ST5209X Assignement1

ZHAO LINGJIE

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`.

```r
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.4.4     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.0
v purrr     1.0.2
-- 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
```

```r
library(fpp3)
```

```
-- Attaching packages ---------------------------------------------- fpp3 0.5 --
v tsibble     1.1.3     v fable       0.3.3
v tsibbledata 0.4.1     v fabletools  0.3.4
v feasts      0.3.1
-- 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)
```

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

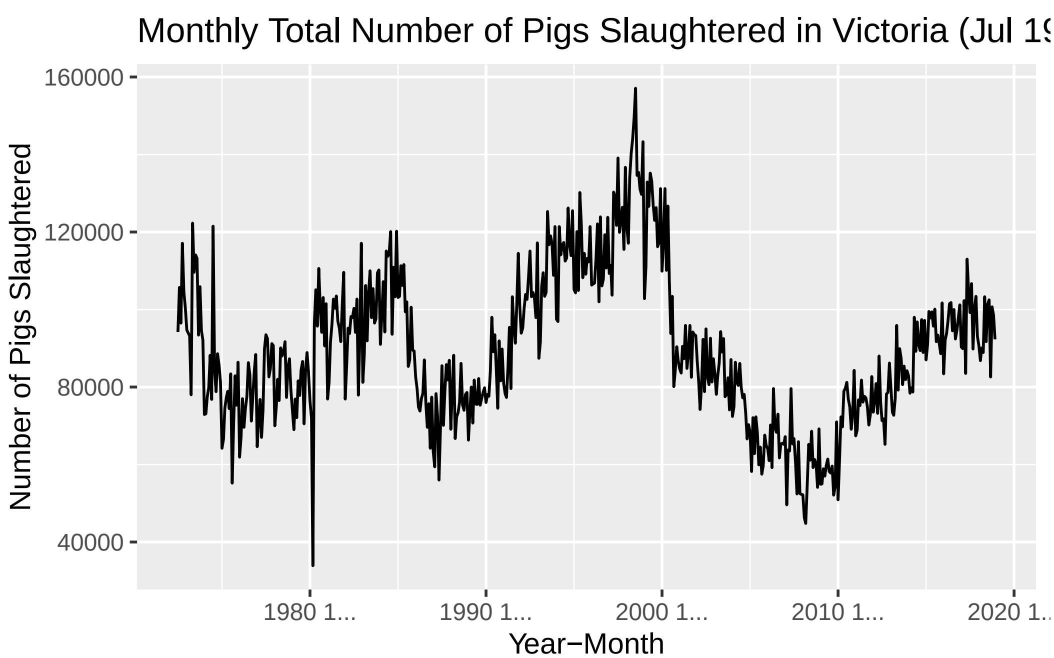
```
# A tsibble: 558 x 4 [1M]
# Key:       Animal, State [1]
       Month Animal State     Count
       <mth> <fct>  <fct>     <dbl>
 1  1972 7  Pigs   Victoria  94200
 2  1972 8  Pigs   Victoria 105700
 3  1972 9  Pigs   Victoria  96500
 4 1972 10  Pigs   Victoria 117100
 5 1972 11  Pigs   Victoria 104600
 6 1972 12  Pigs   Victoria 100500
 7  1973 1  Pigs   Victoria  94700
 8  1973 2  Pigs   Victoria  93900
 9  1973 3  Pigs   Victoria  93200
10  1973 4  Pigs   Victoria  78000
# i 548 more rows
```

```
Slaughtered_Pigs <- aus_livestock |> filter(Animal=='Pigs',State=='Victoria') |>
filter(Month>=yearmonth("1972-07") & Month<=yearmonth("2018-12") )
Slaughtered_Pigs
```

