STA4003_HW1

Yuzhou Peng

2023-09-26

```
library(fpp3)
```

```
## -- Attaching packages ---------------------------------------------- fpp3 0.5 --

## v tibble      3.2.1     v tsibble     1.1.3
## v dplyr       1.1.1     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()
```

##2.10.4 The **USgas** package contains data on the demand for natural gas in the US.

a.Install the **USgas** package.

```
library(USgas)
```

b.Create a tsibble from **us_total**with year as the index and state as the key.

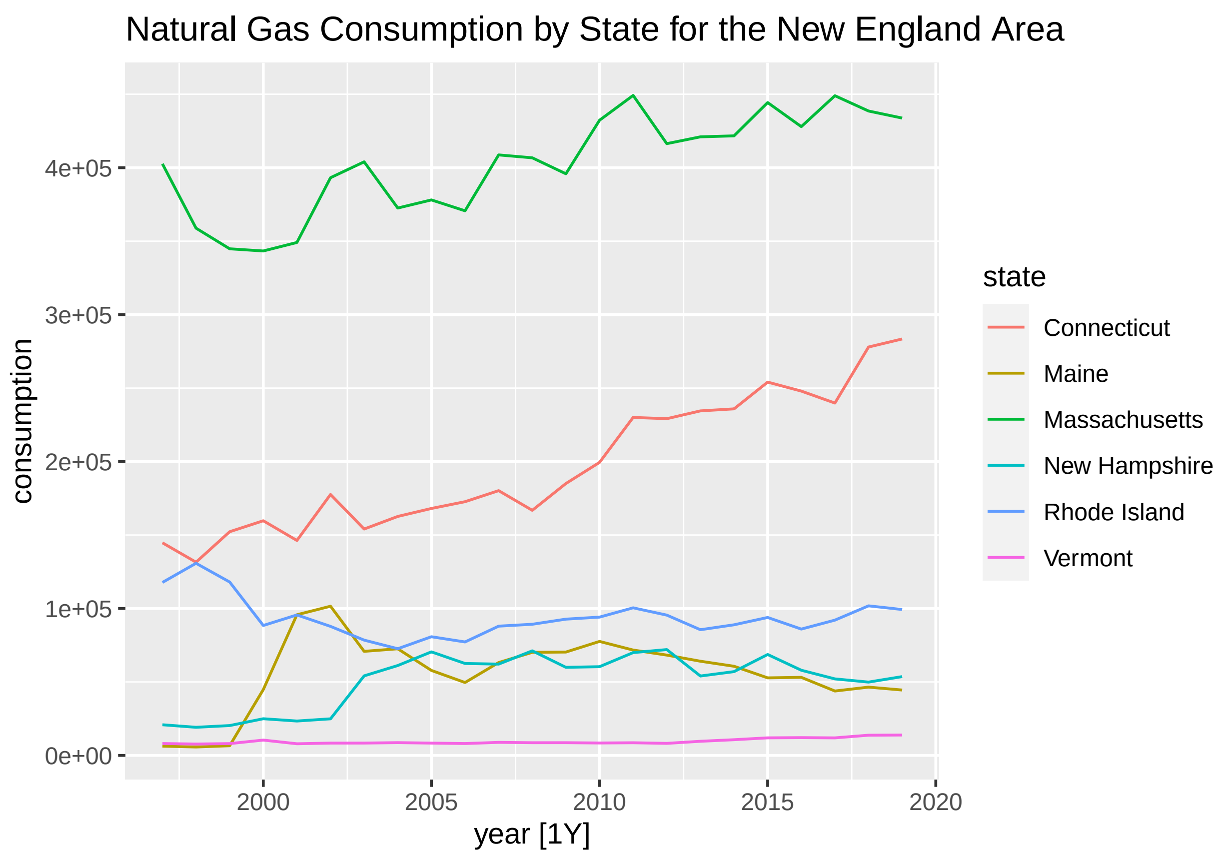
```
data(us_total)
us_total_tsibble <- us_total %>%
  as_tsibble(index = year,
             key = state)
```

c.Plot the annual natural gas consumption by state for the New England area (comprising the states of Maine, Vermont, New Hampshire, Massachusetts, Connecticut and Rhode Island).

```
us_total_by_state <- us_total_tsibble %>%
  group_by(state) %>%
  summarise(consumption = sum(y))

us_total_by_state %>%
```

```
  filter(state %in% c("Maine", "Vermont", "New Hampshire", "Massachusetts", "Connecticut", "Rhode Islan
  autoplot() +
  labs(title = "Natural Gas Consumption by State for the New England Area")
```

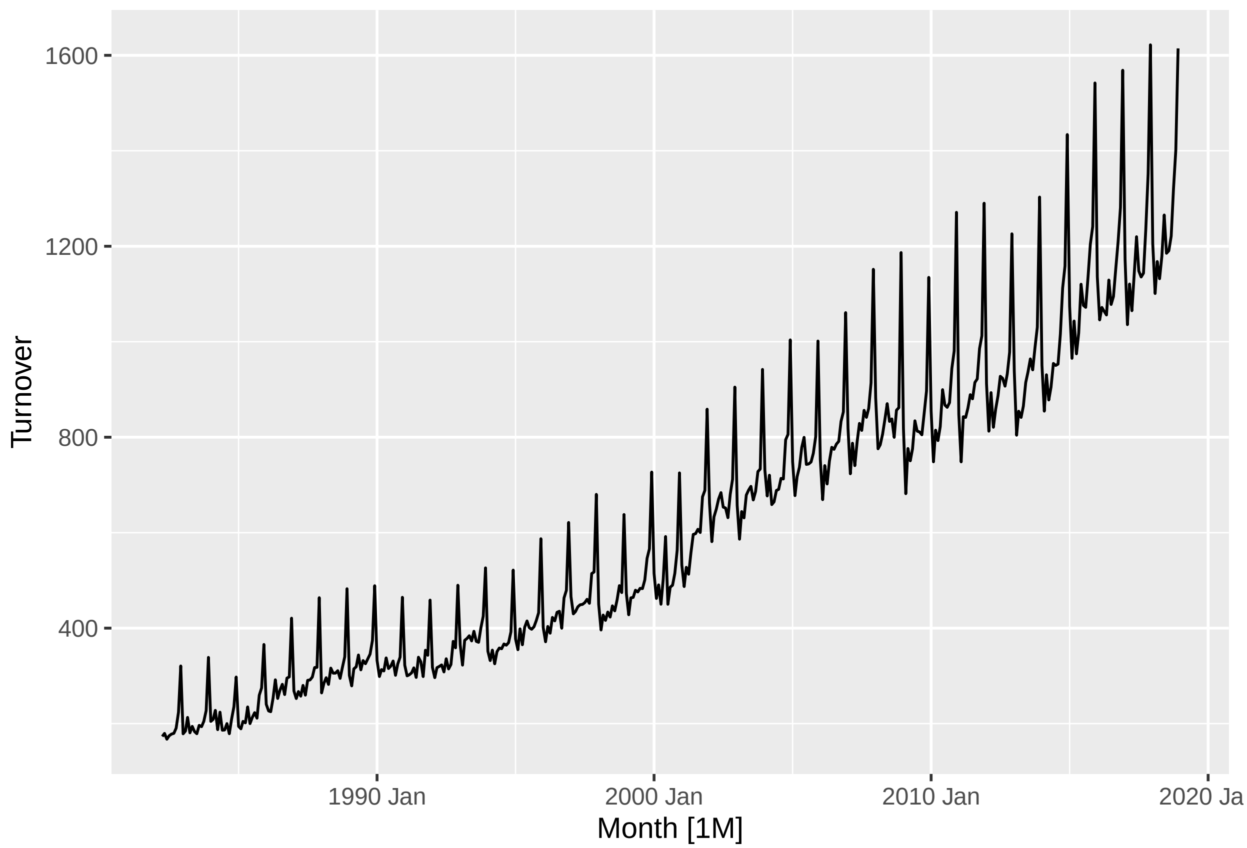
```
## Plot variable not specified, automatically selected `.vars = consumption`
```



