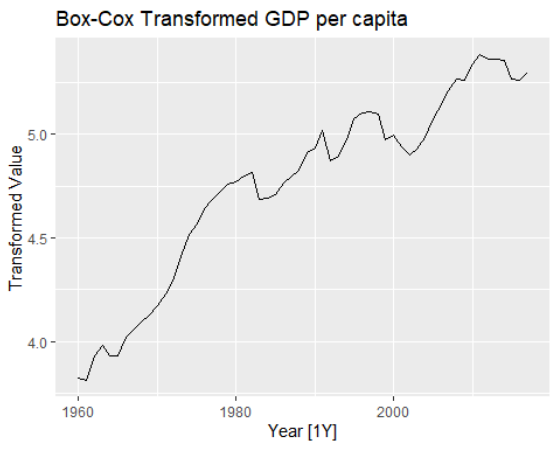
## Question 1

**Use the global economic indicators dataset (*global_economy*) and choose a country of your own choice except USA to perform the following tasks or answer the questions. (20 marks)**

```
# exploring the dataset:
head(global_economy)

## # A tsibble: 6 x 9 [1Y]
## # Key:       Country [1]
##   Country     Code   Year         GDP Growth   CPI Imports Exports Population
##   <fct>       <fct> <dbl>       <dbl>  <dbl> <dbl>   <dbl>   <dbl>      <dbl>
## 1 Afghanistan AFG    1960  537777811.     NA    NA    7.02    4.13    8996351
## 2 Afghanistan AFG    1961  548888896.     NA    NA    8.10    4.45    9166764
## 3 Afghanistan AFG    1962  546666678.     NA    NA    9.35    4.88    9345868
## 4 Afghanistan AFG    1963  751111191.     NA    NA   16.9     9.17    9533954
## 5 Afghanistan AFG    1964  800000044.     NA    NA   18.1     8.89    9731361
## 6 Afghanistan AFG    1965 1006666638.     NA    NA   21.4    11.3     9938414

# Filter Country Brazil as ‘bra_eco’ from ‘global_economy’:
bra_eco <- global_economy |>
  filter(Country == 'Brazil')
head(bra_eco)

## # A tsibble: 6 x 9 [1Y]
## # Key:       Country [1]
##   Country Code   Year          GDP Growth   CPI Imports Exports Population
##   <fct>   <fct> <dbl>        <dbl>  <dbl> <dbl>   <dbl>   <dbl>      <dbl>
## 1 Brazil  BRA    1960 15165569913. NA        NA    7.12    7.06   72207554
## 2 Brazil  BRA    1961 15236854859. 10.3      NA    7.34    7.28   74351763
## 3 Brazil  BRA    1962 19926293839.  5.22     NA    5.19    3.87   76573248
## 4 Brazil  BRA    1963 23021477292.  0.875    NA    9.11    9.04   78854019
## 5 Brazil  BRA    1964 21211892260.  3.49     NA    5.68    6.39   81168654
## 6 Brazil  BRA    1965 21790035117.  3.05     NA    5.56    7.74   83498020

# Finding the GDP per capita of Brazil and adding on the dataset and plotting a   graph:
bra_eco <- bra_eco |>
  mutate(GDP_per_capita = GDP / Population)

bra_eco |>
  autoplot(GDP_per_capita)+
  labs(title= "GDP per capita", y = "$US")
```

```
# The data exhibits fluctuations in overall growth trend, as well as periods of decline and
subsequent growth. To address the variance instability in the data and prepare it for
modeling, we applied the Guerrero transformation. This transformation, with a lambda value
of -0.133254, was chosen to tailor the data to our modeling needs. The transformation
applied serves to stabilize variance, reduce heteroscedasticity, and smooth the trend (see
below).

bra_eco |>
  features(GDP, features = guerrero)

## # A tibble: 1 × 2
##   Country lambda_guerrero
##   <fct>             <dbl>
## 1 Brazil           -0.133

bra_eco_box <- bra_eco |>
  mutate(GDP_per_capita_bx = box_cox(GDP_per_capita, -0.133254))
  autoplot(bra_eco_box, GDP_per_capita_bx) +
  labs(title = "Box-Cox Transformed GDP per capita", y = "Transformed Value")
```

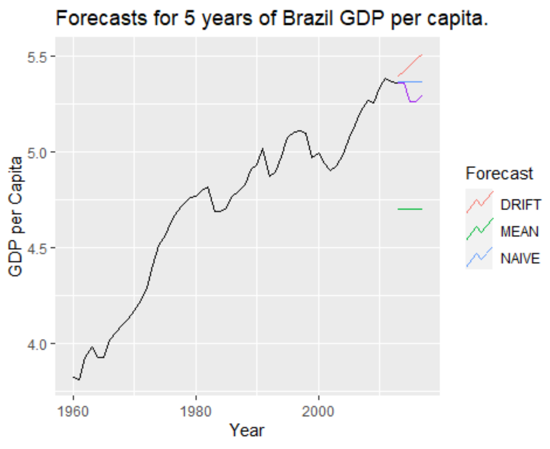


**(a)** **Create a training set by withholding the last five years as a test set. (2 marks)**

```
# Filtering the transformed dataset withholding the last five years as a test set:
train <- bra_eco_box |>
  filter(Year < 2013)
test <- bra_eco_box |>
  filter(Year >= 2013)
```

**(b)** **Fit all the appropriate benchmark methods (i.e. simple forecasting methods) to the training set and forecast the periods covered by the test set. (10 marks)**

```
benchmark_models <- train |>
  model(
    NAIVE = NAIVE(GDP_per_capita_bx),
    MEAN = MEAN(GDP_per_capita_bx),
    DRIFT = RW(GDP_per_capita_bx ~ drift()),
  )
# Forecast the test set periods using the fitted benchmark models:
fc <- benchmark_models |>
  forecast(h = "5 years")

fc |>
  autoplot(bra_eco_box, level = NULL) +
  autolayer(test, GDP_per_capita_bx, colour = "purple")+
  labs(y = "GDP per Capita",
    title = "Forecasts for 5 years of Brazil GDP per capita.") +
  guides(colour = guide_legend(title = "Forecast"))
```



```
# By examining the plot, we can see that the NAIVE forecast closely aligns with the observed values from 2013.
```