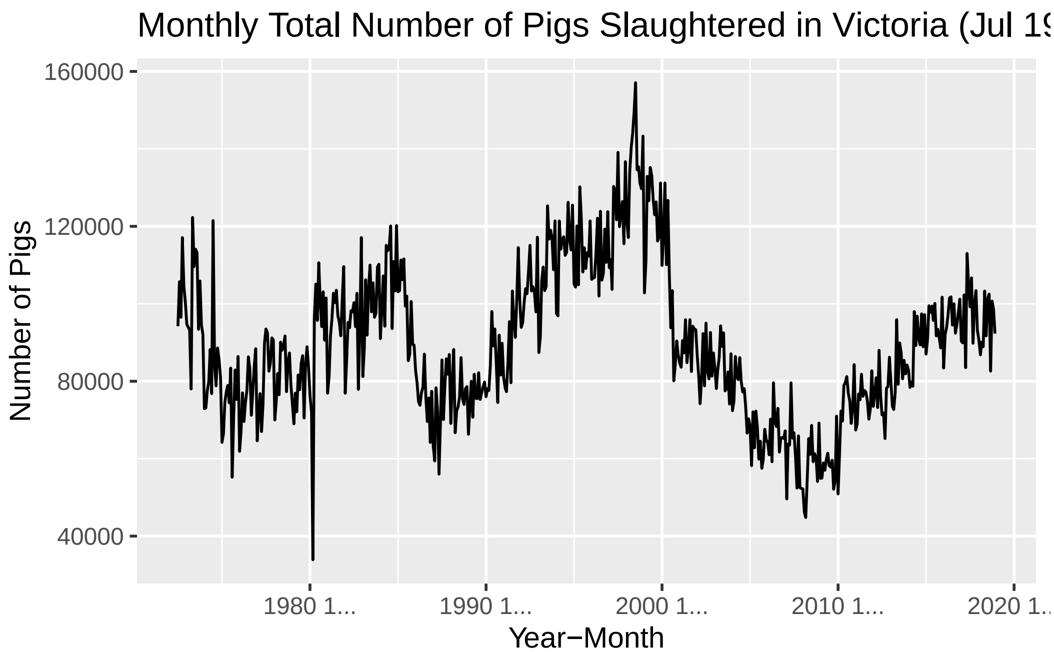
```
# A tsibble: 558 x 4 [1M]
# Key:       Animal, State [1]
       Month Animal State     Count
       <mth> <fct>  <fct>     <dbl>
 1  1972 7  Pigs   Victoria  94200
 2  1972 8  Pigs   Victoria 105700
 3  1972 9  Pigs   Victoria  96500
 4 1972 10  Pigs   Victoria 117100
 5 1972 11  Pigs   Victoria 104600
 6 1972 12  Pigs   Victoria 100500
 7  1973 1  Pigs   Victoria  94700
 8  1973 2  Pigs   Victoria  93900
 9  1973 3  Pigs   Victoria  93200
10  1973 4  Pigs   Victoria  78000
# i 548 more rows
```

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

```r
pigs_vic %>%
  autoplot(Count) +
  labs(title = "Monthly Total Number of Pigs Slaughtered in Victoria (Jul 1972 - Dec 201
       x = "Year-Month",
       y = "Number of Pigs Slaughtered")
```



```
Slaughtered_Pigs |>
ggplot(aes(x = Month, y = Count)) +
geom_line() +
labs(title = "Monthly Total Number of Pigs Slaughtered in Victoria (Jul 1972 - Dec 2018)
x = "Year-Month",
y = "Number of Pigs")
```



Question 3 (Data cleaning)

Inspect the function `process_sgcpi()` located in `_code/clean_data.R`. This function is used to convert the raw Consumer Price Index (CPI) data in `_data/raw/sg-cpi.csv` into a tsibble, stored in `_data/cleaned/sgcpi.rds`.

a) In line 9, what does `skip = 10` and `n_max = 152` do? Why do we need to do this when reading the csv file?

   `skip = 10`: This parameter is used to skip the first 10 lines at the beginning of the CSV file when reading it. This is usually because the first few lines of the file might contain titles, descriptions, or other non-data content. By skipping these lines, you can start reading the actual data rows directly.

   `n_max = 152`: This parameter specifies to read at most 152 rows of data during the reading process. This is very useful for limiting the size of the dataset, especially when you're only interested in a part of the data in the file. It helps avoid loading too much unnecessary data, thereby improving processing efficiency.

b) In line 14, what does `t()` do? Why do we need to do this in order to make a tsibble?

