

# ST5209/X Assignment 1

Teng Siyuan

## Set up

1. Make sure you have the following installed on your system:  $\text{\LaTeX}$ , R4.2.2+, RStudio 2023.12+, and Quarto 1.3.450+.
2. Clone the course [repo](#).
3. Create a separate folder in the root directory of the repo, label it with your name, e.g. `yanshuo-assignments`
4. Copy the `assignment1.qmd` file over to this directory.
5. Modify the duplicated document with your solutions, writing all R code as code chunks.
6. When running code, make sure your working directory is set to be the folder with your assignment `.qmd` file, e.g. `yanshuo-assignments`. This is to ensure that all file paths are valid.<sup>1</sup>

## Submission

1. Render the document to get a `.pdf` printout.
2. Submit both the `.qmd` and `.pdf` files to Canvas.

## Question 1 (Quarto)

Read the [guide](#) on using Quarto with R and answer the following questions:

- a) Write a code chunk that imports `tidyverse` and `fpp3`.

```
library(tidyverse)
```

```
Warning: package 'ggplot2' was built under R version 4.5.2
```

```
Warning: package 'readr' was built under R version 4.5.2
```

---

<sup>1</sup>You may view and set the working directory using `getwd()` and `setwd()`.

```

-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr      1.1.4      v readr      2.1.6
v forcats    1.0.1      v stringr   1.5.1
v ggplot2    4.0.1      v tibble    3.3.0
v lubridate  1.9.4      v tidyr     1.3.1
v purrr      1.1.0
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()     masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to
library(fpp3)

```

```

Registered S3 method overwritten by 'tsibble':
  method      from
  as_tibble.grouped_df dplyr
-- Attaching packages ----- fpp3 1.0.2 --
v tsibble      1.1.6      v feasts      0.4.2
v tsibbledata  0.4.1      v fable       0.4.1
-- 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()

```

- b) Modify the chunk so that only the following output is shown (i.e. the usual output about attaching packages and conflicts is not shown.)

```

library(tidyverse)
library(fpp3)

```

```

library(tidyverse)
library(fpp3)

```

- c) Modify the chunk so that it is executed but no code is shown at all when rendered to a pdf.
- d) Modify the document so that your name is printed on it beneath the title.

## Question 2 (Livestock)

Consider the `aus_livestock` dataset loaded in the `fpp3` package.

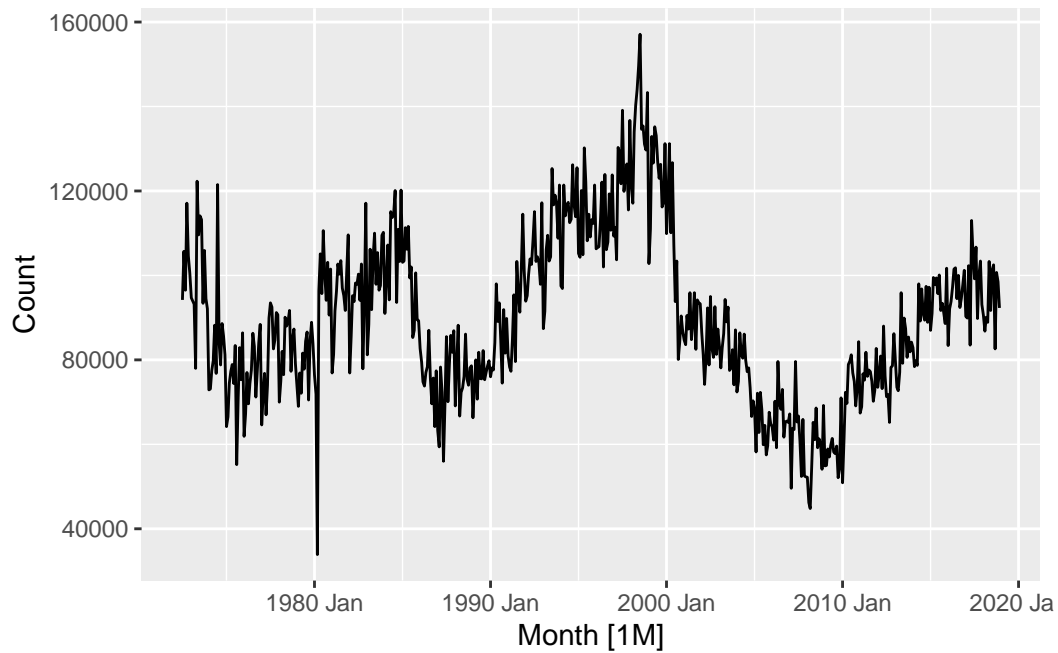
- a) Use `filter()` to extract a time series comprising the monthly total number of pigs slaughtered in Victoria, Australia, from Jul 1972 to Dec 2018.

```
pigs_victoria <- aus_livestock |>
  filter(Animal == "Pigs",
         State == "Victoria") |>
  filter(Month >= yearmonth("1972 Jul") & Month <= yearmonth("2018 Dec"))
pigs_victoria
```

```
# A tsibble: 558 x 4 [1M]
# Key:       Animal, State [1]
   Month Animal State   Count
   <mth> <fct> <fct>   <dbl>
1 1972 Jul Pigs  Victoria  94200
2 1972 Aug Pigs  Victoria 105700
3 1972 Sep Pigs  Victoria  96500
4 1972 Oct Pigs  Victoria 117100
5 1972 Nov Pigs  Victoria 104600
6 1972 Dec Pigs  Victoria 100500
7 1973 Jan Pigs  Victoria  94700
8 1973 Feb Pigs  Victoria  93900
9 1973 Mar Pigs  Victoria  93200
10 1973 Apr Pigs  Victoria  78000
# i 548 more rows
```

