Business Forcasting on Global Economy, Canadian Gas Utilization & Aus Retail Time Series

10/28/2023

Nikhil Eatalpacka

neata1@unh.newhaven.edu

Description: Economic indicators featured by the World Bank from 1960 to 2017.

```
remove(list=ls())
library(fpp3)
```

```
## -- Attaching packages --------------------------------------------- fpp3 0.5 --
## v tibble      3.2.1     v tsibble     1.1.3
## v dplyr       1.1.3     v tsibbledata 0.4.1
## v tidyr       1.3.0     v feasts      0.3.1
## v lubridate   1.9.2     v fable       0.3.3
## v ggplot2     3.4.3     v fabletools  0.3.3
## -- Conflicts ------------------------------------------------ fpp3_conflicts --
## x lubridate::date()    masks base::date()
## x dplyr::filter()      masks stats::filter()
## x tsibble::intersect() masks base::intersect()
## x tsibble::interval()  masks lubridate::interval()
## x dplyr::lag()         masks stats::lag()
## x tsibble::setdiff()   masks base::setdiff()
## x tsibble::union()     masks base::union()
```

```
library(tsibble)
library(knitr)
```

We will be using the GDP information in the global_economy dataset.

```
head(global_economy)
```

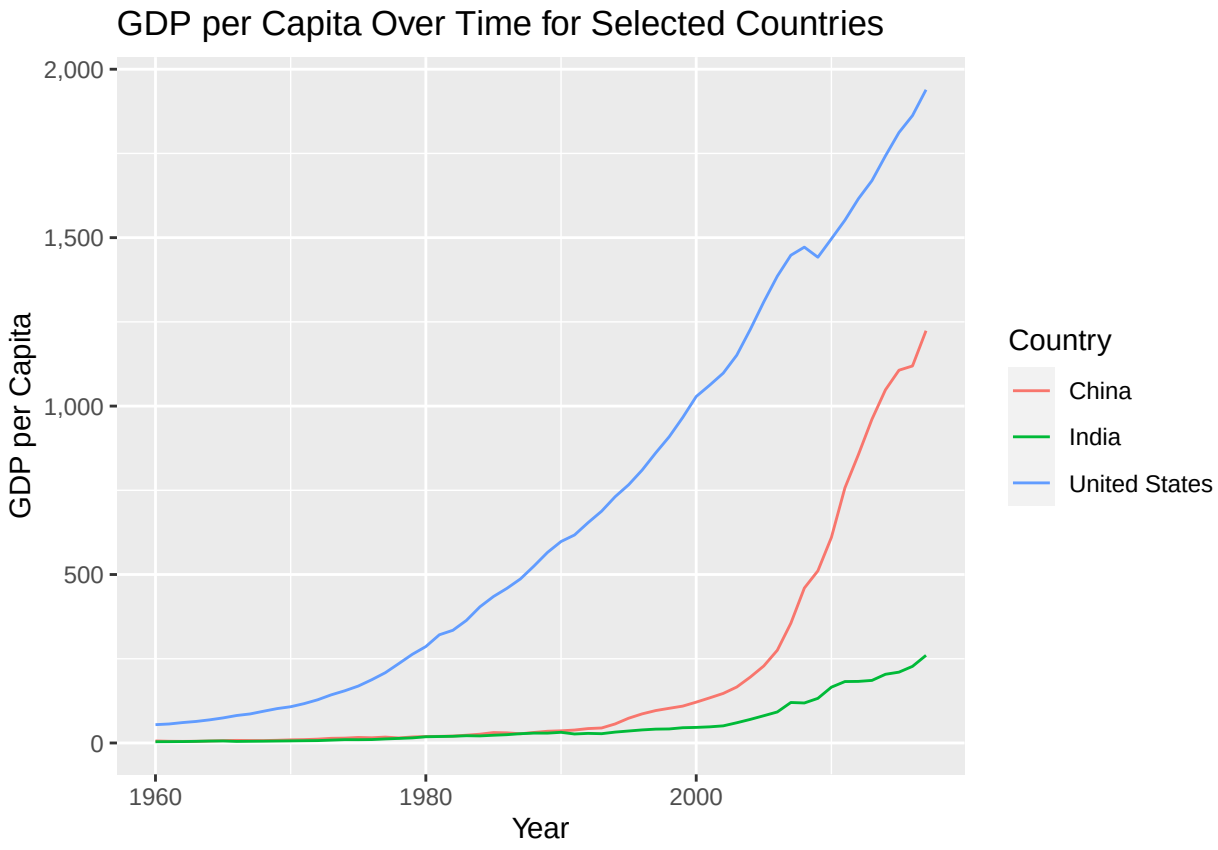
```
## # A tsibble: 6 x 9 [1Y]
## # Key:       Country [1]
##   Country     Code   Year         GDP Growth   CPI Imports Exports Population
##   <fct>       <fct> <dbl>       <dbl>  <dbl> <dbl>   <dbl>   <dbl>      <dbl>
## 1 Afghanistan AFG    1960  537777811.     NA    NA    7.02    4.13    8996351
## 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
```

PART 1

Plot the GDP per capita over time for 3 countries of your choice. Which countries did you choose? AND How has GDP per capita changed over time for these 3 countries?