`t()` is the transpose function in R, used to transpose the rows and columns of a matrix or dataframe. In processing time series data, usually, each column in the original data represents a time point, and each row represents different variables or observations.

To convert the data into a time series format (tsibble), the dataframe needs to be transposed so that each row represents a time point and each column represents different variables. This is a common data format in time series analysis and facilitates subsequent processing and analysis.

Question 4 (Beer production)

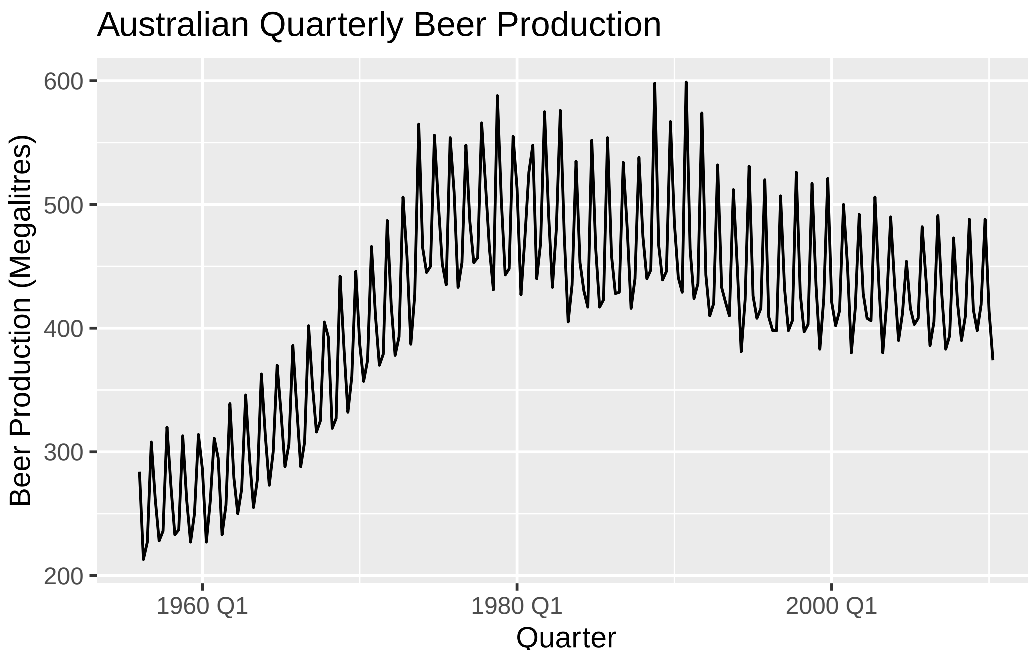
Consider the `aus_production` dataset loaded in the `fpp3` package. We will study the column measuring the production of beer.

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

```
aus_production
```

```
# A tsibble: 218 x 7 [1Q]
   Quarter  Beer Tobacco Bricks Cement Electricity   Gas
     <qtr> <dbl>   <dbl>  <dbl>  <dbl>       <dbl> <dbl>
 1 1956 Q1   284    5225    189    465        3923     5
 2 1956 Q2   213    5178    204    532        4436     6
 3 1956 Q3   227    5297    208    561        4806     7
 4 1956 Q4   308    5681    197    570        4418     6
 5 1957 Q1   262    5577    187    529        4339     5
 6 1957 Q2   228    5651    214    604        4811     7
 7 1957 Q3   236    5317    227    603        5259     7
 8 1957 Q4   320    6152    222    582        4735     6
 9 1958 Q1   272    5758    199    554        4608     5
10 1958 Q2   233    5641    229    620        5196     7
# i 208 more rows
```

```
aus_production |>
autoplot(Beer)+labs(title = "Australian Quarterly Beer Production",
x = "Quarter",
y = "Beer Production (Megalitres)")
```



b) Describe the observed trend.

1. **Trend**: There appears to be an increasing trend in beer production starting from the early years of the series and leveling off in later years. The production increases steadily up to around the 1990s, after which it fluctuates around a more constant level.

2. **Seasonality**: There is a clear seasonal pattern within each year. The production of beer seems to peak in certain quarters regularly, which suggests a seasonal influence on beer production.

