methods of transformation.
my_series %>%
  autoplot(y^(1/3)) + labs(y = "Cube Root Spending ($)")
```




```
##Similarly, the graph looks better and linear,but variations still exist at
different levels.
my_series %>%
  autoplot(log(y)) + labs(y = "Log Spending ($)")
```




```
##Compared to the previous two transformations, this graph shows less
variations,but it is more "curvey" and non-linear.

##In fact, there is not much difference between this graph and the initial
autoplot() graph.
my_series %>%
  features(my_series, features = guerrero)
```

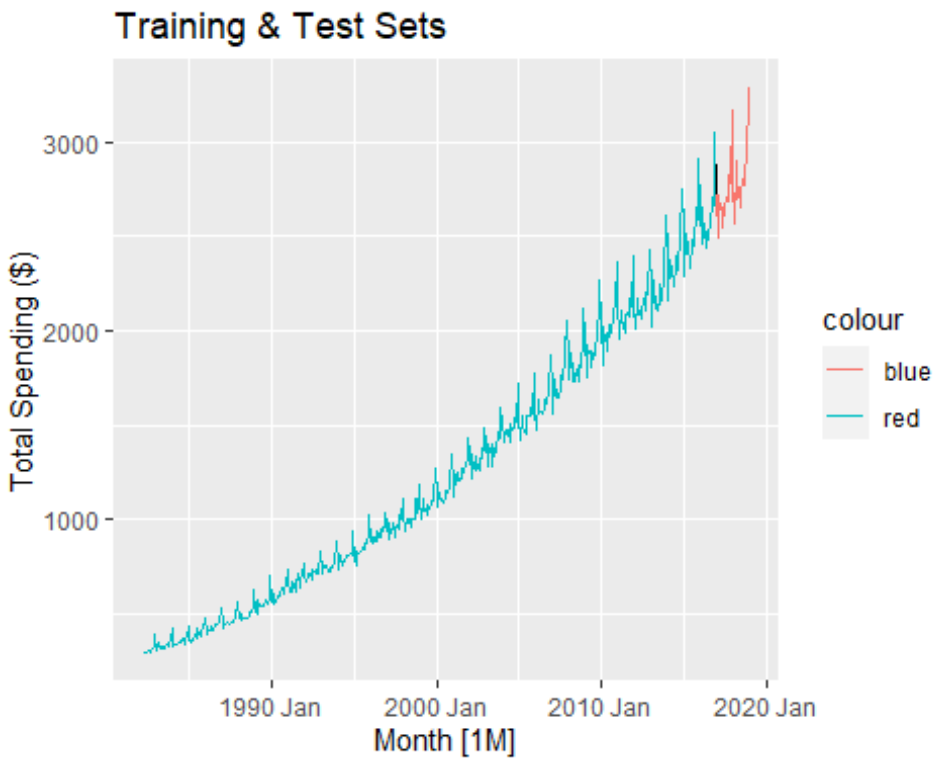
```
## # A tibble: 1 x 1
##   lambda_guerrero
##             <dbl>
## 1            1.02
```

```
##the guerrero function suggests a lambda value of 1.02,

##suggesting that we should "transform" y to y^1,

##which is equivalent to the original y itself.
##Hence, no transformations are required for the data.
```

Question 3

Split your data into training and test sets. Leave the last two years' worth of observations as the test set. Plot these on the same graph to make sure you have done this properly. (3 marks)

```
train <- my_series %>%
  filter(year(Month) <= 2016)
test <- my_series %>%
  filter(year(Month) > 2016)
my_series %>%
  autoplot() +

geom_line(aes(y= y, colour= "red"),data = train) +

geom_line(aes(y=y, colour = "blue"),data = test) +

labs(title = "Training & Test Sets", y = "Total Spending ($)")

