

# Assignment 3

Viyanka Moodley 300565283

2024-08-29

```
hlfs_data <- read.csv("hlfs-jun24qtr-csv.csv", header = TRUE)
# filtering to relevant series
unemployment_rate <- hlfs_data %>%
  filter(Series_reference == "HLFQ.SIQ3")
unemployment_count <- hlfs_data %>%
  filter(Series_reference == "HLFQ.SIK3")
```

## Question 1

a. Convert the unemployment rate and unemployment counts to time series objects and plot them. Make sure you derive year and quarter of each observation from the Period column. The Period value is a combination of the year + the number of the month at the end of the quarter. e.g. 1990.09 is the September (3rd) quarter of 1990.

```
# 1a converting the period column to year and quarter
convert_period <- function(period) {
  year <- as.integer(substr(period, 1, 4))
  month <- as.integer(substr(period, 6, 7))
  quarter <- ifelse(month == 3, 1, ifelse(month == 6, 2, ifelse(month ==
    9, 3, 4)))
  return(paste(year, quarter, sep = " Q"))
}
```

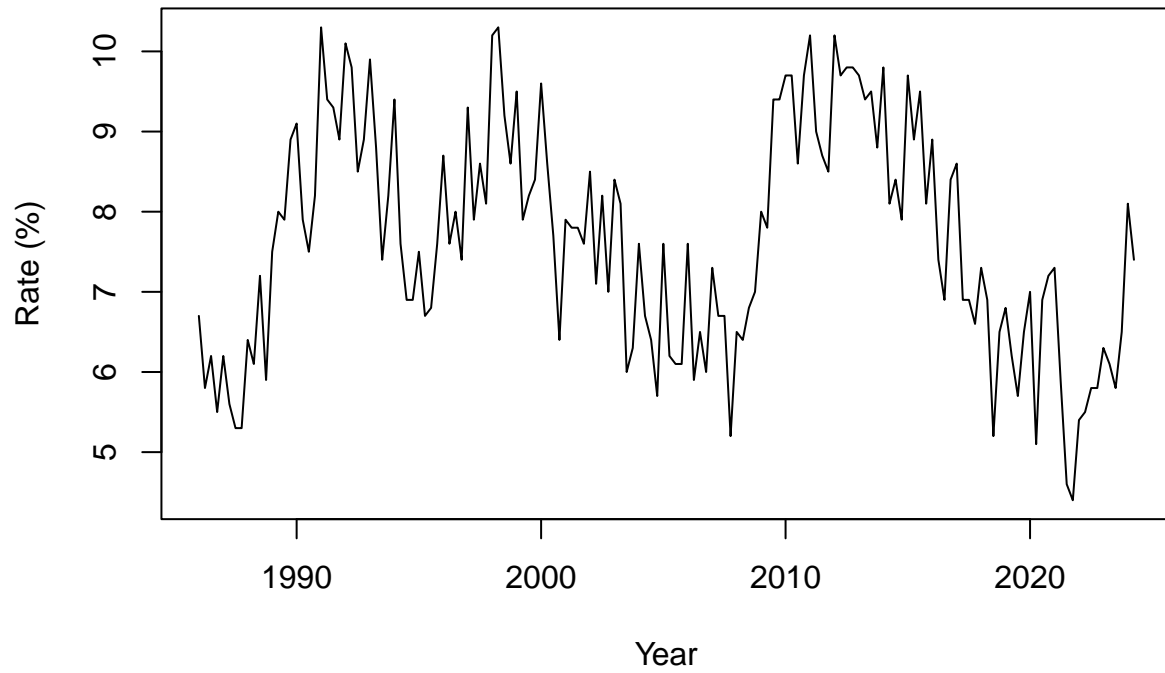
```
unemployment_rate$Year_Quarter <- sapply(unemployment_rate$Period,
  convert_period)

unemployment_count$Year_Quarter <- sapply(unemployment_count$Period,
  convert_period)
```

```
# 1a: converting to time series objects
unemployment_rate_ts <- ts(unemployment_rate$Data_value, start = c(min(as.integer(substr(unemployment_r
  1, 4))), 1), frequency = 4)
unemployment_count_ts <- ts(unemployment_count$Data_value, start = c(min(as.integer(substr(unemployment.
  1, 4))), 1), frequency = 4)

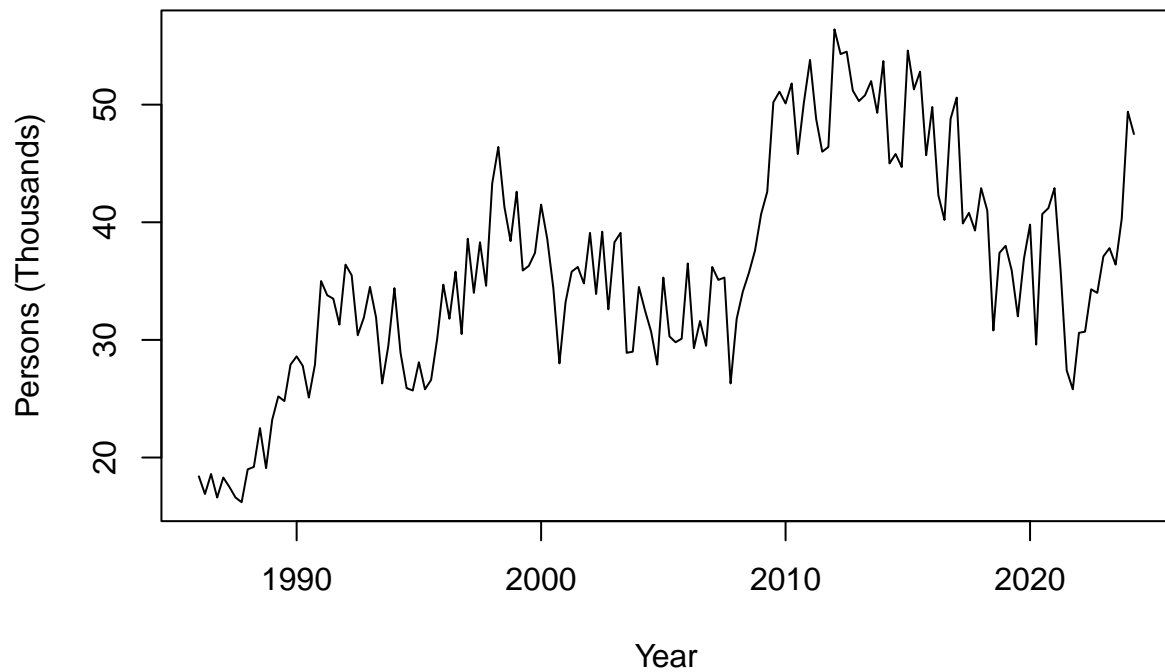
# plotting time series
plot(unemployment_rate_ts, main = "Part-Time Unemployment Rate",
  xlab = "Year", ylab = "Rate (%)")
```