**(c)** **Compute the accuracy of your forecasts. Which method does best? (3 marks)**

```
forecast_results <- benchmark_models |>
  forecast(h = "5 years")

forecast_results |>
   accuracy(bra_eco_box)

## # A tibble: 3 × 11
##   .model Country .type      ME   RMSE    MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>  <fct>   <chr>   <dbl>  <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 DRIFT  Brazil  Test  -0.144  0.164  0.144  -2.72  2.72  2.73  2.59 0.448
## 2 MEAN   Brazil  Test   0.605  0.607  0.605  11.4  11.4  11.5   9.57 0.318
## 3 NAIVE  Brazil  Test  -0.0550 0.0703 0.0550 -1.04  1.04  1.04  1.11 0.318

# When considering the RMSE and MAPE metrics, it becomes evident that the NAÏVE method
exhibits the highest level of accuracy, with an RMSE of 0.0702 and an MAPE of 1.0424. In
contrast, the MEAN method displays substantially higher RMSE and MAPE values, underscoring
its lower accuracy when compared to both DRIFT and NAÏVE.
```

**(d) Do the residuals from the best method resemble white noise? (5 marks)**

```
# Residuals from NAIVE method:
bra_eco_box |>
  model(NAIVE(GDP_per_capita_bx)) |>
  gg_tsresiduals() +
  labs(title = "NAIVE Model Residuals")

## Warning: Removed 1 row(s) containing missing values (geom_path).
## Warning: Removed 1 rows containing missing values (geom_point).
## Warning: Removed 1 rows containing non-finite values (stat_bin).
```

```
# NAIVE exhibit similar patterns in their residuals. Observing the Autocorrelation Function
(ACF) plot of the residuals, the spikes representing potential white noise behavior falling
within the range of +/- 0.25. The presence of white noise spikes within an acceptable range
indicates that the residuals exhibit no significant autocorrelation, a certain degree of
randomness and unpredictability in the residual patterns. Thus, the best method NAIVE do
resemble white noise.
```

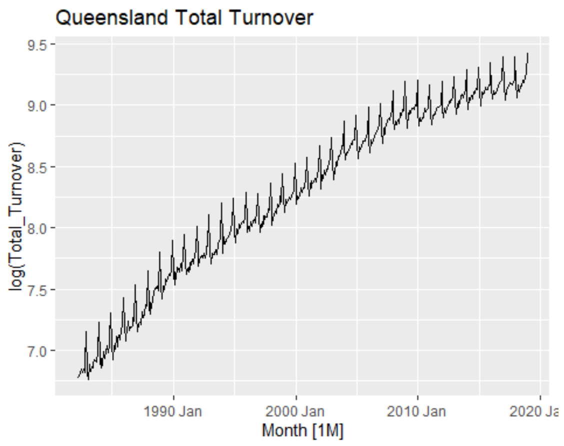
## Question 2

**Use the Australian retail trade turnover dataset (*aus_retail*) to perform the following tasks or answer the questions. (20 marks)**

**(a) Choose a state of your own choice. Aggregate turnover (sum over *Turnover*) using *summarise()*, and discuss the frequency of your data (4 marks)**

```
#  Summarise the turnover data for Queensland, Australia, as well as the aggregated
turnover for Queensland using the summarise() function:
qld <- aus_retail |>
  filter(State == "Queensland")
qld <- qld |>
  summarise(Total_Turnover = sum(Turnover))

qld
## # A tsibble: 441 x 2 [1M]
##       Month Total_Turnover
##       <mth>          <dbl>
##  1 1982 Apr           883.
##  2 1982 May           893.
##  3 1982 Jun           904.
##  4 1982 Jul           940.
##  5 1982 Aug           915.
##  6 1982 Sep           940.
##  7 1982 Oct           918.
##  8 1982 Nov            976
##  9 1982 Dec           1279.
## 10 1983 Jan            918.
## # … with 431 more rows

# Observing the frequency:
qld|>
  autoplot(Total_Turnover)+
  labs(title = "Queensland Total Turnover")
```

```
# The observed frequency pattern indicates that there is a recurring spike in turnover at the end of each month to the beginning of the next, followed by a subsequent drop. As the month progresses, there is a gradual increase in turnover, leading to another spike towards the end of the month.

# Additionally, there is a clear trend of increasing turnover over the years, suggesting a consistent upward movement in turnover values. Applying a logarithmic transformation may yield a more linear trend, while the seasonal variation appears to remain relatively stable across the observed period.

# Applying the log:
qld|>
  autoplot(log(Total_Turnover))+
  labs(title = "Queensland Total Turnover")
```



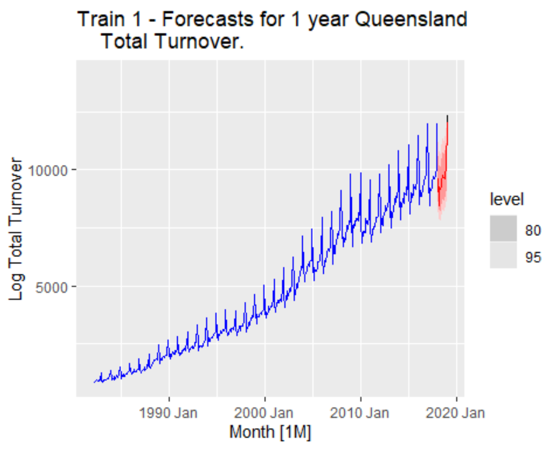
**(b)** **Using *slice()* or *filter()*, create three training sets for this data excluding the last 1, 2 and 3 years. (3 marks) Hint: Look at Lecture 4B slides and the relevant sections in the online textbook for *slice()* and *filter()* functions.**

```
# Creating training set excluding the last 1 year:
train_1 <- qld |>
  filter(year(Month) < 2018)

# Creating training set excluding the last 2 years:
train_2  <- qld |>
  filter(year(Month) < 2017)

# Creating training set excluding the last 3 years:
train_3 <- qld |>
  filter(year(Month) < 2016)
```

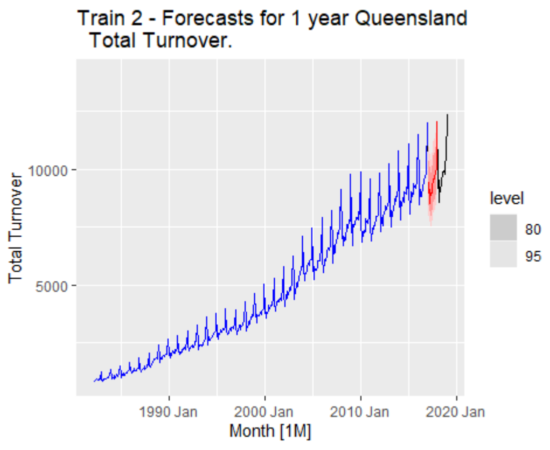
**(c)** **Compute one year of forecasts for each training set using the seasonal naive method. Make a plot of the original series with the forecasts from three sets. (5 marks)**

```
# Forecasting 1 year - train 1:
fit_1 <- train_1 |>
  model(SNAIVE(log(Total_Turnover)))
 fc_1 <- fit_1 |>
```