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



Question 4

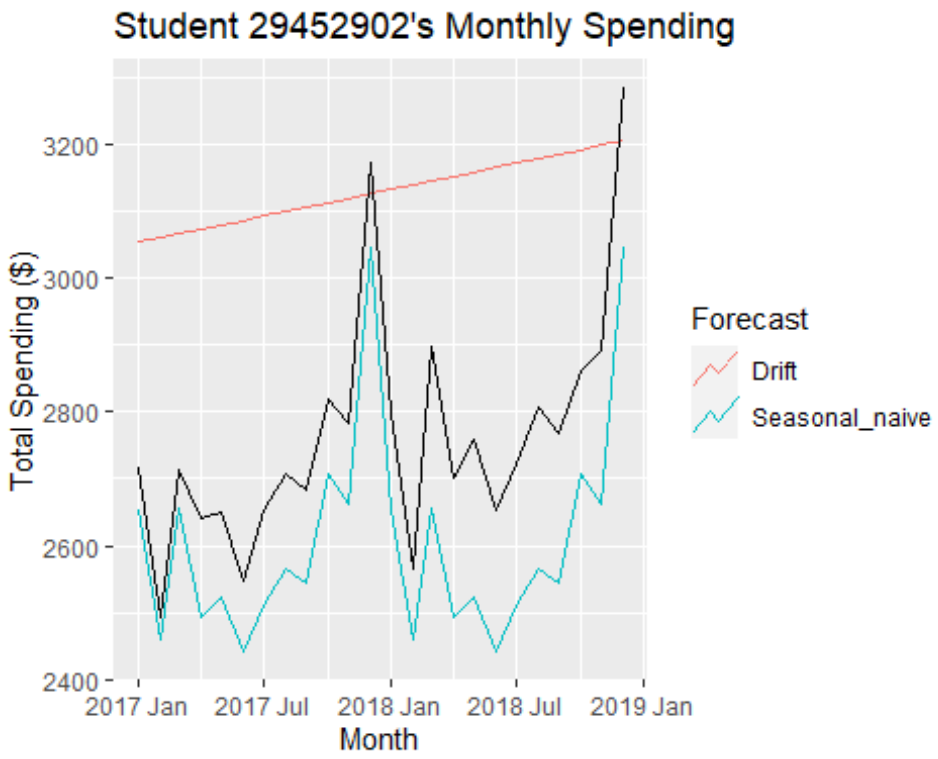
Apply the two most appropriate benchmark methods on the training set. Generate forecasts for the test set and plot them on the same graph. Compare their forecasting performance on the test set. Which method would you choose as the appropriate benchmark? Justify your answer (no more 100 words). (Hint: it will be useful to tabulate your results.) (12 marks)

```
##Since seasonality exists in the data, the Seasonal Naive method might be
useful.
```

```r
#Meanwhile, due to existence of an upward trend, the Drift method might also
be useful.
benchmark <- train %>%
  model(Seasonal_naive = SNAIVE(y),
        Drift = RW(y ~ drift()))
benchmark_acc <- benchmark %>%
  forecast(h = 24)%>%
  accuracy(my_series)
benchmark_acc
```

```
## # A tibble: 2 x 10
##   .model         .type    ME  RMSE   MAE    MPE  MAPE  MASE RMSSE   ACF1
##   <chr>          <chr> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 Drift          Test  -367.  398.  377. -13.6  14.0   5.54  4.96 0.0908
## 2 Seasonal_naive Test   156.  168.  156.   5.63  5.63  2.30  2.10 0.650
```

```r
##The output shows that the Seasonal Drift method have lower Mean Absolute
Error (MAE),

##Mean Absolute Percentage Error(MAPE),Mean Absolute Scaled Error (MASE)

##and Root Mean Squared Scaled Error (RMSSE) values,

##suggesting that the Seasonal Naive method is more accurate and hence

better.
#In addition:
benchmark_fc <- benchmark %>%
  forecast(h = 24)
benchmark_fc %>%
  autoplot(test, level = NULL) +

 labs(title = "Student 29452902's Monthly Spending",

y = "Total Spending ($)") +

 guides(colour = guide_legend(title = "Forecast"))
```