I choose United States, China, India and Sure, Upon analyzing the plot, we can see that GDP per capita has steadily increased in the United States over the years. In contrast, China has witnessed rapid economic growth, resulting in a significant rise in GDP per capita, while India also exhibits an upward trend, indicating economic development and an increase in GDP per capita.

```
library(ggplot2)
library(dplyr)

# I choose United States, China, India
countries_selected <- c("United States", "China", "India")

# Filtering the data for the selected countries
filtered_data <- global_economy %>%
  filter(Country %in% countries_selected)

# Creating a line plot
ggplot(filtered_data, aes(x = Year, y = GDP, color = Country)) +
  geom_line() +
  labs(title = "GDP per Capita Over Time for Selected Countries",
       x = "Year",
       y = "GDP per Capita") +
  scale_y_continuous(labels = scales::comma_format(accuracy = 1, scale = 1e-10))
```



Out of ALL the countries in the dataset, which one had the highest GDP per capita in 2017? Filter and mutate the data as needed.

The highest GDP per capita in 2017 is World with a GDP of 8.073758e+13

```
library(dplyr)

# Filter the data for the year 2017
gdp_2017 <- global_economy %>%
  filter(Year == 2017)

# ReplacING NA values in the "GDP" and "Country" columns
gdp_2017 <- gdp_2017 %>%
  mutate(
    GDP = ifelse(is.na(GDP), 0, GDP), # Replace NA in GDP with 0
    Country = ifelse(is.na(Country), "NA", Country)  # Replace NA in Country with NA
)

# Find the country with the highest GDP per capita in 2017
highest_gdp_country <- gdp_2017 %>%
  filter(GDP == max(GDP))

# Print the country with the highest GDP per capita
print(highest_gdp_country)
```

```
## # A tsibble: 1 x 9 [1Y]
## # Key:       Country [1]
##   Country Code   Year     GDP Growth   CPI Imports Exports Population
##     <int> <fct> <dbl>   <dbl>  <dbl> <dbl>   <dbl>   <dbl>      <dbl>
## 1     260 WLD    2017 8.07e13   3.14    NA    34.7    37.0 7530360149
```

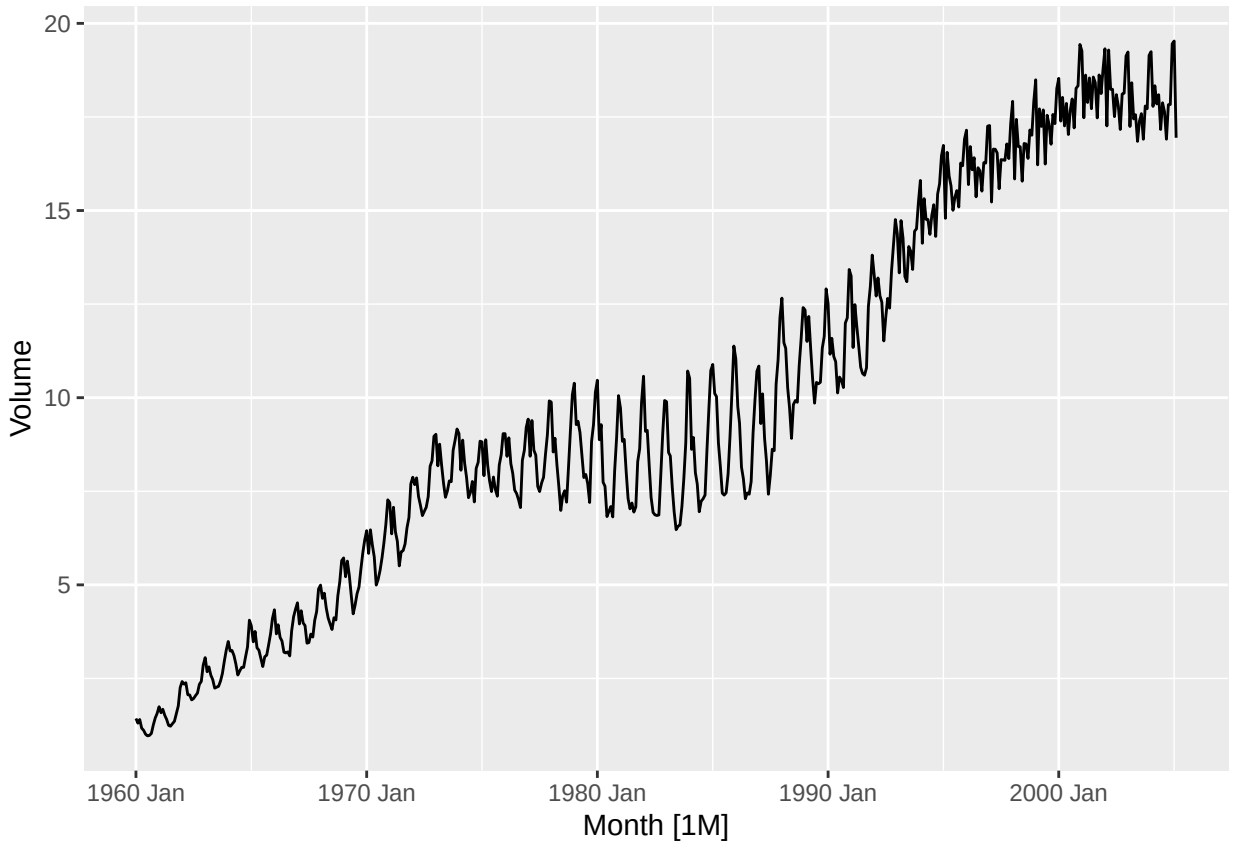
PART 2.

Use the canadian_gas data (monthly Canadian gas production in billions of cubic metres, January 1960 – February 2005).