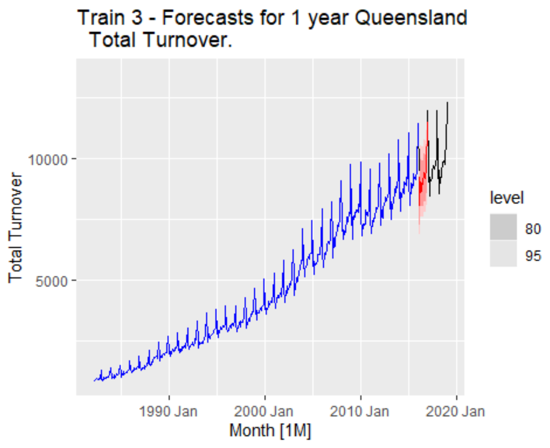
```
  forecast(h = "1 year")
autoplot(qld, Total_Turnover, colour = "black") +
  autolayer(
    train_1, Total_Turnover, colour = "blue")+
   autolayer(fc_1, series = "1-year Forecast", colour = "red") +
labs(y = "Log Total Turnover",
    title = "Train 1 - Forecasts for 1 year Queensland
    Total Turnover.") +
guides(colour = guide_legend(title = "Forecast"))
```



```
# Forecasting 1 year - train 2:
fit_2 <- train_2 |>
  model(SNAIVE(log(Total_Turnover)))
 fc_2 <- fit_2 |>
  forecast(h = "1 year")
autoplot(qld, Total_Turnover, colour = "black") +
  autolayer(
    train_2, Total_Turnover, colour = "blue") +
   autolayer(fc_2, series = "1-year Forecast", colour = "red") +
   labs(y = "Total Turnover",
     title = "Train 2 - Forecasts for 1 year Queensland Total Turnover.") +
guides(colour = guide_legend(title = "Forecast"))
```



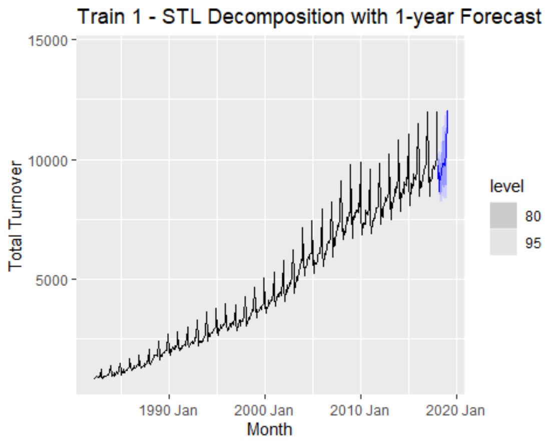
```
# Forecasting 1 year - train 3:
fit_3 <- train_3 |>
  model(SNAIVE(log(Total_Turnover)))
 fc_3 <- fit_3 |>
  forecast(h = "1 year")
autoplot(qld, Total_Turnover, colour = "black") +
  autolayer(train_3, Total_Turnover, colour = "blue") +
  autolayer(fc_3, series = "1-year Forecast", colour = "red") +
  labs(y = "Total Turnover",
    title = "Train 3 - Forecasts for 1 year Queensland Total Turnover.")
```



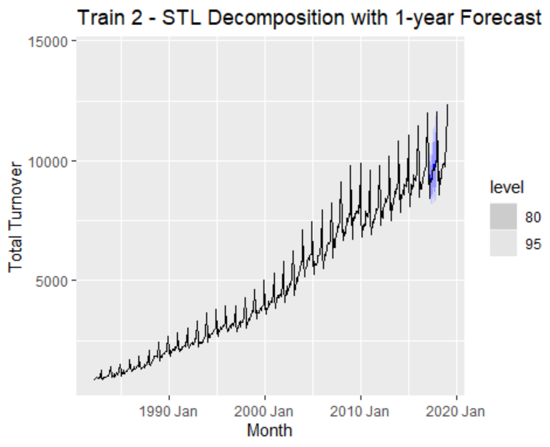
```
# The forecasts from the three sets exhibit similarity and demonstrate strong performance
when employing the seasonal naive method.
```

**(d)** **Compute one year of forecasts for each training set using the STL decomposition method. Make a plot of the original series with the forecasts from three sets. (5 marks)**

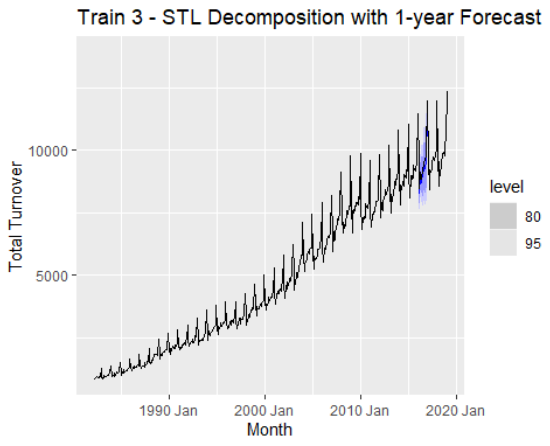
```
# STL Decomposition method for Training 1:
dcmp_1 <- train_1 |>
  model(stlf = decomposition_model(
    STL(log(Total_Turnover) ~ trend(window = 7), robust = TRUE),
    NAIVE(season_adjust)))
dc_fc1 <- dcmp_1 |>
  forecast( h = "1 year")
dc_fc1 |>
  autoplot() +
  autolayer(train_1, Total_Turnover) +
  labs(y = "Total Turnover",
    title = "Train 1 - STL Decomposition with 1-year Forecast")
```



```
# STL Decomposition method for Training 2:
dcmp_2 <- train_2 |>
  model(stlf = decomposition_model(
    STL(log(Total_Turnover) ~ trend(window = 7), robust = TRUE),
    NAIVE(season_adjust)))
dc_fc2 <- dcmp_2 |>
  forecast( h = "1 year")
dc_fc2 |>
  autoplot(qld) +
  autolayer(train_2, Total_Turnover) +
  labs( y = "Total Turnover",
        title = "Train 2 - STL Decomposition with 1-year Forecast")
```



```
# STL Decomposition method for Training 3:
dcmp_3 <- train_3 |>
  model(stlf = decomposition_model(
    STL(log(Total_Turnover) ~ trend(window = 7), robust = TRUE),
    NAIVE(season_adjust)))
dc_fc3 <- dcmp_3 |>
  forecast( h = "1 year")
dc_fc3 |>
  autoplot(qld) +
  autolayer(train_3, Total_Turnover) +
  labs( y = "Total Turnover",
        title = "Train 3 - STL Decomposition with 1-year Forecast")
```



```
# Upon observing the results of the STL Decomposition method applied to all three sets, it
becomes apparent that the forecasts closely mirror the actual dataset.
```