## Part-Time Unemployment Rate



```
plot(unemployment_count_ts, main = "Part-Time Unemployment Count",  
     xlab = "Year", ylab = "Persons (Thousands)")
```

## Part-Time Unemployment Count

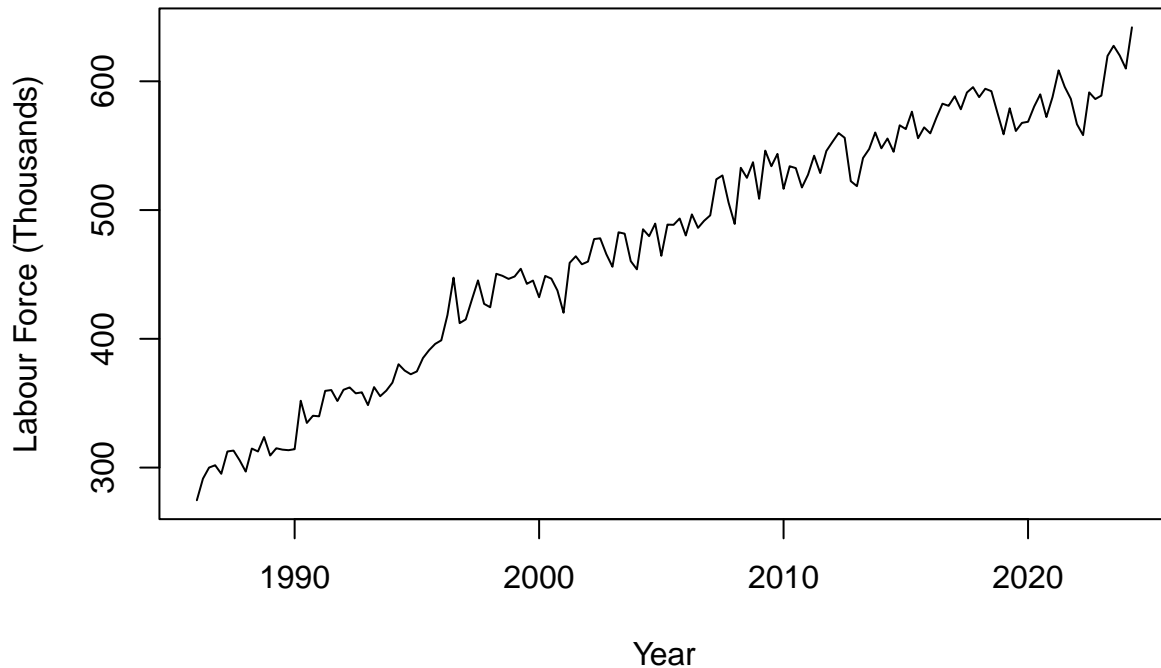


b. Estimate the size of the part time labour force for each quarter from these two time series. Plot and comment on what you find.

```
# 1b: estimating part-time labour force size

part_time_labour_force <- unemployment_count_ts/(unemployment_rate_ts/100)
plot(part_time_labour_force, main = "Part-Time Labour Force Size",
     xlab = "Year", ylab = "Labour Force (Thousands)")
```