##2.10.7 Monthly Australian retail data is provided in **aus_retail**. Select one of the time series as follows (but choose your own seed value):

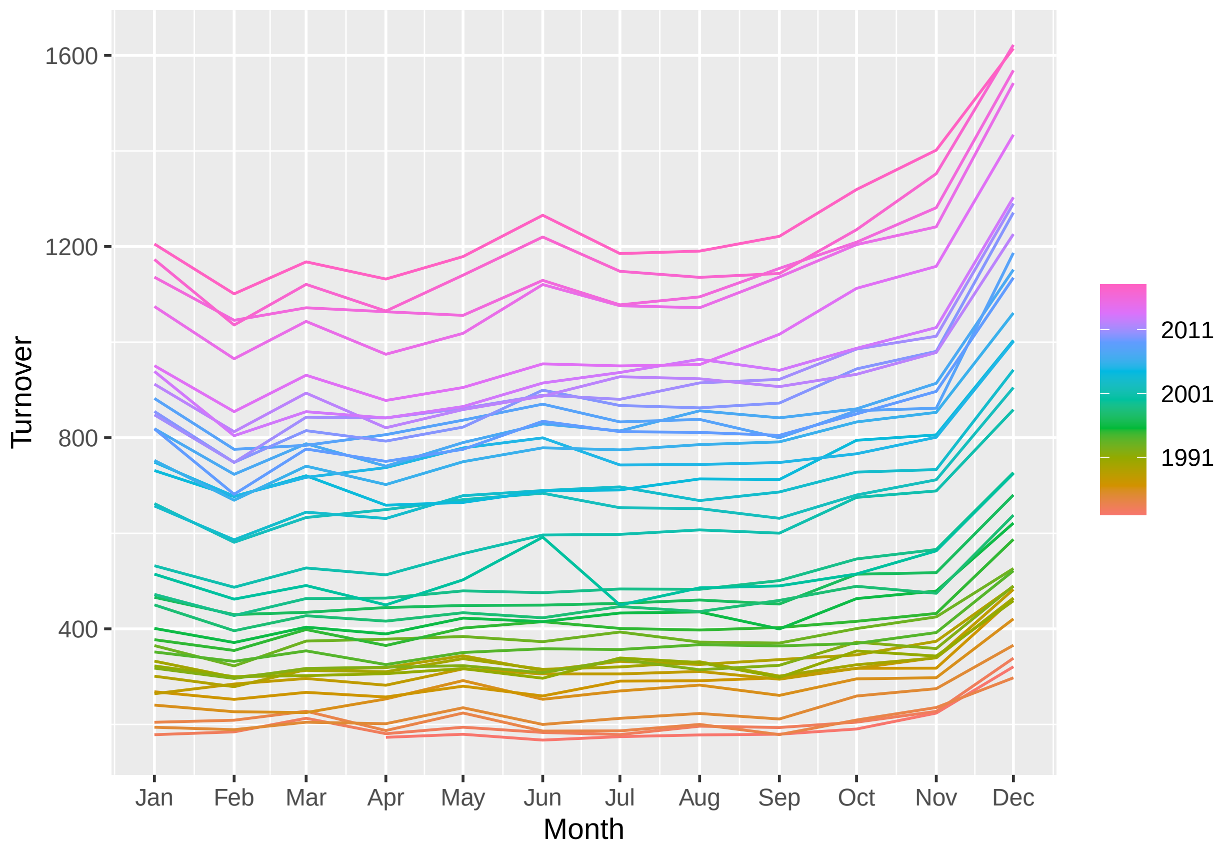
```
data("aus_retail")
set.seed(123)
myseries <- aus_retail |>
  filter(`Series ID` == sample(aus_retail$`Series ID`,1))

myseries %>% autoplot()
```