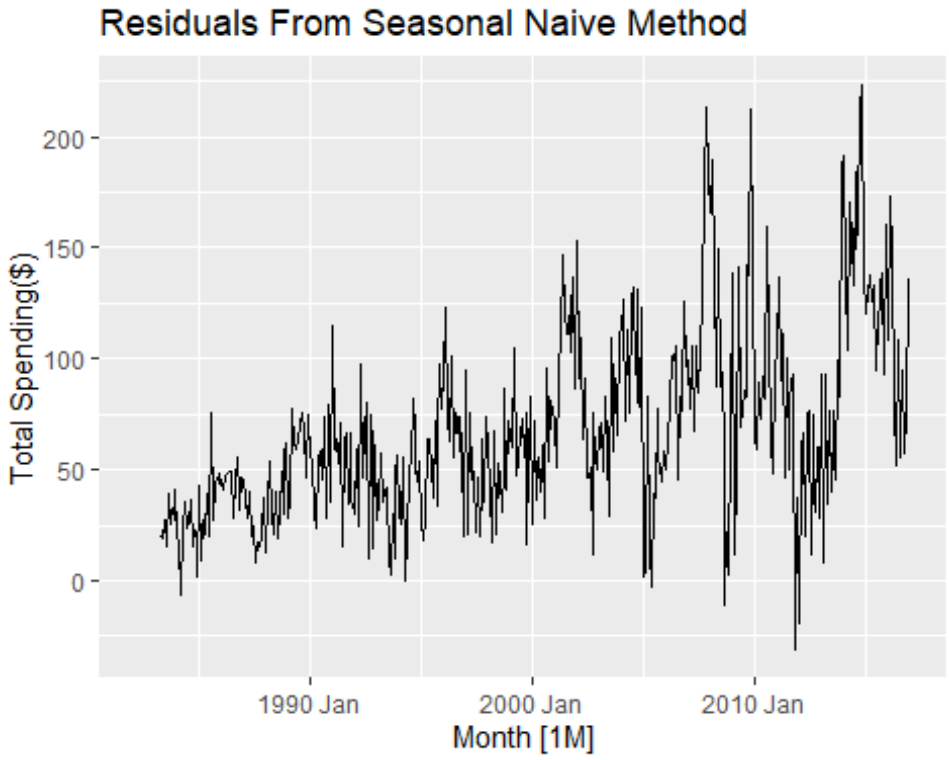



```
##the Seasonal Naive plot looks more similar (in shape) to the test set plot,
##implicitly suggesting that it is more realistic and accurate.
```

Question 5

For the best method, do a residual analysis. Comment on these. What did your forecasting method miss? (no more than 150 words) (8 marks)

```
fit <- train %>%
  model(SNAIVE(y))
aug_fit <- augment(fit)
##Since Snaive's output is influenced by values from previous seasons,

##the .fitted, .resid and .innov values of the first 12 months will be NA
(since there are no data collected prior)
aug_fit %>%
 autoplot(.resid) + labs(

y = "Total Spending($)",

title = "Residuals From Seasonal Naive Method")

## Warning: Removed 12 row(s) containing missing values (geom_path).
```