## Part-Time Labour Force Size

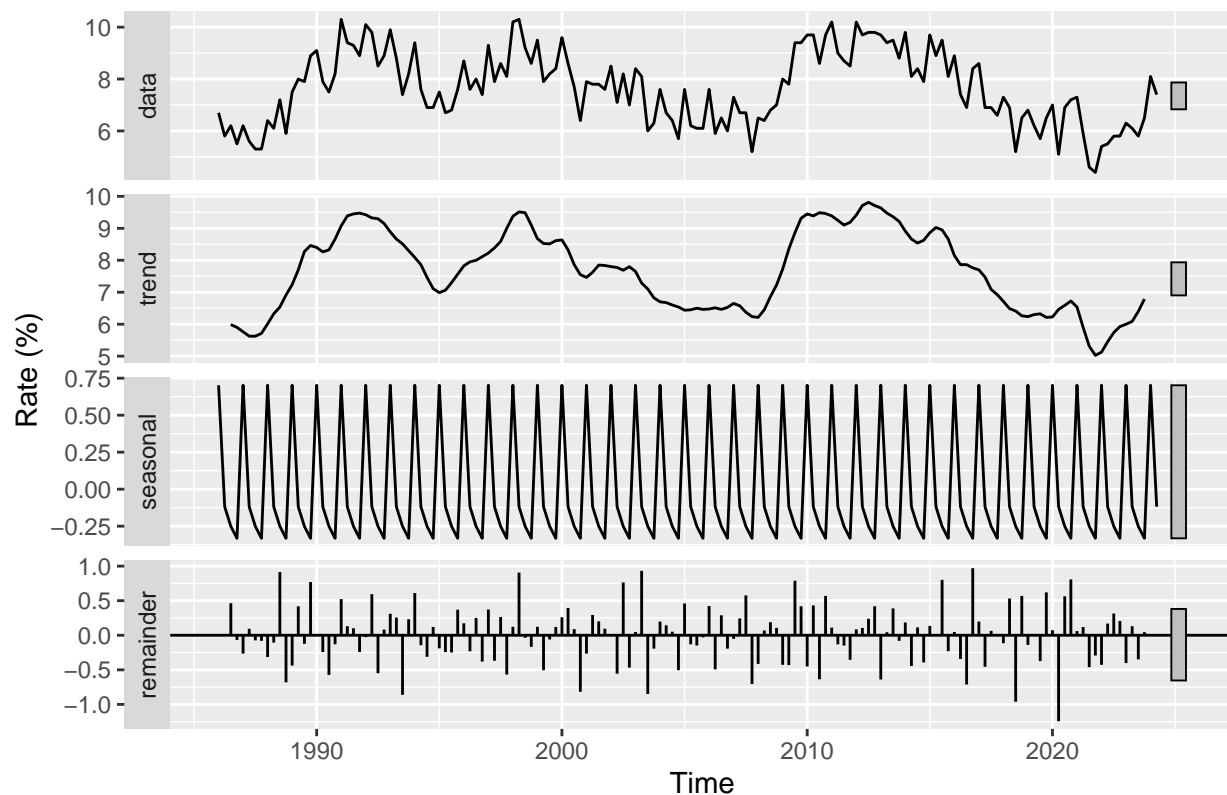


The plot titled “Part-Time Labour Force Size” shows a steady increase in the part-time labour force from around 1988 to 2023, highlighting a growing reliance on or preference for part-time work over time. While the overall trend is upward, the growth is not perfectly linear, with noticeable fluctuations throughout. Around 2009, a significant dip can be observed, likely reflecting the impact of the global financial crisis, followed by a recovery and continued growth. The years following 2020 show increased volatility, potentially linked to the COVID-19 pandemic, which disrupted labour markets worldwide. Despite these fluctuations, the labour force size has roughly doubled, growing from about 300,000 in the late 1980s to over 600,000 by 2023. This trend suggests that part-time work is becoming more common, possibly due to both structural changes in the labour market and its sensitivity to global economic events.

c. Carry out a seasonal decomposition of the part time unemployment rate, and graph the result.

```
unemployment_rate_ts %>%  
  decompose(type = "additive") %>% # using the additive model  
  autoplot() + # using autoplot for auto plotting  
  ggtitle("Seasonal Decomposition of Part-Time Unemployment Rate") +  
  ylab("Rate (%)")
```

## Seasonal Decomposition of Part-Time Unemployment Rate



### d. Test the count of unemployed people for stationarity, also test the first and second differences. Comment on what you find.

```
unemployment_count_ts %>%
  unitroot_kpss()
```

```
##   kpss_stat kpss_pvalue
##   1.568644   0.010000
```

The KPSS test on the original series ( p-value = 0.01) suggests that the series is not stationary.

```
kpss_test_original <- unitroot_kpss(unemployment_count_ts)
```

```
# Test for the first difference
first_diff_ts <- diff(unemployment_count_ts, differences = 1)
kpss_test_diff1 <- unitroot_kpss(first_diff_ts)
print(kpss_test_diff1)
```

```
##   kpss_stat kpss_pvalue
##   0.05636235 0.10000000
```

```
# Test for the second difference
second_diff_ts <- diff(unemployment_count_ts, differences = 2)
kpss_test_diff2 <- unitroot_kpss(second_diff_ts)
print(kpss_test_diff2)
```

```
##    kpss_stat kpss_pvalue
## 0.01769115 0.10000000
```

After applying the first difference, the KPSS statistic drops to 0.056 with a p-value of 0.1, meaning the series becomes stationary. The second difference further confirms stationarity (statistic = 0.018, p-value = 0.1), but it is unnecessary since the first difference already achieves stationarity. Therefore, the series is suitable for time series modeling after the first differencing.

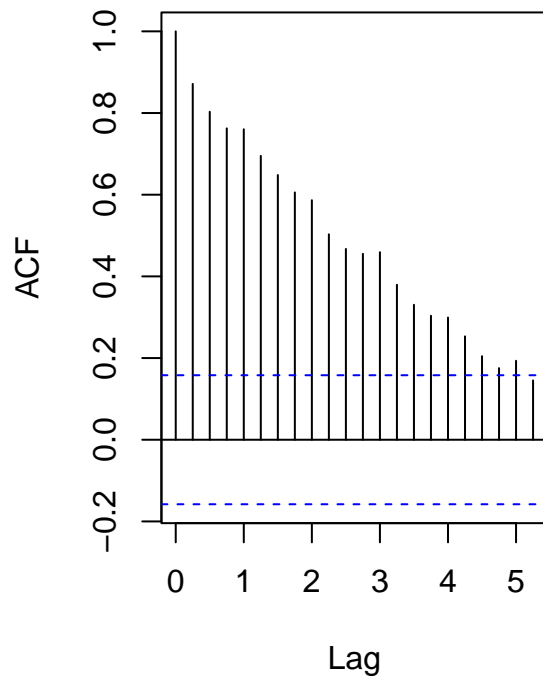
**e. Fit a seasonal ARIMA model to the count of unemployed people, and interpret the result - referring to your answer regarding stationarity.**

```
unemployment_count_ts %>%
  auto.arima()
```