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



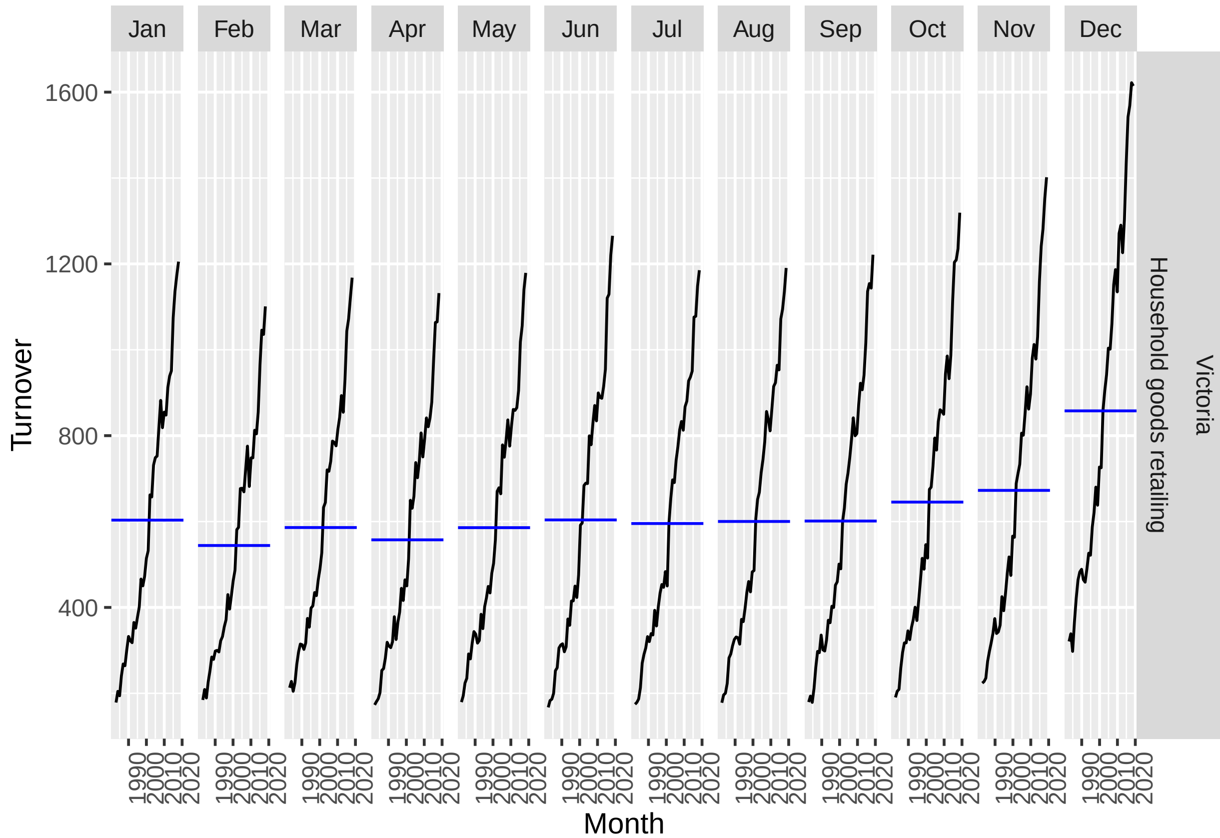

```
myseries %>% gg_season()
```

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



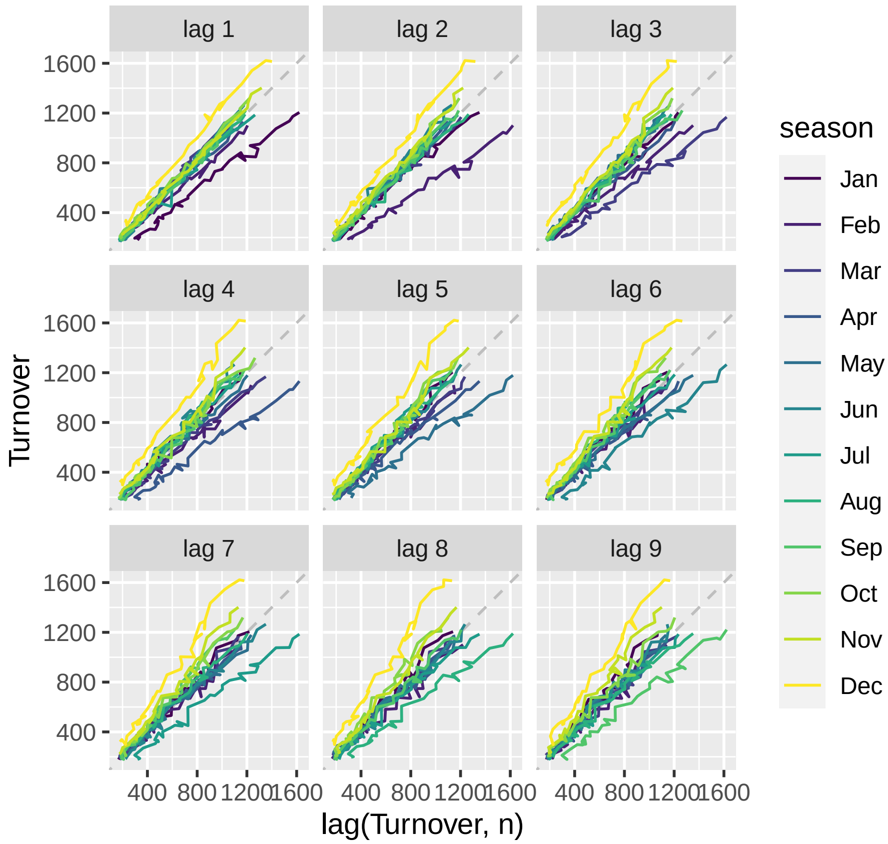

```
myseries %>% gg_subseries()
```

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



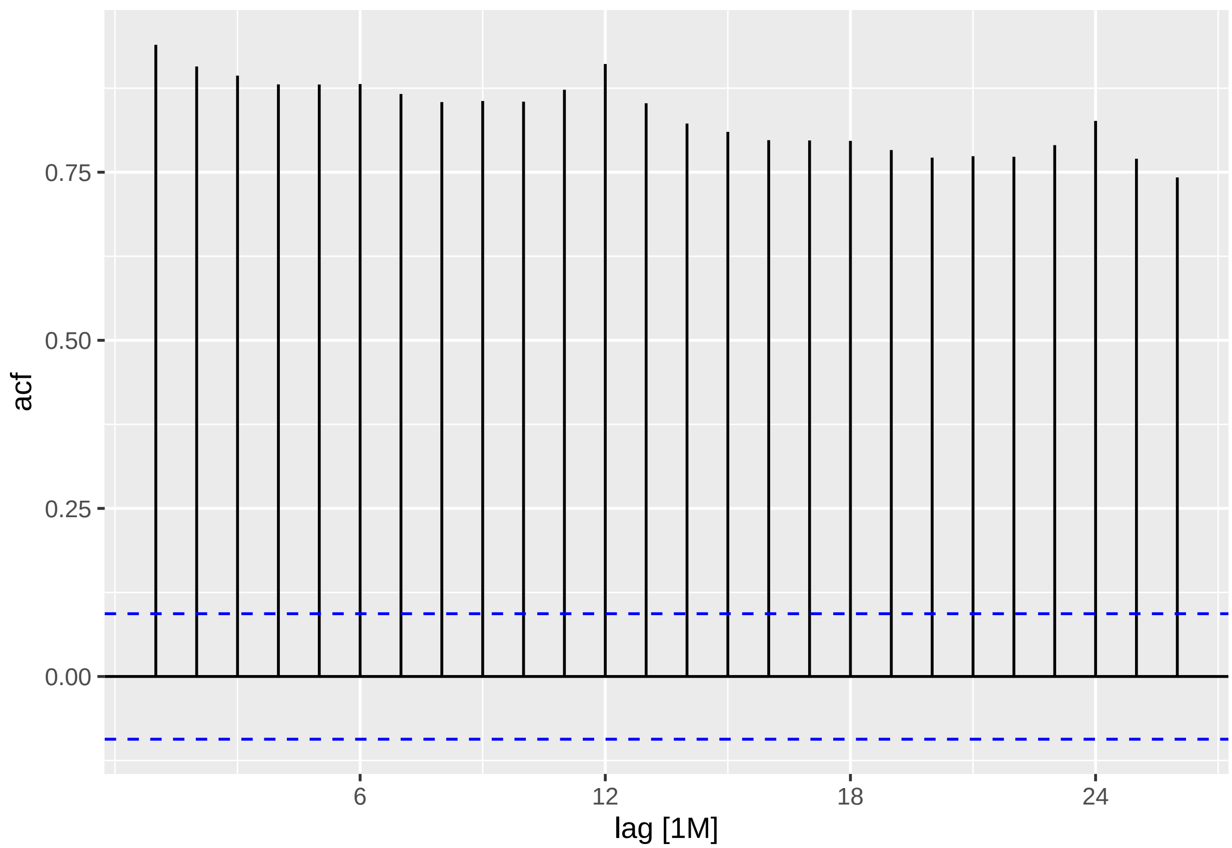

```
myseries %>% gg_lag()
```

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




```
myseries %>%
  ACF(,Turnover) %>%
  autoplot()
```

```
## Warning: The `...` argument of `PACF()` is deprecated as of feasts 0.2.2.
## i ACF variables should be passed to the `y` argument. If multiple variables are
##   to be used, specify them using `vars(...)`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



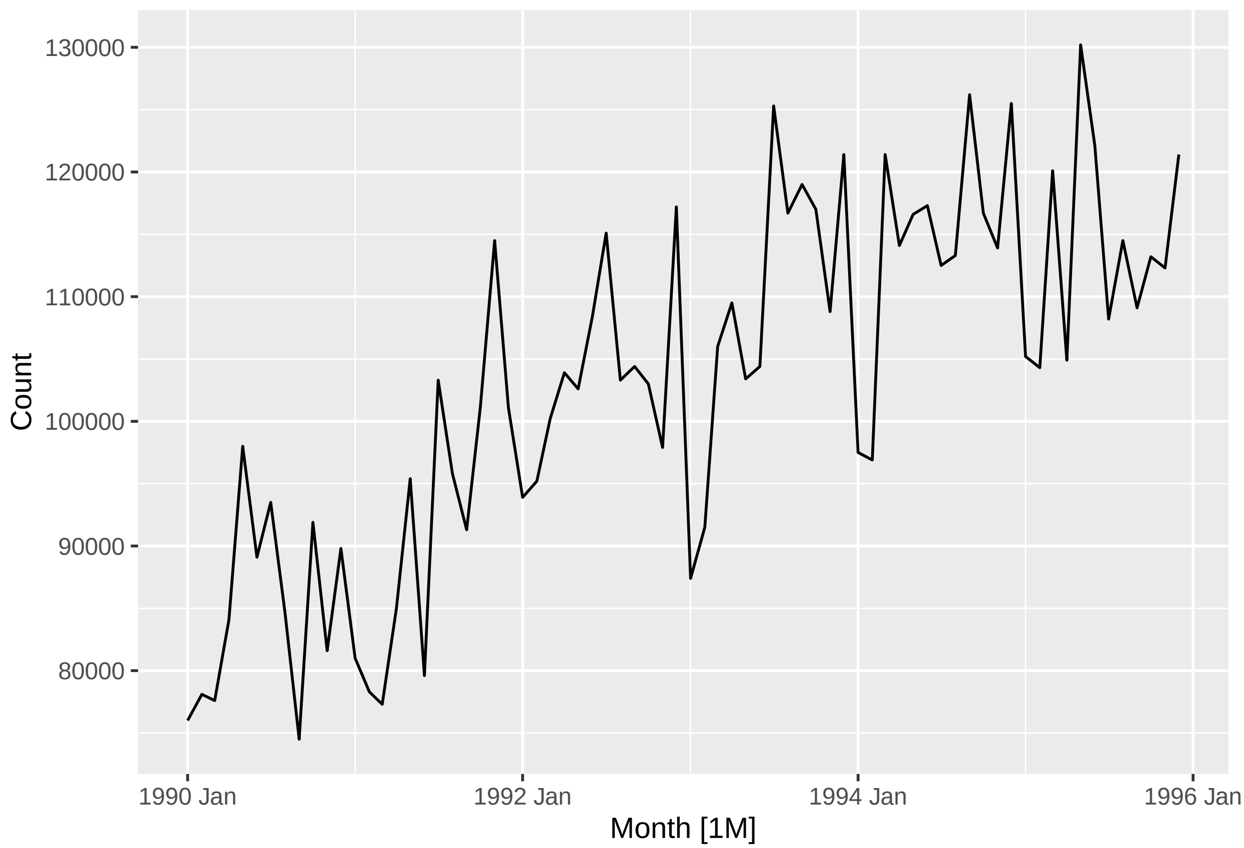