- b) Make a time plot of the resulting time series.

```
pigs_victoria |>
  autoplot(Count)
```

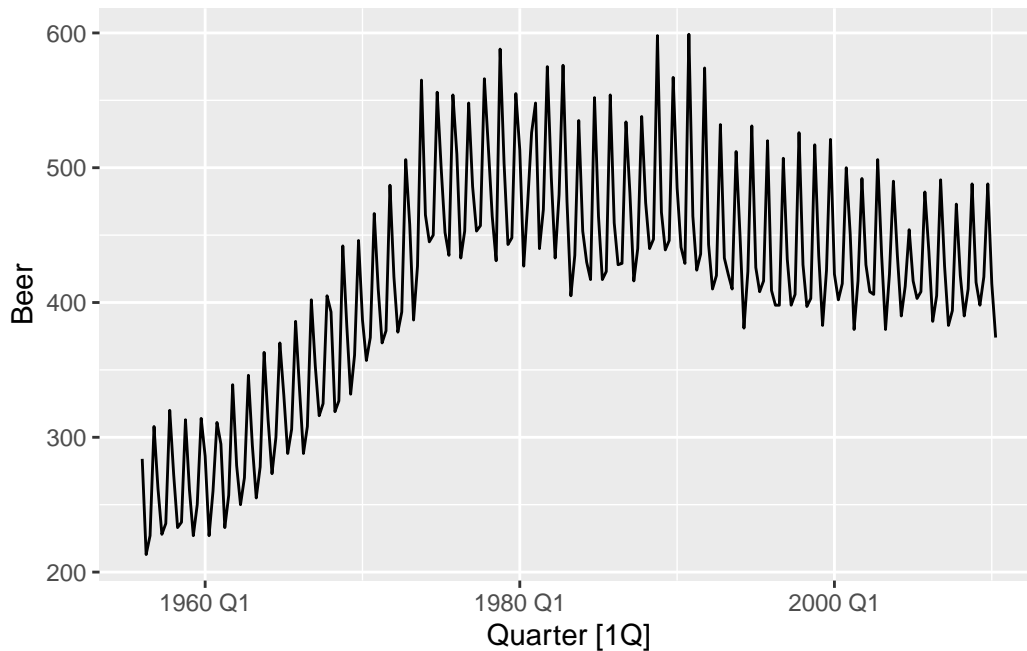


### Question 3 (Beer production)

Consider the `aus_production` dataset loaded in the `fpp3` package. We will study the column measuring the production of beer.

- a) Make a time plot of the beer production time series.

```
beer_pro <- aus_production |>
  select(Quarter, Beer)
beer_pro |>
  autoplot(Beer)
```



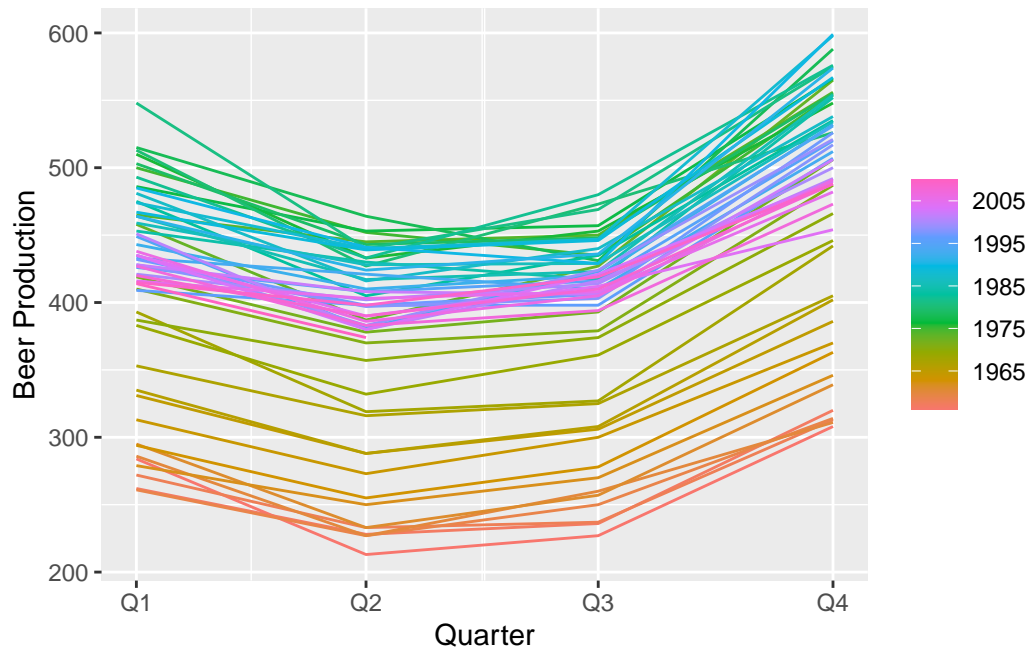
b) Describe the observed trend.

The time series shows an upward trend from 1950s to 1970s, a plateau around late 1970s and 1980s, then a gradual decline thereafter.

c) Make a seasonal plot.

```
beer_pro |>
  gg_season(Beer)+
  labs(y="Beer Production")
```

Warning: `gg\_season()` was deprecated in feasts 0.4.2.  
i Please use `ggtime::gg\_season()` instead.



d) What is the period of the seasonality?

The period of seasonality is 4 quarters(one year)

e) Describe the seasonal behavior.