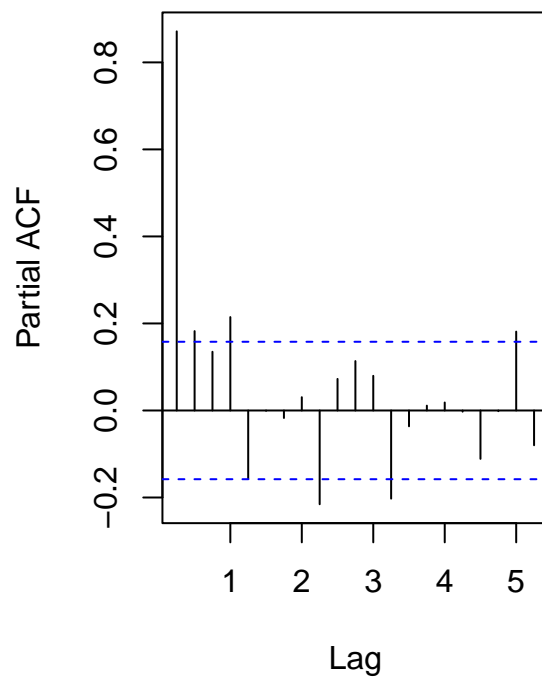
```
## Series: .
## ARIMA(0,1,2)(0,0,2)[4]
##
## Coefficients:
##          ma1      ma2      sma1      sma2
##      -0.3413 -0.1307  0.1653  0.1493
## s.e.   0.0796   0.0776  0.0905  0.0833
##
## sigma^2 = 16.51: log likelihood = -429.79
## AIC=869.57   AICc=869.98   BIC=884.72
```

```
par(mfrow = c(1, 2))
acf(unemployment_count_ts)
pacf(unemployment_count_ts)
```

Series unemployment\_count\_ts



Series unemployment\_count\_ts



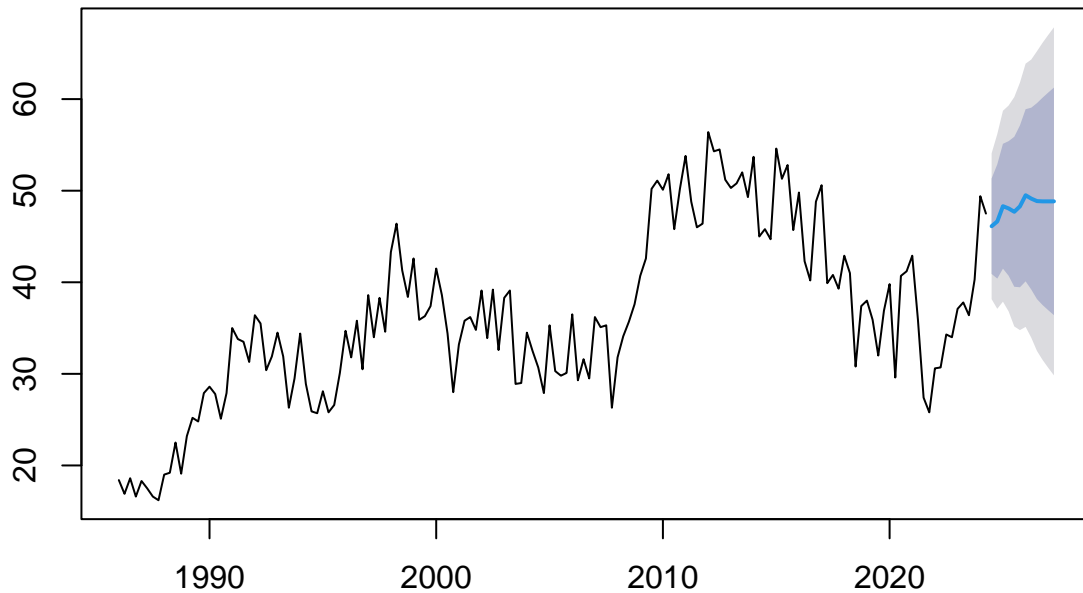
The SARIMA(0,1,2)(0,0,2)[4] model effectively captures both the non-stationary nature of the original series, as identified by the KPSS test, and the seasonality with a period of 4. The first differencing ( $d=1$ ) successfully transformed the series to stationary, while the two non-seasonal moving average terms (MA(2)) and two seasonal moving average terms (SMA(2)) account for the temporal dependencies and moderate seasonal effects, respectively. The negative MA coefficients suggest a dampening effect of past shocks, while the positive SMA coefficients indicate a moderate seasonal influence. The model's fit, indicated by the  $\sigma^2$  value of 16.51 and information criteria (AIC, AICc, BIC), suggests a reasonable balance between complexity and explanatory power. Overall, the model is well suited for forecasting.