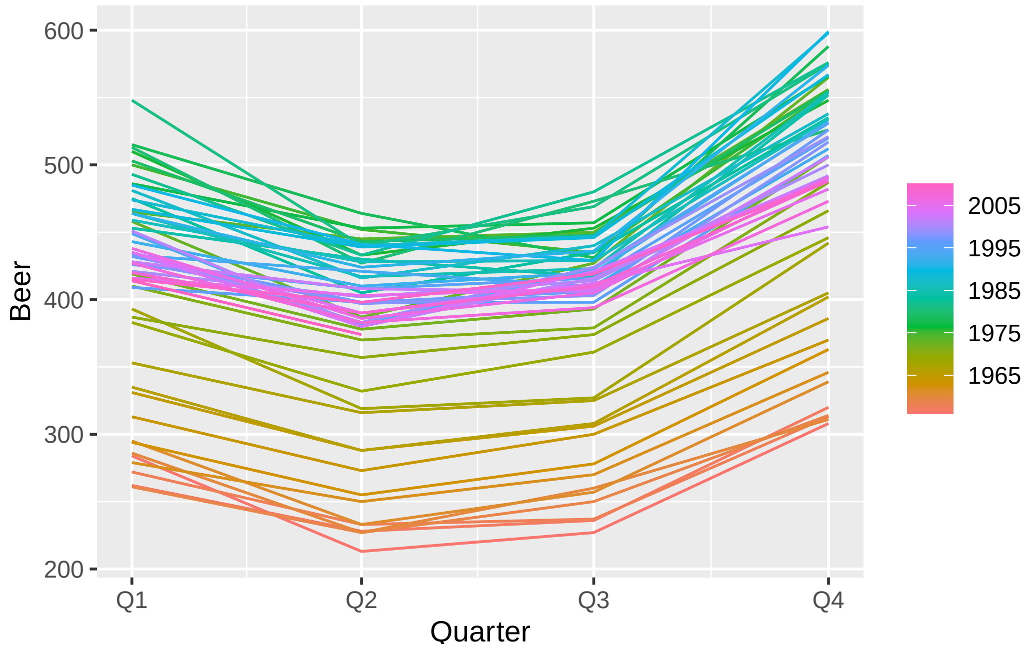
c) Make a seasonal plot.

```
beer_produced <- aus_production |> select(Quarter,Beer) |> as_tsibble(index=Quarter)
beer_produced
```

```
# A tsibble: 218 x 2 [1Q]
   Quarter  Beer
     <qtr> <dbl>
 1 1956 Q1   284
 2 1956 Q2   213
 3 1956 Q3   227
 4 1956 Q4   308
 5 1957 Q1   262
 6 1957 Q2   228
 7 1957 Q3   236
```

```
 8 1957 Q4   320
 9 1958 Q1   272
10 1958 Q2   233
# i 208 more rows
```

```r
gg_season(beer_produced,y=Beer)
```



d) What is the period of the seasonality?

The periodic nature of the seasonality is likely quarterly, as the data is quarterly. This would suggest that the period of the seasonality is one year, with the pattern repeating every four quarters.

e) Describe the seasonal behavior.

There are clear and consistent seasonal patterns within each year. Beer production peaks in certain quarters and troughs in others. Specifically, it appears that there is a peak in the later quarters of the year, which might correspond to increased beer production in anticipation of the summer season in the holiday period.

Question 5 (Pelts)