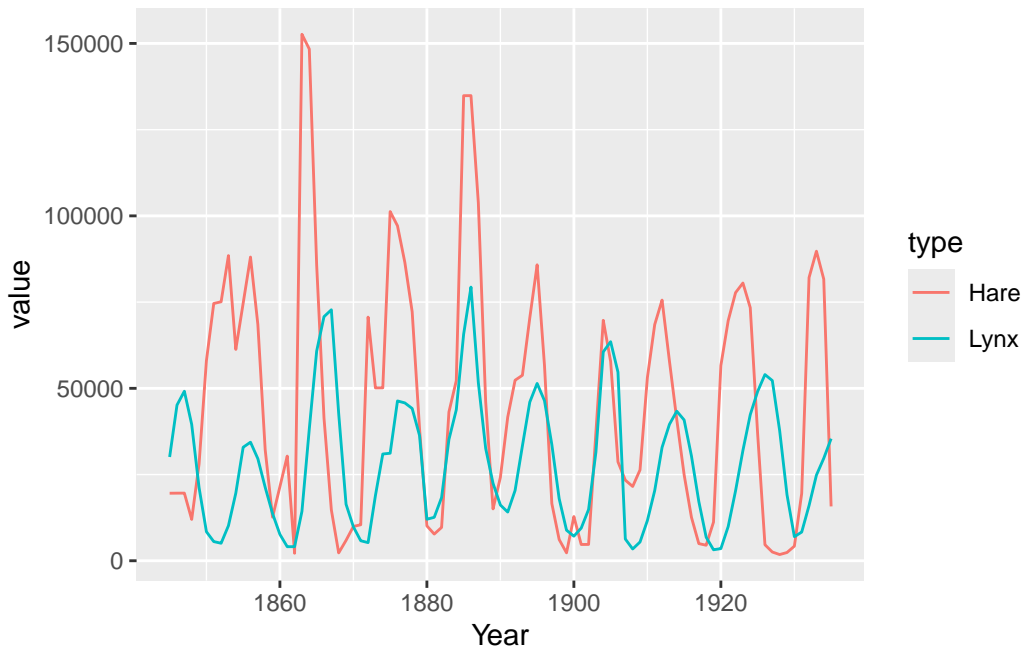
Beer Production is typically lowest in the second quarter(summer) and increases thereafter, reaching its peak in the fourth quarter(winter).

### Question 4 (Pelts)

Consider the `pelt` dataset loaded in the `fpp3` package, which measures the Hudson Bay Company trading records for Snowshoe Hare and Canadian Lynx furs from 1845 to 1935.

a) Plot both time series on the same axes. *Hint: Use `pivot_longer()` to create a key column.*

```
pelt_new <- pelt|>
  pivot_longer(cols=-Year,
               names_to = "type",
               values_to = "value")
ggplot(pelt_new,aes(x=Year,y=value,color=type))+
  geom_line()
```



- b) What happens when you try to use `gg_season()` to the lynx fur time series? What is producing the error?

```
try(feasts::gg_season(pelt, Lynx))
```

```
Error in feasts::gg_season(pelt, Lynx) :
```

```
The data must contain at least one observation per seasonal period.
```

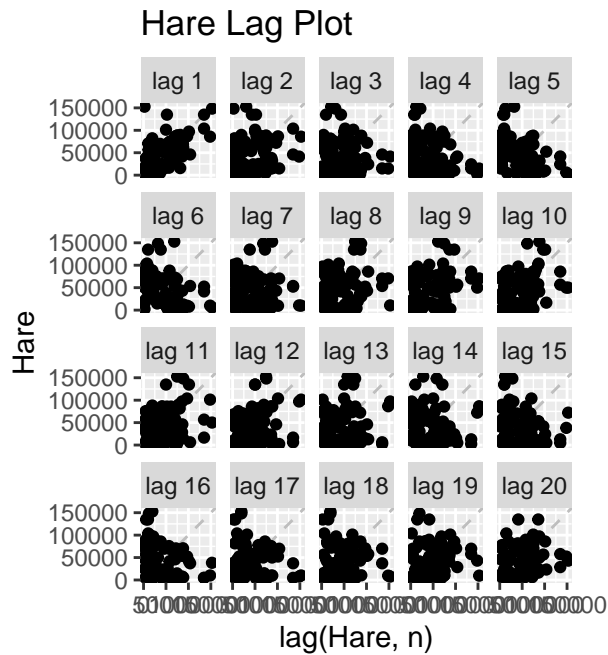
An error occurs because the data are annual and do not contain seasonal structure. There's only one observation per year, the seasonal period is one, which prevents the construction of a seasonal plot.

- c) Make a lag plot with the first 20 lags. Which lags display strong positive correlation? Which lags display strong negative correlation? Verify this with the time plot.

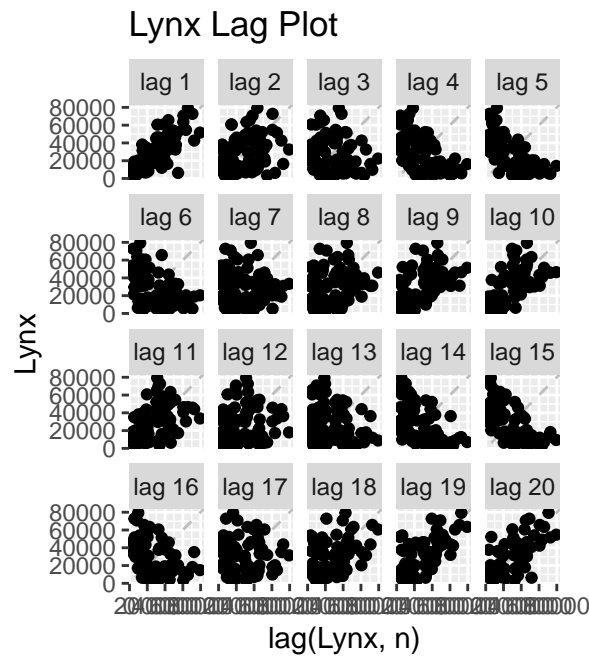
```
pelt |>
  gg_lag(y=Hare, geom='point',lags=1:20)+
  labs(title="Hare Lag Plot")
```

```
Warning: `gg_lag()` was deprecated in feasts 0.4.2.
```

```
i Please use `ggtime::gg_lag()` instead.
```



```
pelt |>
  gg_lag(y=Lynx, geom='point',lags=1:20)+
  labs(title="Lynx Lag Plot")
```



For both hare and lynx series, strong positive correlations are observed around lag1, lag10 and lag 20, while negative correlations are observed around lag5 and lag15. This is



consistent with previous time plot, which suggests a 10-year cycle.