f. Forecast the ARIMA fit three years (12 quarters) ahead. Plot the result, and give a 95% prediction interval for the number of part time unemployed in the March Quarter of 2027, commenting on the quality of the result.

```
fit <- unemployment_count_ts %>%
  auto.arima()
forecast_result <- forecast(fit, h = 12)

# plotting forecast result
plot(forecast_result)
```

## Forecasts from ARIMA(0,1,2)(0,0,2)[4]



```
# getting
march_2027_forecast <- forecast_result$mean[12]
march_2027_lower <- forecast_result$lower[12, 2] # 95% ci lower value
march_2027_upper <- forecast_result$upper[12, 2] # 95% ci upper value

cat("Forecast for the March Quarter of 2027:\n")
```

```
## Forecast for the March Quarter of 2027:
```

```
cat("Point Forecast:", round(march_2027_forecast, 2), "\n")
```

```
## Point Forecast: 48.83
```

```
cat("95% Prediction Interval: [", round(march_2027_lower, 2),
    ",", round(march_2027_upper, 2), "]\n")
```

```
## 95% Prediction Interval: [ 29.83 , 67.84 ]
```

The forecast for the March Quarter of 2027 predicts a point estimate of 48.83 for the number of part-time unemployed people, with a 95% prediction interval between 29.83 and 67.84. The broad range of this interval indicates significant uncertainty, which is common in longer-term forecasts, such as this three-year outlook. Although the model captures key patterns like trends and seasonality, the wide prediction interval suggests that the actual number could vary substantially due to unforeseen factors. Overall, the forecast provides a useful general sense of direction and potential outcomes, but it should be interpreted with caution, especially for making precise short-term predictions.