Observe lag, subseries and ACF plots, there is significant trend and cyclicity. Observe seasonal plot, there is seasonality in the timw series.

##2.10.9 The following time plots and ACF plots correspond to four different time series. Your task is to match each time plot in the first row with one of the ACF plots in the second row.

1. B
2. A
3. D
4. C

##2.10.10 The **aus_livestock** data contains the monthly total number of pigs slaughtered in Victoria, Australia, from Jul 1972 to Dec 2018. Use **filter()** to extract pig slaughters in Victoria between 1990 and 1995. Use **autoplot()** and **ACF()** for this data. How do they differ from white noise? If a longer period of data is used, what difference does it make to the ACF?

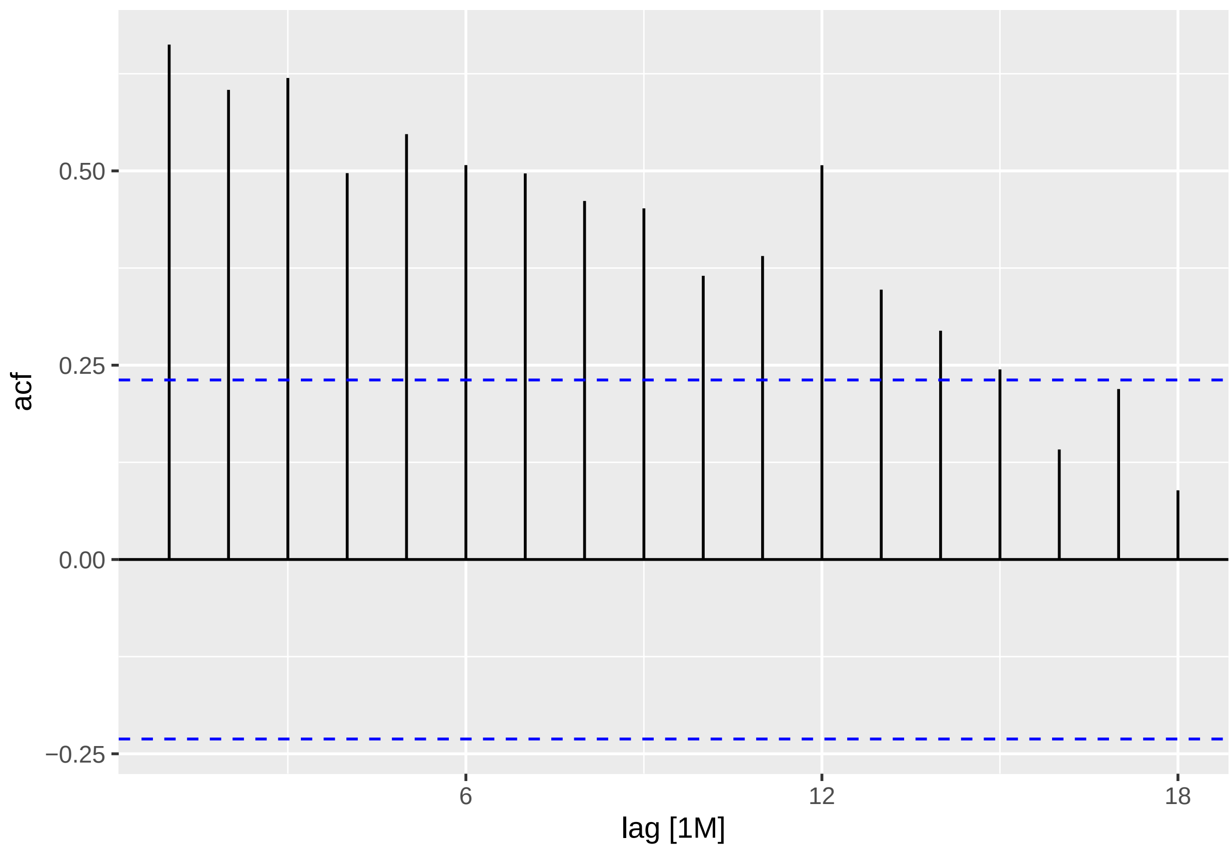
```
data("aus_livestock")
pig_slaughtered <- aus_livestock %>%
  filter(year(Month) >= 1990 & year(Month) <= 1995)  %>%
  filter(Animal == "Pigs") %>%
  filter(State == "Victoria")