- d) If you were to guess the seasonality period based on the lag plot, what would it be?

It would be a 10-year seasonality period.

- e) Use the provided function `gg_custom_season()` in `_code/plot_util.R`<sup>2</sup> to make a seasonal plot for lynx furs with the period that you guessed.<sup>3</sup> Does the resulting plot suggest seasonality? Why or why not?

```
source("../_code/plot_util.R")
```

```
Loading required package: rlang
```

```
Attaching package: 'rlang'
```

```
The following objects are masked from 'package:purrr':
```

```
  %%, flatten, flatten_chr, flatten_dbl, flatten_int, flatten_lgl,  
  flatten_raw, invoke, splice
```

```
Loading required package: magrittr
```

```
Attaching package: 'magrittr'
```

```
The following object is masked from 'package:rlang':
```

```
  set_names
```

```
The following object is masked from 'package:purrr':
```

```
  set_names
```

```
The following object is masked from 'package:tidyr':
```

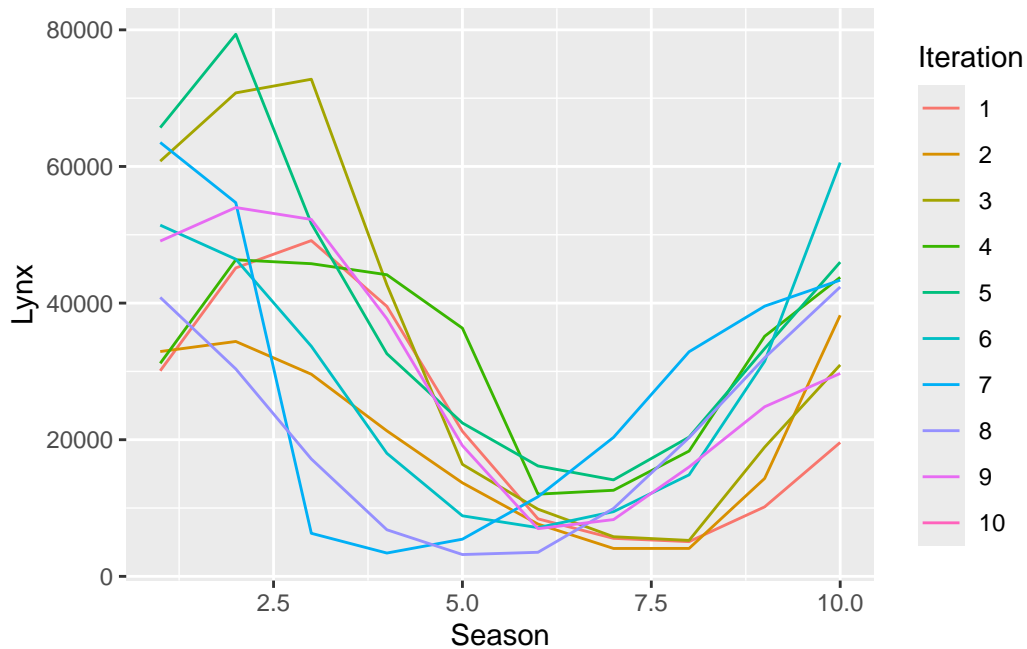
```
  extract
```

```
gg_custom_season(pelt, Lynx, period = 10)
```

---

<sup>2</sup>You can load this function using `source("../_code/plot_util.R")`.

<sup>3</sup>Unfortunately, it seems `gg_season()` does not allow this functionality.



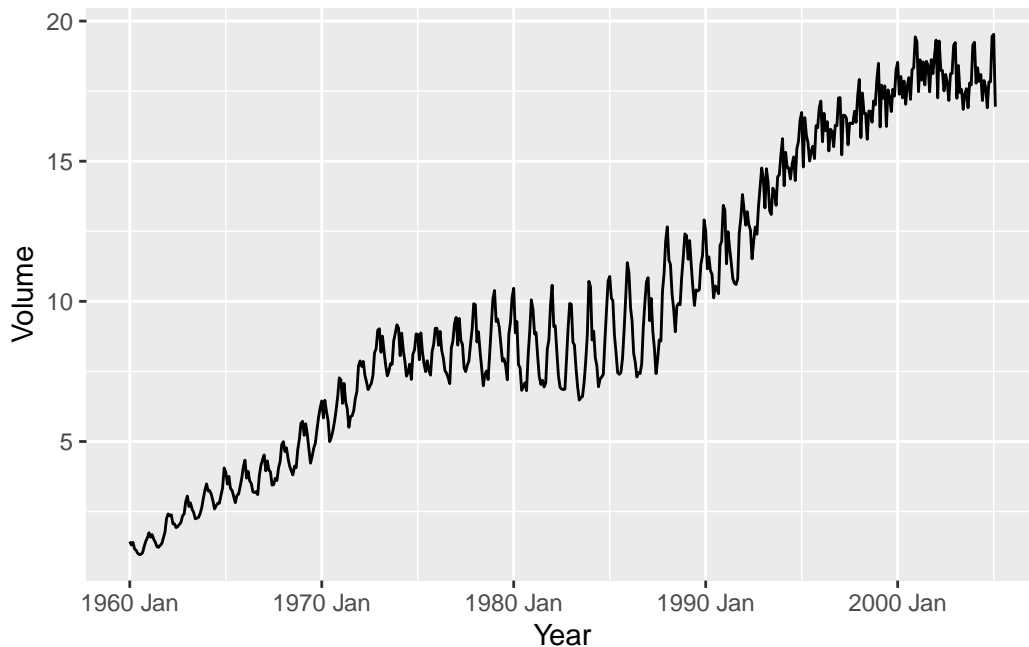
From the plot we can observe that each 10-year interval generally shows similar pattern, However, the lowest point and highest point of within each interval are not consistent across iterations. This indicates that the pattern observed is more likely a result of cycle instead of seasonality.

