Week 3 homework 1

Alex Parra

11/6/2022

```r
library(tsibble)
```

```
## Warning: package 'tsibble' was built under R version 4.1.3

##
## Attaching package: 'tsibble'

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, union
```

```r
library(fpp3)
```

```
## Warning: package 'fpp3' was built under R version 4.1.3

## -- Attaching packages -------------------------------------------- fpp3 0.4.0 --

## v tibble      3.1.6     v ggplot2     3.3.5
## v dplyr       1.0.7     v tsibbledata 0.4.0
## v tidyr       1.2.0     v feasts      0.2.2
## v lubridate   1.8.0     v fable       0.3.1

## Warning: package 'tsibbledata' was built under R version 4.1.3

## Warning: package 'feasts' was built under R version 4.1.3

## Warning: package 'fabletools' was built under R version 4.1.3

## Warning: package 'fable' was built under R version 4.1.3

## -- Conflicts -------------------------------------------------- fpp3_conflicts --
## x lubridate::date()     masks base::date()
## x dplyr::filter()       masks stats::filter()
## x tsibble::intersect()  masks base::intersect()
## x lubridate::interval() masks tsibble::interval()
## x dplyr::lag()          masks stats::lag()
## x tsibble::setdiff()    masks base::setdiff()
## x tsibble::union()      masks base::union()
```

```r
library(seasonal)
```

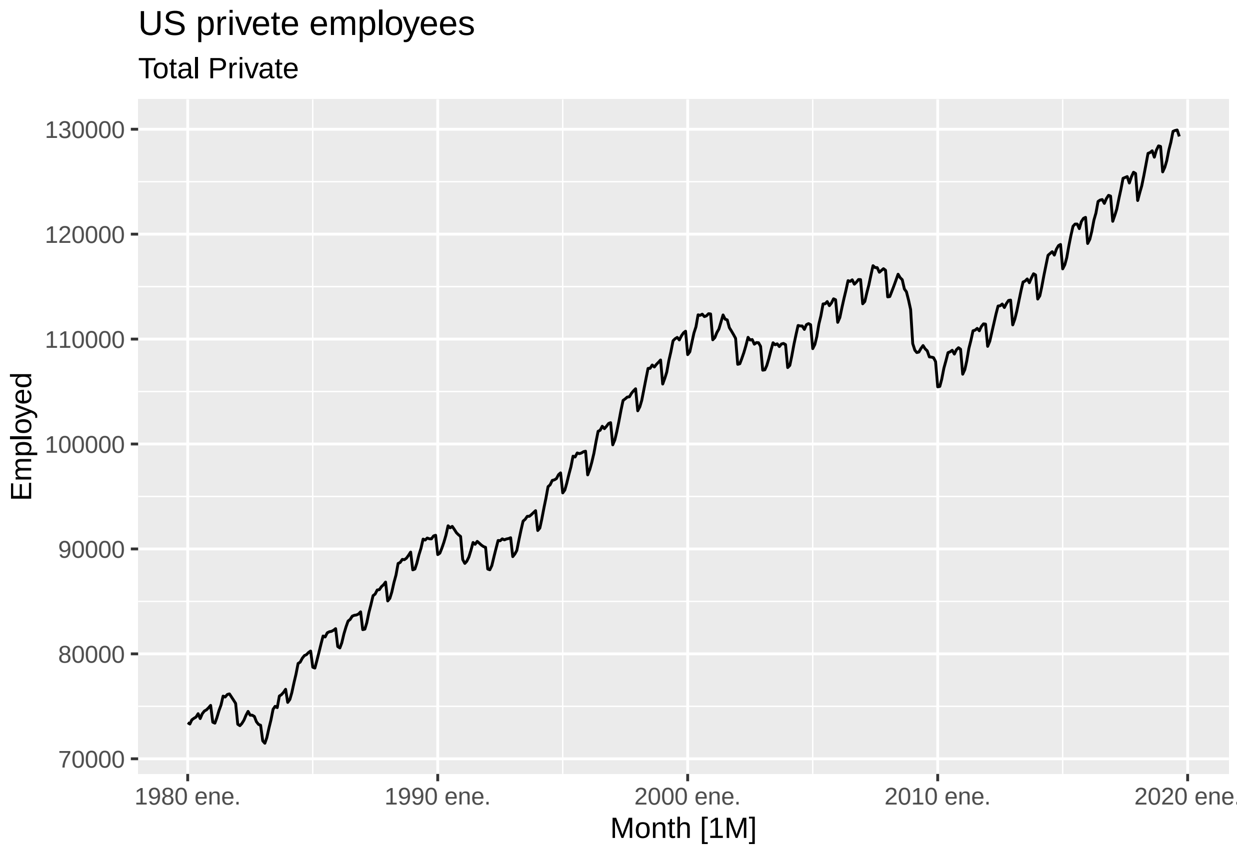
```
## Warning: package 'seasonal' was built under R version 4.1.3

##
## Attaching package: 'seasonal'

## The following object is masked from 'package:tibble':
##
##     view
```

```r
library(glue)
library(GGally)
```

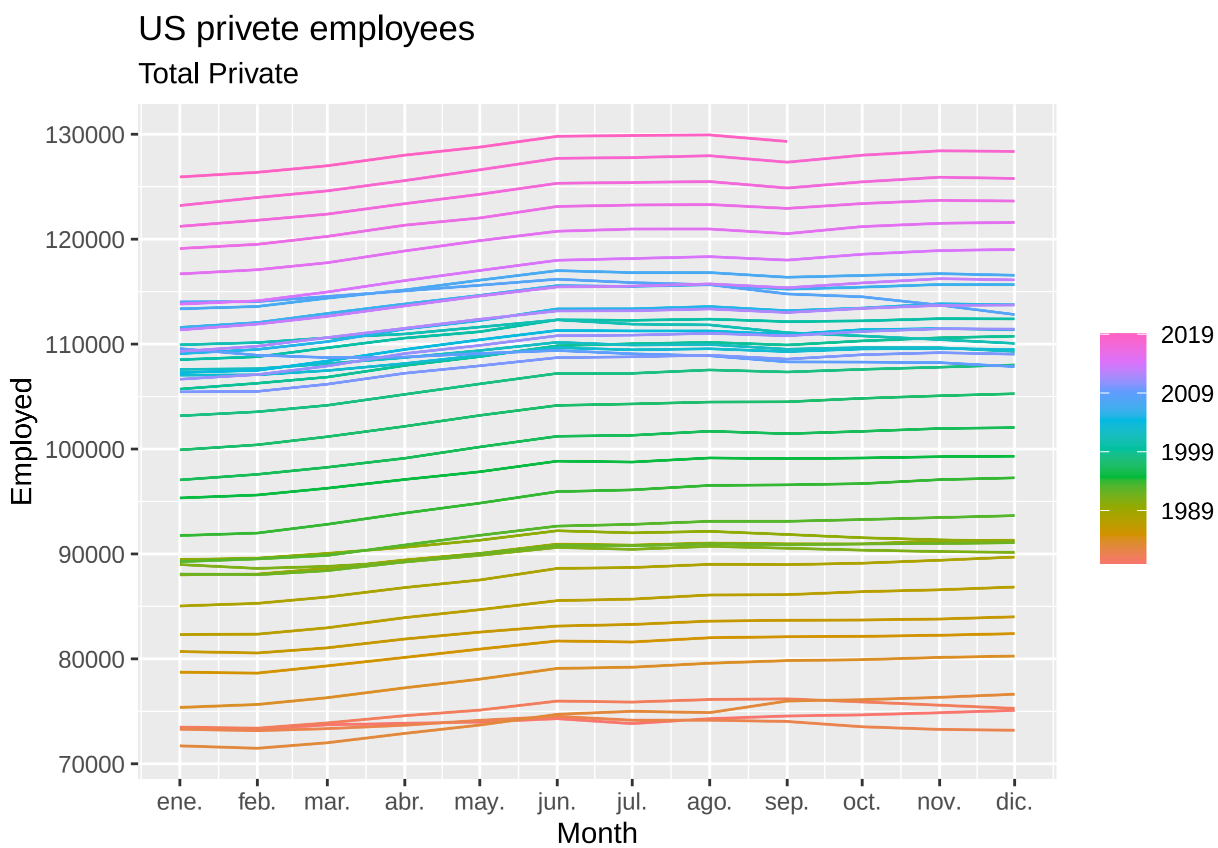
```
## Warning: package 'GGally' was built under R version 4.1.3

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

2.10 exercise 9

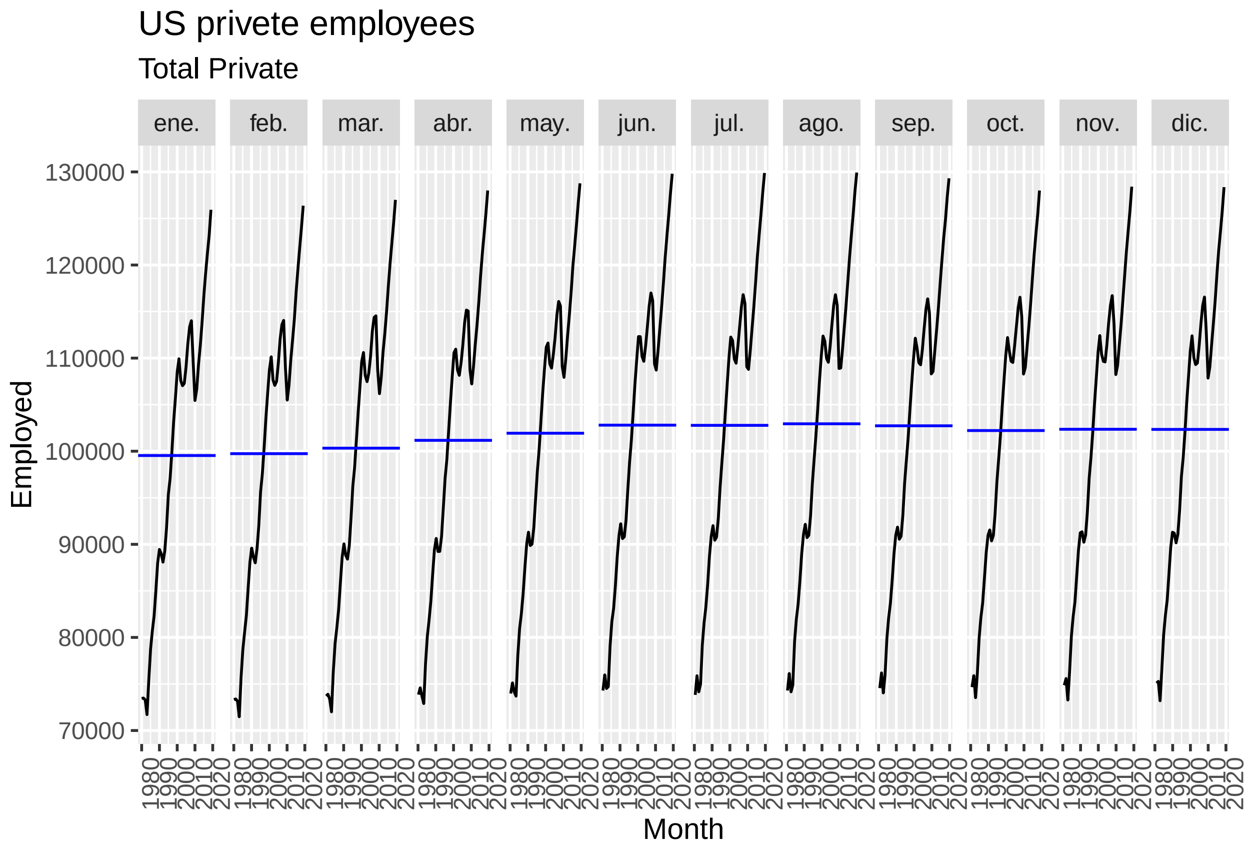
```r
# "Total Private" Employed from us_employment
employed_total_private  <- us_employment  %>%
  filter(Title == "Total Private" & year(Month) >= 1980) %>%
  select(Month, Employed)

autoplot(employed_total_private, Employed) +
  labs(title = "US privete employees",
       subtitle = "Total Private")
```



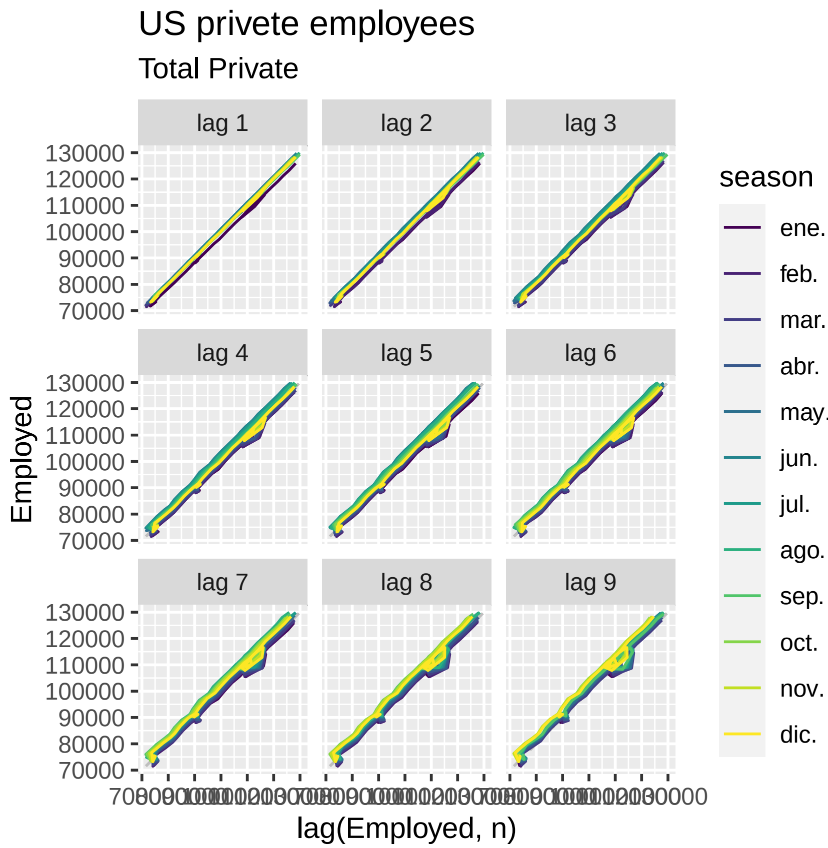

```
gg_season(employed_total_private, Employed) +
  labs(title = "US privete employees",
       subtitle = "Total Private")
```




```
gg_subseries(employed_total_private, Employed) +
  labs(title = "US privete employees",
       subtitle = "Total Private")
```



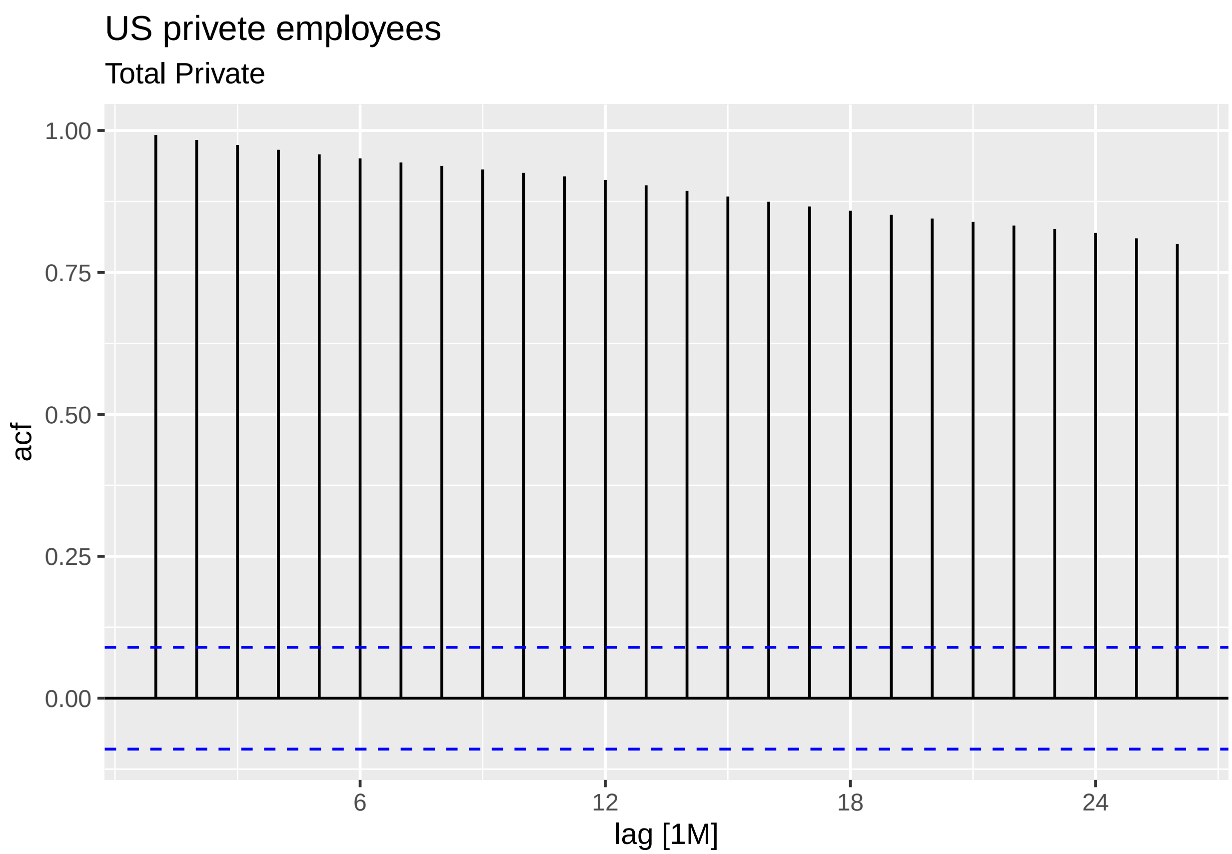

```r
gg_lag(employed_total_private, Employed) +
  labs(title = "US privete employees",
       subtitle = "Total Private")
```