**(e)** **Compare the forecasts between the seasonal naive and the STL decomposition for each training/test set using MAPE. Which one is the best method? (3 marks)**

```
# Calculate MAPE for Seasonal Naive forecasts
mape_snaive <- c(
  accuracy(fc_1, qld)$MAPE,
  accuracy(fc_2, qld)$MAPE,
  accuracy(fc_3, qld)$MAPE)

# Calculate MAPE for STL decomposition forecasts
mape_stl <- c(
  accuracy(dc_fc1, qld)$MAPE,
  accuracy(dc_fc2, qld)$MAPE,
  accuracy(dc_fc3, qld)$MAPE)

# Create a comparison tibble
comparison_table <- tibble(
  Training_Set = c("train 1 - excluded 1 Year", " train 2 - excluded 2 Years", "train 3 -
excluded 3 Years"),
  Seasonal_Naive_MAPE = mape_snaive,
  STL_Decomposition_MAPE = mape_stl)

# Print the comparison table
print(comparison_table)
## # A tibble: 3 × 3
##   Training_Set                     Seasonal_Naive_MAPE STL_Decomposition_MAPE
##   <chr>                                          <dbl>                  <dbl>
## 1 "train 1 - excluded 1 Year"                     2.05                   1.24
## 2 " train 2 - excluded 2 Years"                   1.42                   1.36
## 3 "train 3 - excluded 3 Years"                    2.58                   2.28

# The comparison table provides insight into the Mean Absolute Percentage Error (MAPE)
comparison between the Seasonal Naive and STL Decomposition methods. MAPE is a measure of
the average magnitude of error exhibited by both models.

# It becomes apparent that the STL Decomposition method exhibits the most optimal
performance across all three training sets, boasting the lowest percentage error when
compared to the Seasonal Naive method. Among these training sets, the STL Decomposition
performs particularly well with the train 1 data (excluded 1 year), yielding the lowest
percentage error of 1.2409.

# In summary, the STL Decomposition method demonstrates greater effectiveness in generating
accurate forecasts, displaying a lower MAPE of 1.24%. This establishes it as the favored
choice for attaining optimal model accuracy within this context.
```

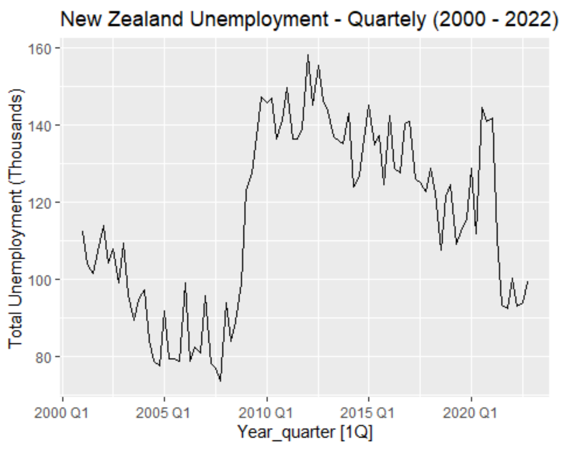
## Question 3

**Use the New Zealand quarterly unemployment data described above. (30 marks)**

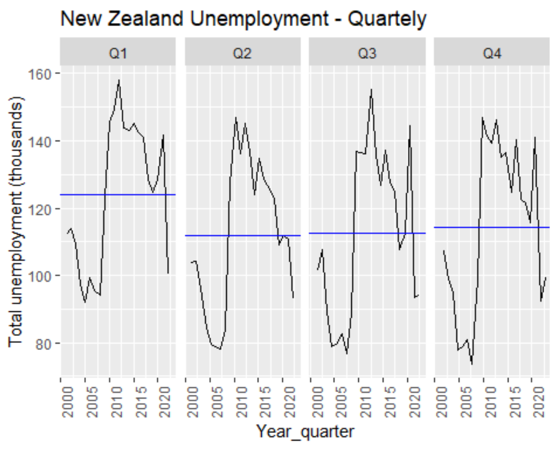
```
# Importing dataset:
# Read the Excel file into a data frame
NZ_Unemployment_Quarterly <-
read.csv("C:/Users/cinar/Desktop/AUT/Forecasting/Assignment_01/NZ_Unemployment_Quarterly.cs
v")

# Preparing the dataset and converting it into a tsibble format:
df <- NZ_Unemployment_Quarterly |>
  mutate(Year_quarter = yearquarter(Year)) |>
  as_tsibble(index = Year_quarter) |>
  select(-Year, -X, -X.1)

head(df)
## # A tsibble: 6 x 2 [1Q]
##   Total_unemployment Year_quarter
##                <dbl>        <qtr>
## 1               112.      2001 Q1
## 2               104.      2001 Q2
## 3               102.      2001 Q3
## 4               107.      2001 Q4
## 5               114.      2002 Q1
## 6               104.      2002 Q2
```

**(a) Plot the series and discuss the main features of the data. (5 marks)**

```
# plotting the data:
autoplot(df) +
  labs(title = "New Zealand Unemployment - Quartely (2000 - 2022)",
       y = "Total Unemployment (Thousands)")

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



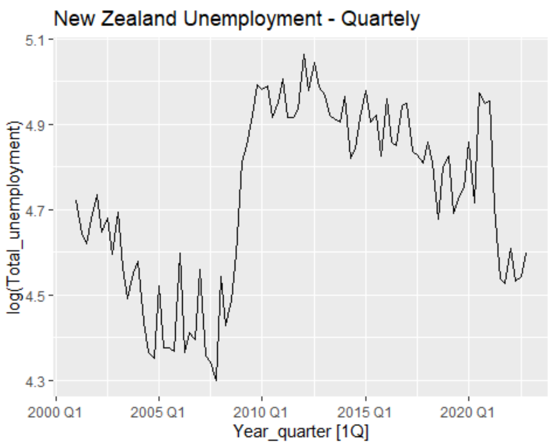
```
# There is a pattern of fluctuation characterized by a declining trend during the initial seven years, followed by a sudden and significant increase in unemployment starting in 2009 which could potentially be attributed to the economic downturn in 2008. Subsequently, there is continued fluctuation from 2010 through the following decade, with a notable sharp decline in unemployment observed in 2021.