autoplot(pig_slaughtered, Count)
```



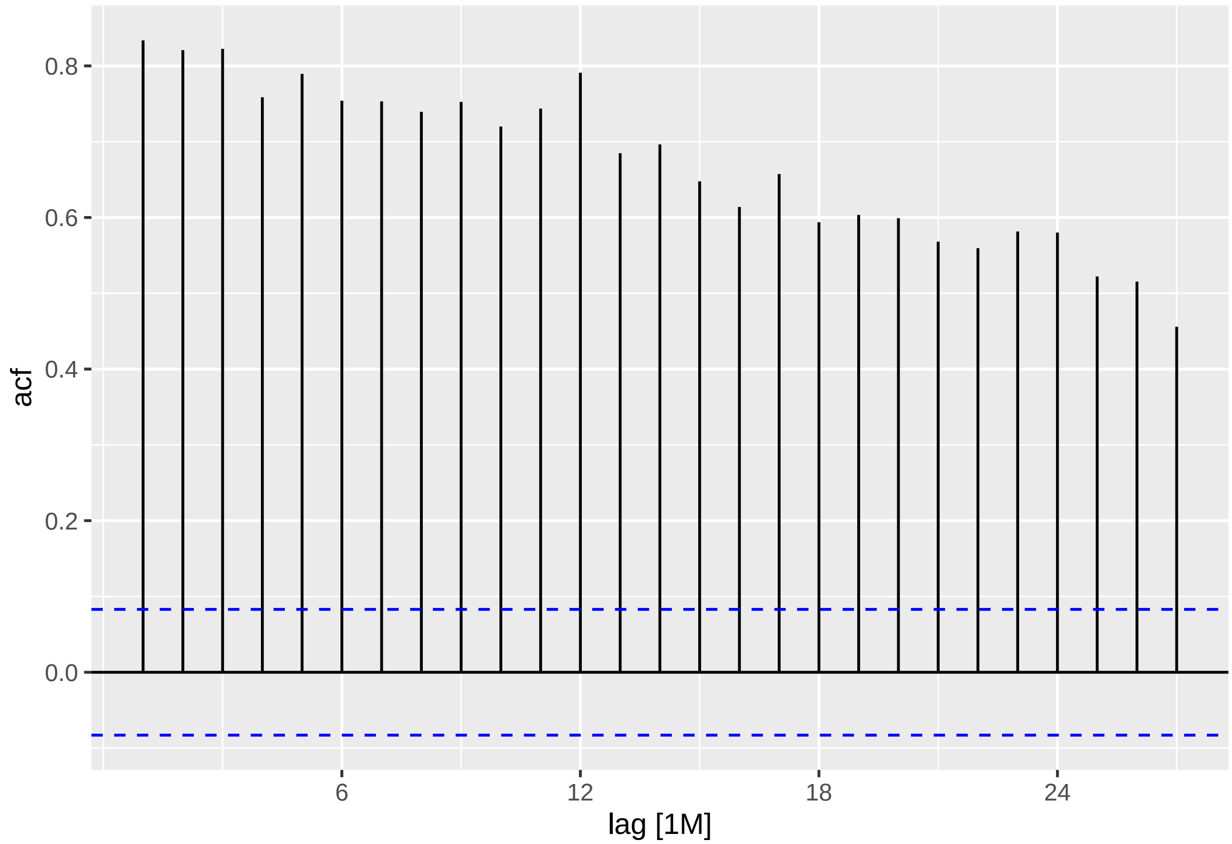

```
ACF(pig_slaughtered, Count)
```

```
## # A tsibble: 18 x 4 [1M]
## # Key:       Animal, State [1]
##    Animal State         lag    acf
##    <fct>  <fct>    <cf_lag>  <dbl>
##  1 Pigs   Victoria       1M 0.663
##  2 Pigs   Victoria       2M 0.604
##  3 Pigs   Victoria       3M 0.620
##  4 Pigs   Victoria       4M 0.497
##  5 Pigs   Victoria       5M 0.547
##  6 Pigs   Victoria       6M 0.507
##  7 Pigs   Victoria       7M 0.497
##  8 Pigs   Victoria       8M 0.461
##  9 Pigs   Victoria       9M 0.452
## 10 Pigs   Victoria      10M 0.365
## 11 Pigs   Victoria      11M 0.390
## 12 Pigs   Victoria      12M 0.507
## 13 Pigs   Victoria      13M 0.347
## 14 Pigs   Victoria      14M 0.294
## 15 Pigs   Victoria      15M 0.245
## 16 Pigs   Victoria      16M 0.142
## 17 Pigs   Victoria      17M 0.219
## 18 Pigs   Victoria      18M 0.0890
```

```r
autoplot(ACF(pig_slaughtered, Count))
```




```r
aus_livestock %>%
  filter(Animal == "Pigs") %>%
  filter(State == "Victoria") %>%
  ACF(Count) %>%
  autoplot()
```



The autoplot shows significant trend in the time series and from ACF,we observe that autocorrelation coefficient within lag 15 are all larger than the threshold while those of white noise are likely to be within the threshold. If a longer period data is used, the autocorrelation coefficients tend to be larger.

##2.10.11 a.Use the following code to compute the daily changes in Google closing stock prices.

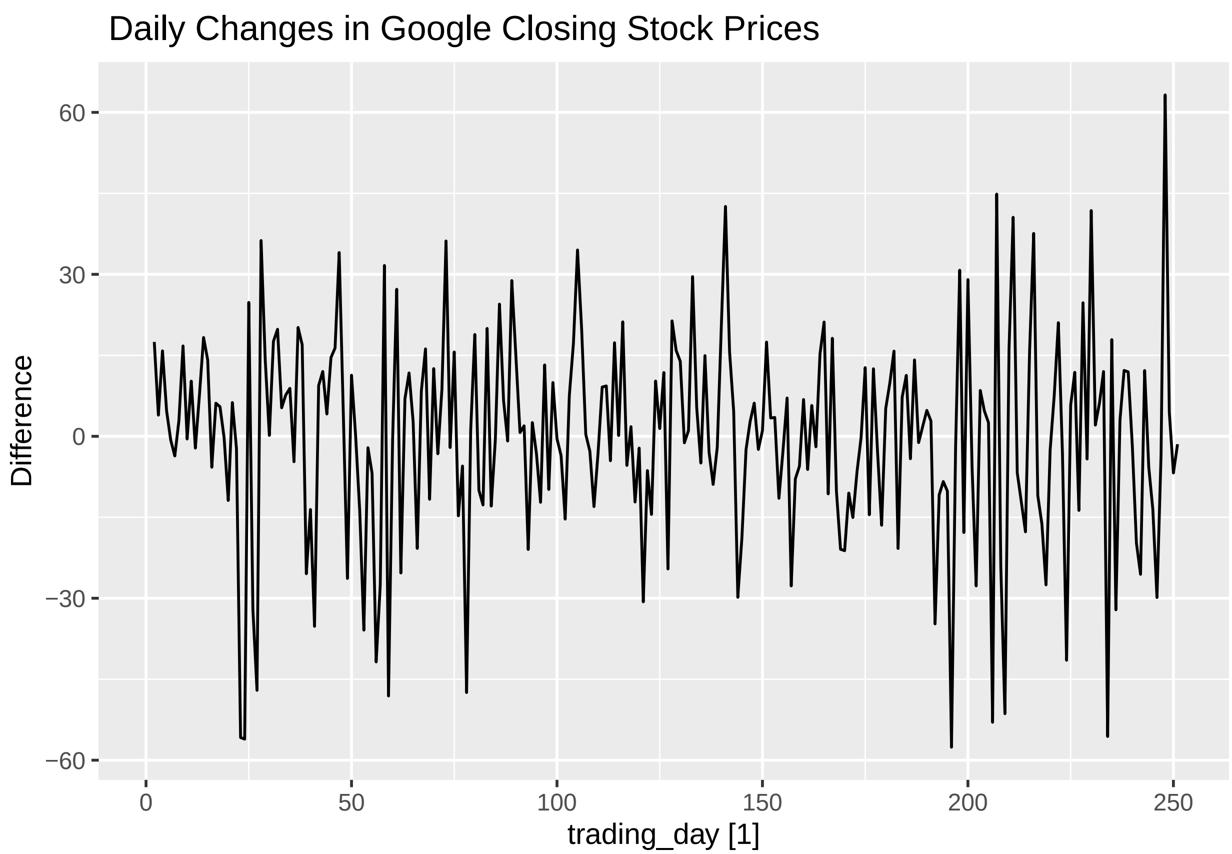
```
dgoog <- gafa_stock |>
  filter(Symbol == "GOOG", year(Date) >= 2018) |>
  mutate(trading_day = row_number()) |>
  update_tsibble(index = trading_day, regular = TRUE) |>
  mutate(diff = difference(Close))
```

b. Why was it necessary to re-index the tsibble? Observe the data, we discover that there are small gaps in **Date**, i.e. there are days that are not trading. To better study data on trading days, we re-index the data.

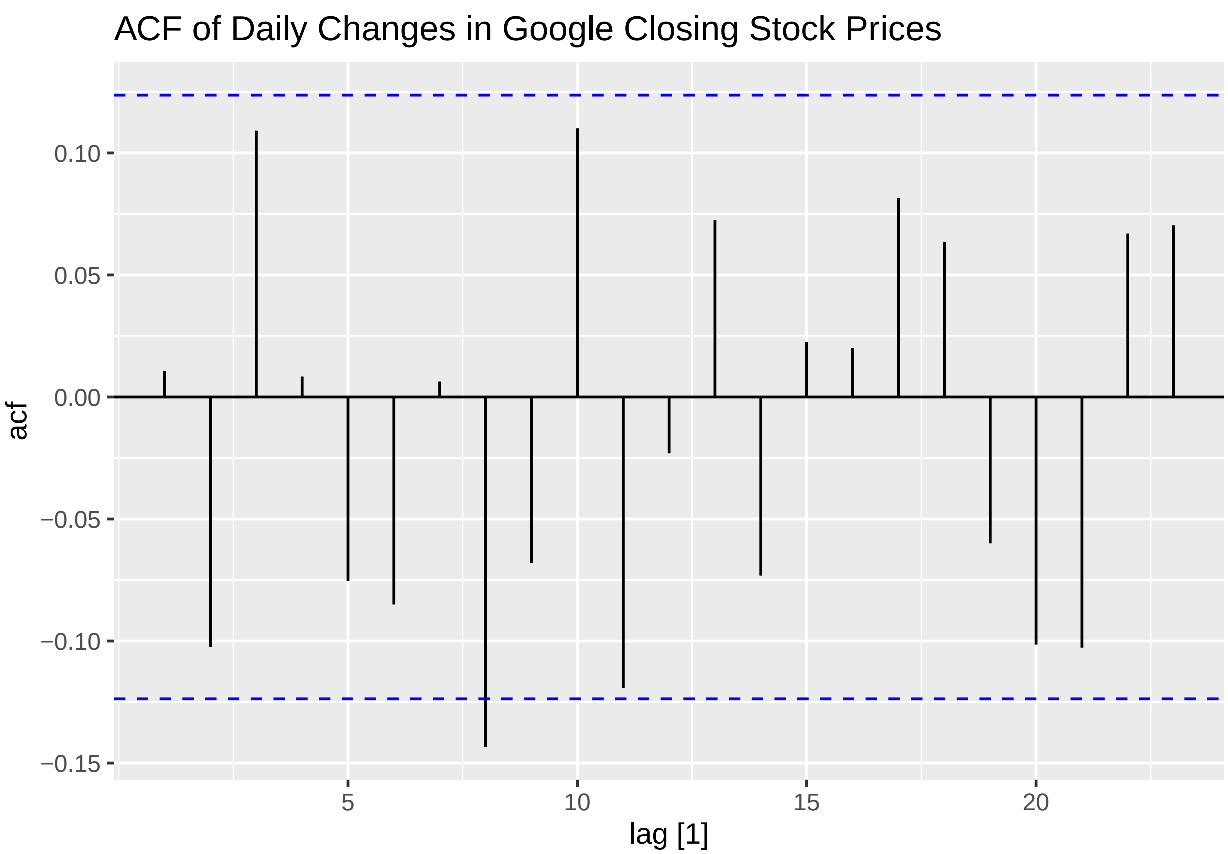
c.Plot these differences and their ACF.