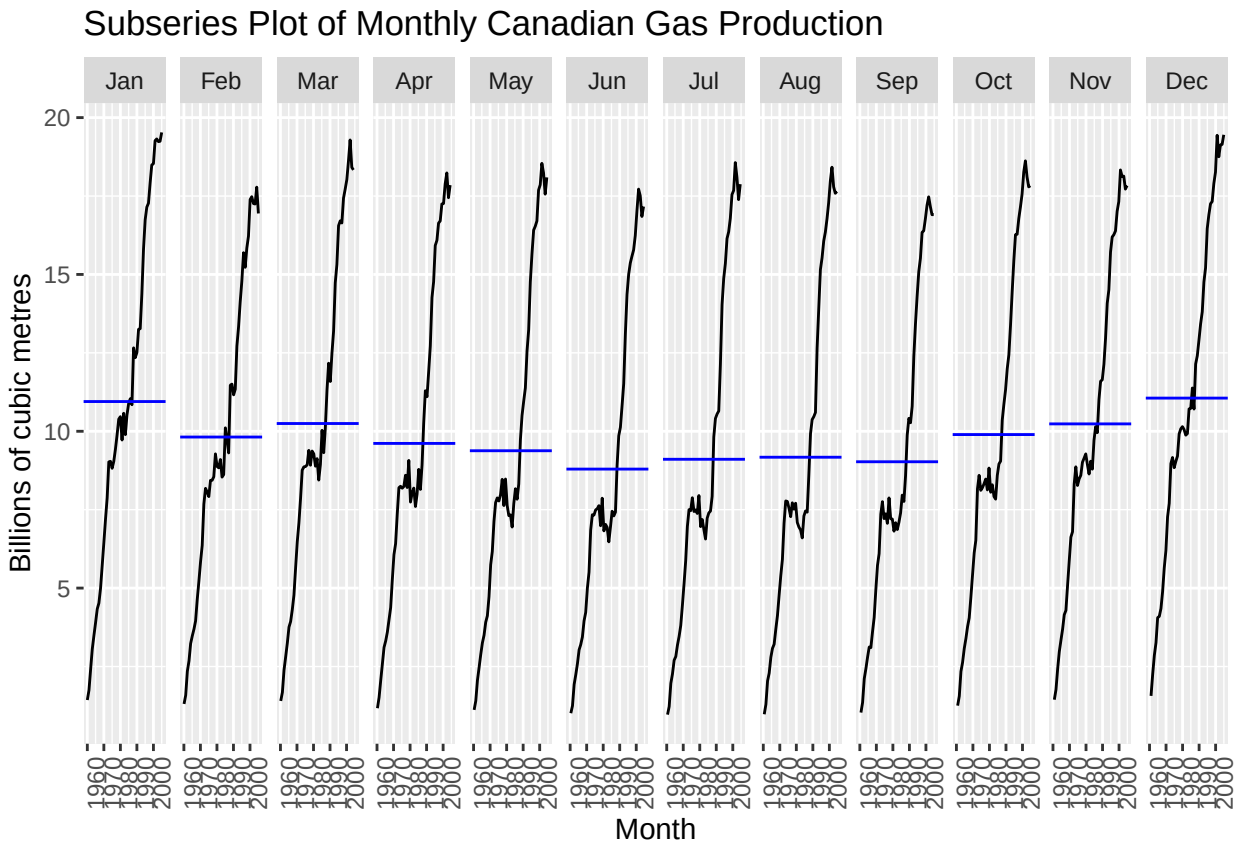
Plot Volume using autoplot, gg_subseries, gg_season to look at the effect of changing seasonality over time. What do you observe with the seasonality?

The analysis reveals that the seasonality of monthly Canadian gas production has decreased over time. This reduction can be attributed to various factors, including technological advancements that improve gas production efficiency in the summer and a higher demand for gas during the summer months, mainly for air conditioning purposes. However, it is important to note that the decrease in seasonality is not consistent across all years, with some periods experiencing a more significant decrease than others, notably in the late 1970s and early 1980s. The seasonal plot confirms this trend by illustrating a diminishing amplitude of the seasonal component, indicating a less pronounced seasonal pattern over time. Overall, the findings suggest a notable decline in the seasonality of monthly Canadian gas production, likely influenced by a combination of technological advancements and changing demand patterns.