# exploring the data applying the subseries plots:
df |>
  gg_subseries() +
  labs(y = "Total unemployment (thousands)",
    title = "New Zealand Unemployment - Quartely")

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



```
# The subseries plots indicate that the most significant increase in unemployment occurred during Q1, with little to no variation observed in Q2, Q3, and Q4.
```

**(b) Discuss whether a transformation is needed. If yes, do so and describe the effect. (6 marks)**

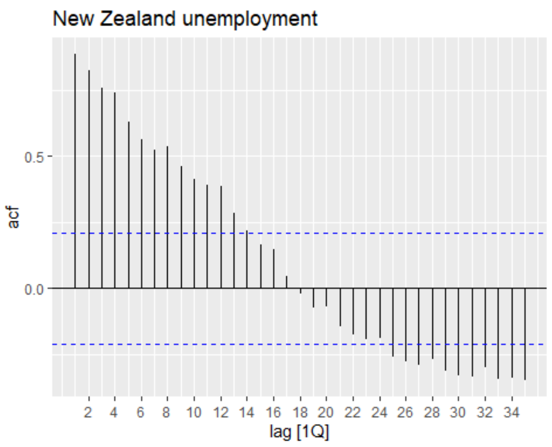
```
# Transformation applying log:
df |>
  autoplot(log(Total_unemployment))+
  labs(title = "New Zealand Unemployment - Quarterly")
```



```
# Observing the log transformation becomes evident that there is a resemblance to the original series. This leads to the conclusion that the transformation has no discernible effect, consequently it is unnecessary to apply a transformation.
```

**(c) Find and discuss whether the autocorrelation exists in this time series. (4 marks)**

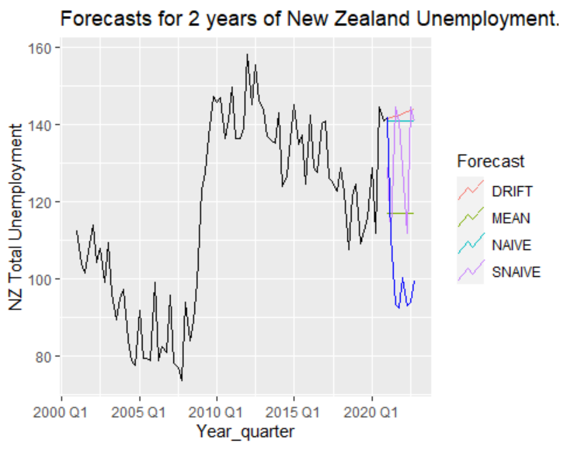
```
# Computing the Autocorrelation Coefficient Function (ACF) for NZ Unemployment, to cover a reasonable range of lags, it is applied lag_max of 20:
df |>
  ACF(Total_unemployment, lag_max = 20)

## # A tsibble: 20 x 2 [1Q]
##          lag     acf
##    <cf_lag>   <dbl>
##  1        1Q  0.885
##  2        2Q  0.826
##  3        3Q  0.756
##  4        4Q  0.740
##  5        5Q  0.630
##  6        6Q  0.565
##  7        7Q  0.523
##  8        8Q  0.536
##  9        9Q  0.462
## 10       10Q  0.414
## 11       11Q  0.392
## 12       12Q  0.388
## 13       13Q  0.283
## 14       14Q  0.218
## 15       15Q  0.166
## 16       16Q  0.148
## 17       17Q  0.0468
## 18       18Q -0.0217
## 19       19Q -0.0720
## 20       20Q -0.0695

# It is observed that there is a strong autocorrelation at 1Q and slow decrease towards to 20Q.

# Plotting ACF to see how the correlation change with lag_max of 35:
df |>
  ACF(Total_unemployment, lag_max = 35) |>
  autoplot() + labs(title="New Zealand unemployment")
```



```
# The autocorrelation function (ACF) plot reveals several key insights about the time series Total unemployment data. Thus, a strong positive autocorrelation at lag 1, with a correlation coefficient of 0.88. This suggests a significant relationship between the current quarter's unemployment rate and the previous quarter's rate. The ACF coefficients gradually decline as the number of lags increases, eventually crossing into negative territory at lag 18, just below zero.

# Notably, there is a lack of significant autocorrelation between lag 14 and 24. This range of lags appears to be uncorrelated, suggesting that the total unemployment for these quarters is relatively independent of each other. Beyond lag 21, the ACF coefficients become negative, with notable negative autocorrelation observed from lag 25 to lag 35. This negative autocorrelation suggests that unemployment in these quarters tend to move in the opposite direction compared to earlier periods, possibly indicating cyclical or seasonal patterns. Therefore, it can be concluded that autocorrelation exists in this time series.
```

**(d)** **Compute two years of forecasts (i.e. holding the last two years of data out as the test set) using the four methods: (1) mean, (2) naive, (3) seasonal naive, and (4) drift. Plot the series and the forecasts and discuss the results. (10 marks)**

```r
# training data by holding two years as the test set:
df_train <- df |>
  filter(year(Year_quarter) <= 2020)
df_test <- df |>
  filter(year(Year_quarter) > 2020)

# Fitting:
fit_train <- df_train |>
  model(
    NAIVE = NAIVE(Total_unemployment),
    SNAIVE = SNAIVE(Total_unemployment),
    MEAN = MEAN(Total_unemployment),
    DRIFT = RW(Total_unemployment ~ drift()),
  )

# Forecast the test set periods using the fitted benchmark models:
fct <- fit_train |>
  forecast(h = 8) # Forecasting for 2 years (8 quarters)

fct |>
  autoplot(df, level = NULL, colour = 'black') +
  autolayer(
    df_test, Total_unemployment, colour = "blue")+
  labs(y = "NZ Total Unemployment",
    title = "Forecasts for 2 years of New Zealand Unemployment.") +
  guides(colour = guide_legend(title = "Forecast"))
```



```
# When examining the forecasts for the upcoming two years starting from the end of 2020, a noteworthy observation emerges: none of the forecasting methods align precisely with the original dataset in blue. Notably, in 2020, it is noticeable a spike in unemployment, potentially attributed to the outbreak of the COVID-19 pandemic. As we project forward from this elevated point of unemployment, it becomes apparent that the forecasts fail to account for a subsequent decline in unemployment during 2021, a trend that was not anticipated. This decline in unemployment may be indicative of businesses' efforts to recover from the economic impact of the pandemic.

# Interestingly, the DRIFT method suggests a prediction of an ongoing increase in unemployment, suggesting the presence of a sustained trend. Meanwhile, the NAÏVE method relies on projecting directly from the most recent data point, which, given the high unemployment reading at the end of 2020, results in an overestimation of future unemployment levels. Finally, the SEASONAL NAIVE method reveals a recurrent pattern of seasonal fluctuations in high unemployment numbers, indicating a predictable cyclical trend.
```