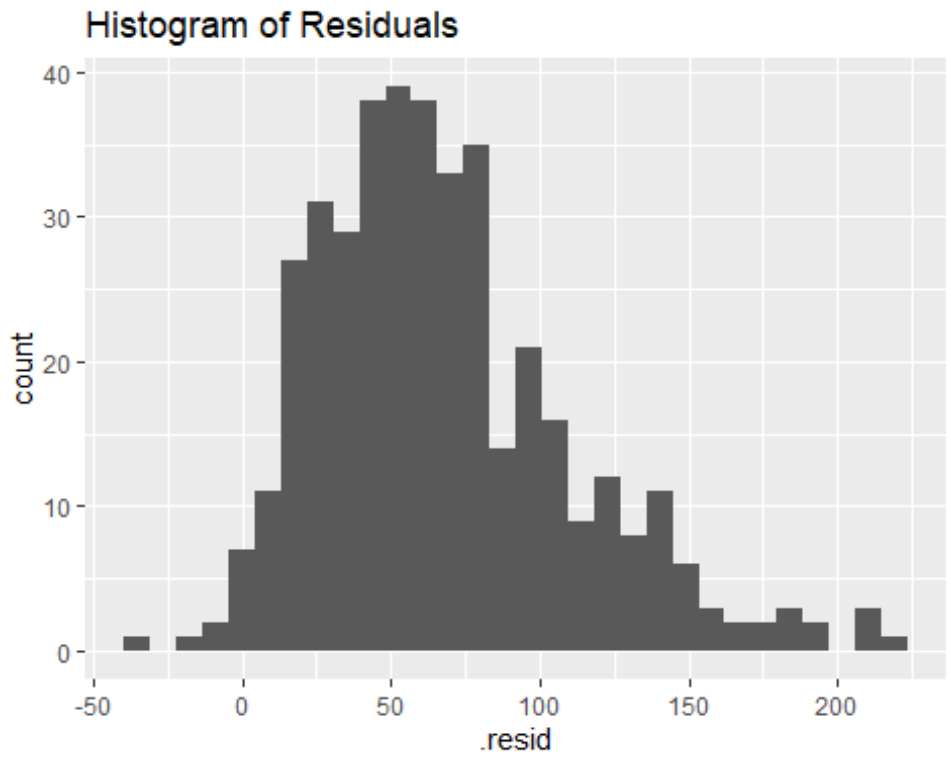



```
##Based on the residuals autoplot,

## we can see that the residuals do not have a mean of zero and are not
uncorrelated.

##To confirm:
```

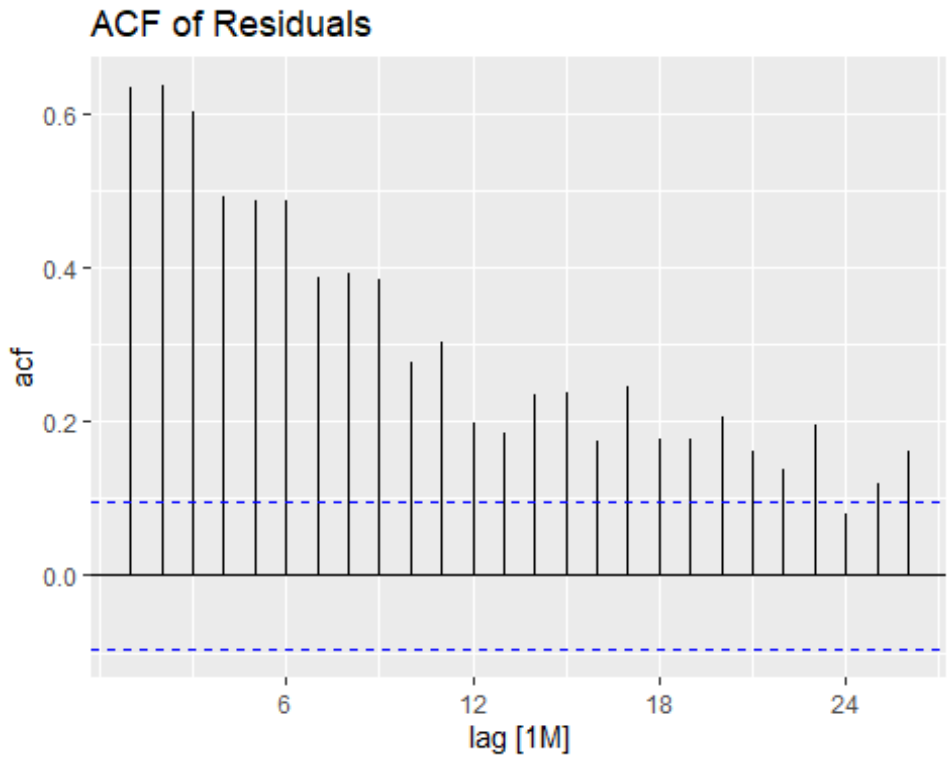
```
aug_fit %>%
  ggplot(aes(x = .resid)) +

  geom_histogram() +

  ggtitle("Histogram of Residuals")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 12 rows containing non-finite values (stat_bin).
```




```
##Here, we can see that the mean of the residuals is not zero.

##This also suggests a non-normal distribution.
aug_fit %>%
  ACF(.resid) %>%
  autoplot() + labs(title = "ACF of Residuals")
```