```
dgoog %>%
  autoplot(diff) +
  labs(y = "Difference", title = " Daily Changes in Google Closing Stock Prices")
```

```
## Warning: Removed 1 row containing missing values (`geom_line()`).
```




```
dgoog %>%
  ACF(diff) %>%
  autoplot +
  labs(title = "ACF of Daily Changes in Google Closing Stock Prices")
```



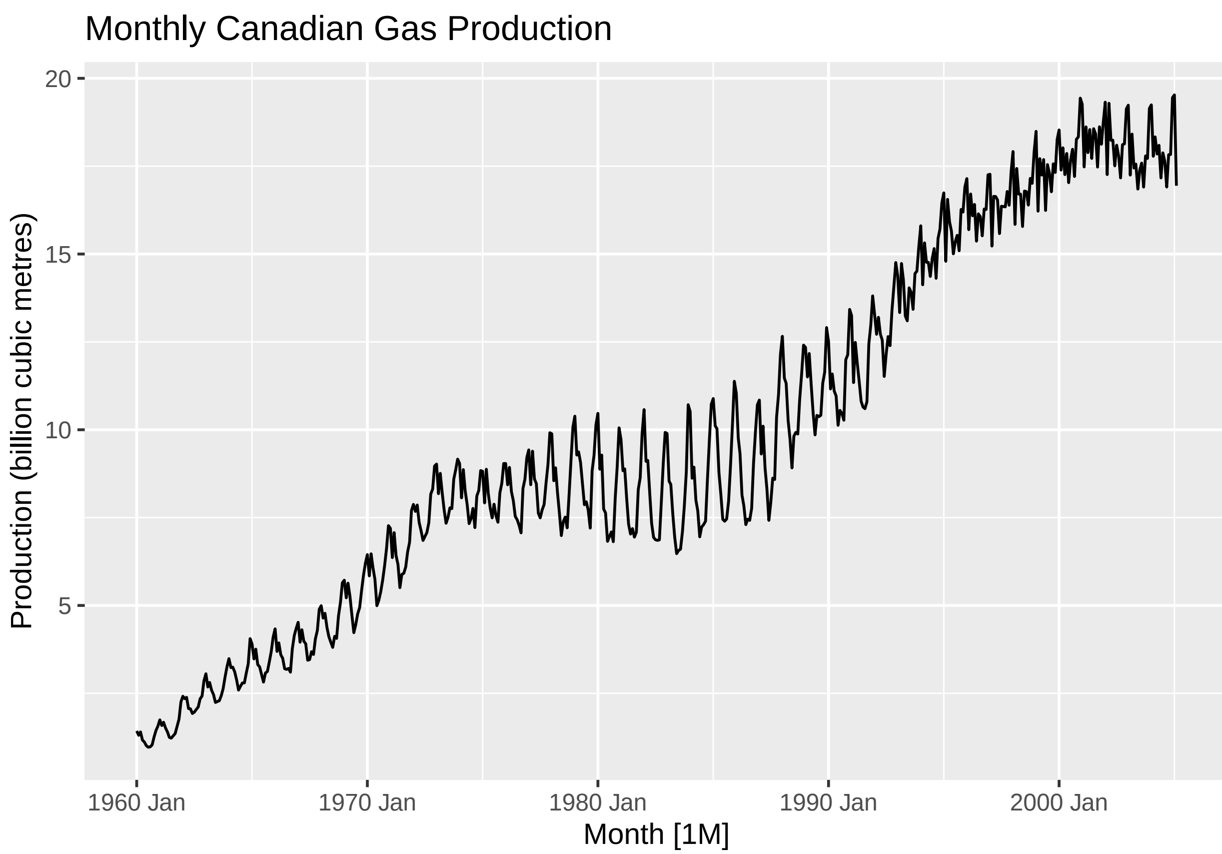
d.Do the changes in the stock prices look like white noise? According to the ACF plot, most of autocorrelation coefficients are within the threshold (except lag = 8), behaving similarly as white noise.

##3.7.10 This exercise uses the **canadian_gas** data (monthly Canadian gas production in billions of cubic metres, January 1960 – February 2005).

a.Plot the data using **autoplot()**, **gg_subseries()** and **gg_season()** to look at the effect of the changing seasonality over time.

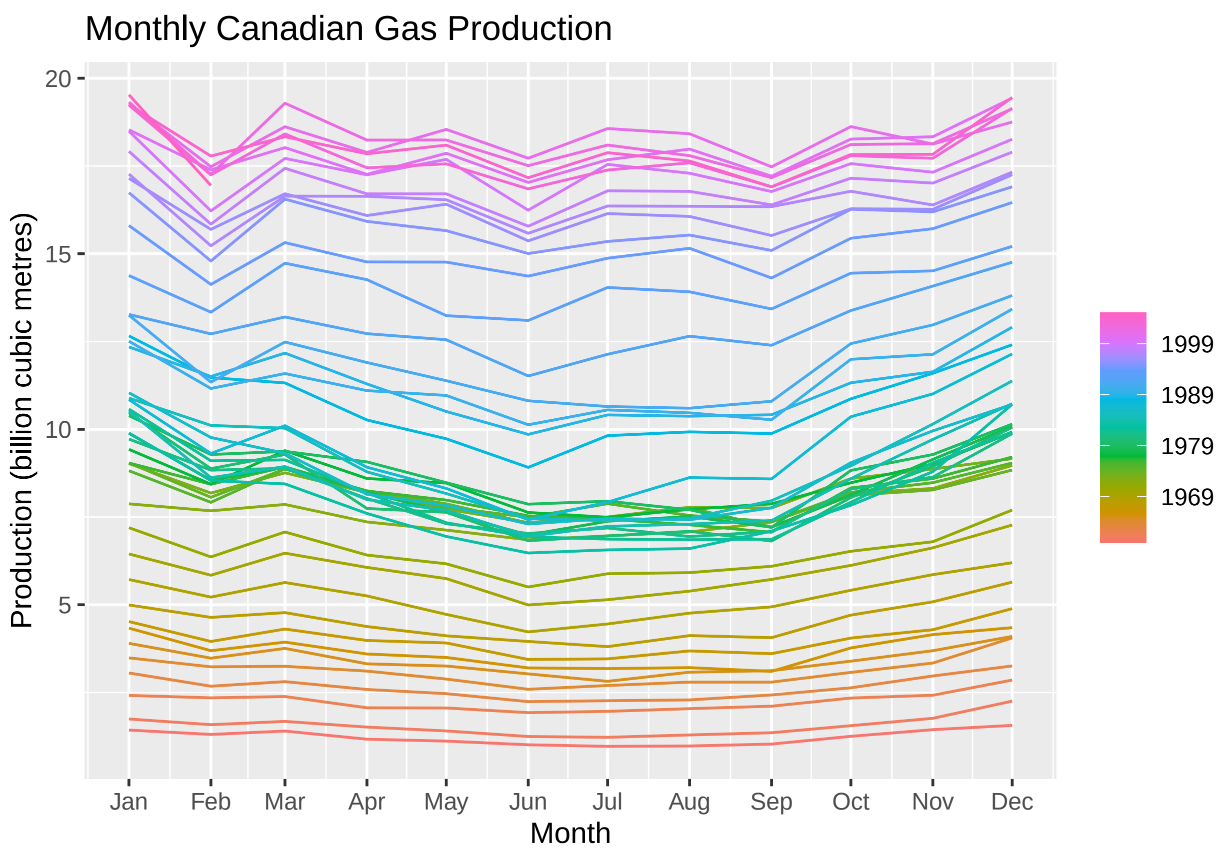
```
data("canadian_gas")
canadian_gas %>%
  autoplot() +
  labs(title = "Monthly Canadian Gas Production", y = "Production (billion cubic metres)")
```

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



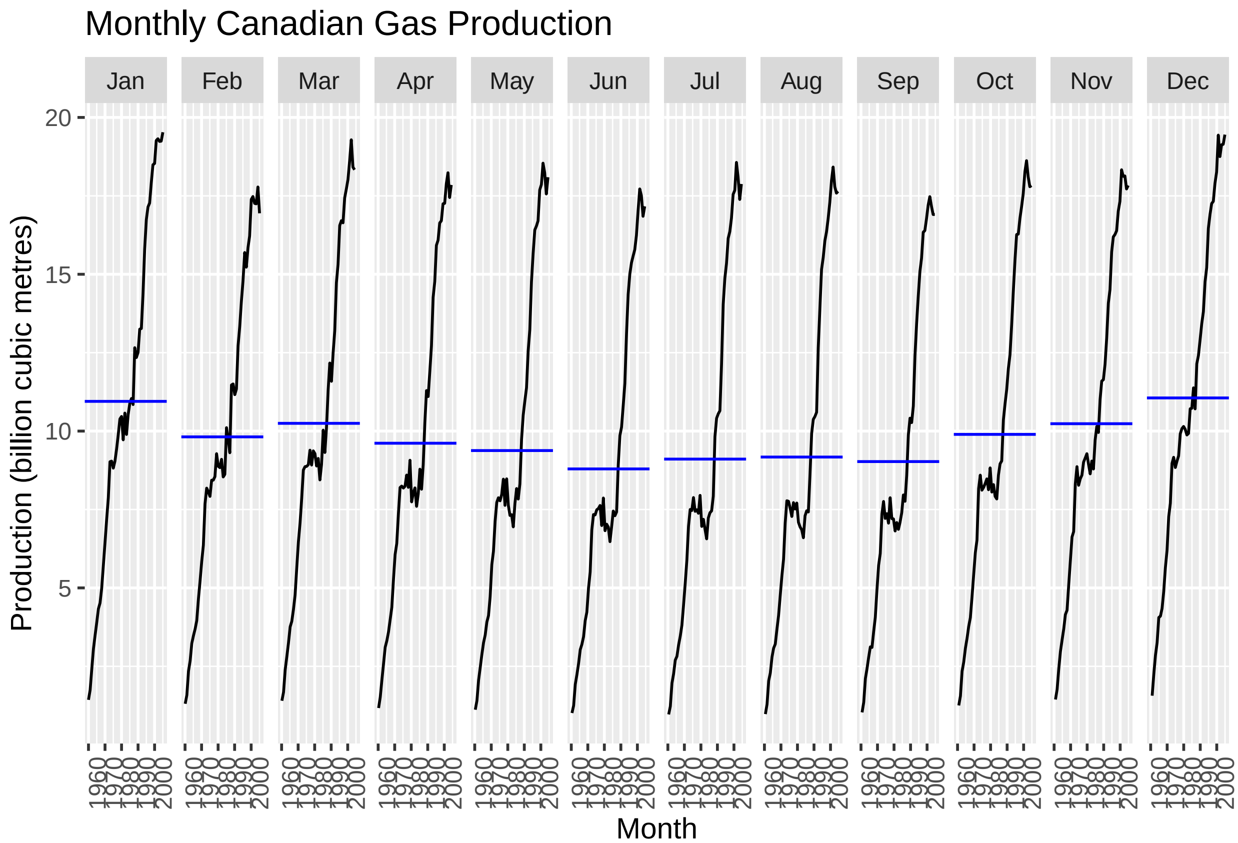