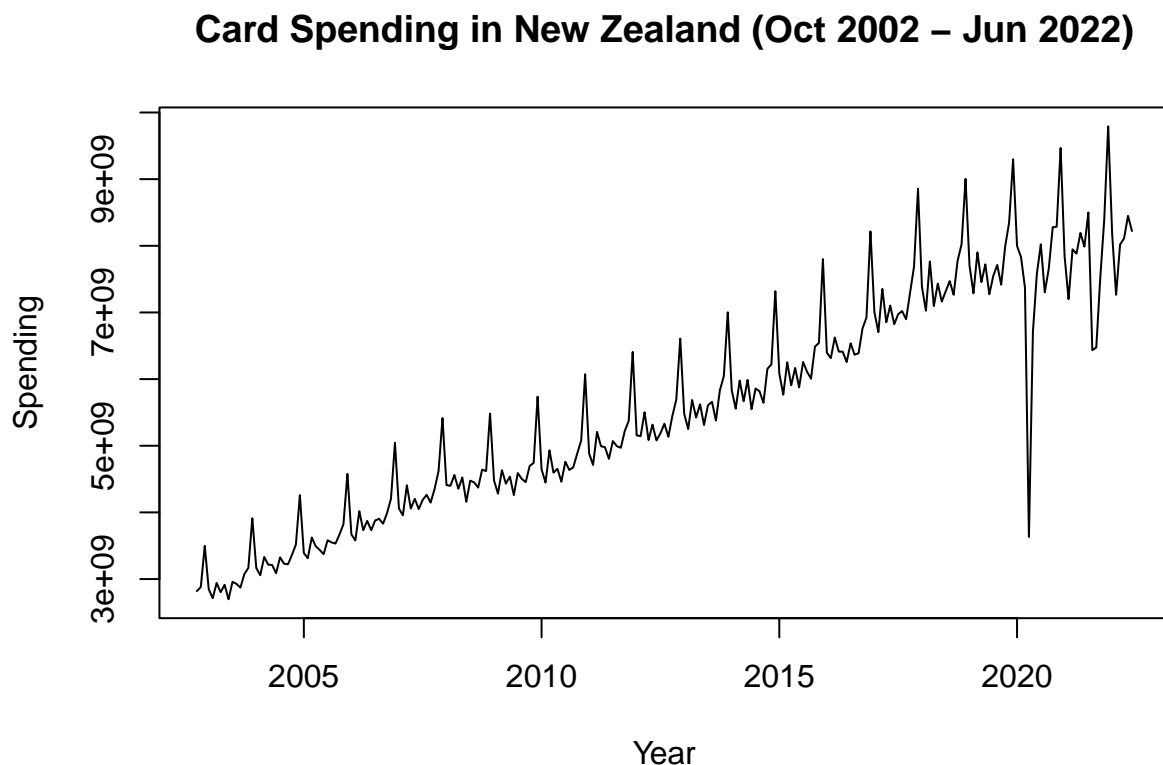


## Question 2.

a. Read the 'card.csv' file which contains data on retail card spending for New Zealand from 2002. Convert the data into a time series object. Plot the data and comment on the general features of it.

```
card <- read.csv("card-1.csv", header = TRUE)
card_ts <- ts(card[, 2], start = c(2002, 10), end = c(2022, 6),
  frequency = 12)
```

```
plot(card_ts, main = "Card Spending in New Zealand (Oct 2002 - Jun 2022)",
  xlab = "Year", ylab = "Spending")
```