```r
employed_total_private %>% ACF(Employed, lag_max = 9)
```

```
## # A tsibble: 9 x 2 [1M]
##      lag   acf
##    <lag> <dbl>
## 1     1M 0.992
## 2     2M 0.983
## 3     3M 0.974
## 4     4M 0.966
## 5     5M 0.958
## 6     6M 0.951
## 7     7M 0.944
## 8     8M 0.938
## 9     9M 0.932
```

```r
employed_total_private %>%
  ACF(Employed) %>%
  autoplot() +
  labs(title="US privete employees",
       subtitle = "Total Private")
```

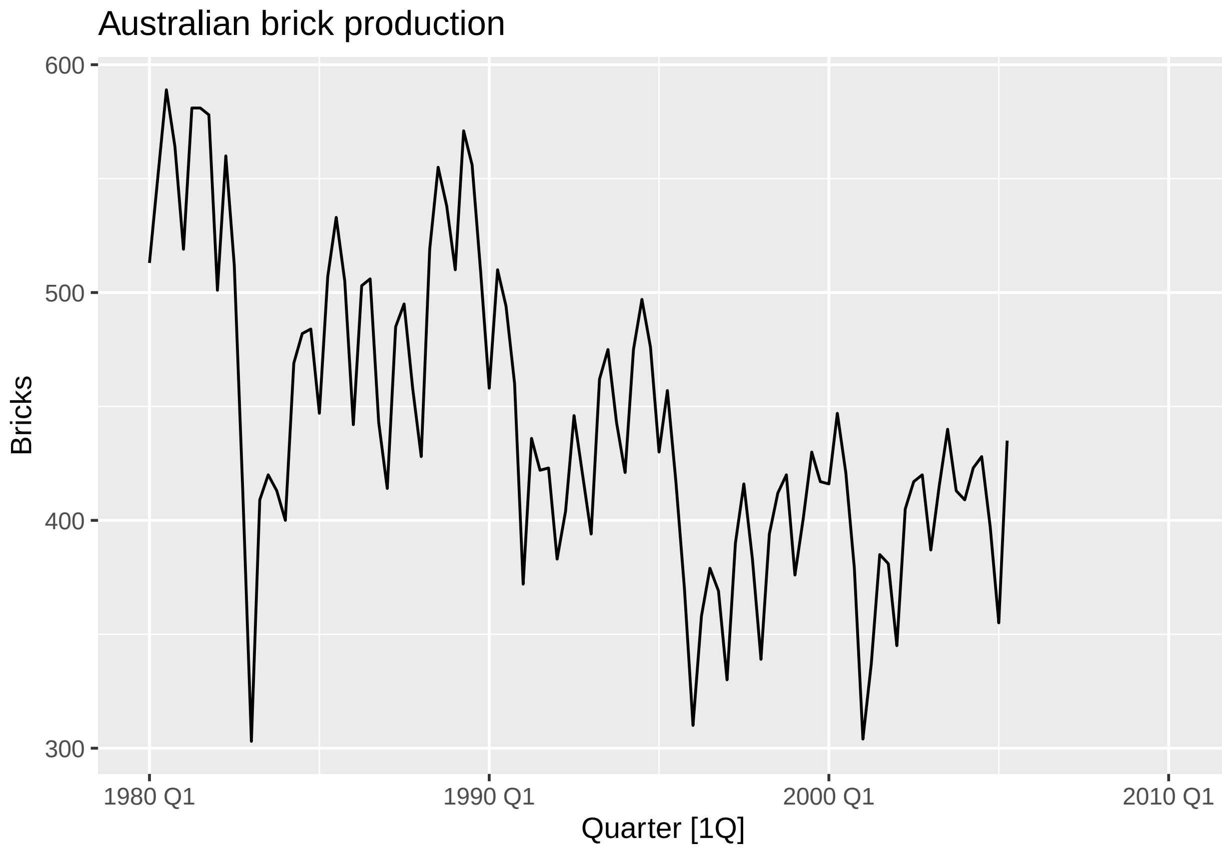


We can see that there is seasonality on the data, the number of employees start as a lower value in January a slowly grow over the month until it gets a peek in the summer month, and then it starts decreasing again. We can see that there is no cyclicity on the data, as there are no longer cycles. The trend is increasing over time, except for the 2007-2008 crises where the employment decrease.

We see an increasing series that grow over time as the country grows, there is an important seasonal pattern over the months, where the employment grows during the first half of the year and decreases over the second half. There are a couple of unusual years during the early 80 decade, the early 90 decade, the early and late 2020 decade, where employment decrease, as it was periods of economic crisis.

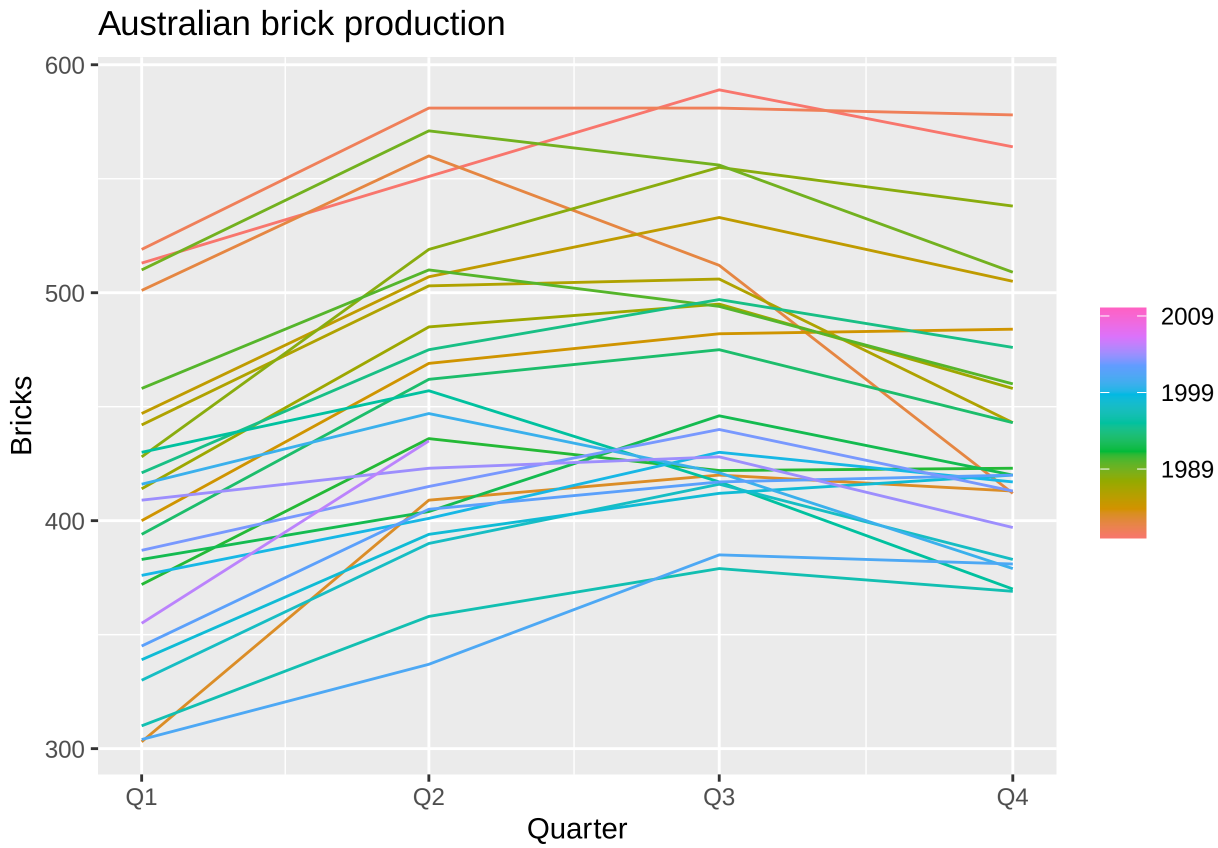
```
# Bricks from aus_production
bricks_aus_production   <- aus_production  %>%
  filter(year(Quarter) >= 1980) %>% #& Title == "Total Private"
  select(Quarter, Bricks)

autoplot(bricks_aus_production, Bricks) +
  labs(title = "Australian brick production")
```

```
## Warning: Removed 20 row(s) containing missing values (geom_path).
```



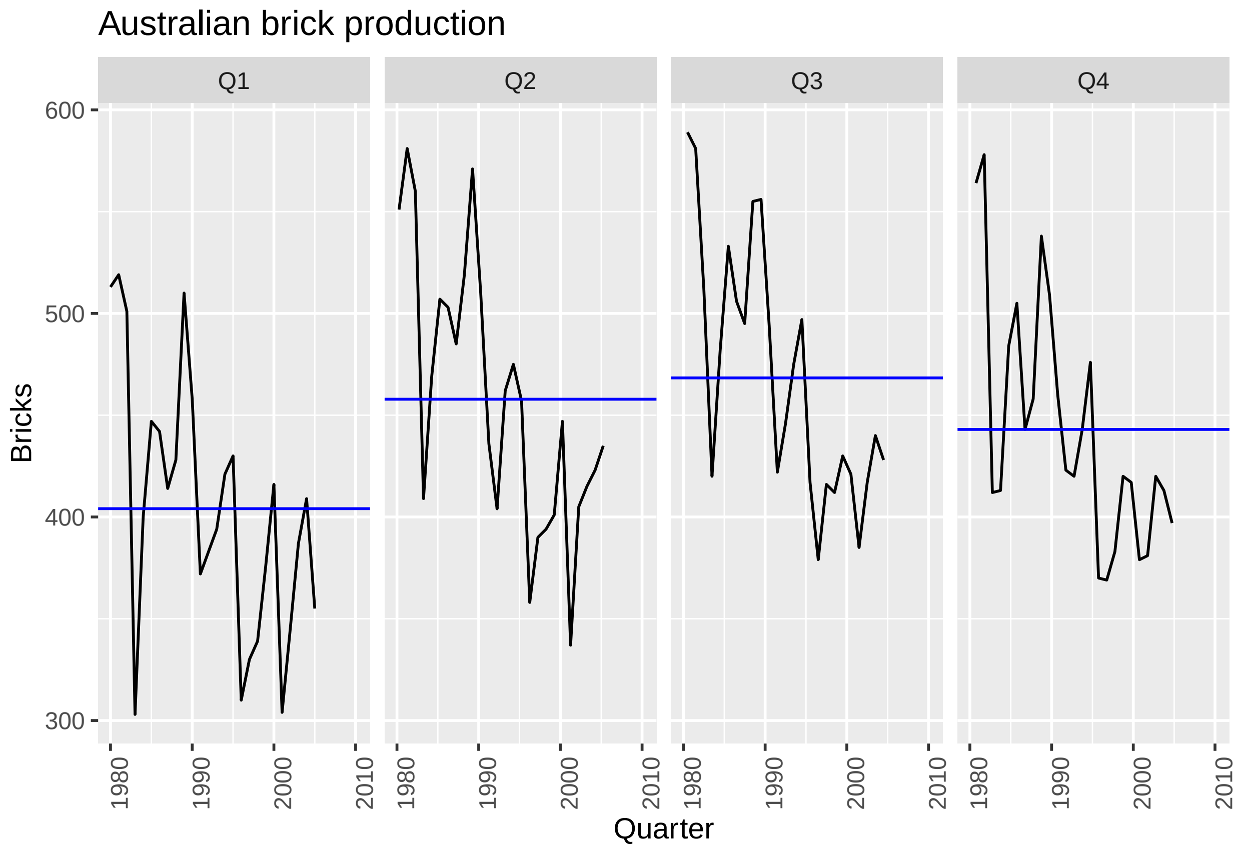

```
gg_season(bricks_aus_production, Bricks) +
  labs(title = "Australian brick production")
```

```
## Warning: Removed 20 row(s) containing missing values (geom_path).
```



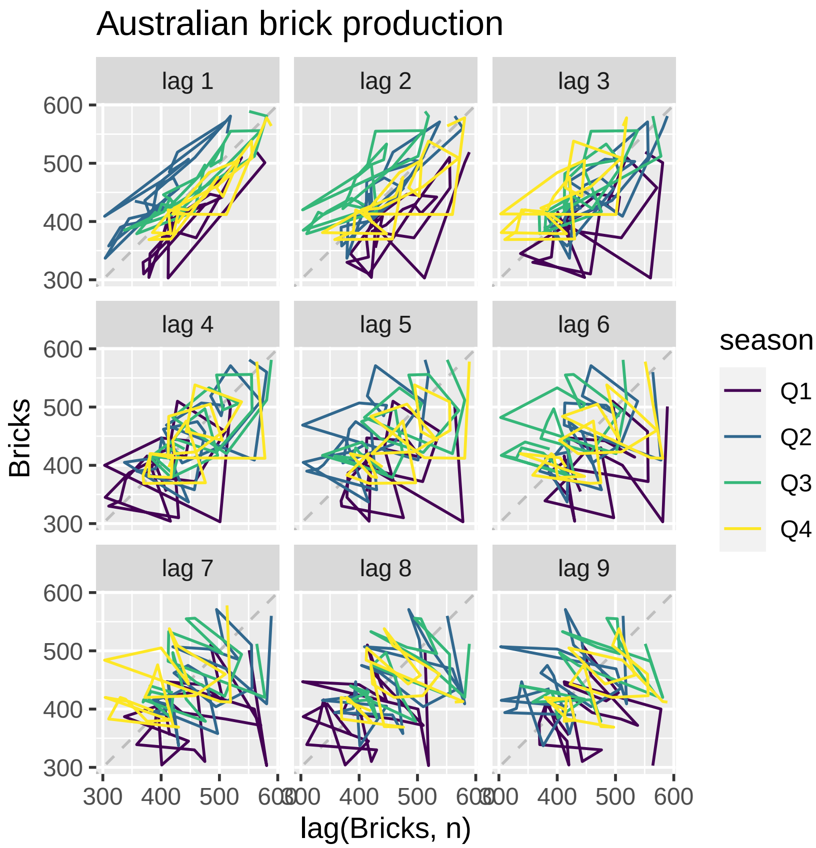

```
gg_subseries(bricks_aus_production, Bricks) +
  labs(title = "Australian brick production")
```

```
## Warning: Removed 5 row(s) containing missing values (geom_path).
```




```
gg_lag(bricks_aus_production, Bricks) +
  labs(title = "Australian brick production")
```

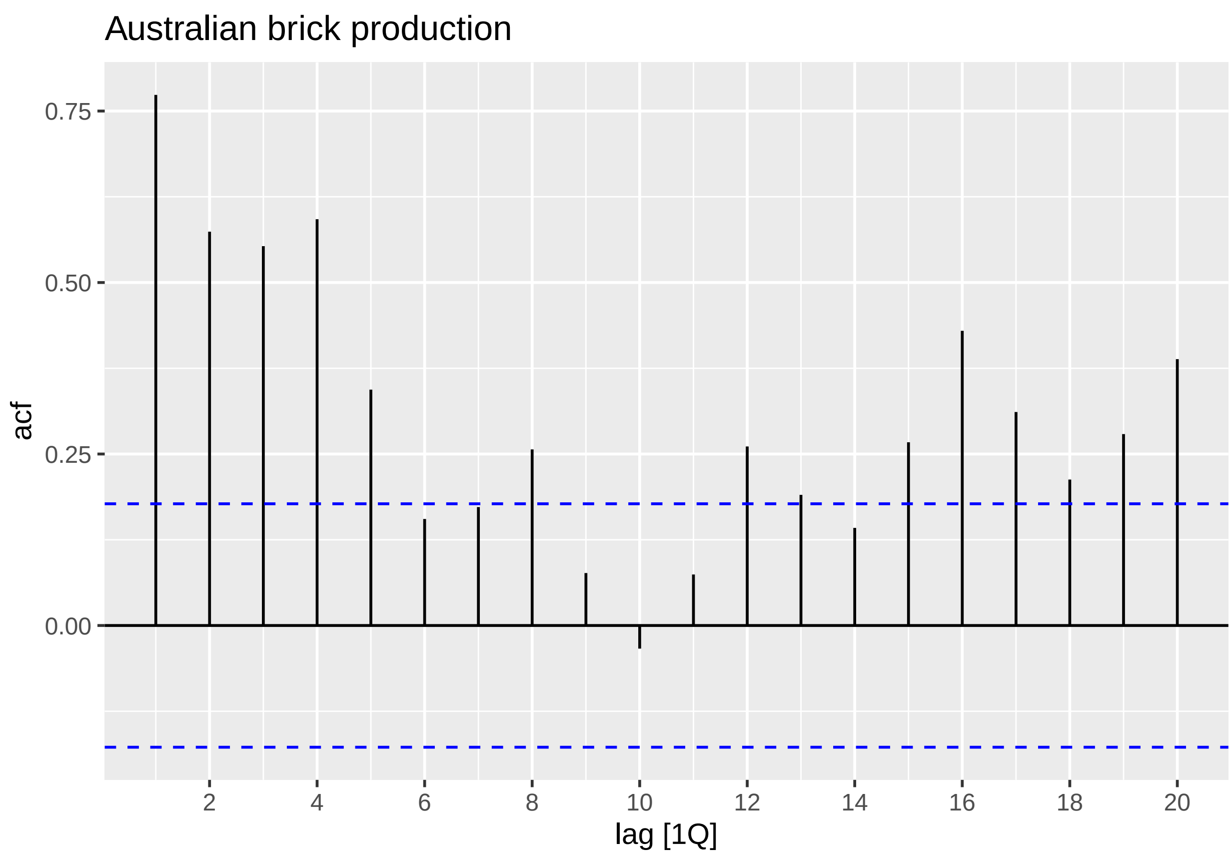
```
## Warning: Removed 20 rows containing missing values (gg_lag).
```