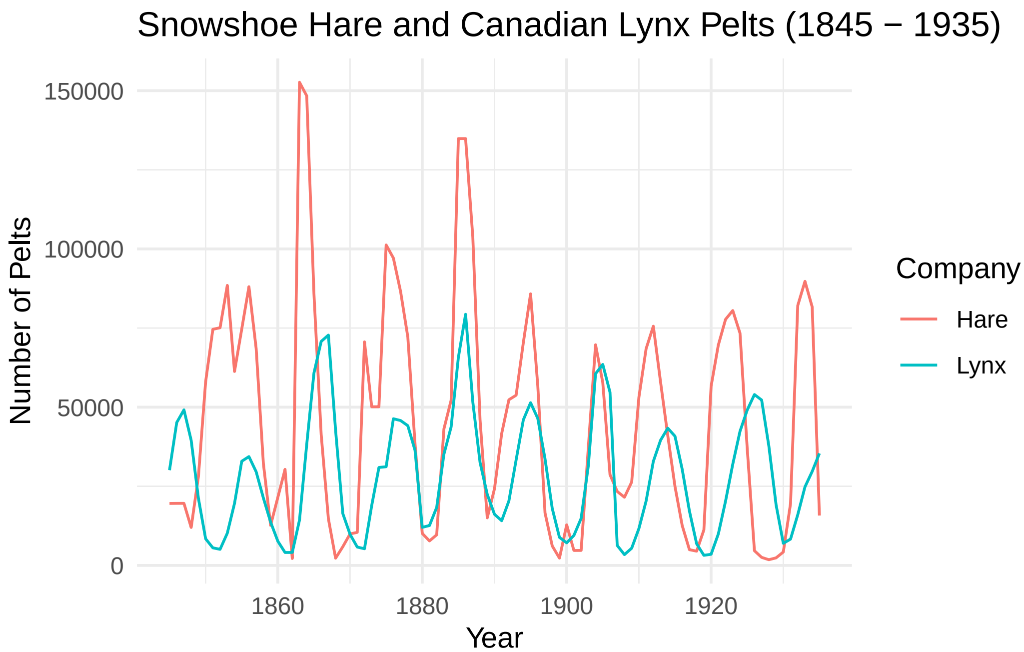
Consider the `pelt` dataset loaded in the `fpp3` package, which measures the Hudson Bay Company trading records for Snowshoe Hare and Canadian Lynx furs from 1845 to 1935.

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

```
pelt_long <- pelt|>
  pivot_longer(cols = c('Hare', 'Lynx'), names_to = "Company", values_to = "Count")
 pelt_long
```

```
# A tsibble: 182 x 3 [1Y]
# Key:       Company [2]
    Year Company Count
   <dbl> <chr>   <dbl>
 1  1845 Hare    19580
 2  1845 Lynx    30090
 3  1846 Hare    19600
 4  1846 Lynx    45150
 5  1847 Hare    19610
 6  1847 Lynx    49150
 7  1848 Hare    11990
 8  1848 Lynx    39520
 9  1849 Hare    28040
10  1849 Lynx    21230
# i 172 more rows
```

```
pelt_long |>
ggplot(aes(x = Year, y = Count, color = Company)) +
geom_line() +
labs(title = "Snowshoe Hare and Canadian Lynx Pelts (1845 - 1935)",
x = "Year",
y = "Number of Pelts") +
theme_minimal()
```