### Question 5 (Box-Cox, Q3.3 in FPP)

Why is the Box-Cox transform unhelpful for the `canadian_gas` data?

```
canadian_gas |>
  autoplot()+labs(x='Year')
```

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



The Box-Cox transformation is unhelpful because we can observe that the variance of Canadian gas production data does not increase explicitly with the level of series. As a result, Box-Cox transformation does not stabilize the variance of the series.

### Question 6 (Decomposition with outliers, Q3.7 in FPP)

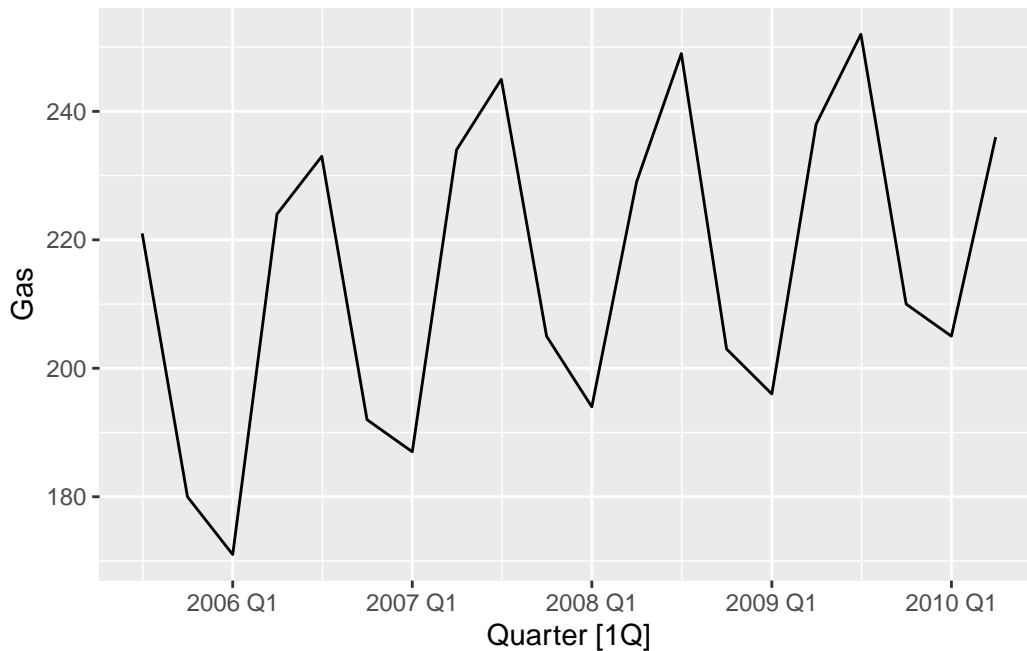
Consider the last five years of the Gas data from `aus_production`.

```
gas <- tail(aus_production, 5*4) |> select(Gas)
```

- Plot the time series. Can you identify seasonal fluctuations and/or a trend-cycle?

```
autoplot(gas)
```

Plot variable not specified, automatically selected ``.vars = Gas``



From the plot, we can observe a up-and-down pattern which repeats each year that indicates the seasonal fluctuations. And a slight upward trend can also be observed.

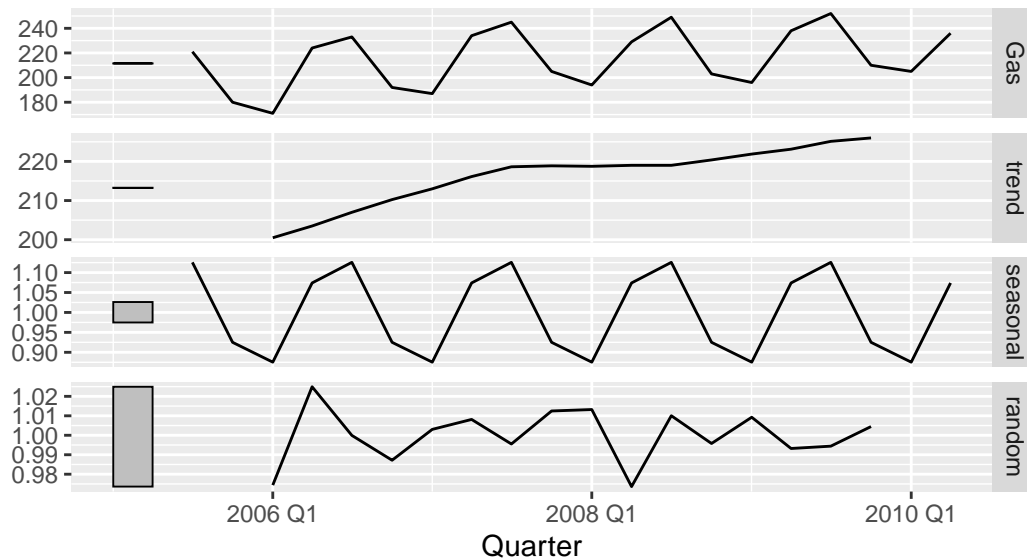
- b. Use `classical_decomposition` with `type=multiplicative` to calculate the trend-cycle and seasonal indices.

```
decomp <- gas|>
  model(classical_decomposition(Gas,type="multiplicative"))|>
  components()
autoplot(decomp)
```

Warning: Removed 8 rows containing missing values or values outside the scale range (``geom_line()``).