```
bricks_aus_production %>% ACF(Bricks, lag_max = 9)
```

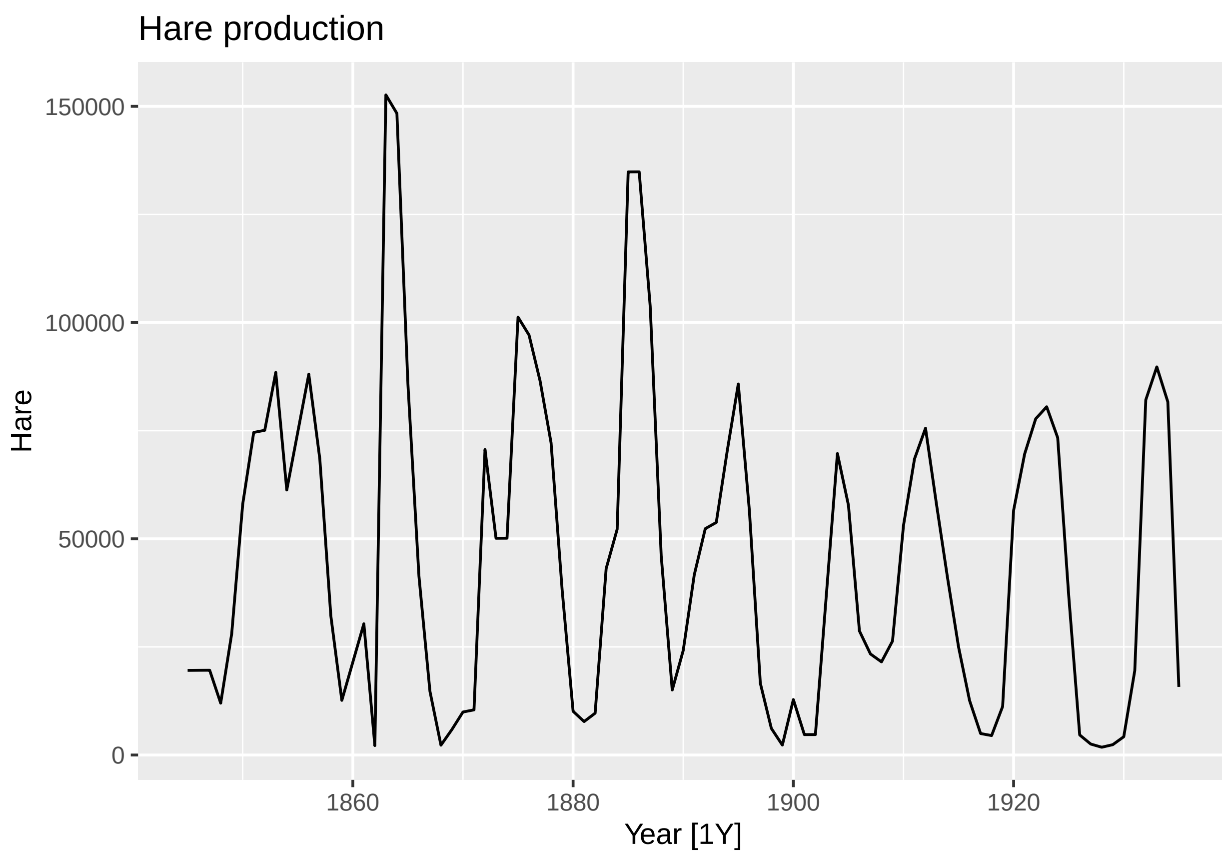
```
## # A tsibble: 9 x 2 [1Q]
##      lag    acf
##    <lag>  <dbl>
## 1     1Q 0.774
## 2     2Q 0.574
## 3     3Q 0.553
## 4     4Q 0.592
## 5     5Q 0.344
## 6     6Q 0.155
## 7     7Q 0.173
## 8     8Q 0.257
## 9     9Q 0.0765
```

```
bricks_aus_production %>%
  ACF(Bricks) %>%
  autoplot() +
  labs(title="Australian brick production")
```

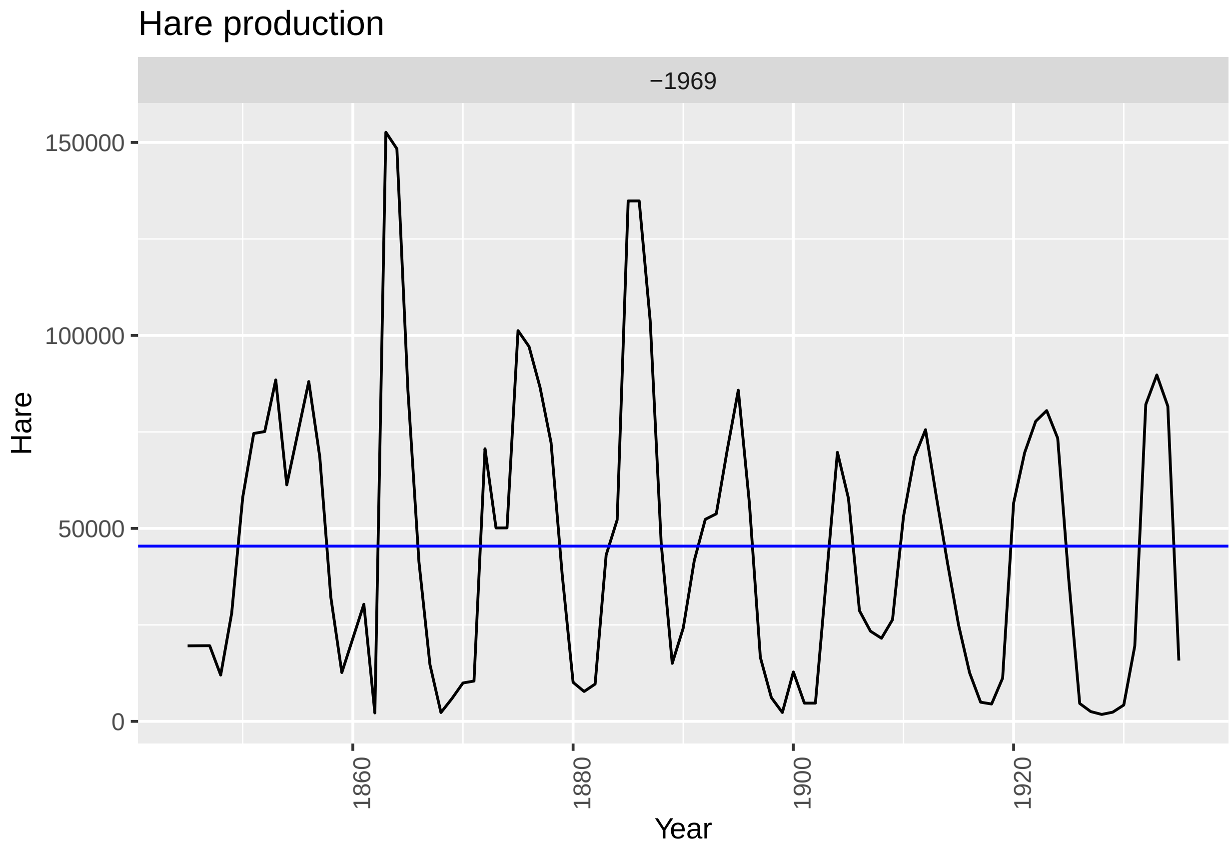


We can see that there is also seasonality, as the bricks production increases over the quarters, and start decreasing in the last quarter. We can see that there is a decreasing trend in the data. From the series we can see that the bricks production is greater over the middle of the year with respect to the start and the end. We can see that there is an unusual yar that the first half of the 80's, as the production dropped to 300, from the 450-550 value it is supposed to be in.

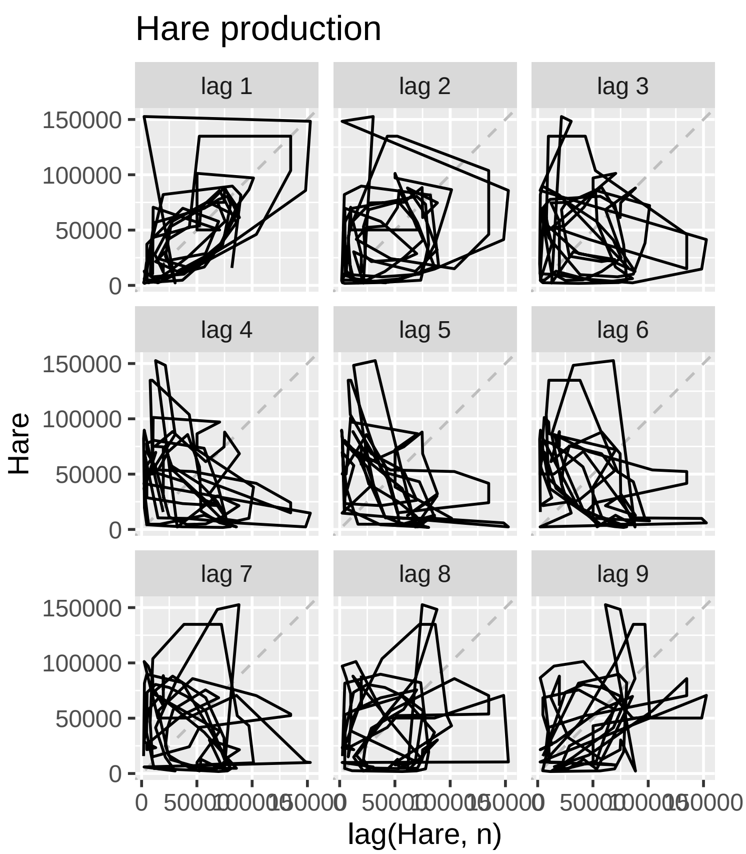
```
# Hare from pelt
hare_production   <- pelt  %>%
  #filter(Year >= 1980) %>%
  select(Year, Hare)

autoplot(hare_production, Hare) +
  labs(title = "Hare production")
```




```r
#gg_season(hare_production, Hare) +
#  labs(title = "Hare production")

gg_subseries(hare_production, Hare) +
  labs(title = "Hare production")
```



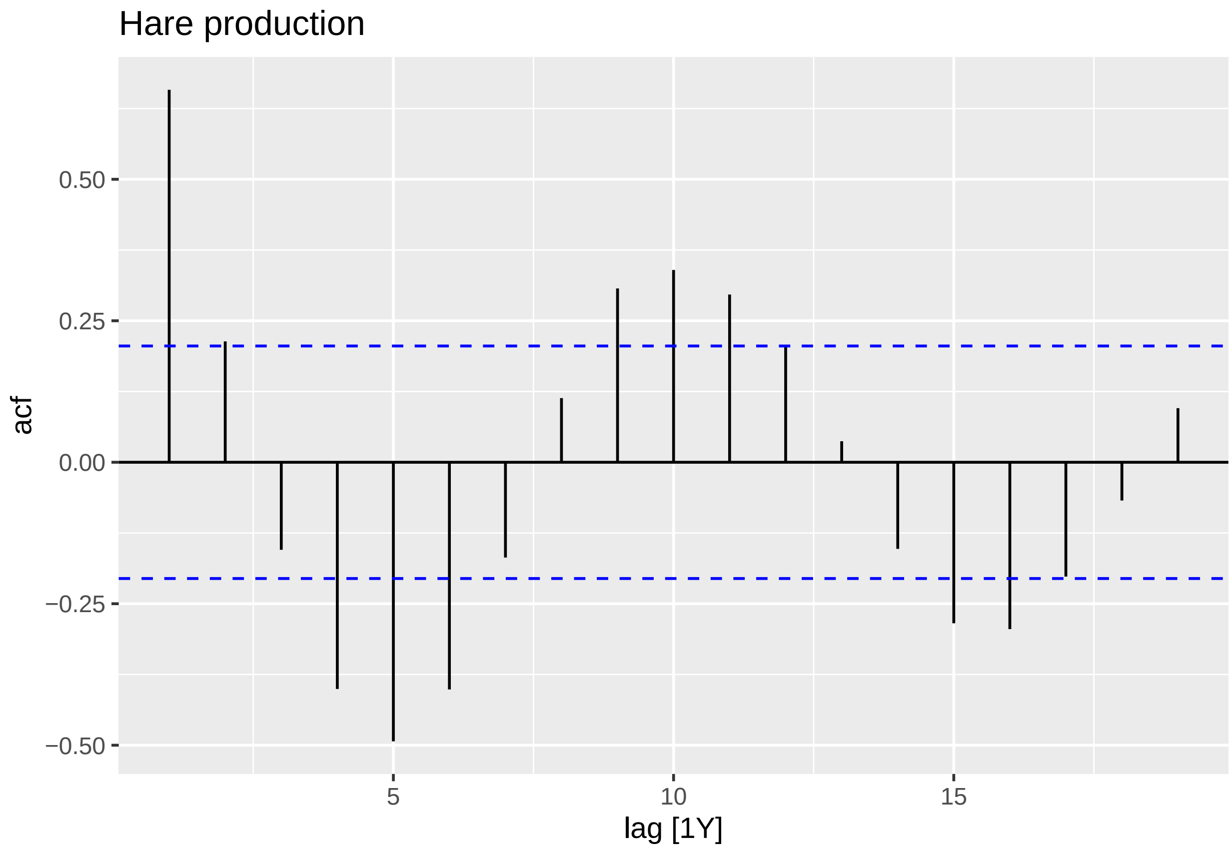

```
gg_lag(hare_production, Hare) +
  labs(title = "Hare production")
```




```
hare_production %>% ACF(Hare, lag_max = 9)
```

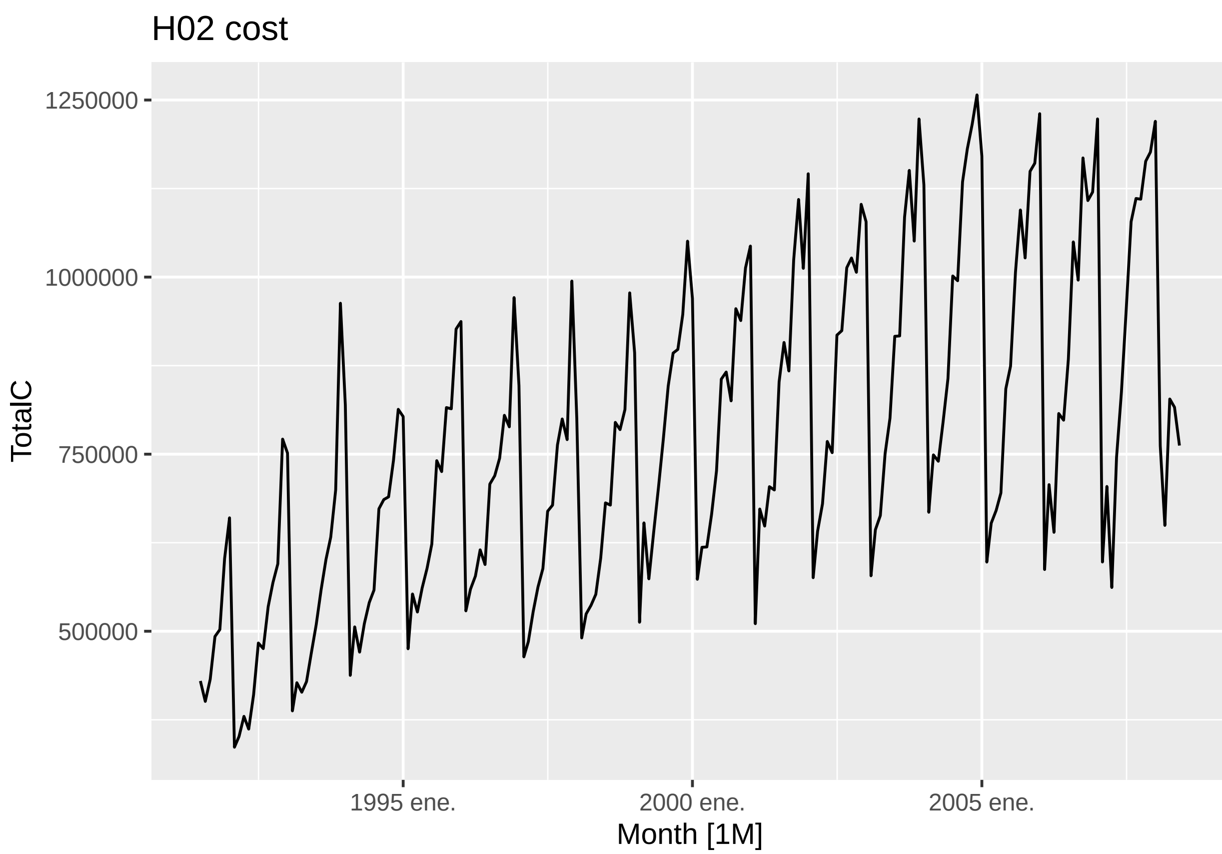
```
## # A tsibble: 9 x 2 [1Y]
##      lag    acf
##    <lag>  <dbl>
## 1     1Y  0.658
## 2     2Y  0.214
## 3     3Y -0.155
## 4     4Y -0.401
## 5     5Y -0.493
## 6     6Y -0.401
## 7     7Y -0.168
## 8     8Y  0.113
## 9     9Y  0.307
```