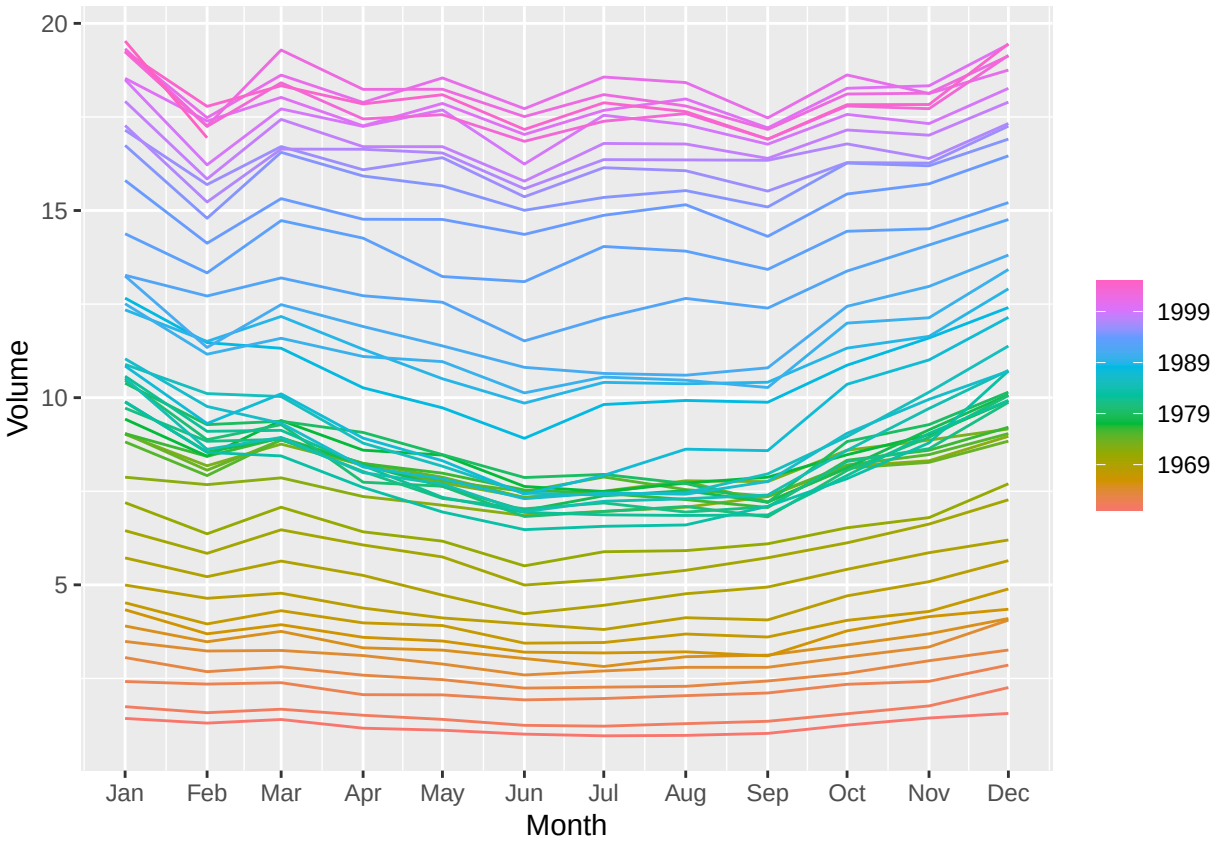
```
library(fpp3)

canadian_gas %>% autoplot(Volume)
```




```
# Subseries plot
canadian_gas %>% gg_subseries(Volume) +
  ggtitle("Subseries Plot of Monthly Canadian Gas Production") +
  xlab("Month") +
  ylab("Billions of cubic metres")
```




```
# Seasonal plot
canadian_gas %>% gg_season(Volume)
```



Do an STL decomposition of the data. You will need to choose a seasonal window to allow for the changing shape of the seasonal component

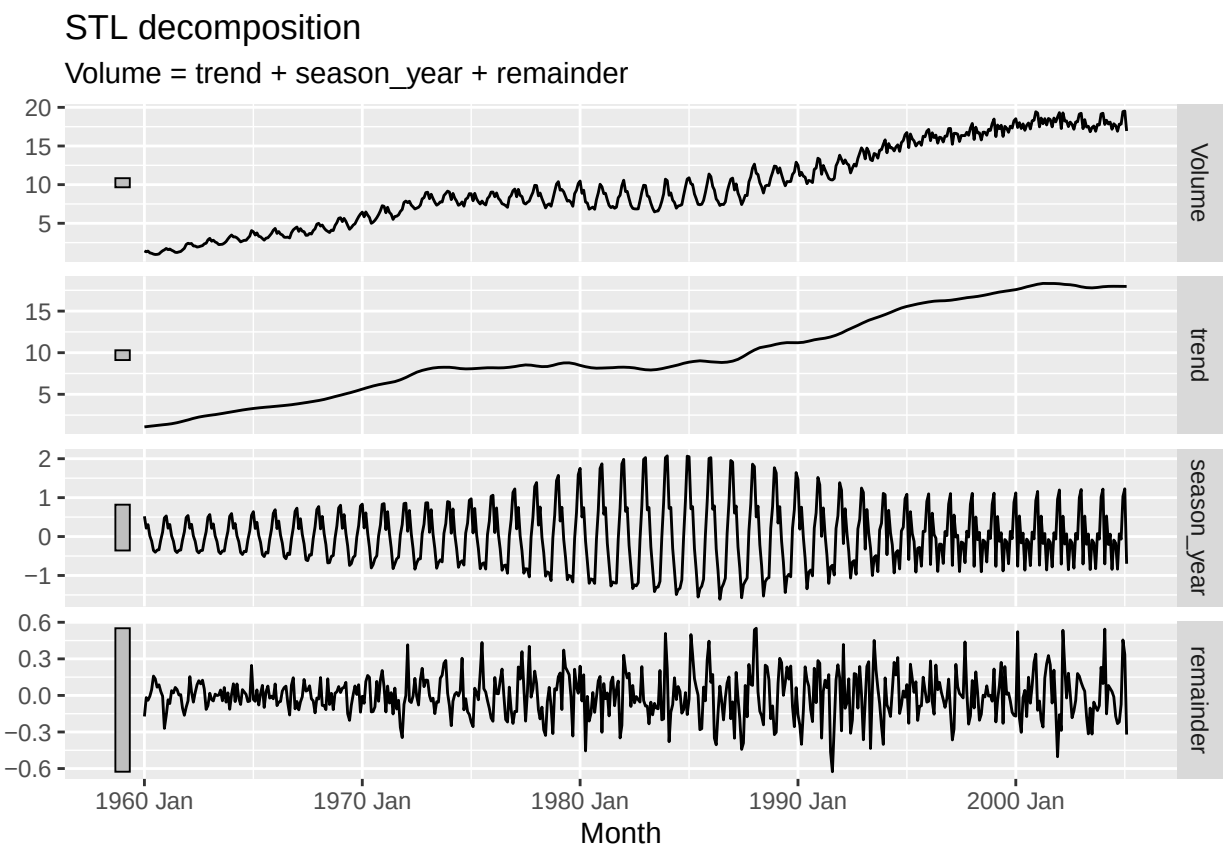
```
# Fit a Seasonal and Trend decomposition using STL model to the "Volume" series in the "canadian_gas" d
fit <- canadian_gas %>%
 model(STL(Volume)) %>%
 components()

# Plotting the decomposed series
fit
```

```
## # A dable: 542 x 7 [1M]
## # Key:     .model [1]
## # :        Volume = trend + season_year + remainder
##    .model          Month Volume trend season_year remainder season_adjust
##    <chr>           <mth>  <dbl> <dbl>       <dbl>     <dbl>          <dbl>
##  1 STL(Volume) 1960 Jan  1.43   1.08      0.520   -0.172           0.911
##  2 STL(Volume) 1960 Feb  1.31   1.11      0.215   -0.0178          1.09
##  3 STL(Volume) 1960 Mar  1.40   1.13      0.307   -0.0395          1.09
##  4 STL(Volume) 1960 Apr  1.17   1.16      0.0161  -0.00627         1.15
##  5 STL(Volume) 1960 May  1.12   1.18     -0.116    0.0476          1.23
##  6 STL(Volume) 1960 Jun  1.01   1.21     -0.356    0.159           1.37
##  7 STL(Volume) 1960 Jul  0.966  1.23     -0.403    0.136           1.37
##  8 STL(Volume) 1960 Aug  0.977  1.26     -0.349    0.0677          1.33
##  9 STL(Volume) 1960 Sep  1.03   1.28     -0.340    0.0870          1.37
## 10 STL(Volume) 1960 Oct  1.25   1.31     -0.0899   0.0329          1.34
## # i 532 more rows
```

```
# Retrieve the names of the components in the decomposition
names(fit)
```

```
## [1] ".model"        "Month"         "Volume"        "trend"
## [5] "season_year"   "remainder"     "season_adjust"
```