## Classical decomposition

Gas = trend \* seasonal \* random

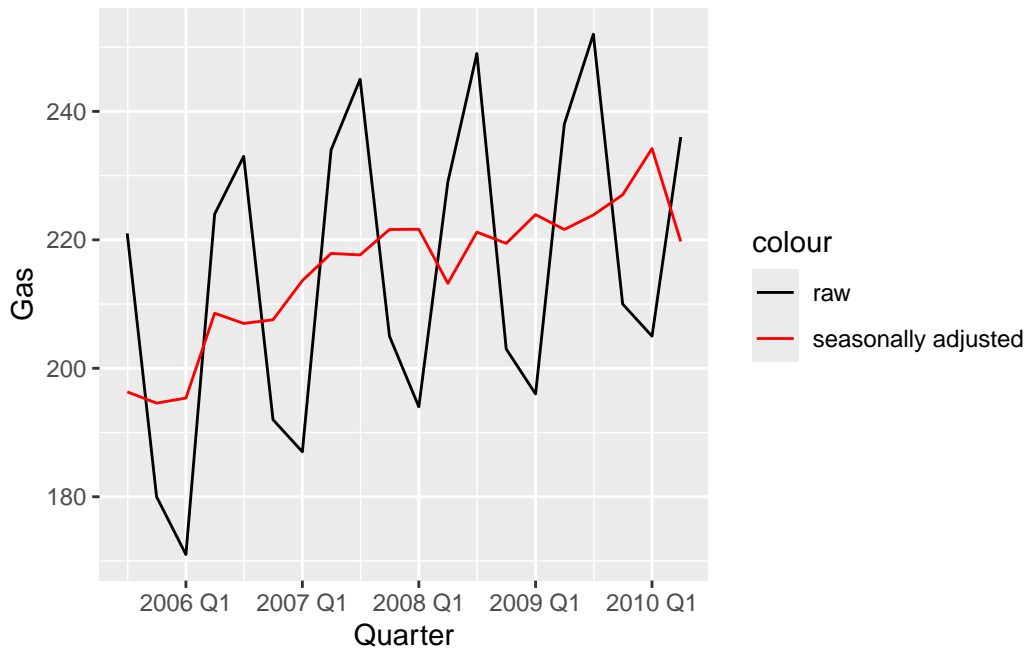


c. Do the results support the graphical interpretation from part a?

The results indicate the existence of a slight upward trend and seasonality.

d. Compute and plot the seasonally adjusted data.

```
decomp|>
  ggplot(aes(x=Quarter))+geom_line(aes(y = Gas,colour="raw"))+
  geom_line(aes(y = season_adjust,colour ="seasonally adjusted"))+
  scale_colour_manual(values = c("black","red"),
                      breaks = c("raw","seasonally adjusted"))
```

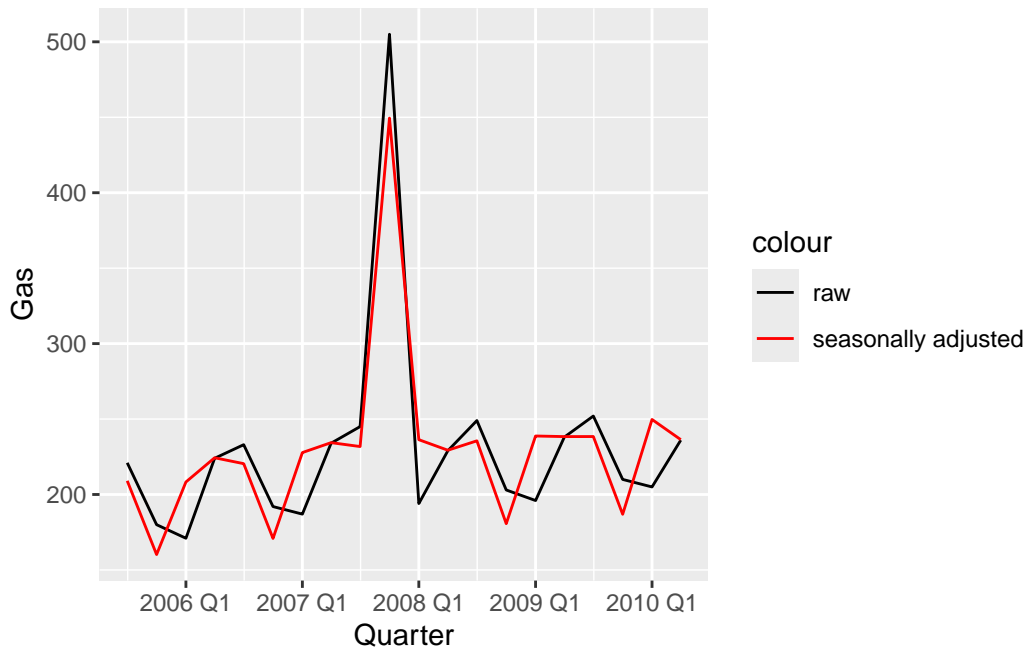


- e. Change one observation to be an outlier by running the following snippet:

```
gas_outlier<- gas |>
  mutate(Gas = if_else(Quarter == yearquarter("2007Q4"), Gas + 300, Gas))
```

Recompute the decomposition. What is the effect of the outlier on the seasonally adjusted data?