```
hare_production %>%
  ACF(Hare) %>%
  autoplot() +
  labs(title="Hare production")
```

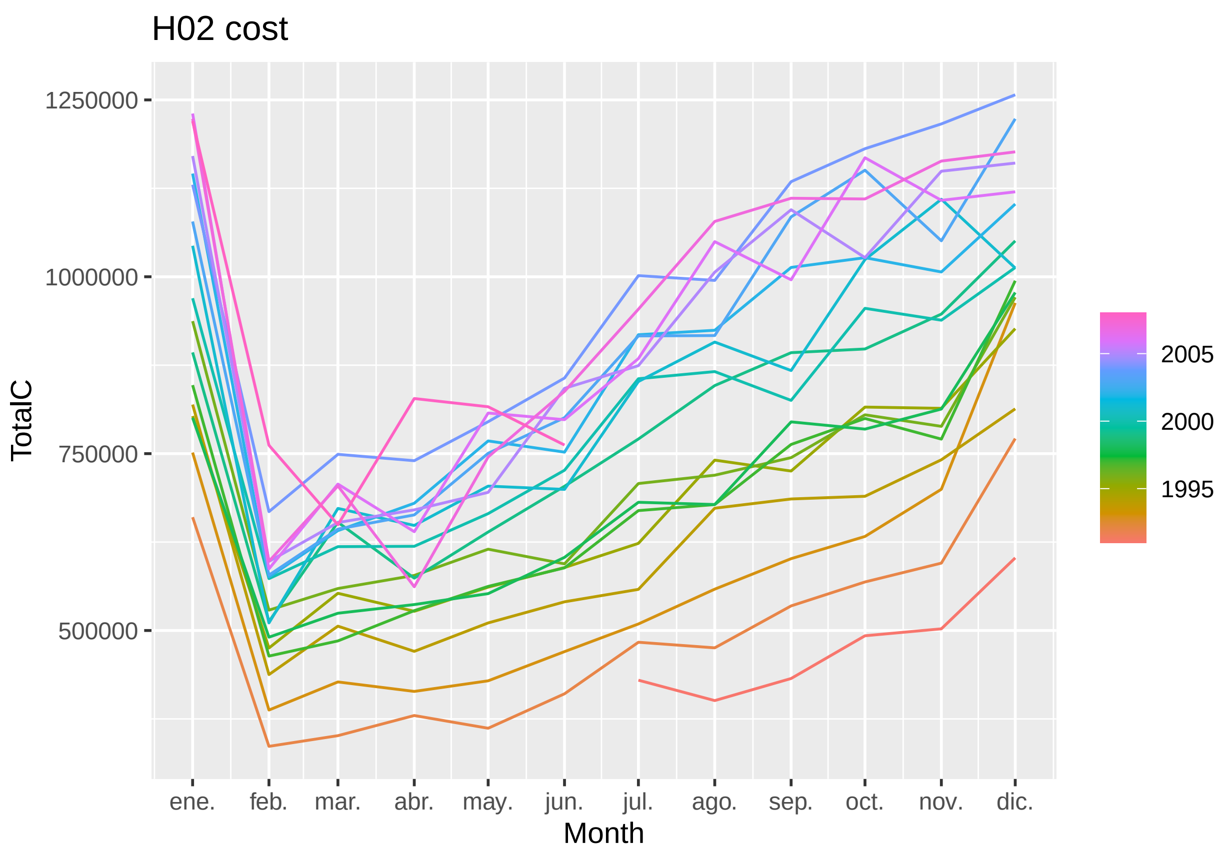


In this series, we only have access to yearly data, so we cannot test if there is any seasonality. Although we can see that there is cyclicity on the data, as the values grows and falls in a period of 5 years, there is also no trend during this period. The only remarkable year is in the early 60's where the data experience the highest value in all the time series.

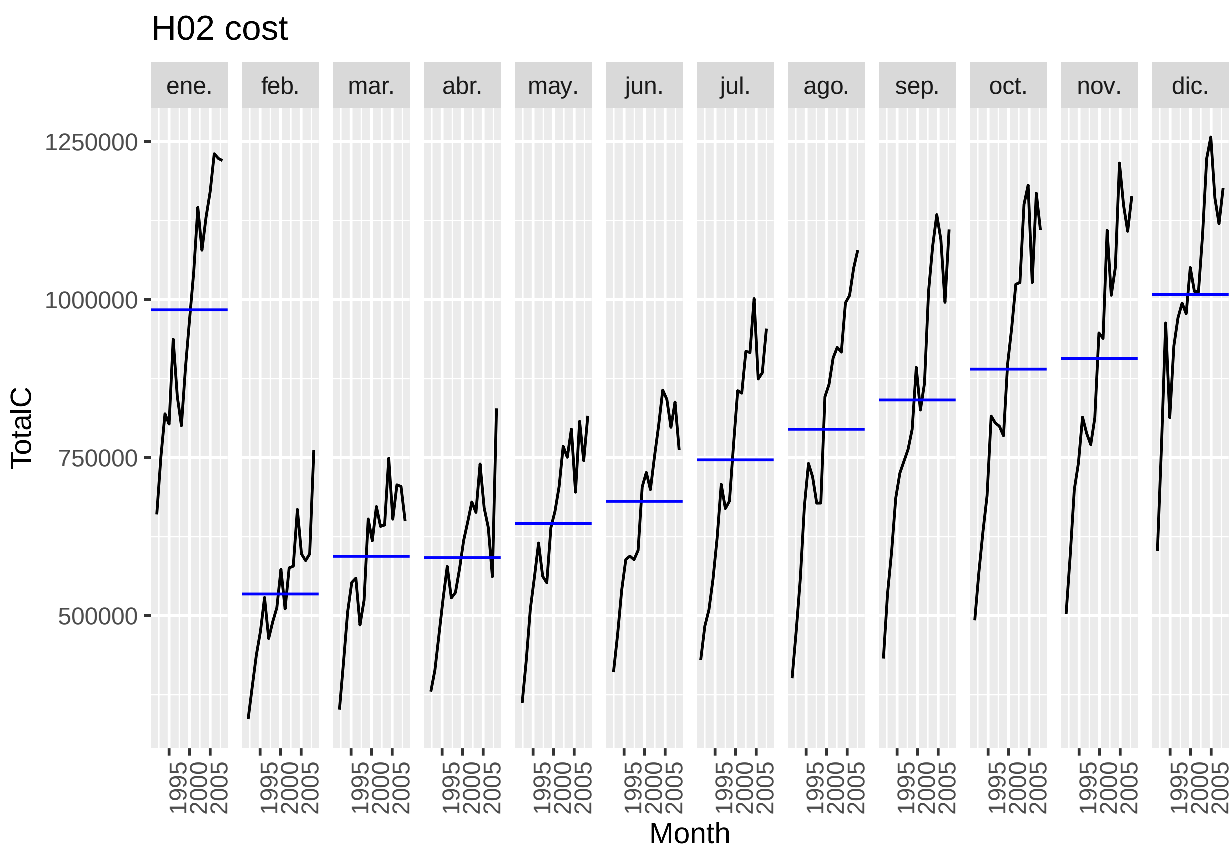
```
# "H02" Cost from PBS
h02_cost   <- PBS  %>%
  filter(ATC2 == "H02") %>% #Year >= 1980
  select(Month, Cost)  %>%
  summarise(TotalC = sum(Cost))

autoplot(h02_cost, TotalC) +
  labs(title = "H02 cost")
```



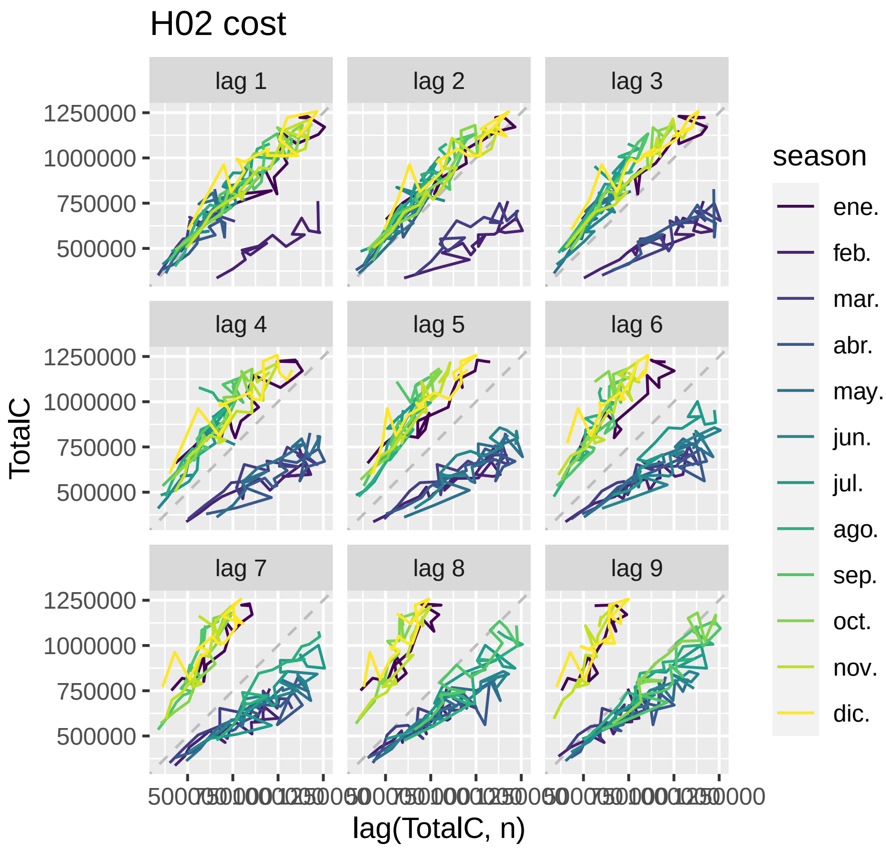

```
gg_season(h02_cost, TotalC) +
  labs(title = "H02 cost")
```




```r
gg_subseries(h02_cost, TotalC) +
  labs(title = "H02 cost")
```



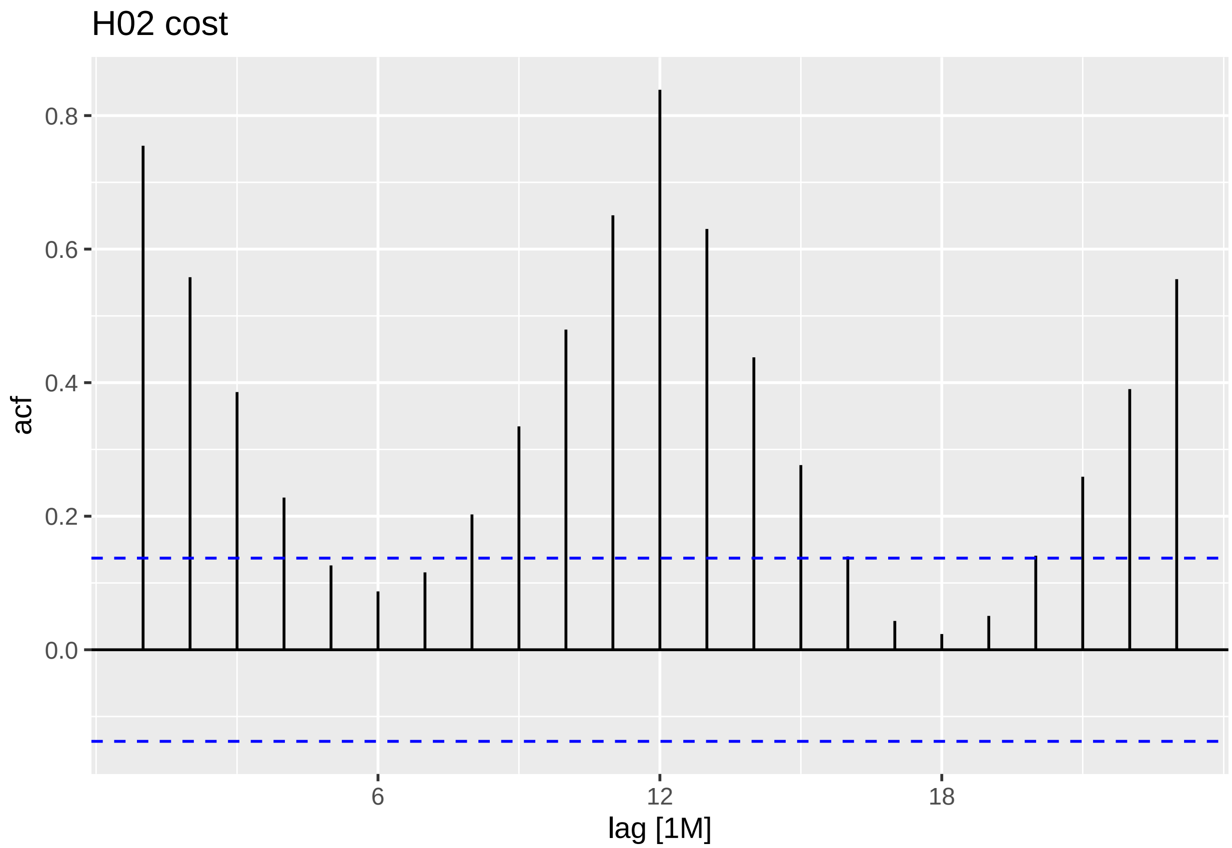

```r
gg_lag(h02_cost, TotalC) +
  labs(title = "H02 cost")
```




```r
h02_cost %>% ACF(TotalC, lag_max = 9)
```

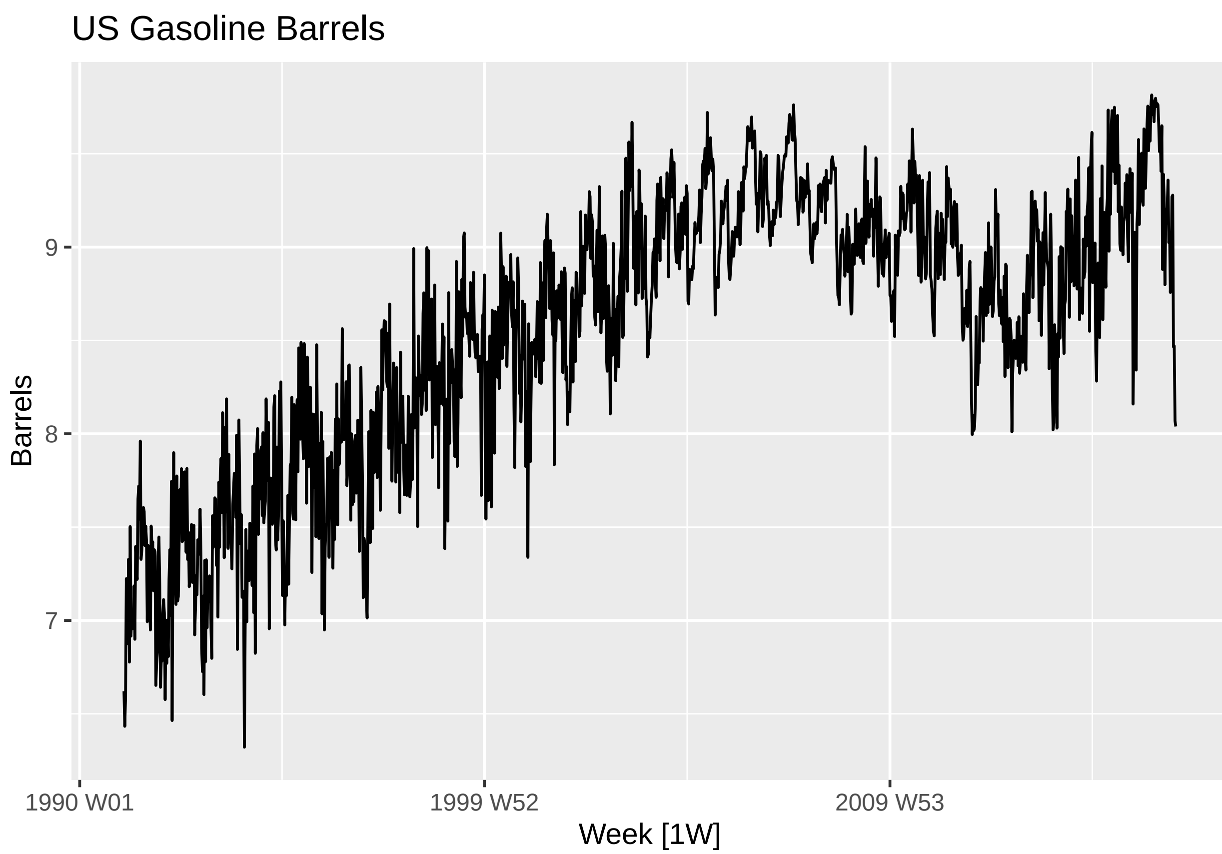
```
## # A tsibble: 9 x 2 [1M]
##      lag    acf
##    <lag>  <dbl>
## 1     1M 0.755
## 2     2M 0.558
## 3     3M 0.386
## 4     4M 0.228
## 5     5M 0.126
## 6     6M 0.0874
## 7     7M 0.116
## 8     8M 0.203
## 9     9M 0.335
```