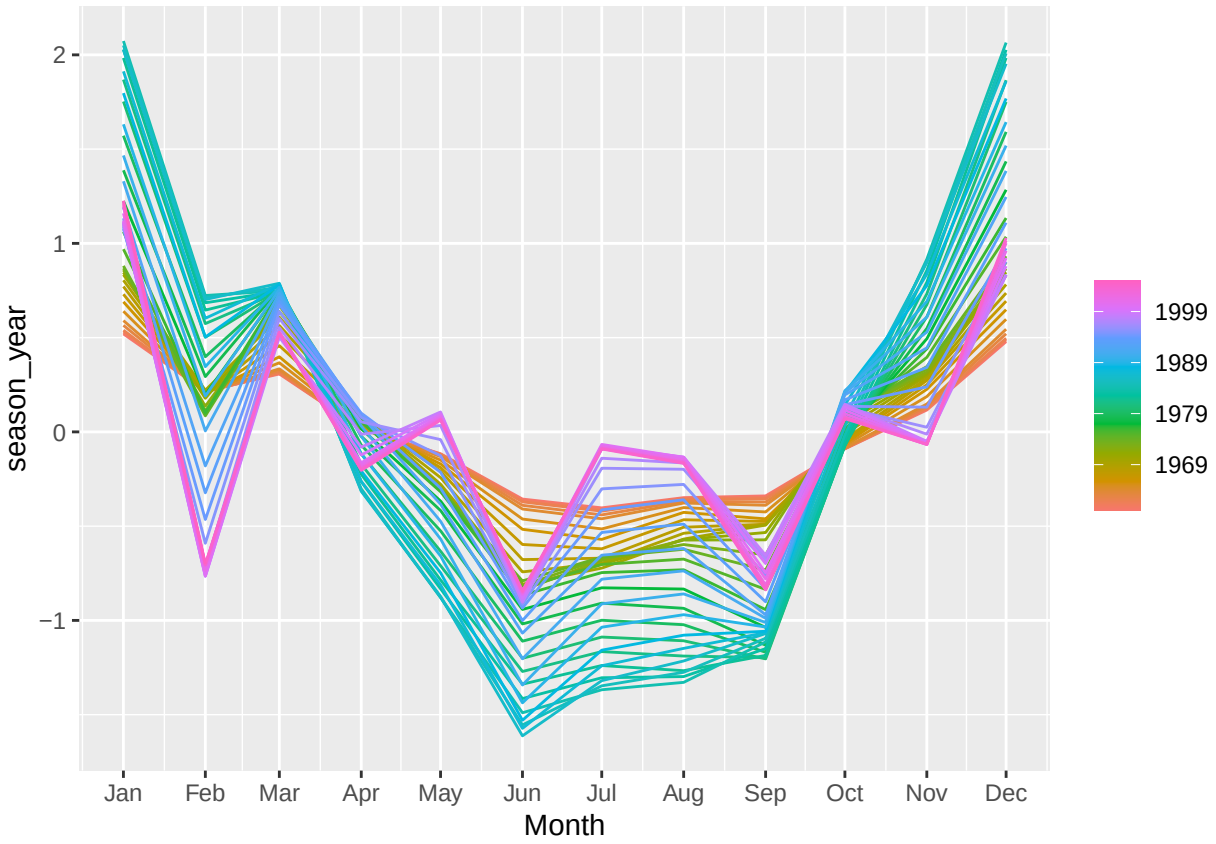
```
# Create an autoplot of the decomposition, visualizing the observed data and its components
fit %>% autoplot()
```



How does seasonal SHAPE change over time? Plot season_year using gg_season().