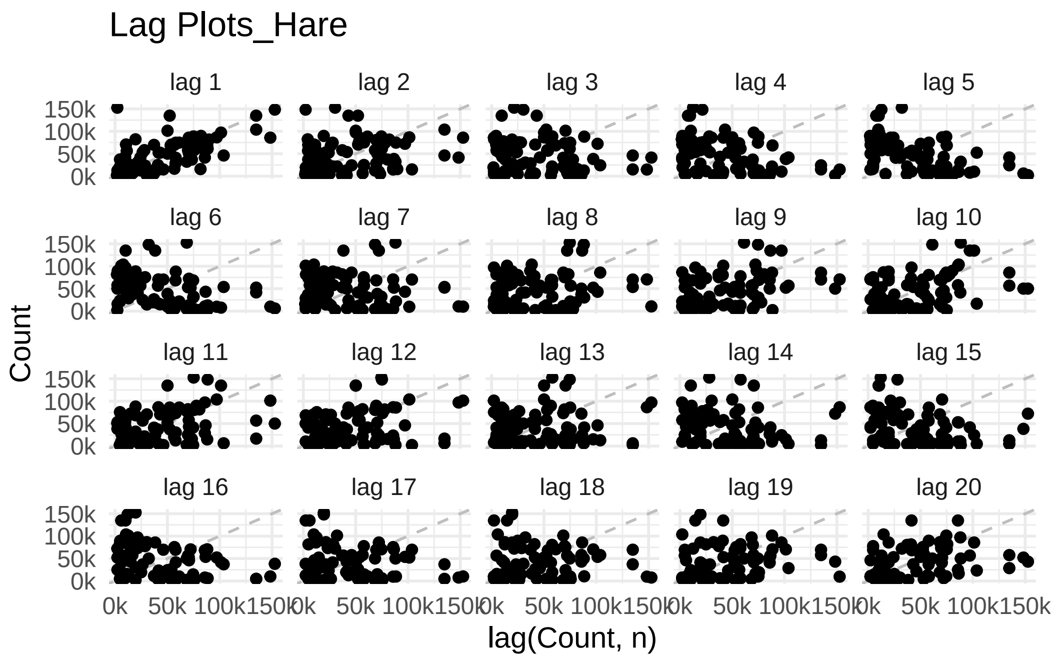


b) What happens when you try to use `gg_season()` to the lynx fur time series? What is producing the error?

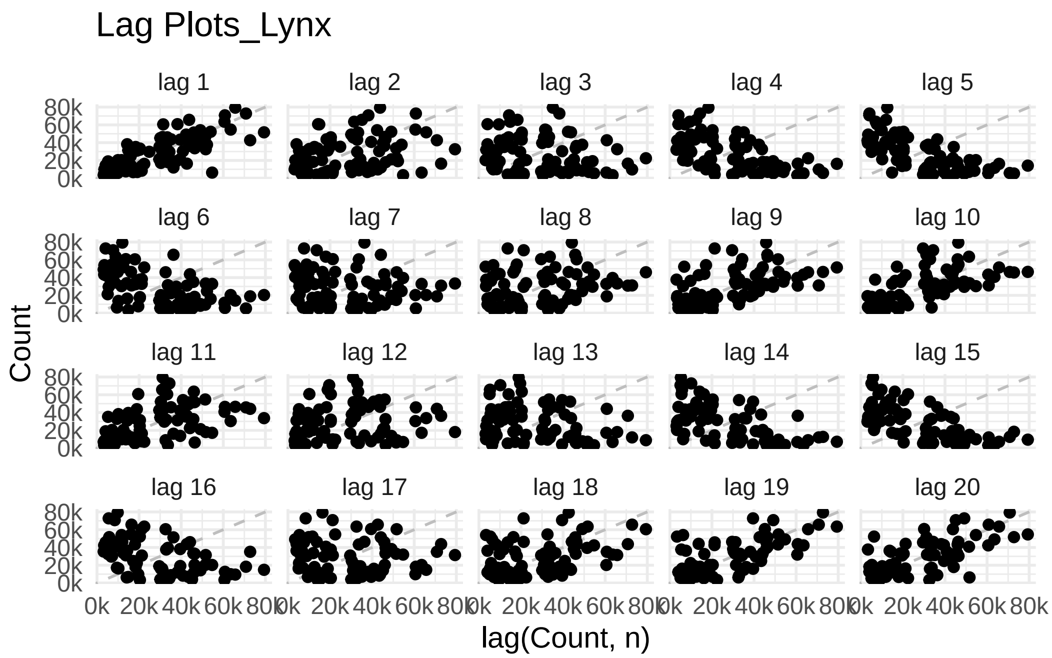
It said Error in gg_season(pelt,y=lynx), the data must contain at least one observation per seasonal period. Though gg_season() can estimate the period itself and pelt is a tsibble dataset, it looks like lynx doesn't have so called seasonality, at least with time unit–year.

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_long|>filter(Company=="Hare")%>%
gg_lag(y=Count,"geom"="point",lags=1:20) +
scale_x_continuous(labels = function(x) paste0(x / 1000, "k")) +
scale_y_continuous(labels = function(x) paste0(x / 1000, "k"))+
labs(title = "Lag Plots_Hare")+
theme_minimal()
```