```r
h02_cost %>%
  ACF(TotalC) %>%
  autoplot() +
  labs(title="H02 cost")
```

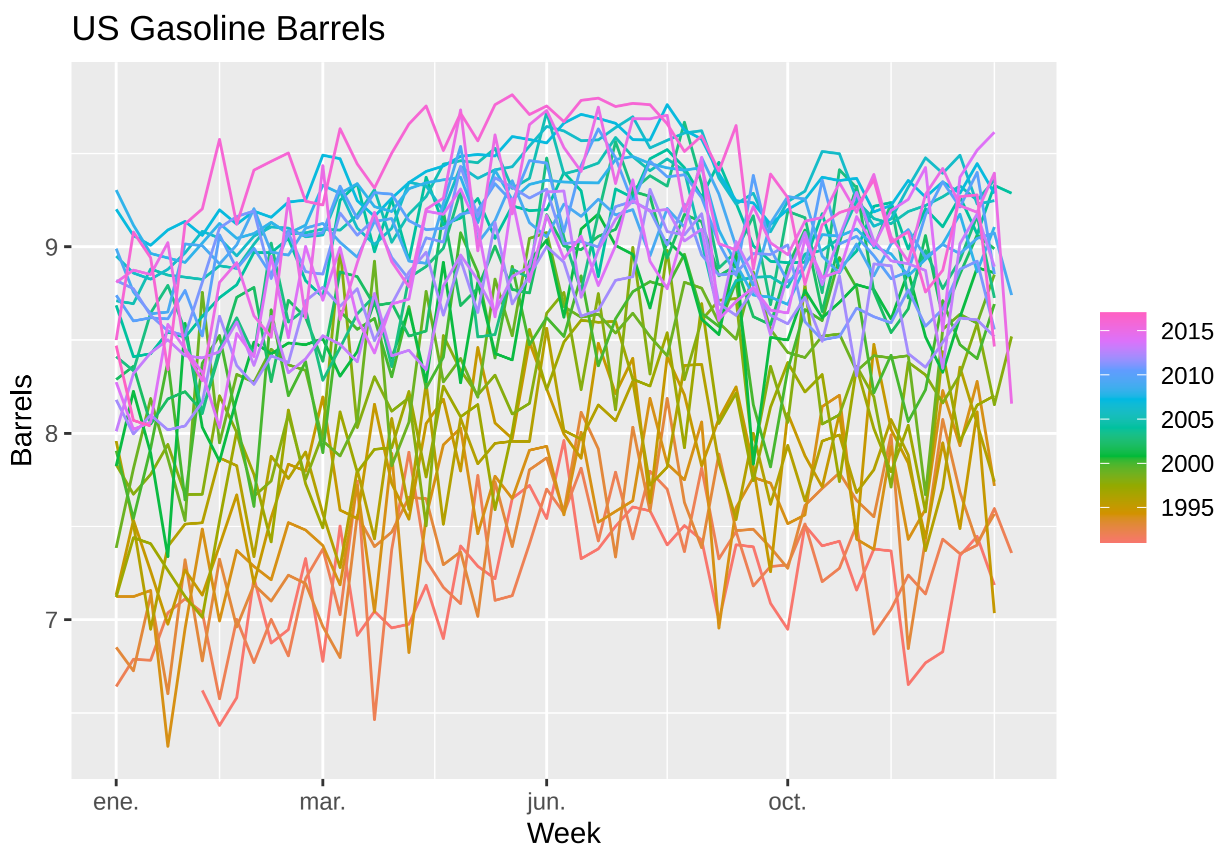


We can see that the series has seasonality, it starts as a time high during January, follow by a huge drop in February a slow increase over the rest of the year forward the January values. We can also see that there is no cyclicity, and the general trend is increasing, it is interesting the huge drop on the beginning of the year, this can be related to how medicine is administered in the USA (I'm not so familiarize with how this works). There is no unusual year on the time series.

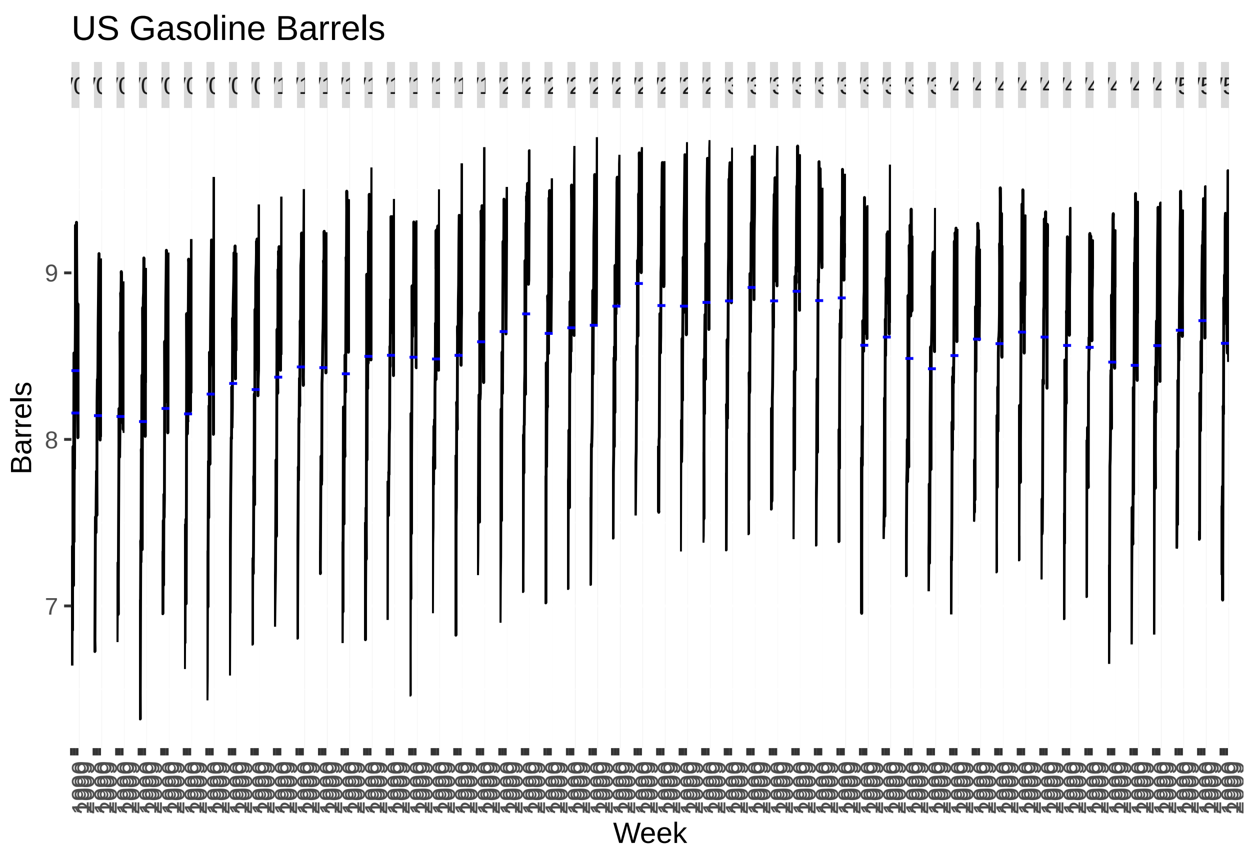
```
# "H02" Cost from PBS
us_Barrels   <- us_gasoline  #%>%
  #filter(ATC2 == "H02") %>% #Year >= 1980
  #select(Month, Cost)  %>%
  #summarise(TotalC = sum(Cost))

autoplot(us_Barrels, Barrels) +
  labs(title = "US Gasoline Barrels")
```



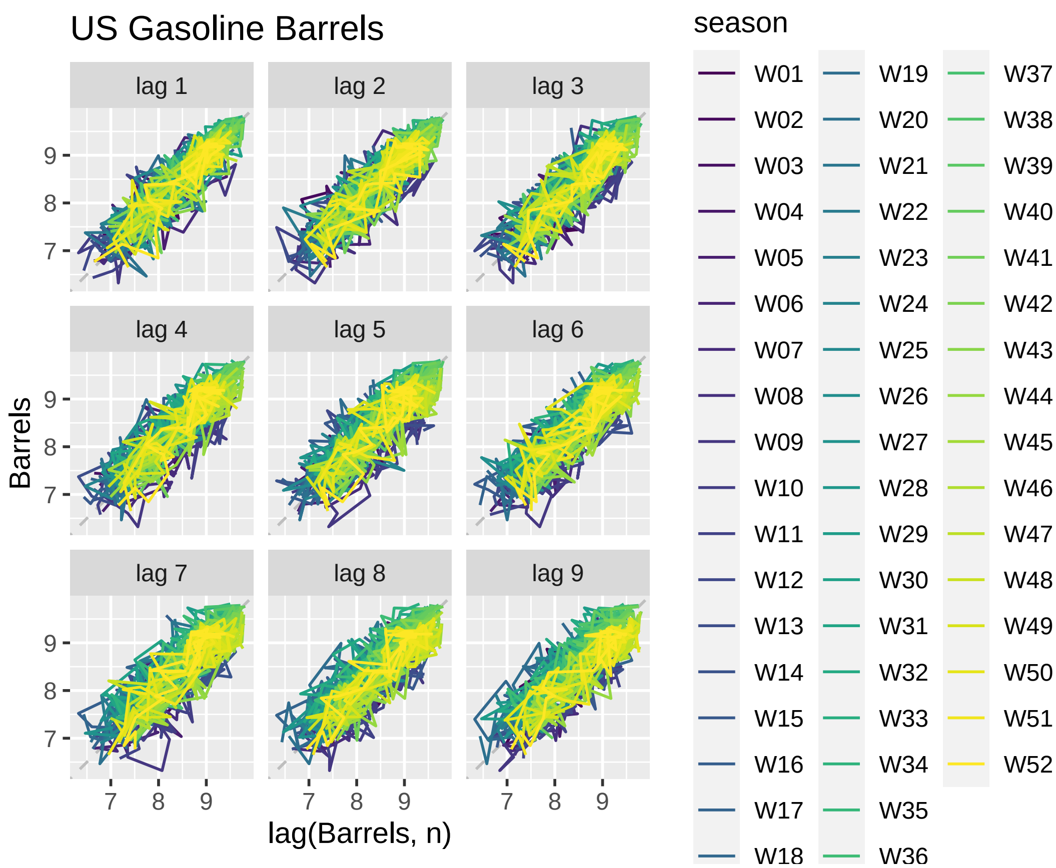

```
gg_season(us_Barrels, Barrels) +
  labs(title = "US Gasoline Barrels")
```




```
gg_subseries(us_Barrels, Barrels) +
  labs(title = "US Gasoline Barrels")
```



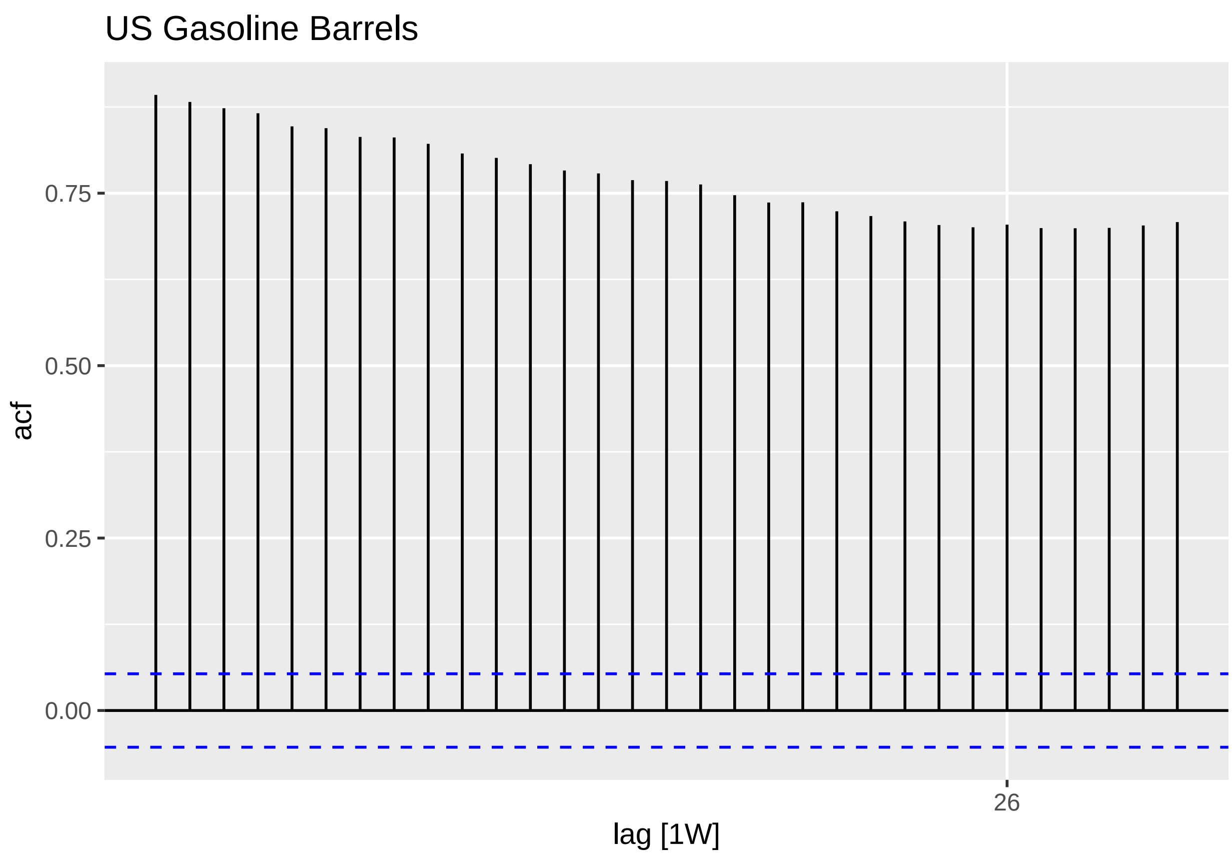

```
gg_lag(us_Barrels, Barrels) +
  labs(title = "US Gasoline Barrels")
```




```
us_Barrels %>% ACF(Barrels, lag_max = 9)
```

```
## # A tsibble: 9 x 2 [1W]
##      lag   acf
##    <lag> <dbl>
## 1     1W 0.893
## 2     2W 0.882
## 3     3W 0.873
## 4     4W 0.866
## 5     5W 0.847
## 6     6W 0.844
## 7     7W 0.832
## 8     8W 0.831
## 9     9W 0.822
```

```
us_Barrels %>%
  ACF(Barrels) %>%
  autoplot() +
  labs(title="US Gasoline Barrels")
```



In this case there seems to be some seasonality, although it is not extremely clear, we can see, that the barrels increase over the weeks, and then decrease over the end of the year. There is no cyclicity, and there is a positive trend, we can see that the number of barrels increase over time, although in the last part of the series the values stabilize, and the growing pattern dis smaller. There is no unusual year on the series.

3.7 exercises 9

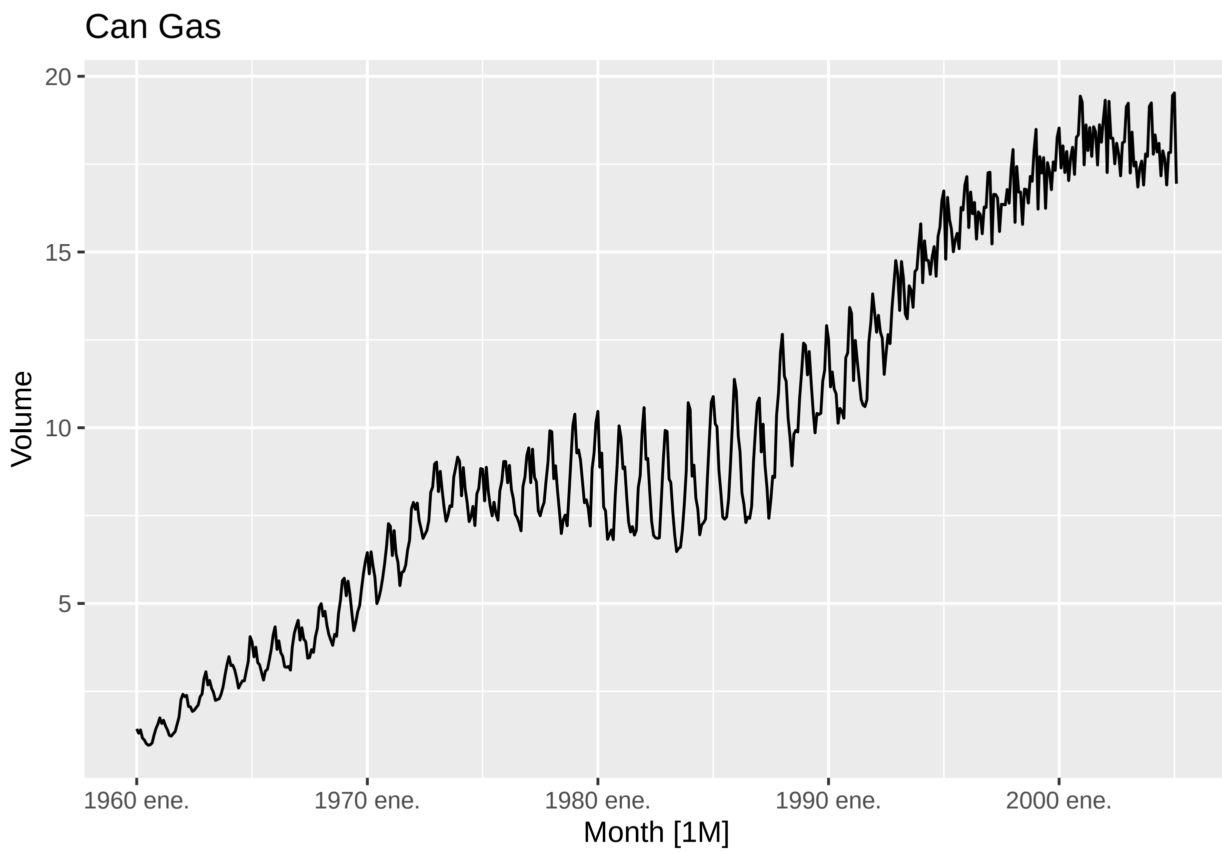
a) Here we can see an increasing time series, we can clearly see that the trend is positive in every year. If we take a look at the seasonality of the time series, we can find that there is a seasonal component, in this case is follows a complex path of increasing in values during the firsts months of the year, followed by a decrease until the month of august, where it reaches a time low, then in increase in September, followed by another fall during October and November, and finally in ends at the highest value during the last month of the year in September.
b) The recession during 1991/1992 is visible under the remainder component of the decomposition, as we can see a huge drop during that time period, while the normal residual is close to cero, and with a deviation no greater than 100, during that period it drop to -400

3.7 exercises 10

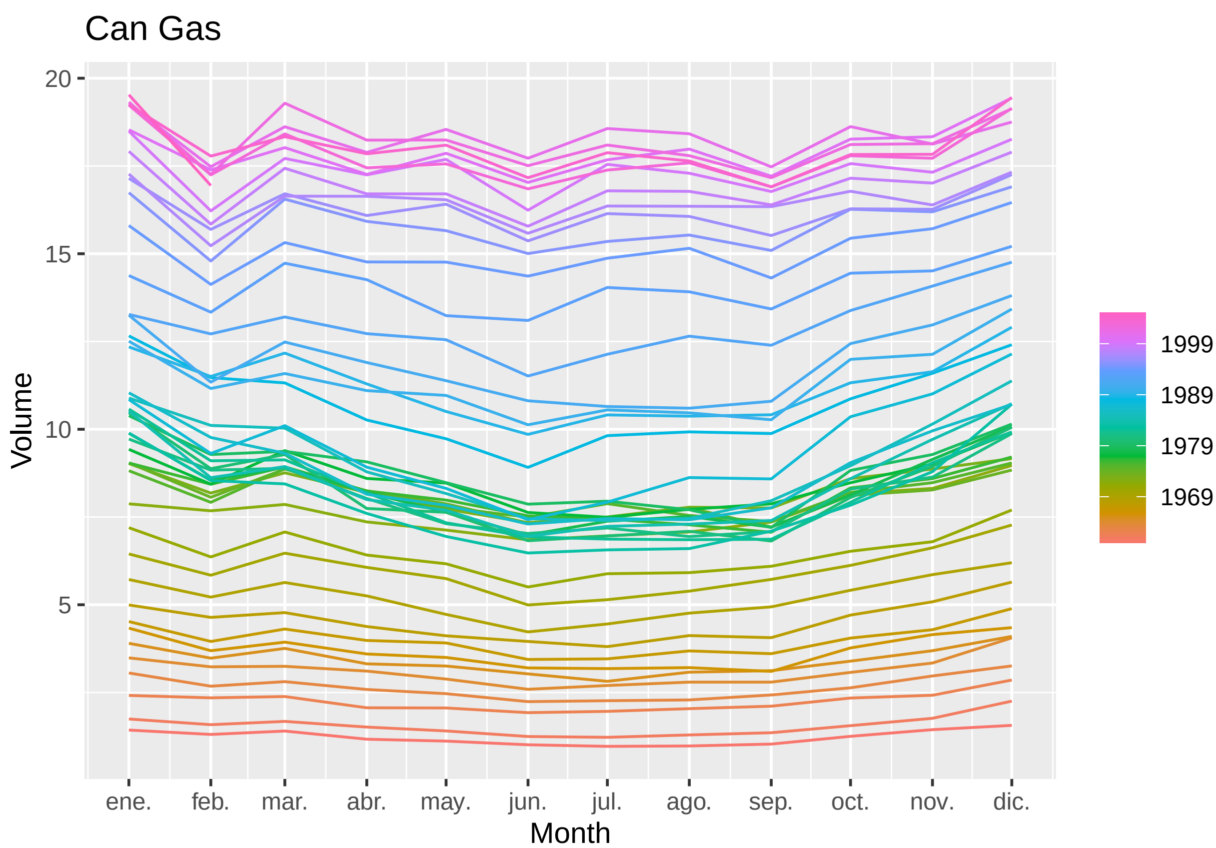
```
# "Total Private" Employed from us_employment
can_gas  <- canadian_gas  #%>%
  #filter(Title == "Total Private" & year(Month) >= 1980) %>%
  #select(Month, Employed)
```

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

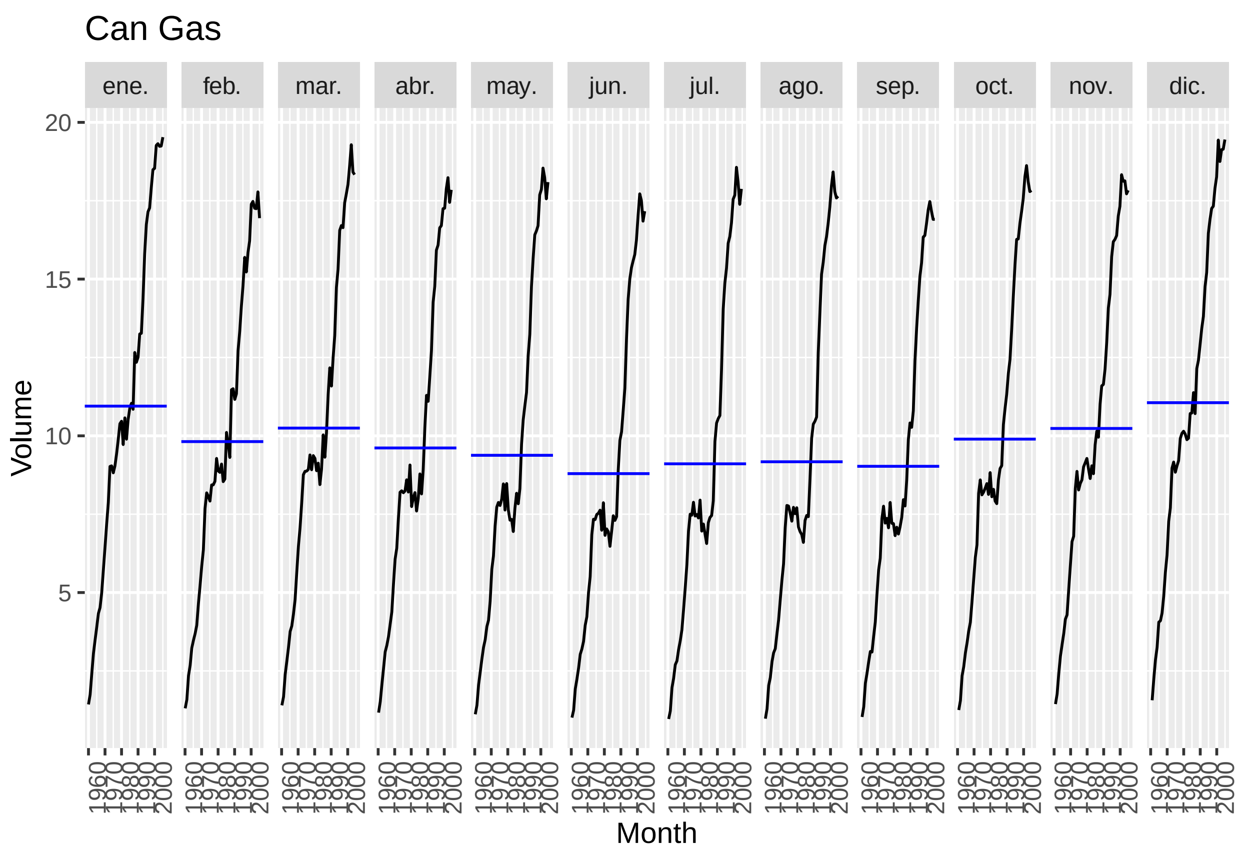
```
autoplot(can_gas, Volume) +
  labs(title = "Can Gas")
```