**(e)** **Compare the root mean squared error (RMSE) and the mean absolute percentage error (MAPE) of forecasts from the four methods in (d). Which method do you think is best for this time series? (5 marks)**

```
# Calculate accuracy using accuracy() function and store the result:
 accuracy(fct, df_test)

## # A tibble: 4 × 10
##   .model .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 DRIFT  Test  -39.5  42.7  39.6 -40.9  41.0   NaN   NaN 0.278
## 2 MEAN   Test  -13.8  20.9  20.0 -15.5  19.9   NaN   NaN 0.262
## 3 NAIVE  Test  -37.8  41.0  38.0 -39.3  39.4   NaN   NaN 0.262
## 4 SNAIVE Test  -28.4  36.4  31.6 -30.4  32.7   NaN   NaN 0.253

# The evaluation of forecast accuracy, as measured by root mean squared error (RMSE) and mean absolute percentage error (MAPE), supports the conclusion that the MEAN method outperformed the other forecasting approaches for the two-year prediction. It is noteworthy that the MEAN method consistently displayed the best performance, yielding the smallest errors of 20.88 for RMSE and 19.88 for MAPE across both evaluation periods.
```

## Question 4

**Time series regression models (30 marks)**

**(a)** **Fit a regression model to the quarterly unemployment data with a linear trend and seasonal dummies. Discuss the results. (4 marks)**

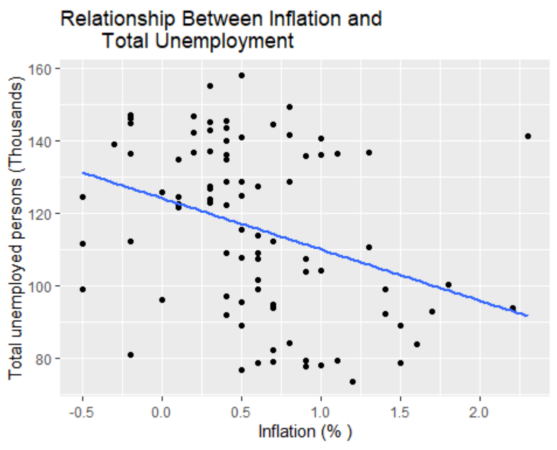
```
fit_df <- df |>
  model(TSLM(Total_unemployment ~ trend() + season()))
report(fit_df)

## Series: Total_unemployment
## Model: TSLM
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.465 -17.015   3.059  17.803  42.034
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   107.97052    5.96615  18.097  < 2e-16 ***
## trend()         0.37405    0.08969   4.171 7.43e-05 ***
## season()year2 -12.48768    6.43825  -1.940   0.0558 .
## season()year3 -12.28446    6.44013  -1.907   0.0599 .
## season()year4 -10.86305    6.44325  -1.686   0.0956 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21.35 on 83 degrees of freedom
## Multiple R-squared: 0.2097,  Adjusted R-squared: 0.1716
## F-statistic: 5.506 on 4 and 83 DF, p-value: 0.00055596

# Coefficient estimate for the intercept shows an estimate of 107.97, which represents the
estimated baseline or initial level of Total Unemployment once trend and seasonality are
not considered.

# Regarding the trend coefficient, an increment of one unit of time, each quarter,
corresponds to an approximate increase of 0.37405 units in Total Unemployment.
This observation suggests a positive linear trend in unemployment over time.

# The coefficients for season year 2, 3, and 4 represent the seasonal dummies for the
second, third, and fourth quarters, respectively. They signify deviations in Total
Unemployment during these quarters compared to the reference, which is the   first quarter.
The negative coefficients imply lower unemployment during these quarters when compared to
the first quarter. On average, this translates to the second quarter having approximately
12.49 thousand less unemployed individuals than the first quarter, the third quarter
exhibiting about 12.28 thousand less unemployed individuals than the first quarter, and the
fourth quarter reflecting roughly 10.86 thousand less unemployed individuals than the first
quarter.

# However, the adjusted R-squared value of 0.1716 indicates that the model's predictive
power is limited, explaining only 17% of the variation in the data.
```

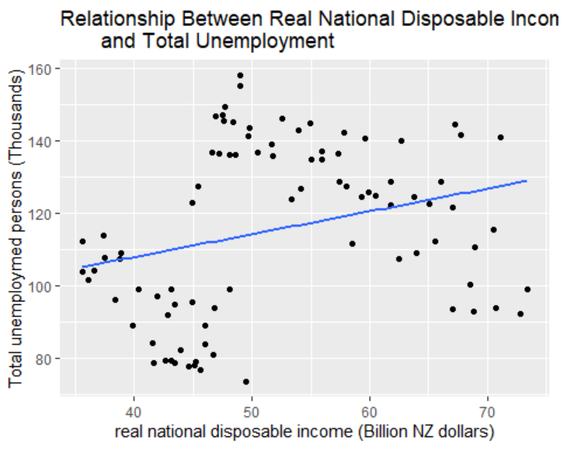
**(b)** **Plot the quarterly unemployment with the quarterly inflation and real national disposable income data. Perform the correlation analysis and discuss the results. (6 marks)**

```
# Importing dataset NZ_DispIncome_Quarterly:
NZ_DispIncome_Quarterly <-
read.csv("C:/Users/cinar/Desktop/AUT/Forecasting/Assignment_01/NZ_DispIncome_Quarterly.csv"
)
NZ_DispIncome_Quarterly <- NZ_DispIncome_Quarterly|>
  select(-X,-X.1)

# View the first few rows of the dataset to verify the import
head(NZ_DispIncome_Quarterly)
##   Year_quarter D_Income
## 1       2001Q1     35.6
## 2       2001Q2     35.6
## 3       2001Q3     36.1
## 4       2001Q4     38.8
## 5       2002Q1     37.4
## 6       2002Q2     36.6

# Importing dataset Inflation:
NZ_Inflation_Quarterly <-
read.csv("C:/Users/cinar/Desktop/AUT/Forecasting/Assignment_01/NZ_Inflation_Quarterly.csv")
NZ_Inflation_Quarterly <- NZ_Inflation_Quarterly |>
  select(-X,-X.1)
```