In my analysis, it is evident that until approximately 1990, there is a noticeable pattern of strong seasonality in Canadian gas production, with the highest volumes occurring during the winter months. Furthermore, the seasonality becomes more pronounced over time. However, after 1990, there is a shift in the seasonality pattern, potentially influenced by factors such as the month length towards the end of the series.

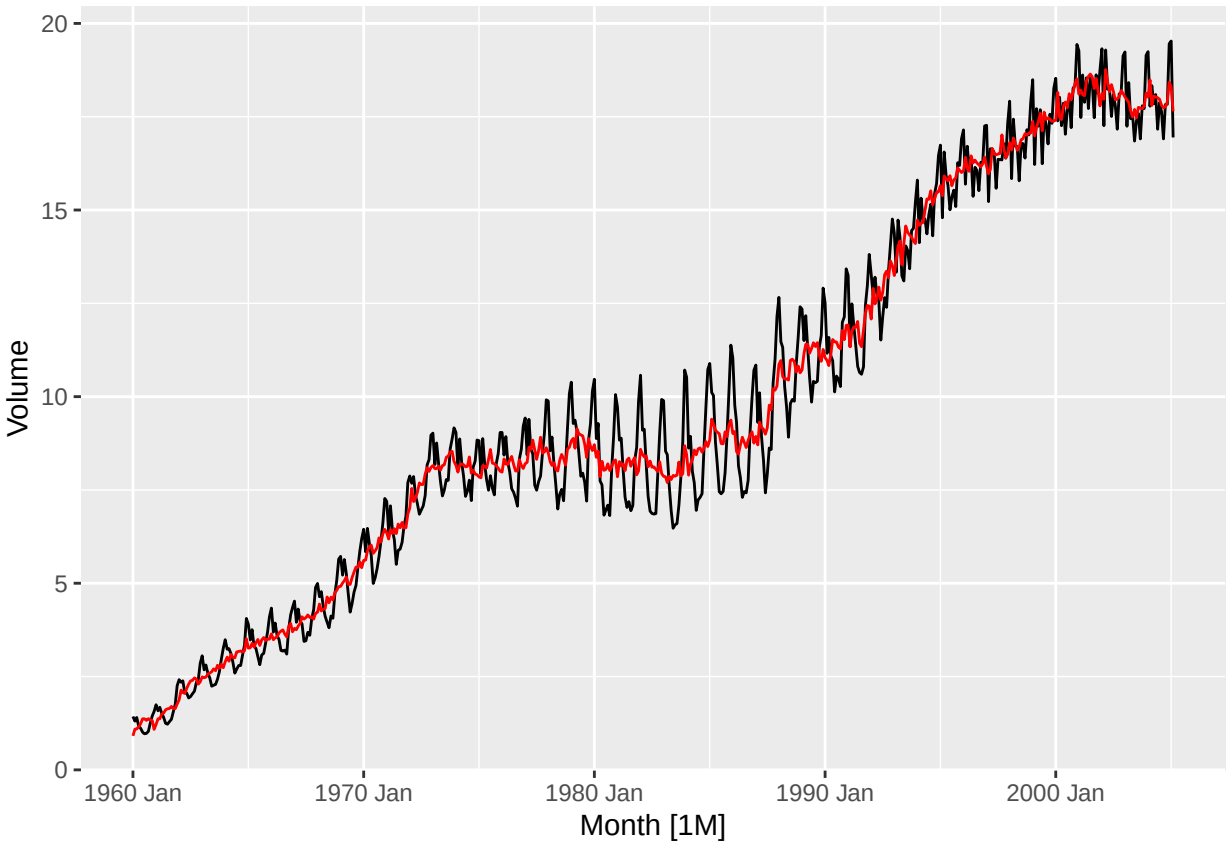
```
fit %>% gg_season(season_year)
```



Can you produce a plausible seasonally adjusted series?

Based on the graph, it is not possible to produce a plausible seasonally adjusted series. The graph shows that the seasonality of the data has changed over time, and the magnitude of the seasonal variation is relatively small. This makes it difficult to develop a reliable model of the seasonal component of the data.

```
#
seasonal_adj <- fit$season_adjust

# Plotting on the original series: Retrieve and save the seasonally adj series
canadian_gas %>%
autoplot(Volume) +
autolayer(fit, season_adjust, col = "red")
```



PART 3

Aus Retail Time Series We will use aus_rail dataset Using the code below, get a series (it gets a series randomly by using sample() function):

```
set.seed(1234567)

myseries <- aus_retail %>%
  filter(`Series ID` == sample(aus_retail$`Series ID`,1))

head(myseries)
```

```
## # A tsibble: 6 x 5 [1M]
## # Key:       State, Industry [1]
##   State    Industry                                `Series ID`    Month Turnover
##   <chr>    <chr>                                   <chr>          <mth>    <dbl>
## 1 Victoria Cafes, restaurants and takeaway food s~ A3349417W   1982 Apr     85.1
## 2 Victoria Cafes, restaurants and takeaway food s~ A3349417W   1982 May     85.1
## 3 Victoria Cafes, restaurants and takeaway food s~ A3349417W   1982 Jun     82.8
## 4 Victoria Cafes, restaurants and takeaway food s~ A3349417W   1982 Jul     82.1
## 5 Victoria Cafes, restaurants and takeaway food s~ A3349417W   1982 Aug     81.8
## 6 Victoria Cafes, restaurants and takeaway food s~ A3349417W   1982 Sep     84.6
```

```
# remover NA's in the series with below:
myseries =  myseries %>% filter(!is.na(`Series ID`))
nrow(myseries)
```

```
## [1] 441
```

```r
# rename the column name  `Series ID` with MyRandomSeries
rename(myseries, MyRandomSeries = `Series ID`)
```

```
## # A tsibble: 441 x 5 [1M]
## # Key:       State, Industry [1]
##    State    Industry                              MyRandomSeries    Month Turnover
##    <chr>    <chr>                                 <chr>             <mth>    <dbl>
##  1 Victoria Cafes, restaurants and takeaway fo~ A3349417W      1982 Apr     85.1
##  2 Victoria Cafes, restaurants and takeaway fo~ A3349417W      1982 May     85.1
##  3 Victoria Cafes, restaurants and takeaway fo~ A3349417W      1982 Jun     82.8
##  4 Victoria Cafes, restaurants and takeaway fo~ A3349417W      1982 Jul     82.1
##  5 Victoria Cafes, restaurants and takeaway fo~ A3349417W      1982 Aug     81.8
##  6 Victoria Cafes, restaurants and takeaway fo~ A3349417W      1982 Sep     84.6
##  7 Victoria Cafes, restaurants and takeaway fo~ A3349417W      1982 Oct     91.7
##  8 Victoria Cafes, restaurants and takeaway fo~ A3349417W      1982 Nov     97.7
##  9 Victoria Cafes, restaurants and takeaway fo~ A3349417W      1982 Dec    109.
## 10 Victoria Cafes, restaurants and takeaway fo~ A3349417W      1983 Jan     94.6
## # i 431 more rows
```