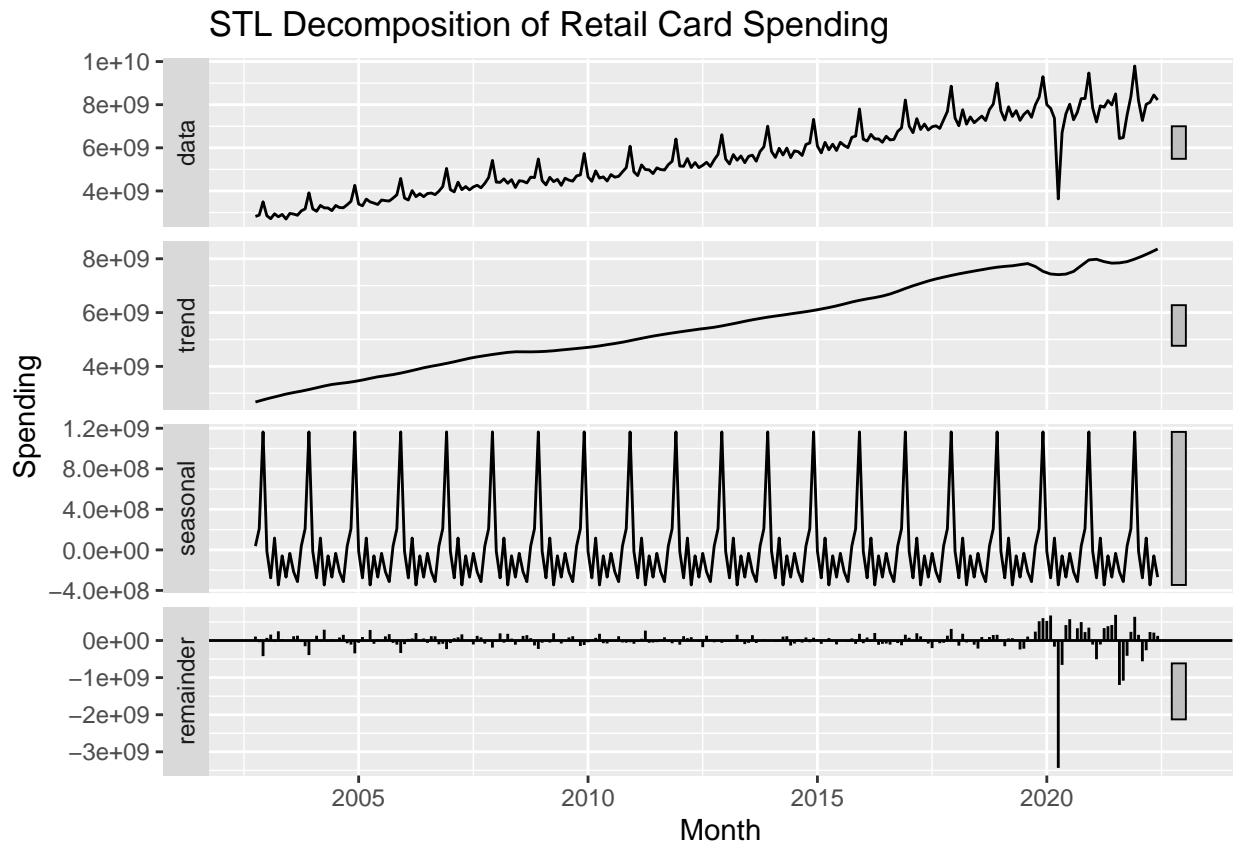


The plot of retail card spending in New Zealand from October 2002 to June 2022 shows a steady upward trend, indicating consistent growth in consumer spending over time. There is a clear seasonal pattern with regular peaks and dips which are likely tied to predictable periods of high spending such as holidays, followed by lower spending in other months. Around 2019, there was a noticeable increase in volatility, which could be linked to various economic factors or changes in consumer behavior.