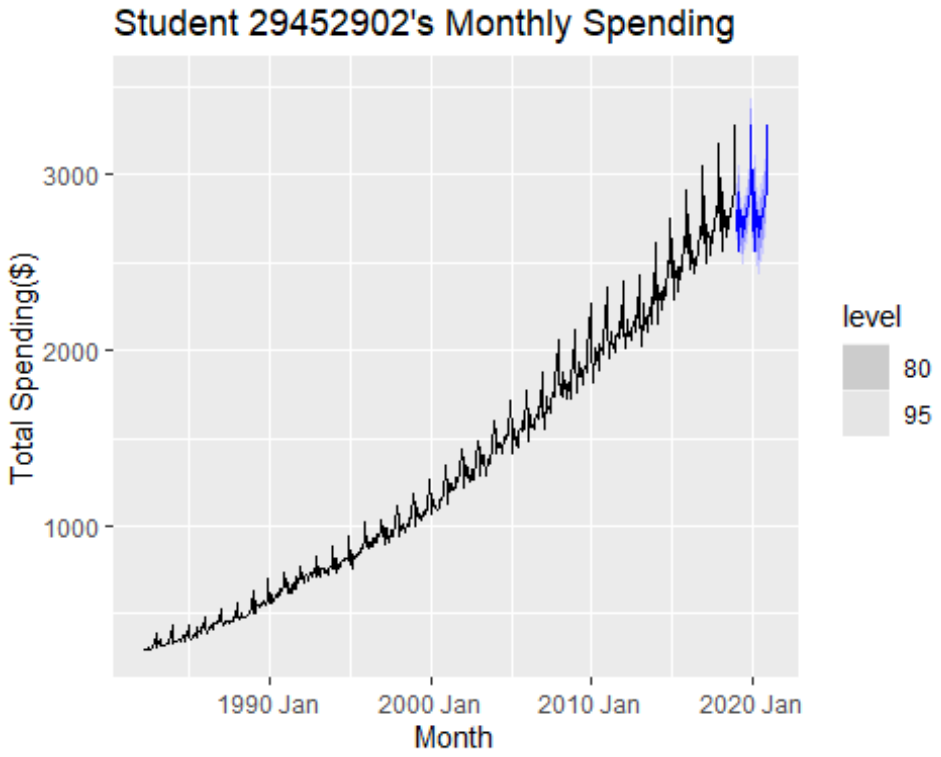



```
##Except for lag 24, all the spikes lie beyond the blue line,

##suggesting that autocorrelation exists.
aug_fit %>%
  features(.resid, ljung_box, h= 2, dof = 0)

## # A tibble: 1 x 3
##   .model    lb_stat lb_pvalue
##   <chr>       <dbl>     <dbl>
## 1 SNAIVE(y)    164.         0

## In addition, the Ljung-Box Test suggests a p-value of 0,

## which is enough to reject the hypothesis that the model

##does not show lack of fit (ie autocorrelation exists).

##In conclusion, the Snaive method violates all assumptions of Forecasting
residuals,

##suggesting that the forecasts are biased

##and extra information are required to compute forecasts.
```