Run a linear regression of Turnover on its trend, by using TSLM() and trend() functions)

```r
# save the regression model results in "fit".
fit = myseries %>% model(TSLM(Turnover~ trend()))
report(fit)
```

```
## Series: Turnover
## Model: TSLM
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -125.471  -50.951   -9.889   48.598  242.364
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -23.529      6.121  -3.844 0.000139 ***
## trend()        1.921      0.024  80.057  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 64.17 on 439 degrees of freedom
## Multiple R-squared: 0.9359,  Adjusted R-squared: 0.9357
## F-statistic:  6409 on 1 and 439 DF, p-value: < 2.22e-16
```

Forecast for next 3 years. What are the values for the next 3 years are they monthly values?

Yes, they are in monthly intervals of 36!

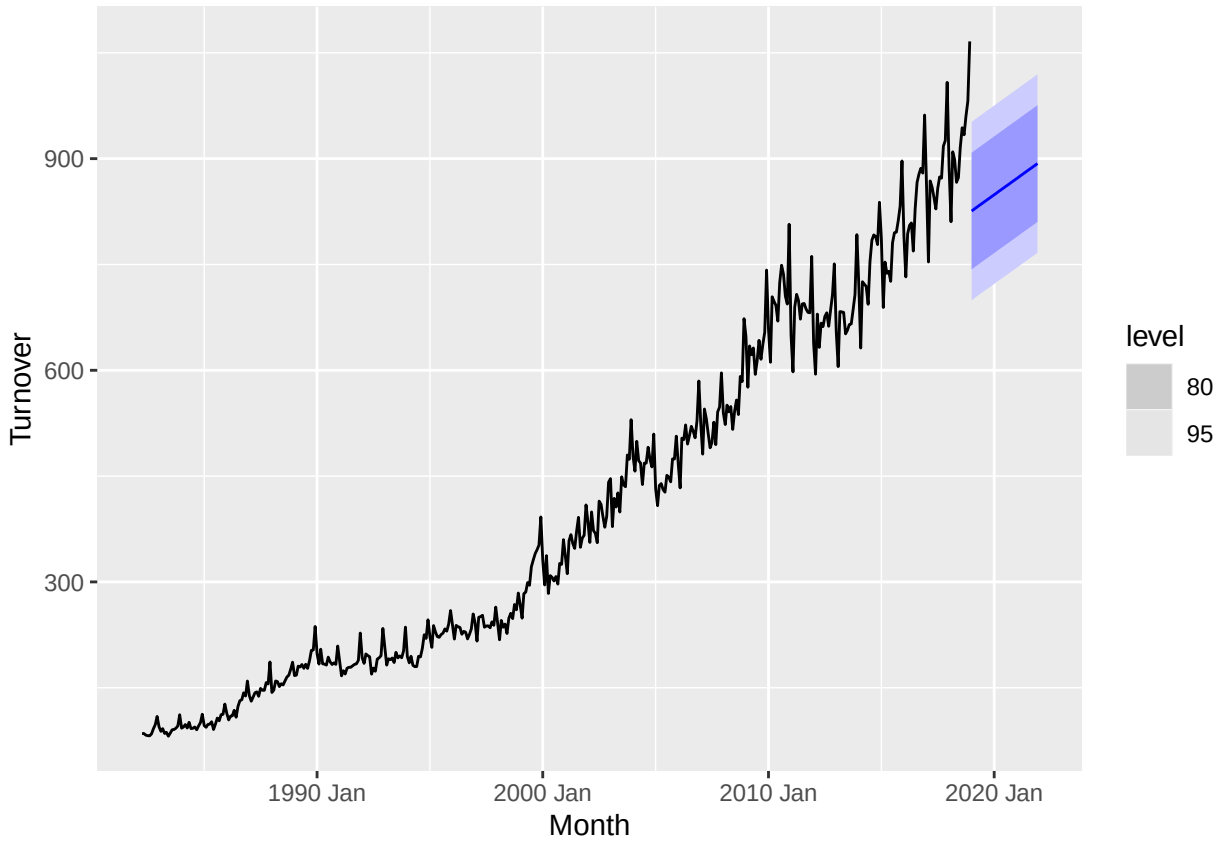
```r
# since, data is monthly, we should take 12 months * 3 years = 36 months to forecast
fit %>% forecast(h=36)
```

```
## # A fable: 36 x 6 [1M]
## # Key:     State, Industry, .model [1]
##    State    Industry                              .model    Month     Turnover .mean
##    <chr>    <chr>                                 <chr>     <mth>       <dist> <dbl>
##  1 Victoria Cafes, restaurants and takeaway ~ TSLM(~ 2019 Jan N(826, 4155)  826.
##  2 Victoria Cafes, restaurants and takeaway ~ TSLM(~ 2019 Feb N(828, 4155)  828.
##  3 Victoria Cafes, restaurants and takeaway ~ TSLM(~ 2019 Mar N(830, 4155)  830.
```

```
##  4 Victoria Cafes, restaurants and takeaway ~ TSLM(~ 2019 Apr N(832, 4155)  832.
##  5 Victoria Cafes, restaurants and takeaway ~ TSLM(~ 2019 May N(833, 4156)  833.
##  6 Victoria Cafes, restaurants and takeaway ~ TSLM(~ 2019 Jun N(835, 4156)  835.
##  7 Victoria Cafes, restaurants and takeaway ~ TSLM(~ 2019 Jul N(837, 4156)  837.
##  8 Victoria Cafes, restaurants and takeaway ~ TSLM(~ 2019 Aug N(839, 4156)  839.
##  9 Victoria Cafes, restaurants and takeaway ~ TSLM(~ 2019 Sep N(841, 4157)  841.
## 10 Victoria Cafes, restaurants and takeaway ~ TSLM(~ 2019 Oct N(843, 4157)  843.
## # i 26 more rows
```