```
pelt_long|>filter(Company=="Lynx")%>%
gg_lag(y=Count,"geom"="point",lags=1:20) +
scale_x_continuous(labels = function(x) paste0(x / 1000, "k")) +
scale_y_continuous(labels = function(x) paste0(x / 1000, "k"))+
labs(title = "Lag Plots_Lynx")+
theme_minimal()
```



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

- **Strong Positive Correlation**:
  Appears at lags where the points cluster along a line running from the bottom left to the top right.
  This is particularly visible at lags like 1, 2, and possibly 11 and 12.

- **Strong Negative Correlation**:
  This is indicated by points clustering along a line running from the top left to the bottom right. This pattern is less pronounced in the Lynx series but might be suggested at lags like 4 and 5.

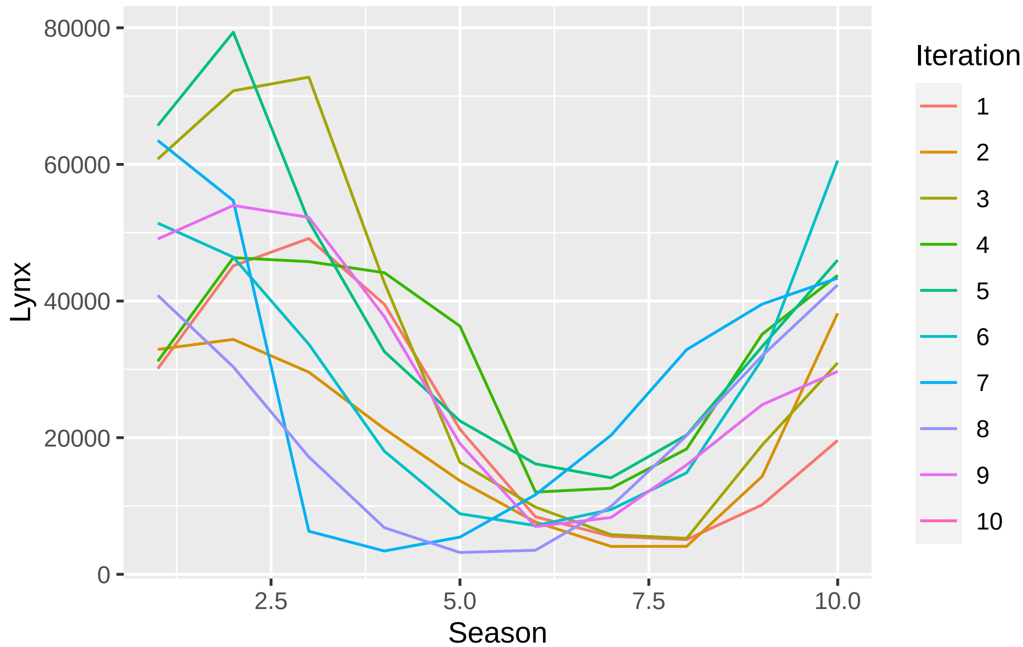
e) Use the provided function ‘`gg_custom_season()` in `_code/plot_util.R`[1] to make a seasonal plot for lynx furs with the period that you guessed.[2] Does the resulting plot suggest seasonality? Why or why not?

```
gg_custom_season <- function(data, y, period, start = 1) {
  # Make a seasonal plot with period specified in integer
  # start argument specifies the row number that will be the first season
  # in the period
  y <- enquo(y)
  data |>
    mutate(Season = (row_number() - start) %% period + start,
```

[1]You can load this function using `source("../_code/plot.util.R")`.
[2]Unfortunately, it seems ‘`gg_season()` does not allow this functionality.

```
            Iteration = as.factor((row_number() - start) %/% period + 1)) |>
    ggplot(aes(x = Season, y = !!y, color = Iteration)) +
    geom_line()
}
```

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



To determine if the plot suggests seasonality:

1. **Consistency**: We look for a consistent pattern that repeats every cycle. Seasonality is suggested if the same pattern of movement (e.g., peaks and troughs) appears at the same point in each cycle.

2. **Pattern**: The plot should show regularity in the data points, meaning the values for each season (or point in the cycle) should follow a predictable pattern.

From the image:

- There is a **variation** in the lynx counts, indicating there are changes over time.
- However, the lines representing different iterations do not seem to follow a **consistent pattern**. The peaks and troughs do not align in a way that suggests a clear, predictable cycle.

Given the irregularity and lack of a clear repeating pattern, the plot does not strongly suggest seasonality, at least not with a 10-year cycle. Instead, it might suggest other forms of cyclical behavior or external factors affecting the population dynamics of lynx.