Question 6

Generate point forecasts for the next 2 years (future) from the benchmark method you considered best and plot them. Comment on the point and prediction intervals (no more than 50 words).(4 marks)

```
fit2 <- my_series %>%
  model(SNAIVE(y))
fc <- fit2 %>%
  forecast(h = 24)
fc %>% autoplot(my_series) + labs(

title = "Student 29452902's Monthly Spending",

y= "Total Spending($)")
```




```
fc %>% hilo(level = 95)
```

```
## # A tsibble: 24 x 5 [1M]
## # Key:       .model [1]
##    .model       Month             y .mean                  `95%`
##    <chr>        <mth>        <dist> <dbl>                 <hilo>
##  1 SNAIVE(y) 2019 Jan N(2798, 6705) 2798. [2637.816, 2958.784]95
##  2 SNAIVE(y) 2019 Feb N(2564, 6705) 2564. [2404.016, 2724.984]95
##  3 SNAIVE(y) 2019 Mar N(2897, 6705) 2897. [2736.316, 3057.284]95
##  4 SNAIVE(y) 2019 Apr N(2700, 6705) 2700. [2539.616, 2860.584]95
##  5 SNAIVE(y) 2019 May N(2759, 6705) 2759  [2598.516, 2919.484]95
```

```
##  6 SNAIVE(y) 2019 Jun N(2652, 6705) 2652. [2491.616, 2812.584]95
##  7 SNAIVE(y) 2019 Jul N(2722, 6705) 2722. [2561.016, 2881.984]95
##  8 SNAIVE(y) 2019 Aug N(2806, 6705) 2806. [2645.216, 2966.184]95
##  9 SNAIVE(y) 2019 Sep N(2767, 6705) 2767. [2606.716, 2927.684]95
## 10 SNAIVE(y) 2019 Oct N(2862, 6705) 2862. [2701.816, 3022.784]95
## # ... with 14 more rows
```

```
fc %>% hilo(level = 80)
```

```
## # A tsibble: 24 x 5 [1M]
## # Key:       .model [1]
##    .model       Month             y .mean                 `80%`
##    <chr>        <mth>        <dist> <dbl>                <hilo>
##  1 SNAIVE(y) 2019 Jan N(2798, 6705) 2798. [2693.365, 2903.235]80
##  2 SNAIVE(y) 2019 Feb N(2564, 6705) 2564. [2459.565, 2669.435]80
##  3 SNAIVE(y) 2019 Mar N(2897, 6705) 2897. [2791.865, 3001.735]80
##  4 SNAIVE(y) 2019 Apr N(2700, 6705) 2700. [2595.165, 2805.035]80
##  5 SNAIVE(y) 2019 May N(2759, 6705) 2759  [2654.065, 2863.935]80
##  6 SNAIVE(y) 2019 Jun N(2652, 6705) 2652. [2547.165, 2757.035]80
##  7 SNAIVE(y) 2019 Jul N(2722, 6705) 2722. [2616.565, 2826.435]80
##  8 SNAIVE(y) 2019 Aug N(2806, 6705) 2806. [2700.765, 2910.635]80
##  9 SNAIVE(y) 2019 Sep N(2767, 6705) 2767. [2662.265, 2872.135]80
## 10 SNAIVE(y) 2019 Oct N(2862, 6705) 2862. [2757.365, 2967.235]80
## # ... with 14 more rows
```

```
##The forecasts on both Confidence Levels are similar:

##Monthly spending lies within the range of $2000s and $3000s.


##Since I used the Snaive method,

##monthly spending are similar and will be affected by spendings on the
corresponding seasons,

##explaining why rises and falls are observed in the forecasts.
```

library(knitr) library(ggplot2) tinytex::install_tinytex()