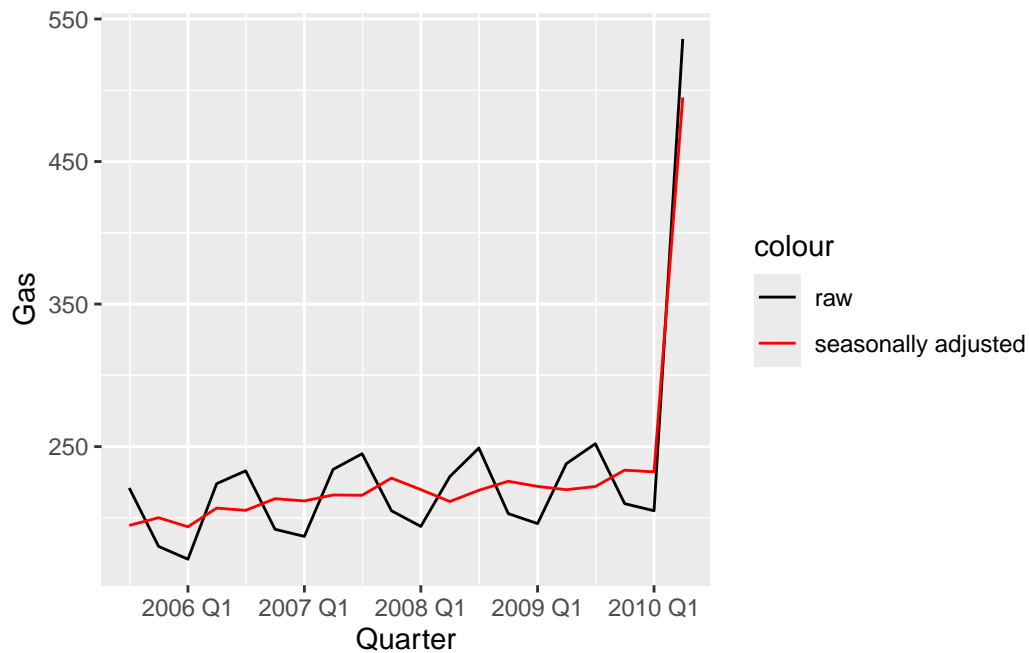
```
decomp_new<-gas_outlier|>
  model(classical_decomposition(Gas,type="multiplicative"))|>
  components()
decomp_new|>
  ggplot(aes(x=Quarter))+geom_line(aes(y = Gas,colour="raw"))+
  geom_line(aes(y = season_adjust,colour ="seasonally adjusted"))+
  scale_colour_manual(values = c("black","red"),
                      breaks = c("raw","seasonally adjusted"))
```



It can be observed that a large spike is produced by the outlier, and the seasonally adjusted data shows some seasonality because the outlier has affected the estimation of seasonal component in decomposition.

- f. Does it make any difference if the outlier is near the end rather than in the middle of the time series?

```
gas_outlier2<- gas |>
  mutate(Gas = if_else(Quarter == yearquarter("2010Q2"), Gas + 300, Gas))
decomp_new2<-gas_outlier2|>
  model(classical_decomposition(Gas,type="multiplicative"))|>
  components()
decomp_new2|>
  ggplot(aes(x=Quarter))+geom_line(aes(y = Gas,colour="raw"))+
  geom_line(aes(y = season_adjust,colour ="seasonally adjusted"))+
  scale_colour_manual(values = c("black","red"),
                      breaks = c("raw","seasonally adjusted"))
```



It can be observed that outlier near the end has a different effect: there's no seasonality in seasonally adjusted data. Its effect is more localised and has smaller influence on the estimation of seasonal component.

### Question 7 (STL decomposition, Q3.10 in FPP)

Consider the `canadian_gas` dataset.

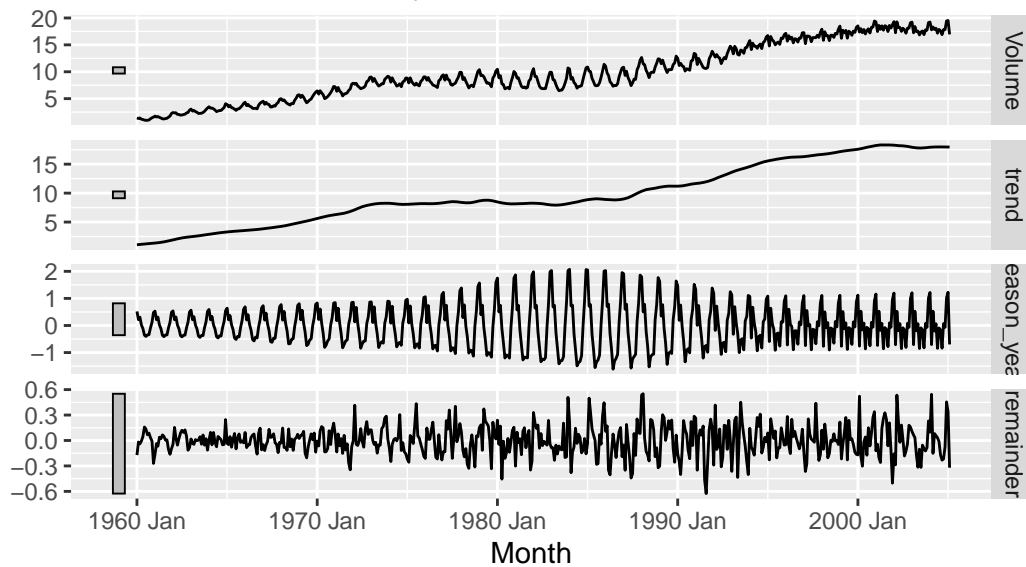
- Do an STL decomposition of the data.

```
decp <- canadian_gas|>
  model(STL(Volume))|>
  components()
autoplot(decp)
```