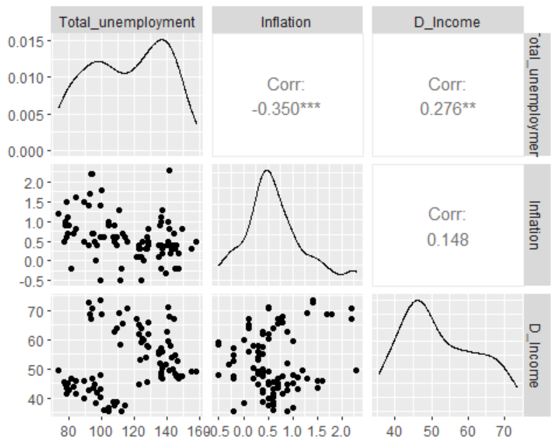
```
# View the first few rows of the dataset to verify the import
head(NZ_Inflation_Quarterly)

##   Year_quarter Inflation
## 1       2001Q1      -0.2
## 2       2001Q2       0.9
## 3       2001Q3       0.6
## 4       2001Q4       0.6
## 5       2002Q1       0.6
## 6       2002Q2       1.0

# Converting in to tsibble for inflation dataset:
df_inf <- NZ_Inflation_Quarterly |>
  mutate(Year_quarter = yearquarter(Year_quarter)) |>
  as_tsibble(index = Year_quarter)

# Converting in to tsibble for Disposable income dataset:
df_inc <- NZ_DispIncome_Quarterly |>
  mutate(Year_quarter = yearquarter(Year_quarter)) |>
  as_tsibble(index = Year_quarter)

# Merging the quarterly Unemployment, Inflation and real national disposable income data:
data <- df |>
  left_join(df_inf, by = "Year_quarter") |>
  left_join(df_inc, by = "Year_quarter") |>
  as_tsibble(index = Year_quarter)

# Reorder columns with "Year_quarter" as the first column
data <- data |>
  select(Year_quarter, everything())

head(data)

## # A tsibble: 6 x 4 [1Q]
##   Year_quarter Total_unemployment Inflation D_Income
##          <qtr>              <dbl>     <dbl>    <dbl>
## 1      2001 Q1               112.      -0.2     35.6
## 2      2001 Q2               104.       0.9     35.6
## 3      2001 Q3               102.       0.6     36.1
## 4      2001 Q4               107.       0.6     38.8
## 5      2002 Q1               114.       0.6     37.4
## 6      2002 Q2               104.       1        36.6

# Plot quarterly unemployment with inflation:
data |>
  ggplot(aes(x = Inflation, y = Total_unemployment)) +
  labs(title = "Relationship Between Inflation and
       Total Unemployment",
       y = "Total unemployed persons (Thousands)",
       x = "Inflation (% )") +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)

## `geom_smooth()` using formula 'y ~ x'
```



```
# The connection between Total Unemployment and Inflation demonstrates a slight negative linear correlation. Specifically, as inflation rises, there is a minor decrease in unemployment.

# Plot quarterly unemployment with real national disposable income data:
data |>
  ggplot(aes(x = D_Income, y = Total_unemployment)) +
  labs(title = "Relationship Between Real National Disposable Income
       and Total Unemployment",
       y = "Total unemploymed persons (Thousands)",
       x = "real national disposable income (Billion NZ dollars)") +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)

## `geom_smooth()` using formula 'y ~ x'
```



```
# There exists a subtle positive linear correlation between Total Unemployment and real national disposable income. In other words, when disposable income increases, there is a slight uptick in unemployment. This relationship stands in contrast to the inverse correlation typically observed with inflation.

# Correlation analysis of relationships between Total Unemployment (forecast variable) and each of the predictors (Inflation and Disposable Income):

data |>
   GGally::ggpairs(columns = 2:4)

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```



```
# The visual analysis of the plots reveals several key observations:

# There is a modest negative correlation (-0.35) between Total Unemployment and Inflation,
suggesting that as inflation rises, unemployment tends to decrease slightly.

# Conversely, there is a slight positive correlation (0.276) between Total Unemployment and
Disposable Income, implying that an increase in disposable income is associated with a
minor uptick in unemployment. Notably, there appears to be no discernible correlation
between Inflation and Disposable Income.

# In the context of forecasting, it is essential to consider certain assumptions regarding
the relationships between the forecast variable and predictor variables:

# The predictor variables should exhibit a mean of zero and must be uncorrelated with each
other. This ensure that multicollinearity issues are minimized and avoiding biased.

# The forecast variable should be independent of the predictor variables, meaning that it
should not be influenced or driven by them. This assumption ensures that the forecast
remains reliable and unbiased.
```

**(c) Fit a regression model to the quarterly unemployment data with the quarterly inflation and real national disposable income data as the explanatory variables. Discuss the results. (6 marks)**

```
# Fit a multiple linear regression model (mrm) including inflation and disposable income:
fit_mrm <- data |>
  model(lm = TSLM(Total_unemployment ~ Inflation + D_Income)) |>
  report()

## Series: Total_unemployment
## Model: TSLM
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -43.4082 -15.7584  -0.7312  14.7152  54.8611
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  85.6146    11.5109   7.438 7.56e-11 ***
## Inflation   -16.1442     3.8765  -4.165 7.45e-05 ***
## D_Income      0.7637     0.2185   3.495 0.000756 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.79 on 85 degrees of freedom
## Multiple R-squared: 0.2329,   Adjusted R-squared: 0.2149
## F-statistic: 12.91 on 2 and 85 DF, p-value: 1.2754e-05

# The fitted regression model incorporates two predictor variables: Inflation and D_Income
(Disposable Income). The model intercept is estimated at 85.6146, representing the expected
```

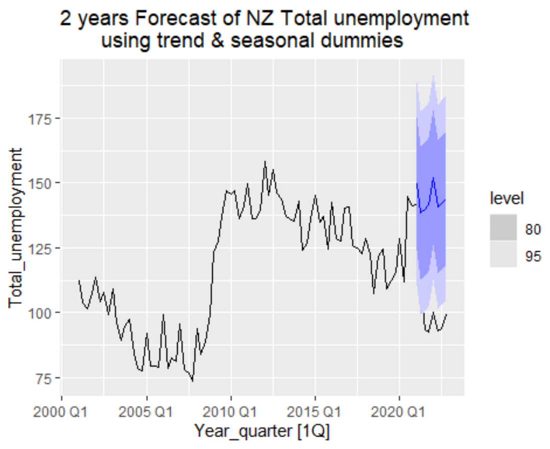
```
value of Total unemployment when both predictor variables are zero. The coefficient for
Inflation stands at -16.1442, suggesting that, on average, each one-unit increase in
inflation corresponds to anticipated decrease of approximately 16.14 units in Total
unemployment. Conversely, the coefficient for D_Income is 0.7637, indicating that, on
average, a one-unit increase in D_Income is associated with an expected rise of about 0.76
units in Total unemployment.