```
canadian_gas %>%
  gg_season() +
  labs(title = "Monthly Canadian Gas Production", y = "Production (billion cubic metres)")
```

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



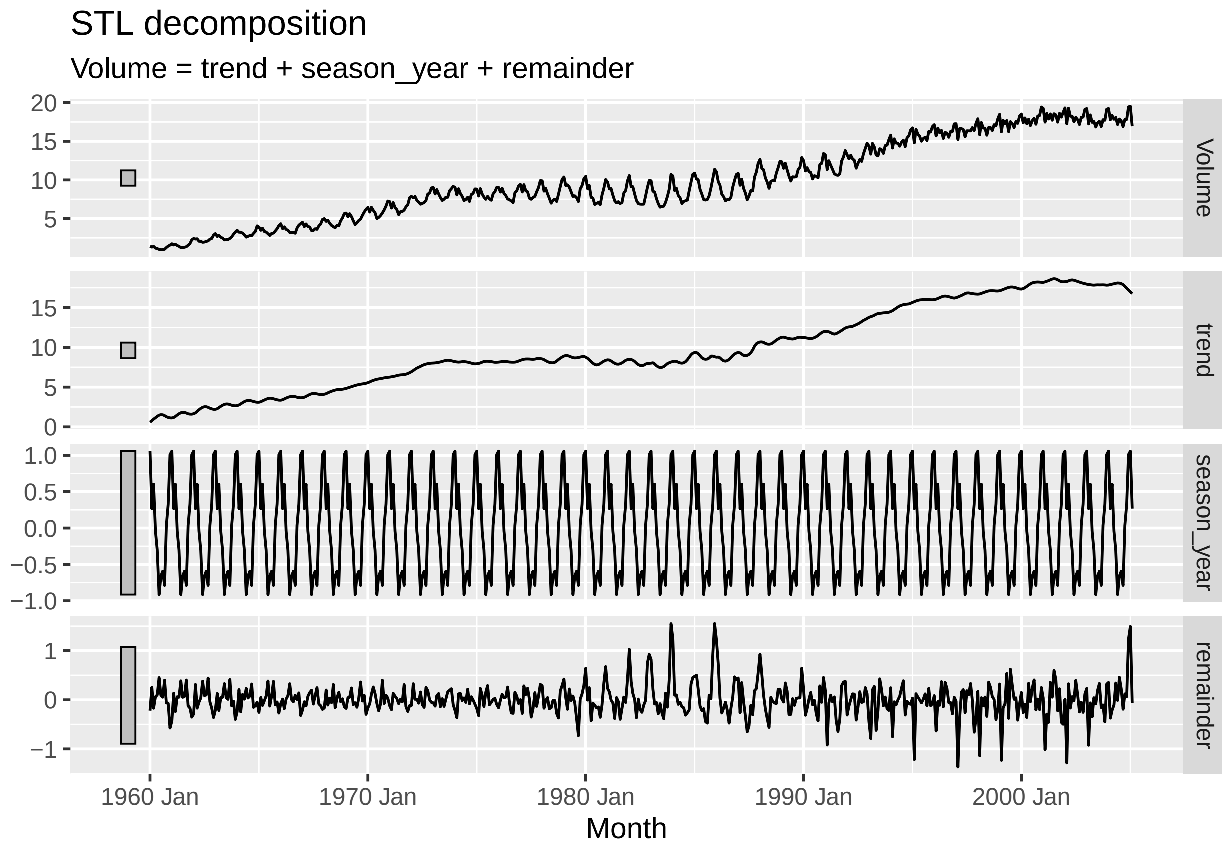