```
gg_season(can_gas, Volume) +
  labs(title = "Can Gas")
```



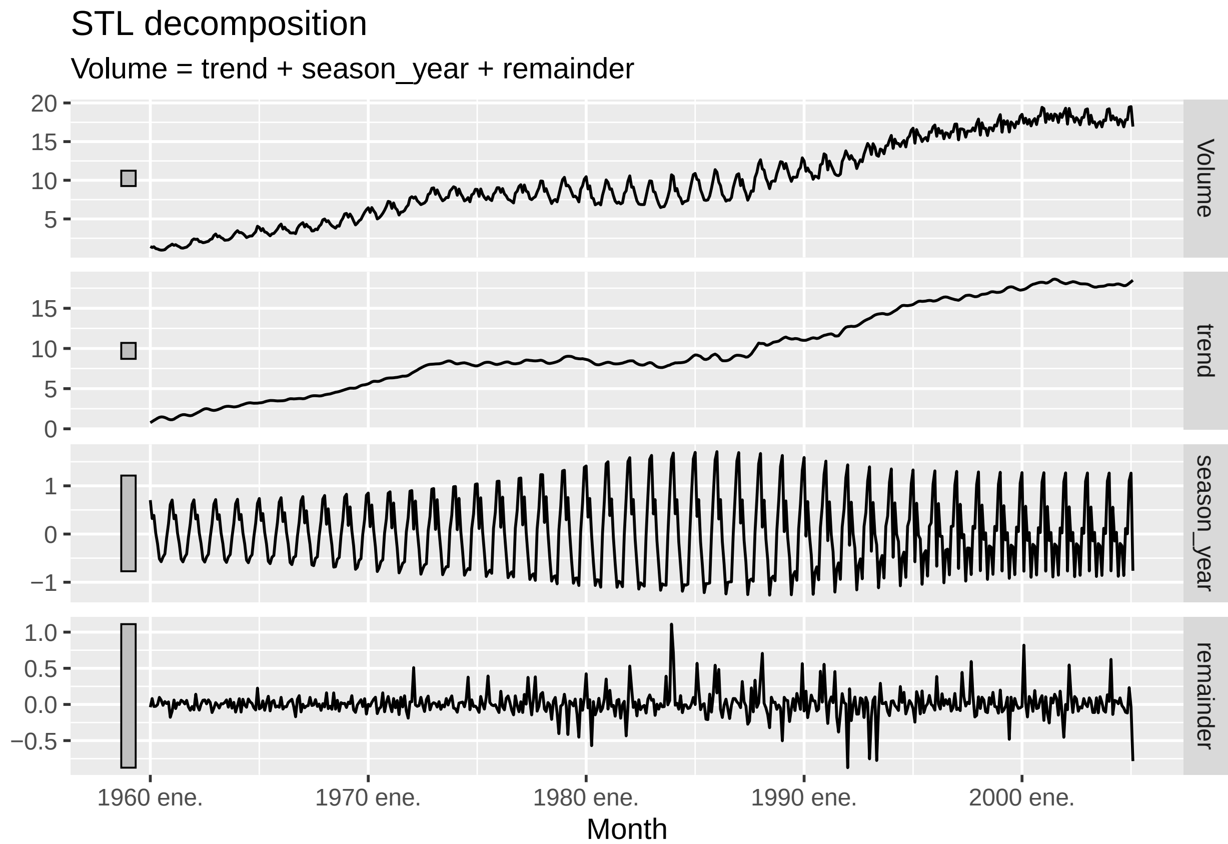

```
gg_subseries(can_gas, Volume) +
  labs(title = "Can Gas")
```



We can see there is seasonality, as the values start high at the biggening of the first month of the year, and slowly drop over the month, until the month of June, after that the values start increasing towards the end of the year, this pattern is similar over all the years.

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.

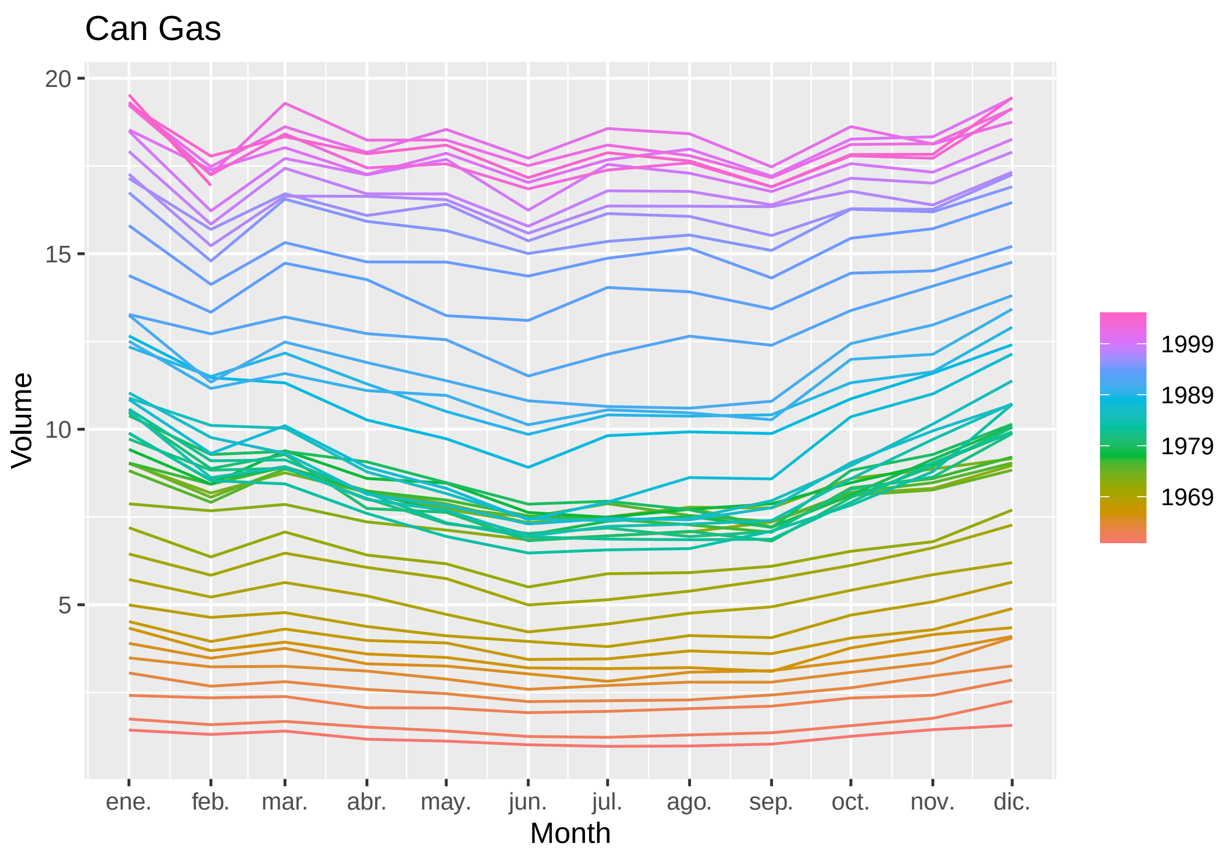
```
can_gas %>%
  model(
    STL(Volume ~ trend(window = 7) +
                 season(window = 13),
    robust = TRUE)) %>%
  components() %>%
  autoplot()
```



We can see that the seasonal component starts to increase over the years until it achieves a higher peak in the mid 80's, after then it starts decreasing.

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

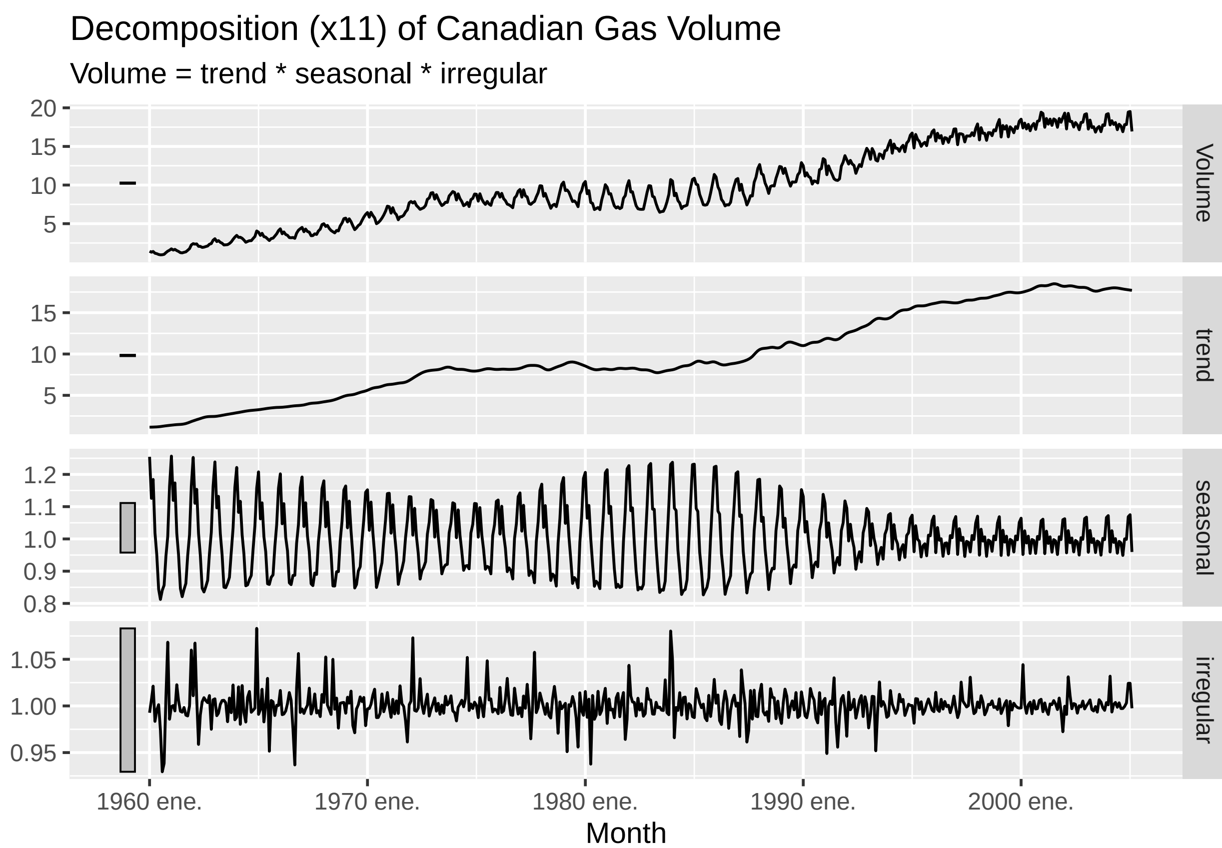
```
gg_season(can_gas, Volume) +
  labs(title = "Can Gas")
```



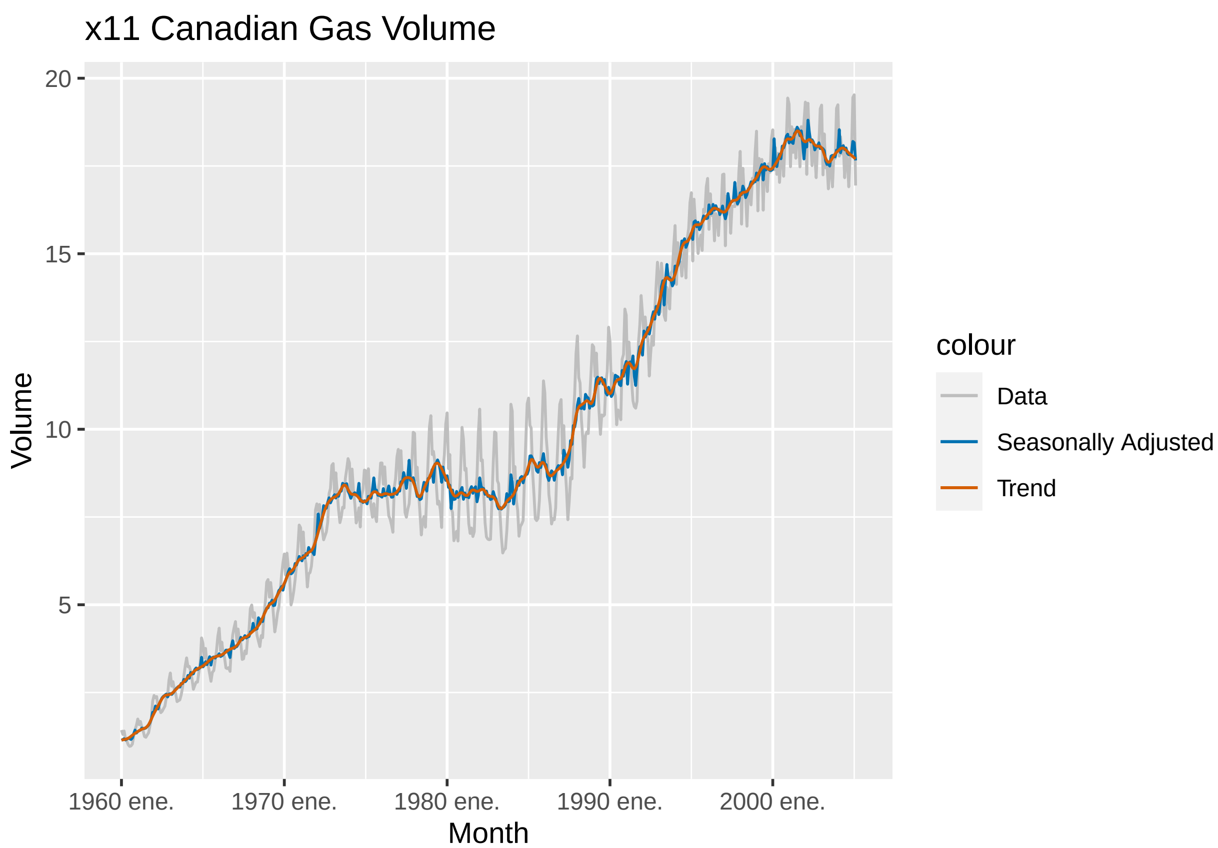
We can see that the seasonal component start to get noisier as time advances, at the beginning it was a simple curve, while the last years it had way more jumps and Sharpe edges.

d) Can you produce a plausible seasonally adjusted series?

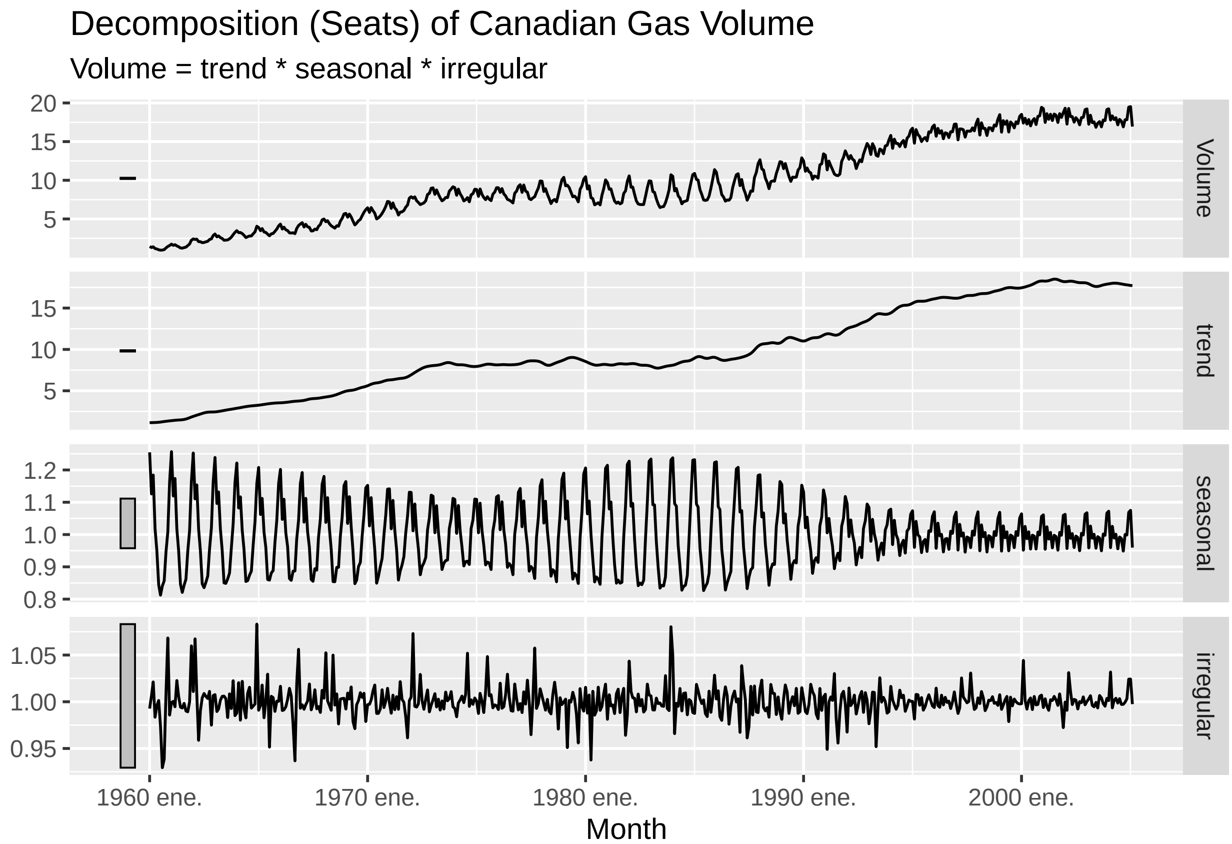
```
x11_dcmp <- can_gas %>%
  model(x11 = X_13ARIMA_SEATS(Volume ~ x11())) %>%
  components()
autoplot(x11_dcmp) +
  labs(title =
    "Decomposition (x11) of Canadian Gas Volume")
```