Autoplot the forecast with original data

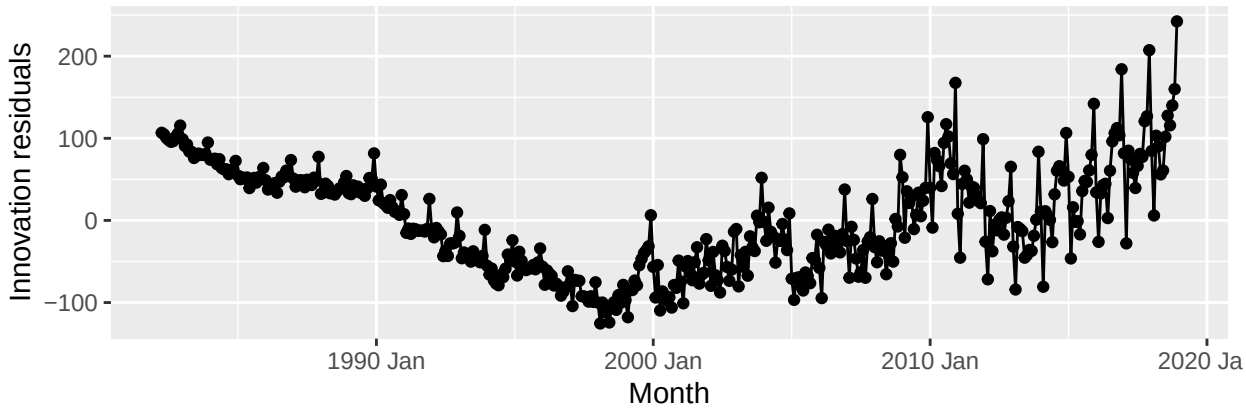
```
fit %>% forecast(h=36 )%>% autoplot(myseries)
```

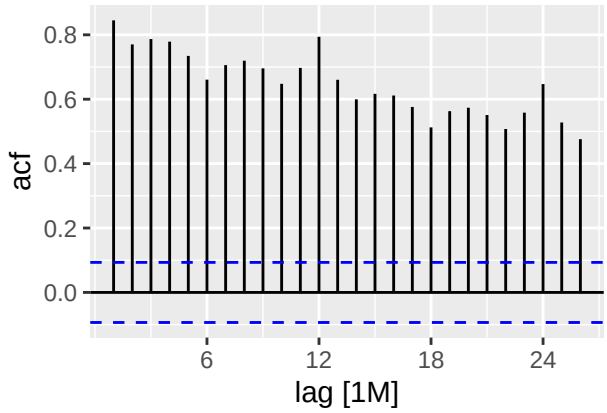


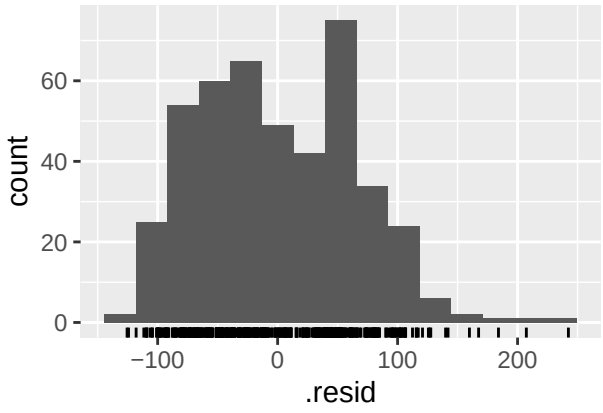
Get the residuals, does it satisfy requirements for white noise error terms? Hint: gg_tsresiduals()

The residuals of the original model do not satisfy the requirements for white noise error terms. To improve the model, it would be necessary to include a seasonal component in the model and to transform the model to make the residuals more normally distributed.

```
fit %>% gg_tsresiduals()
```


Innovation residuals
200
100
0
−100
1990 Jan
2000 Jan
2010 Jan
2020 Ja
Month


acf
0.8
0.6
0.4
0.2
0.0
6
12
18
24
lag [1M]


count
60
40
20
0
−100
0
100
200
.resid