One of the data's most prominent features is the sharp drop in spending around 2020, which aligns with the impact of the COVID-19 pandemic. This decline reflects the effects of lockdowns, shopping restrictions, and economic uncertainty on consumer activity. After this drop, there is a rapid rebound, but with continued fluctuations, suggesting that while spending recovered, it remained more unpredictable than in previous years. Overall, the data highlights a long term growth trend with strong seasonal effects, temporarily interrupted by a significant economic shock, followed by a period of increased volatility.

b. Carry out a seasonal decomposition, and comment on the features of each of the components.

```
decomp_card <- stl(card_ts, s.window = "periodic")
autoplot(decomp_card) + ggtitle("STL Decomposition of Retail Card Spending") +
  labs(x = "Month", y = "Spending")
```



The STL decomposition of retail card spending in New Zealand provides several insights into the data's underlying patterns. The data component shows a general upward trend from October 2002 to June 2022, with some fluctuations and a sharp decline around 2020, likely reflecting the impact of the COVID-19 pandemic. After this drop, spending quickly rebounds but remains more volatile than in earlier years. The trend component reinforces the steady rise in spending over time, suggesting economic growth, inflation, or changing consumer habits. However, there is a slight flattening around 2019-2020 which could be due to the temporary economic downturn caused by the pandemic.

The seasonal component highlights a clear and recurring yearly pattern, with regular peaks and dips that align with predictable consumer behavior, such as increased spending during holidays. The relatively consistent size of these fluctuations over the years suggests that the seasonal trends have remained stable. Meanwhile, the remainder (residual) component captures irregular changes that cannot be explained by the trend or seasonality, such as the sharp dip in 2020 and the increased variability afterward. This heightened variability after 2020 reflects the uncertainty caused by external shocks, like the pandemic, and indicates a period of greater unpredictability in spending patterns. Overall, the decomposition reveals a long term growth trend, stable seasonal effects, and significant disruptions due to external events.

c. Make a rough estimate the amount of card spending volume that was lost during the month of April 2020. (Give your answer in billions of dollars lost.) There are several ways you could do this: briefly explain the method you choose to use.

```
start_year <- 2002
start_month <- 10

# getting the index for April 2020
index_april_2020 <- (2020 - start_year) * 12 + (4 - start_month) +
  1

# getting actual spending for april 2020
april_2020_actual <- card_ts[index_april_2020]

# getting trend/seasonal comps for april 2020
april_2020_trend <- decomp_card$time.series[index_april_2020,
  "trend"]
april_2020_seasonal <- decomp_card$time.series[index_april_2020,
  "seasonal"]

# estimating expected spending
april_2020_expected <- april_2020_trend + april_2020_seasonal

# getting the amount lost
april_2020_loss <- april_2020_expected - april_2020_actual

# converting to billions
april_2020_loss_billion <- april_2020_loss/1e+09

# printing estimated loss
cat("Estimated card spending volume lost in April 2020: ", round(april_2020_loss_billion,
  2), "billion dollars\n")
```

```
## Estimated card spending volume lost in April 2020:  3.43 billion dollars
```

Explanation: To estimate the card spending volume lost in April 2020, I used the difference method based on the STL decomposition of the time series. This approach involves calculating the expected spending for April 2020 by summing the trend and seasonal components, which represent what spending would have been under normal conditions, without extraordinary disruptions. I then compared this expected spending to the actual spending recorded in April 2020, with the difference between these two values representing the estimated loss due to the COVID-19 pandemic's impact. This method allows us to quantify the extent of the spending reduction by identifying the deviation from the usual seasonal patterns and long-term trends. Comments: The estimated card spending volume lost in April 2020 is approximately 3.43 billion dollars. This estimate reflects the difference between the actual card spending during that month and the expected spending based on normal trends and seasonal patterns, excluding the effects of external shocks like the COVID-19 pandemic. The substantial loss indicates a significant impact on consumer behavior and spending activity during that period, likely due to lockdowns, economic uncertainty, and restrictions on retail activities. It highlights the magnitude of the economic disruption caused by the pandemic and the resulting sharp decline in retail activity.