```
x11_dcmp %>%
  ggplot(aes(x = Month)) +
  geom_line(aes(y = Volume, colour = "Data")) +
  geom_line(aes(y = season_adjust,
                colour = "Seasonally Adjusted")) +
  geom_line(aes(y = trend, colour = "Trend")) +
  labs(y = "Volume",
       title = "x11 Canadian Gas Volume") +
  scale_colour_manual(
    values = c("gray", "#0072B2", "#D55E00"),
    breaks = c("Data", "Seasonally Adjusted", "Trend")
  )
```



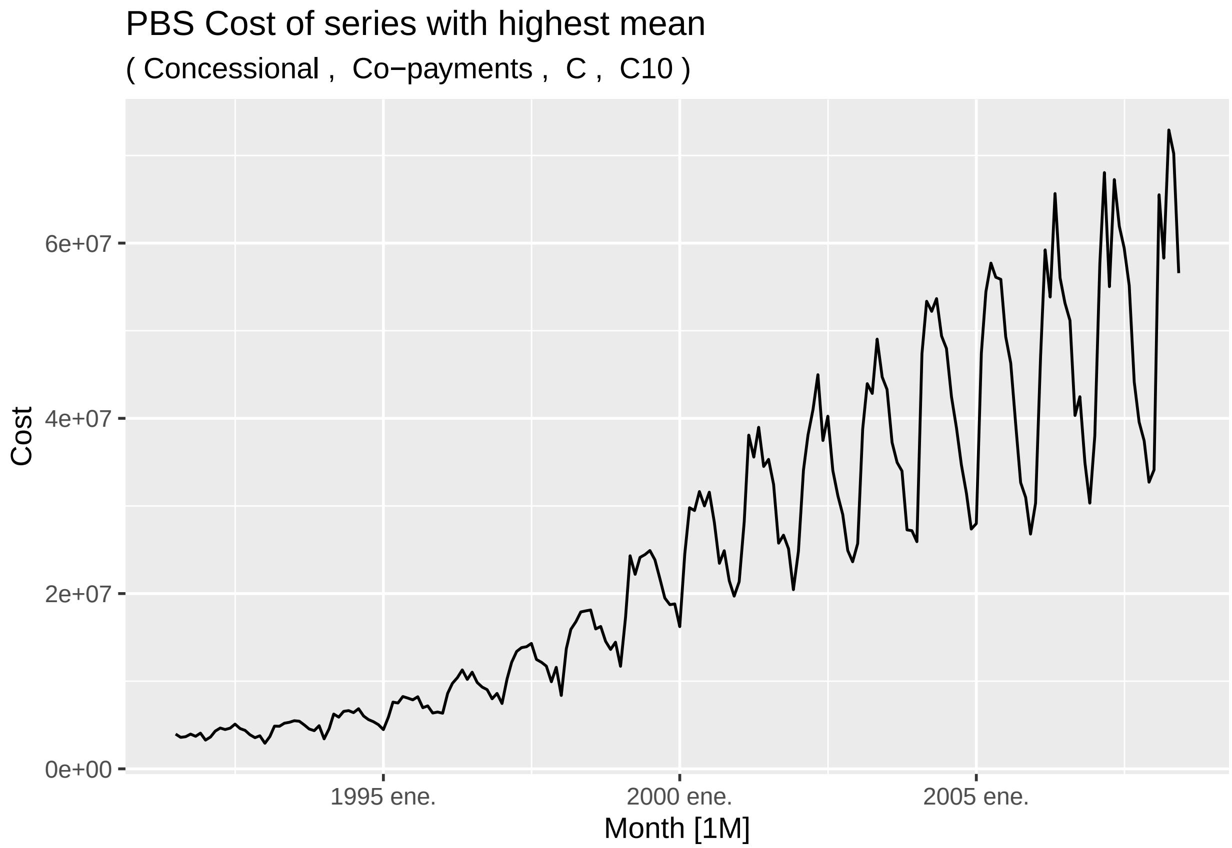

```
x11_dcmp_seats <- can_gas %>%
  model(seats = X_13ARIMA_SEATS(Volume ~ seats())) %>%
  components()
autoplot(x11_dcmp) +
  labs(title =
    "Decomposition (Seats) of Canadian Gas Volume")
```



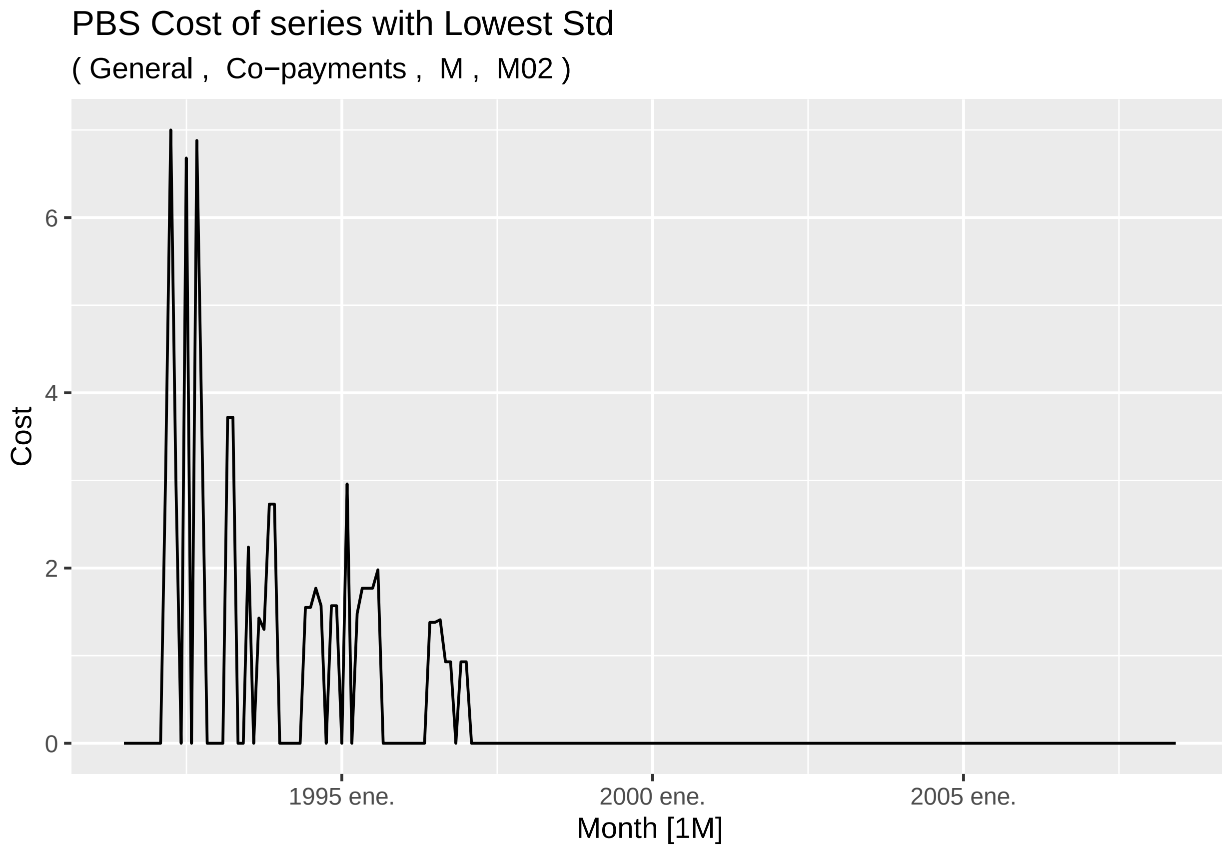
e) Compare the results with those obtained using SEATS and X-11. How are they different? If we take a look at the results, they are pretty much identical, between X-11 and SEATS method

4.6 exercise 1

```
pbs_mean <- PBS %>% features(Cost, list(mean = mean)) %>%
  arrange(desc(mean))
#pbs_mean

pbs_HighMean   <- PBS  %>%
  filter(Concession == pbs_mean$Concession[1] & Type == pbs_mean$Type[1] & ATC1 == pbs_mean$ATC1[1] & AT
#pbs_HighMean

autoplot(pbs_HighMean, Cost) +
  labs(title = "PBS Cost of series with highest mean",
       subtitle = paste("(", pbs_mean$Concession[1], ", ",pbs_mean$Type[1],", ",pbs_mean$ATC1[1],", ",pb
```




```
pbs_std <- PBS %>% features(Cost, list(sd = sd)) %>%
  filter(sd != 0) %>%
  arrange(sd)
#pbs_std

pbs_LowestStd <- PBS  %>%
  filter(Concession == pbs_std$Concession[1] & Type == pbs_std$Type[1] & ATC1 == pbs_std$ATC1[1] & ATC2
#pbs_LowestStd

autoplot(pbs_LowestStd, Cost) +
  labs(title = "PBS Cost of series with Lowest Std",
       subtitle = paste("(", pbs_std$Concession[1], ", ",pbs_std$Type[1],", ",pbs_std$ATC1[1],", ",pbs_s
```



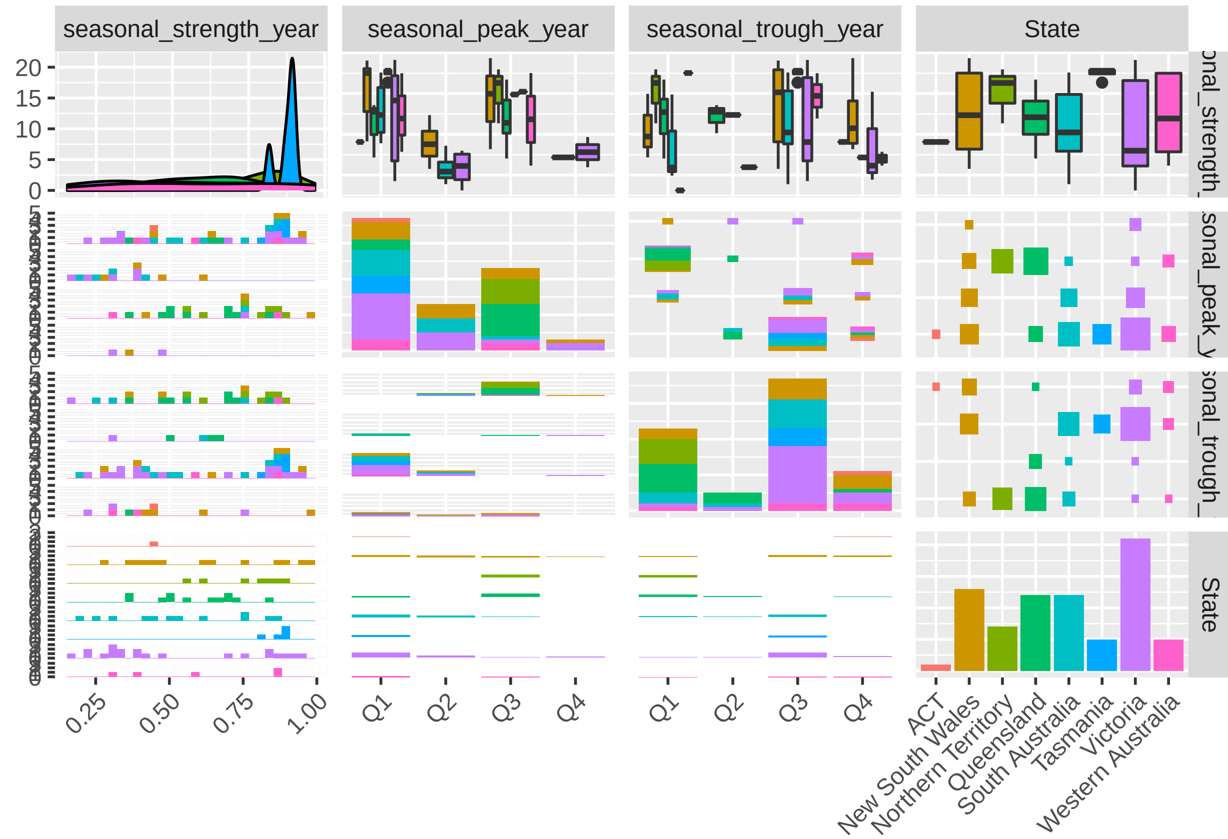
4.6 exercise 2

```
tourism %>%
  filter(Purpose == "Holiday")
```

```
## # A tsibble: 6,080 x 5 [1Q]
## # Key:       Region, State, Purpose [76]
##    Quarter Region   State           Purpose Trips
##      <qtr> <chr>    <chr>           <chr>   <dbl>
##  1 1998 Q1 Adelaide South Australia Holiday  224.
##  2 1998 Q2 Adelaide South Australia Holiday  130.
##  3 1998 Q3 Adelaide South Australia Holiday  156.
##  4 1998 Q4 Adelaide South Australia Holiday  182.
##  5 1999 Q1 Adelaide South Australia Holiday  185.
##  6 1999 Q2 Adelaide South Australia Holiday  135.
##  7 1999 Q3 Adelaide South Australia Holiday  136.
##  8 1999 Q4 Adelaide South Australia Holiday  169.
##  9 2000 Q1 Adelaide South Australia Holiday  184.
## 10 2000 Q2 Adelaide South Australia Holiday  134.
## # ... with 6,070 more rows
```

```r
tourism_features <- tourism %>%
  #filter(Purpose == "Holiday")%>%
  features(Trips, feat_stl)

tourism_features %>%
  filter(Purpose == "Holiday") %>%
  select_at(vars(contains("season"), State)) %>%
  mutate(
    seasonal_peak_year = seasonal_peak_year +
      4*(seasonal_peak_year==0),
    seasonal_trough_year = seasonal_trough_year +
      4*(seasonal_trough_year==0),
    seasonal_peak_year = glue("Q{seasonal_peak_year}"),
    seasonal_trough_year = glue("Q{seasonal_trough_year}"),
  ) %>%
  GGally::ggpairs(mapping = aes(colour = State)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: Groups with fewer than two data points have been dropped.