```
canadian_gas %>%
  gg_subseries() +
  labs(title = "Monthly Canadian Gas Production", y = "Production (billion cubic metres)")
```

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



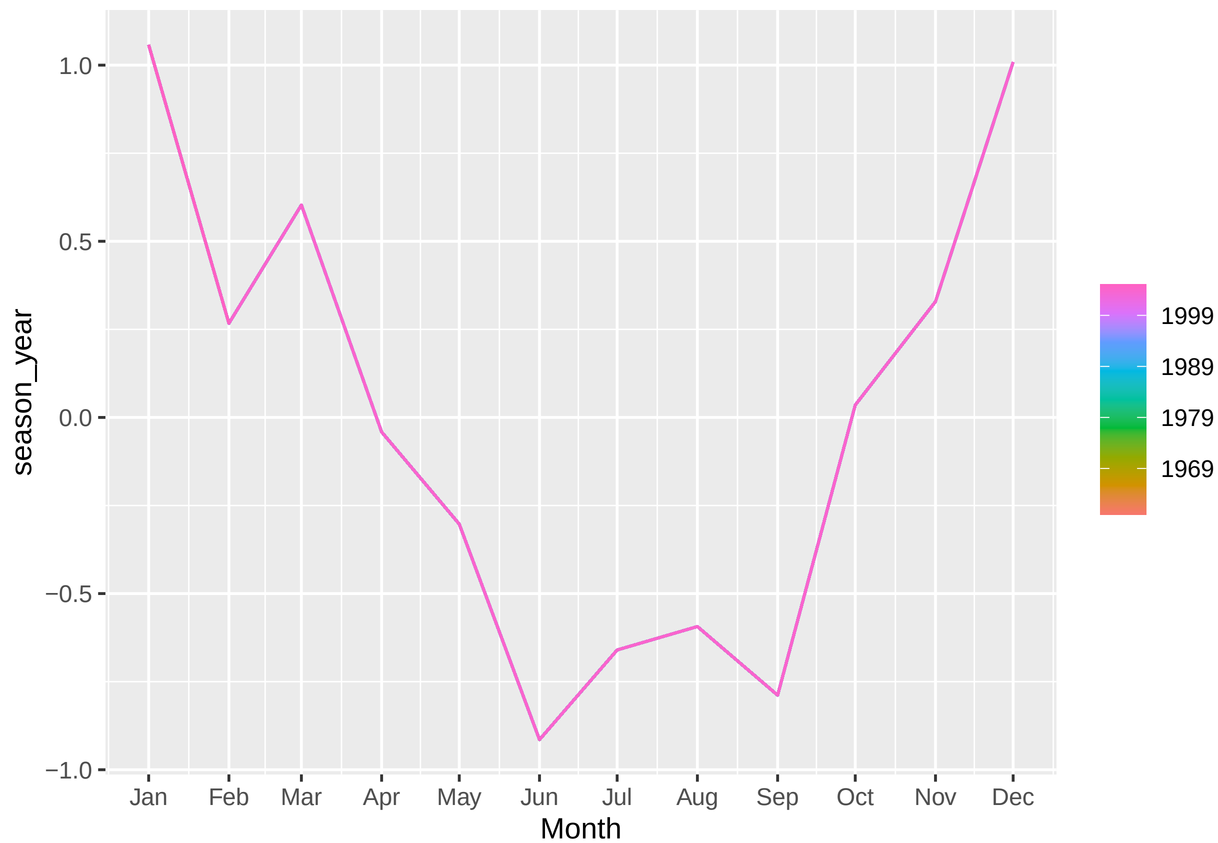
b.Do an STL decomposition of the data. You will need to choose a seasonal window to allow for the changing shape of the seasonal component.

```
canadian_gas_stl <- canadian_gas %>%
  model(
    STL(Volume ~ trend(window = 10) +
          season(window = "periodic"),
        robust = T)) %>%
  components()

canadian_gas_stl %>%
  autoplot()
```



c.How does the seasonal shape change over time? [Hint: Try plotting the seasonal component using **gg_season()**.]

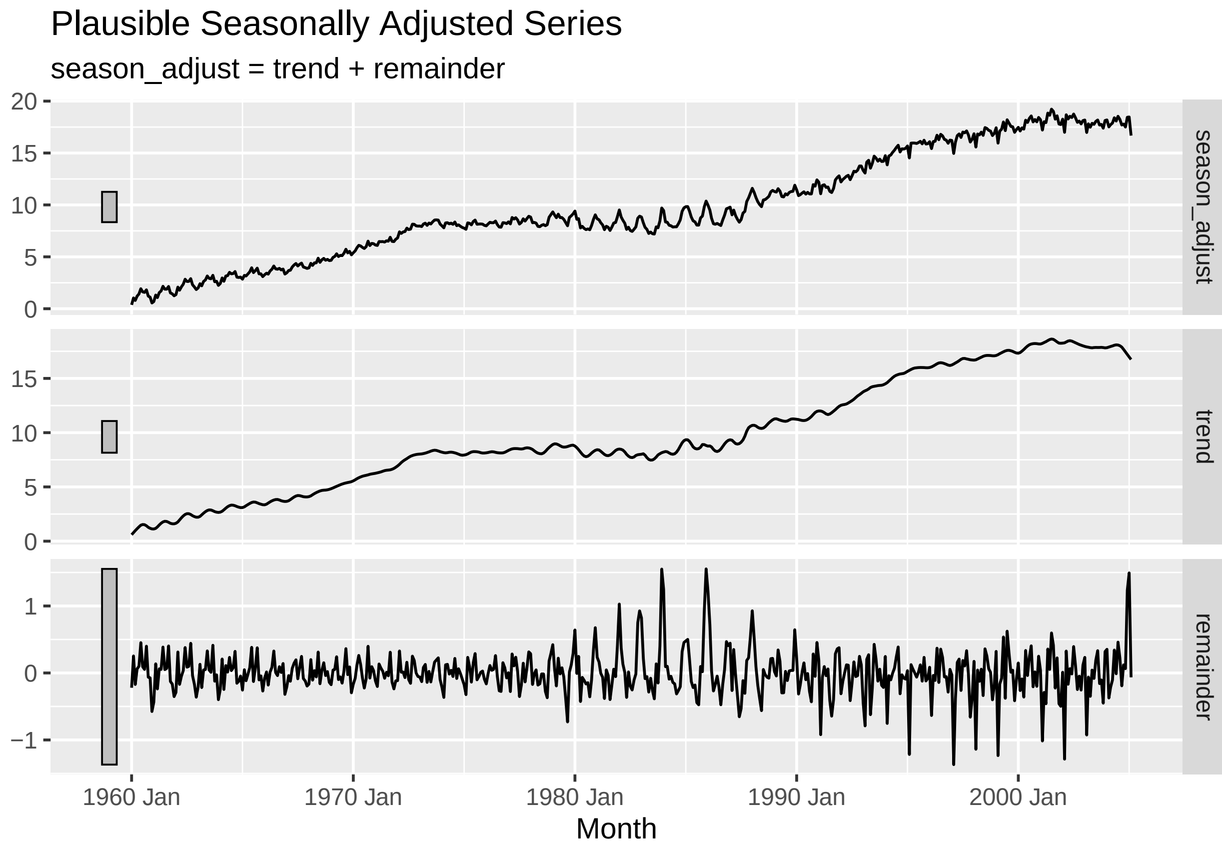
```
canadian_gas_stl %>%
  gg_season(season_year)
```



According to the seasonal plot of seasonal component, we observe a decrease from January to June but with March being a local maximum. The trend is followed by a temporary climb-ups in July and August and drop in September. The data in the last quarter of a year experience steep increase and finally reach the level of January.

d.Can you produce a plausible seasonally adjusted series?

```
canadian_gas_stl %>%
  autoplot(season_adjust) +
  labs(title = "Plausible Seasonally Adjusted Series")
```


Plausible Seasonally Adjusted Series
season_adjust = trend + remainder
20
15
10
5
0
season_adjust
15
10
5
0
trend
1
0
−1
remainder
1960 Jan
1970 Jan
1980 Jan
1990 Jan
2000 Jan
Month