# The model adjusted R-squared value of 0.2149, presents that the model explains
approximately 21.49% of the variance in the data. This relatively low value suggests that
the model has low performance.
```

**(d) Do we need to include the linear trend and seasonal dummies in the regression model in (c)? Perform a relevant analysis and discuss the results. (6 marks)**

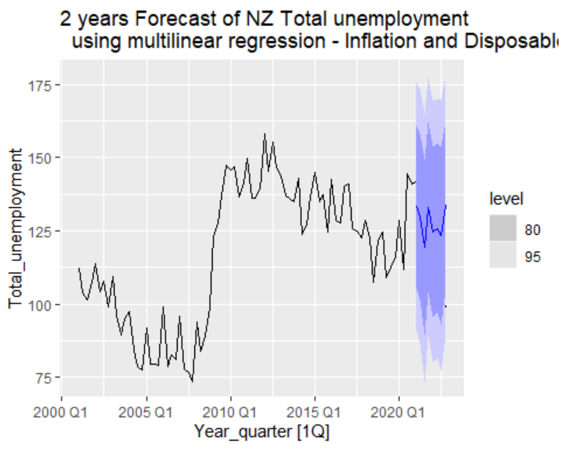
```
# Fit a regression model with a linear trend and seasonal dummies:
model_trend_season <- data |>
  model(TSLM(Total_unemployment ~ Inflation + D_Income + trend() + season())) |>
  report()

## Series: Total_unemployment
## Model: TSLM
##
## Residuals:
##      Min      1Q  Median      3Q     Max
## -59.488 -10.595   2.220   9.324  32.365
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    341.9621    32.0616  10.666  < 2e-16 ***
## Inflation       -7.3986     3.1785  -2.328 0.022424 *
## D_Income        -6.7390     0.9403  -7.167 3.20e-10 ***
## trend()          3.0290     0.3705   8.176 3.38e-12 ***
## season()year2 -16.5081     4.7130  -3.503 0.000753 ***
## season()year3 -10.9825     4.7221  -2.326 0.022532 *
## season()year4   7.8319     5.3942   1.452 0.150390
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.26 on 81 degrees of freedom
## Multiple R-squared: 0.6062,   Adjusted R-squared: 0.577
## F-statistic: 20.78 on 6 and 81 DF, p-value: 1.3624e-14

# Incorporating a linear trend and seasonal dummies into the regression model has resulted
in significant alterations in the findings. Notably, the adjusted R-squared value has seen
a substantial increase, reaching 0.577, indicating that the model now explains
approximately 57.7% of the variability in total unemployment. Thus, represents a marked
improvement compared to the previous model lacking trend and seasonality components.

# The coefficient for Inflation, measuring -7.3986, continues to emphasize a statistically
significant negative relationship with unemployment. Furthermore, the coefficient for
disposable income, which had previously indicated a positive relationship with unemployment
in earlier models, is now estimated at -6.7390, indicating a negative relationship.

# In summary, the incorporation of a linear trend and seasonal dummies has substantially
improved the model's performance, as indicated by the increased adjusted R-squared value.
This updated model adeptly captures both temporal trends and seasonal fluctuations within
the unemployment data. It reaffirms the negative correlation between inflation and
unemployment, while the influence of disposable income on unemployment has shifted to a
negative relationship.

# In conclusion, the incorporation of a linear trend and seasonal dummies appears to have
advantages in enhancing the accuracy and effectiveness of the unemployment model.
```

**(e)** **Compute two year of forecasts for the regression models in (a) and (c). Evaluate the forecast accuracy and compare with those in Question 3 parts (d)-(e). (8 marks)**

```
# two year of forecasts for the linear trend and seasonal dummies model for total
unemployment in (a):
data_train <- data |>
  filter(year(Year_quarter) <= 2020)
data_test <- data |>
  filter(year(Year_quarter) > 2020)
fit_m_a <- data_train |>
  model(lm = TSLM(Total_unemployment ~ trend() + season()))
fc_m_a <- forecast(fit_m_a, new_data = data_test)
data |>
  autoplot(Total_unemployment) +
  autolayer(fc_m_a) +
  labs(title = "2 years Forecast of NZ Total unemployment
       using trend & seasonal dummies")
```



```
accuracy(fc_m_a, data_test)

## # A tibble: 1 × 10
##   .model .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 lm     Test  -40.4  42.8  40.4 -41.5  41.5   NaN   NaN 0.420

#two year of forecasts for the regression model from question 'c', multiple linear
regression:
fit_m_c <- data_train |>
  model(lm = TSLM(Total_unemployment ~ Inflation + D_Income))
fc_m_c <- forecast(fit_m_c, new_data = data_test)
data |>
  autoplot(Total_unemployment) +
  autolayer(fc_m_c) +
  labs(title = "2 years Forecast of NZ Total unemployment
  using multilinear regression - Inflation and Disposable income.")
```



```
accuracy(fc_m_c, data_test)

## # A tibble: 1 × 10
##   .model .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 lm     Test  -24.7  28.4  26.9 -26.0  27.5   NaN   NaN 0.176

table <- tibble(
  models = c("MEAN", "Linear trend and seasonal dummies","Multiple Linear regression
model"),
  RMSE = c(20.88301, 42.79197, 28.43394),
  MAPE = c(19.88245, 41.52487, 27.50914))

table

## # A tibble: 3 × 3
##   models                              RMSE  MAPE
##   <chr>                              <dbl> <dbl>
## 1 MEAN                                20.9  19.9
## 2 Linear trend and seasonal dummies   42.8  41.5
## 3 Multiple Linear regression model    28.4  27.5

# Analyzing the results from the table, we observe the following forecast accuracy metrics
(RMSE and MAPE) for three models: MEAN, Linear trend & seasonal dummies, and Multiple
linear regression models (as displayed in the table). These metrics  provide insights into
the forecasting accuracy of each model.

# The benchmark MEAN achieves the lowest RMSE and MAPE values, indicating reasonably good
accuracy in forecasting (RMSE - 20.8830, MAPE - 19.8824). However, it is important to note
that the MEAN model oversimplifies the data. In contrast, the Linear Trend and Seasonal
Dummies Model shows the highest RMSE and MAPE values (42.7920, 41.5249), indicating less
accuracy in forecasting. The Multiple Linear Regression Model falls in between the two
other models in terms of forecast accuracy.

# That can conclude, while the MEAN model provides the best performance in terms  of RMSE
and MAPE, it lacks complexity. The other models, despite having higher errors, offer more
sophisticated ways to capture the data's underlying patterns and complexities.
```