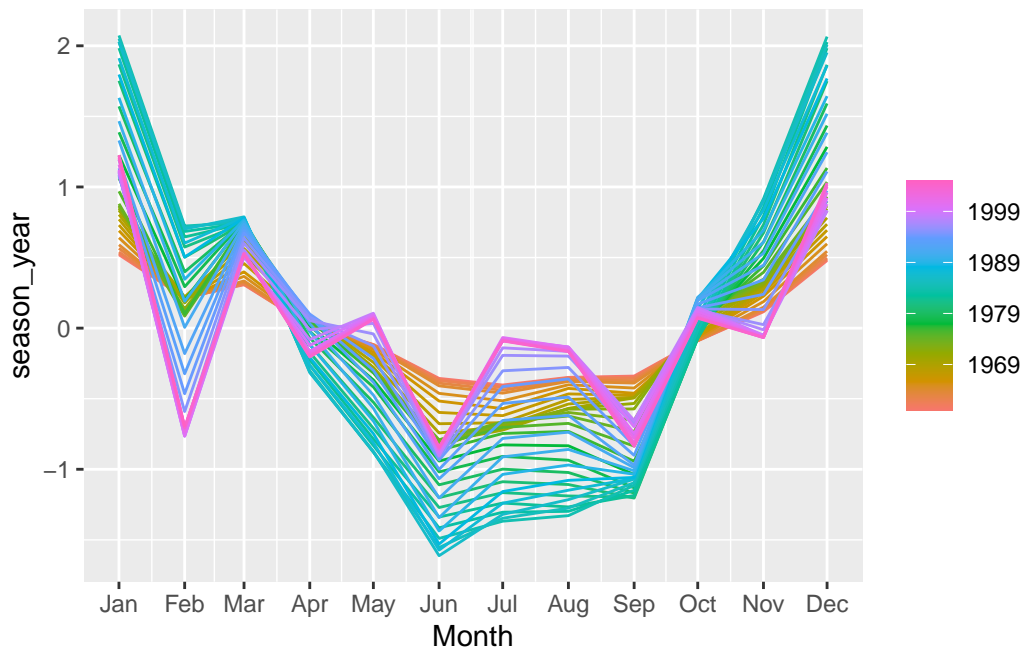
## STL decomposition

Volume = trend + season\_year + remainder



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

```
decpl >  
gg_season(season_year)
```



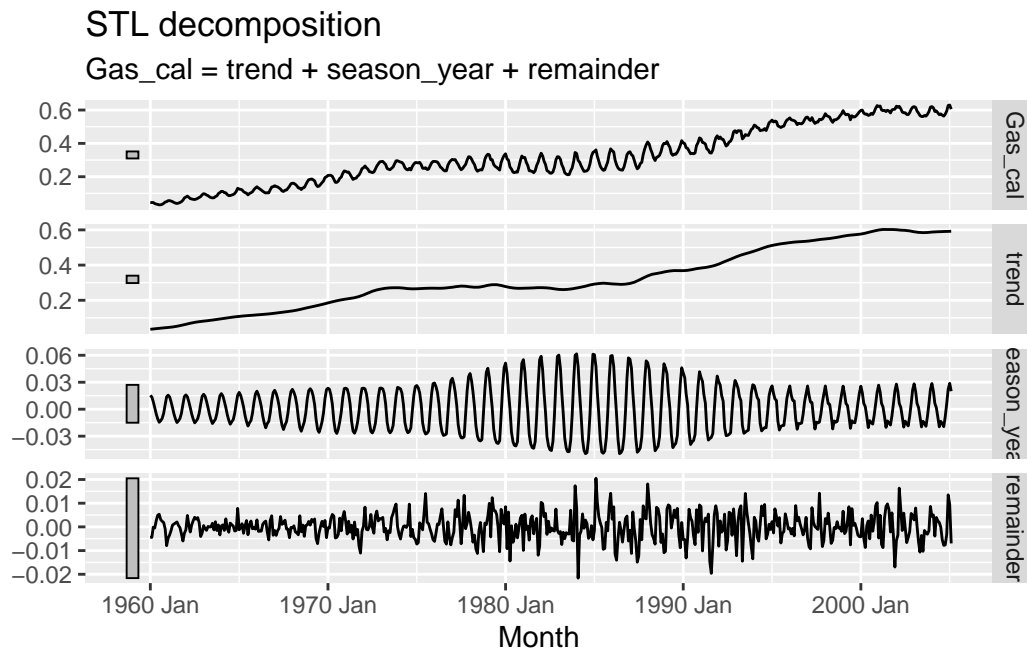
It can be observed that up to around 1989 the seasonality had been stable. After 1989,

the amplitude of it increase, with stronger winter peaks and deeper summer troughs.

- c. Apply a calendar adjustment and compute the STL decomposition again. What is the effect on the seasonal shape?

```
gas_cal<- canadian_gas|>
  mutate(Gas_cal=Volume/days_in_month(Month))
```

```
decp_cal<- gas_cal|>
  model(STL(Gas_cal))|>
  components()
autoplot(decp_cal)
```



It can be observed that after calendar adjustment, the seasonal component becomes smoother and less distorted.