## Warning in max(ids, na.rm = TRUE): ningun argumento finito para max; retornando
## -Inf

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



We can see that the peak quarter for each state is: ACT: Q1 New South Wales: Q1 Northern Territory: Q3 Queensland: Q3 South Australia: Q1 Tasmania: Q1 Victoria: Q1 Western Australia: Q1

4.6 exercise 3

```
PBS_features <- PBS %>%
  #filter(Purpose == "Holiday")%>%
  features(Cost, feat_stl)
PBS_features
```

```
## # A tibble: 336 x 13
##    Concession Type  ATC1  ATC2  trend_strength seasonal_streng~ seasonal_peak_y~
##    <chr>      <chr> <chr> <chr>          <dbl>            <dbl>            <dbl>
##  1 Concessio~ Co-p~ A     A01            0.837            0.899                9
##  2 Concessio~ Co-p~ A     A02            0.973            0.937               11
##  3 Concessio~ Co-p~ A     A03            0.978            0.845               11
##  4 Concessio~ Co-p~ A     A04            0.944            0.843                9
##  5 Concessio~ Co-p~ A     A05            0.960            0.870               11
##  6 Concessio~ Co-p~ A     A06            0.960            0.951               11
##  7 Concessio~ Co-p~ A     A07            0.971            0.910               11
##  8 Concessio~ Co-p~ A     A09            0.942            0.891               11
##  9 Concessio~ Co-p~ A     A10            0.975            0.917               11
## 10 Concessio~ Co-p~ A     A11            0.982            0.898               11
## # ... with 326 more rows, and 6 more variables: seasonal_trough_year <dbl>,
```

```
## #   spikiness <dbl>, linearity <dbl>, curvature <dbl>, stl_e_acf1 <dbl>,
## #   stl_e_acf10 <dbl>
```

```
PBS_features %>%
  select_at(vars(contains("season"))) %>%
  mutate(
    seasonal_peak_year = seasonal_peak_year +
      4*(seasonal_peak_year==0),
    seasonal_trough_year = seasonal_trough_year +
      4*(seasonal_trough_year==0),
    seasonal_peak_year = glue("Q{seasonal_peak_year}"),
    seasonal_trough_year = glue("Q{seasonal_trough_year}"),
  ) %>%
  GGally::ggpairs(mapping = aes()) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: Removed 2 rows containing non-finite values (stat_density).
```

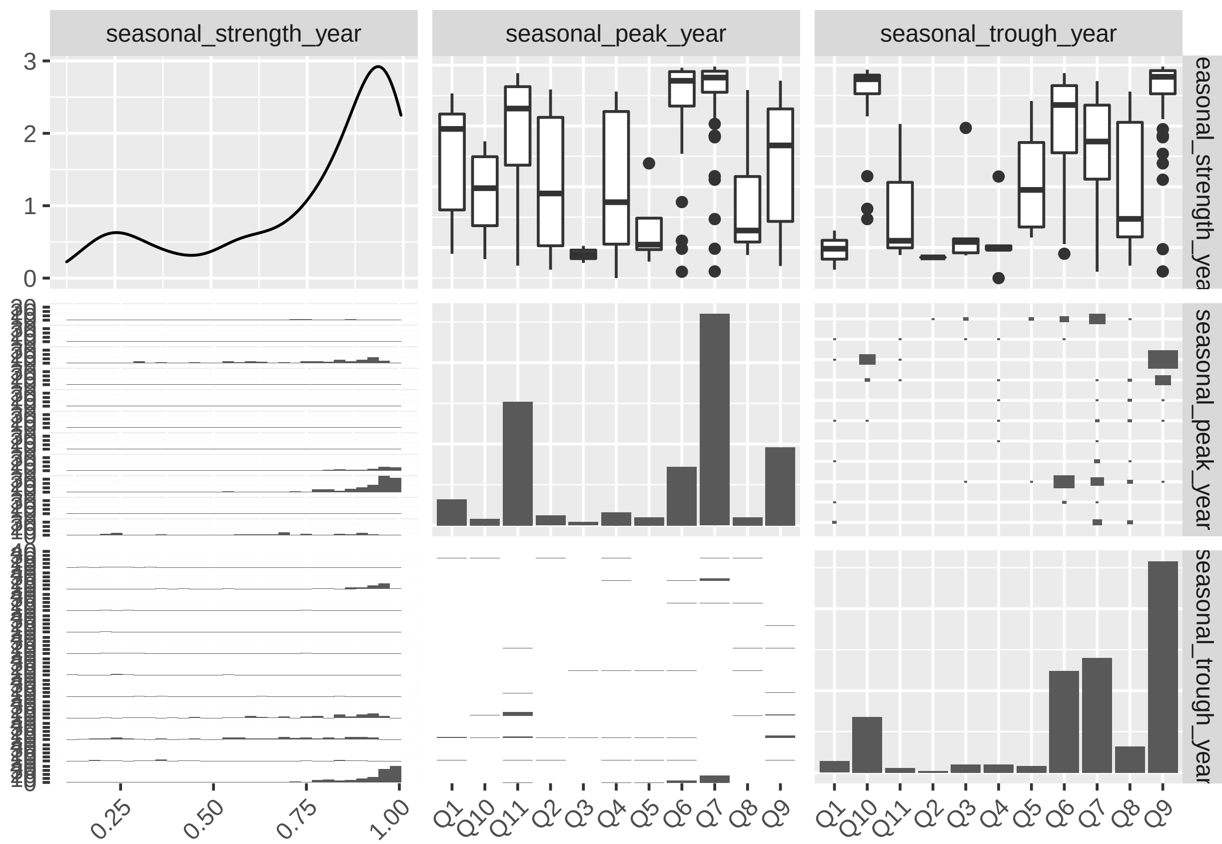
```
## Warning: Removed 2 rows containing non-finite values (stat_boxplot).
## Removed 2 rows containing non-finite values (stat_boxplot).
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 2 rows containing non-finite values (stat_bin).
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 2 rows containing non-finite values (stat_bin).
```



5.11 exercises 9

a) Create a training set for household wealth (hh_budget) by withholding the last four years as a test set.

```
hh_budget
```

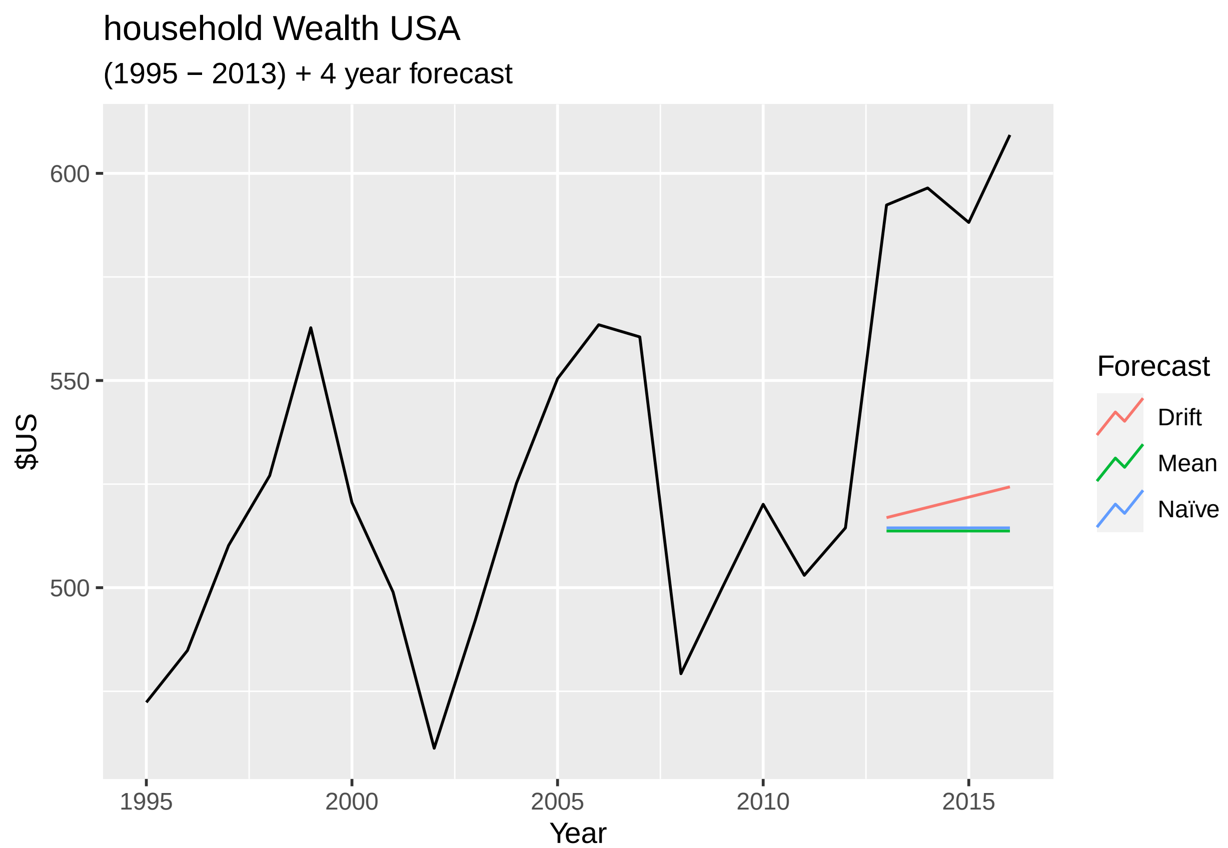
```
## # A tsibble: 88 x 8 [1Y]
## # Key:       Country [4]
##    Country    Year  Debt     DI Expenditure Savings Wealth Unemployment
##    <chr>     <dbl> <dbl>  <dbl>       <dbl>   <dbl>  <dbl>        <dbl>
##  1 Australia  1995  95.7 3.72          3.40   5.24    315.          8.47
##  2 Australia  1996  99.5 3.98          2.97   6.47    315.          8.51
##  3 Australia  1997 108.  2.52          4.95   3.74    323.          8.36
##  4 Australia  1998 115.  4.02          5.73   1.29    339.          7.68
##  5 Australia  1999 121.  3.84          4.26   0.638   354.          6.87
##  6 Australia  2000 126.  3.77          3.18   1.99    350.          6.29
##  7 Australia  2001 132.  4.36          3.10   3.24    348.          6.74
##  8 Australia  2002 149.  0.0218        4.03  -1.15    349.          6.37
##  9 Australia  2003 159.  6.06          5.04  -0.413   360.          5.93
## 10 Australia  2004 170.  5.53          4.54   0.657   379.          5.40
## # ... with 78 more rows
```

```r
train <- hh_budget %>%
  filter(Year <= 2012 & Country=="USA") #& Country=="__" | Australia, Canada, Japan, USA
train
```

```
## # A tsibble: 18 x 8 [1Y]
## # Key:       Country [1]
##    Country  Year  Debt      DI Expenditure Savings Wealth Unemployment
##    <chr>   <dbl> <dbl>   <dbl>       <dbl>   <dbl>  <dbl>        <dbl>
##  1 USA      1995  94.3  3.00         2.95     7.24   472.         5.61
##  2 USA      1996  96.2  2.86         3.47     6.79   485.         5.42
##  3 USA      1997  97.3  3.46         3.77     6.58   510.         4.95
##  4 USA      1998  98.5  5.66         5.31     7.06   527.         4.51
##  5 USA      1999 103.   3.20         5.27     5.27   563.         4.22
##  6 USA      2000 104.   4.69         5.08     5.03   521.         3.99
##  7 USA      2001 108.   2.82         2.52     5.25   499.         4.73
##  8 USA      2002 113.   3.29         2.57     6.06   461.         5.78
##  9 USA      2003 121.   2.79         3.18     5.75   492.         5.99
## 10 USA      2004 128.   3.28         3.75     5.35   525.         5.53
## 11 USA      2005 136.   1.34         3.56     3.28   550.         5.07
## 12 USA      2006 141.   3.56         3.06     4.00   563.         4.62
## 13 USA      2007 144.   1.78         2.22     3.88   561.         4.62
## 14 USA      2008 137.   1.74        -0.212    5.18   479.         5.78
## 15 USA      2009 136.  -0.0516      -1.25     6.34   500.         9.27
## 16 USA      2010 128.   1.01         1.75     6.78   520.         9.62
## 17 USA      2011 121.   2.23         1.89     7.40   503.         8.95
## 18 USA      2012 115.   2.96         1.50     9.14   514.         8.07
```

b) Fit all the appropriate benchmark methods to the training set and forecast the periods covered by the test set.

```r
# Fit the models
budget_fit <- train %>%
  model(
    Mean = MEAN(Wealth),
    `Naïve` = NAIVE(Wealth),
    Drift = RW(Wealth ~ drift())
  )

# Generate forecasts for 14 quarters
budget_fc <- budget_fit %>% forecast(h = 4)

#Plot
budget_fc %>%
  autoplot(hh_budget,level = NULL) +
  labs(y = "$US",
       title = "household Wealth USA",
       subtitle = "(1995 - 2013) + 4 year forecast") +
  guides(colour = guide_legend(title = "Forecast"))
```



c) Compute the accuracy of your forecasts. Which method does best?

```
accuracy(budget_fc, hh_budget)
```

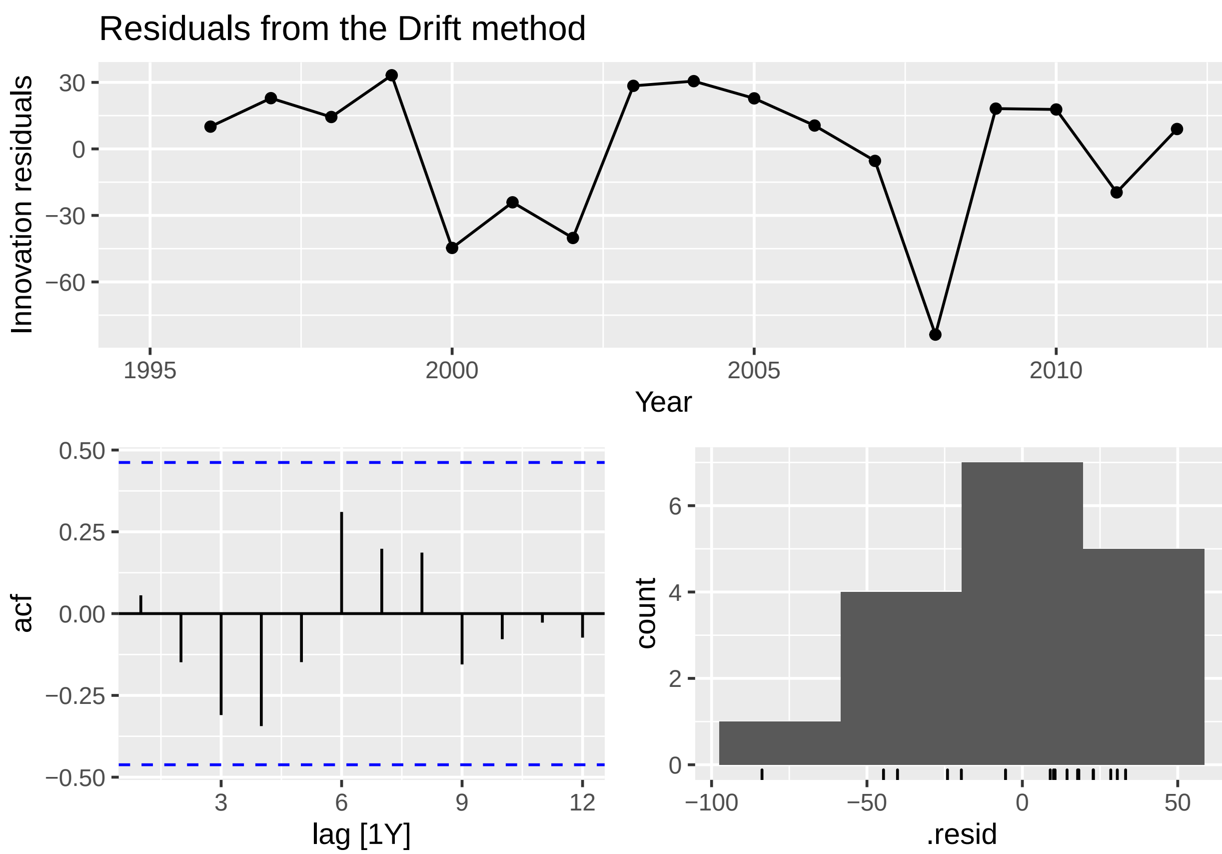
```
## # A tibble: 3 x 11
##   .model Country .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE   ACF1
##   <chr>  <chr>   <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 Drift  USA     Test   75.9  76.2  75.9  12.7  12.7  2.88  2.43 -0.561
## 2 Mean   USA     Test   82.9  83.3  82.9  13.9  13.9  3.15  2.65 -0.423
## 3 Naïve  USA     Test   82.1  82.5  82.1  13.8  13.8  3.12  2.63 -0.423
```

Drift method achieve better metrics than the other methods

d) Do the residuals from the best method resemble white noise?

```
train %>%
  model(Drift = RW(Wealth ~ drift())) %>%
  gg_tsresiduals() +
  labs(title = "Residuals from the Drift method")
```

```
## 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).
```



Yes, they resemble white noise, as they are concentrated close to the 0 value, although they seam to be a little skewed to the positive size. From the ACF of the residuals we can see that there is no correlation, meaning that the forecast is good.

5.11 exercises 10

a) Create a training set for Australian takeaway food turnover (aus_retail) by withholding the last four years as a test set.

```
#aus_retail
test_set <- aus_retail %>%
  filter(Industry=="Takeaway food services") %>%
  summarise(AVGTurnover = mean(Turnover))
test_set
```

```
## # A tsibble: 441 x 2 [1M]
##         Month AVGTurnover
##         <mth>       <dbl>
##   1 1982 abr.        27.7
##   2 1982 may.        27.7
##   3 1982 jun.        26.6
##   4 1982 jul.        27.1
##   5 1982 ago.        27.2
##   6 1982 sep.        27.9
##   7 1982 oct.        29.9
```

```
##   8 1982 nov.          30.4
##   9 1982 dic.          34.0
##  10 1983 ene.          32.1
## # ... with 431 more rows
```

```
train <- test_set %>%
  filter(year(Month) <= 2014)

train
```

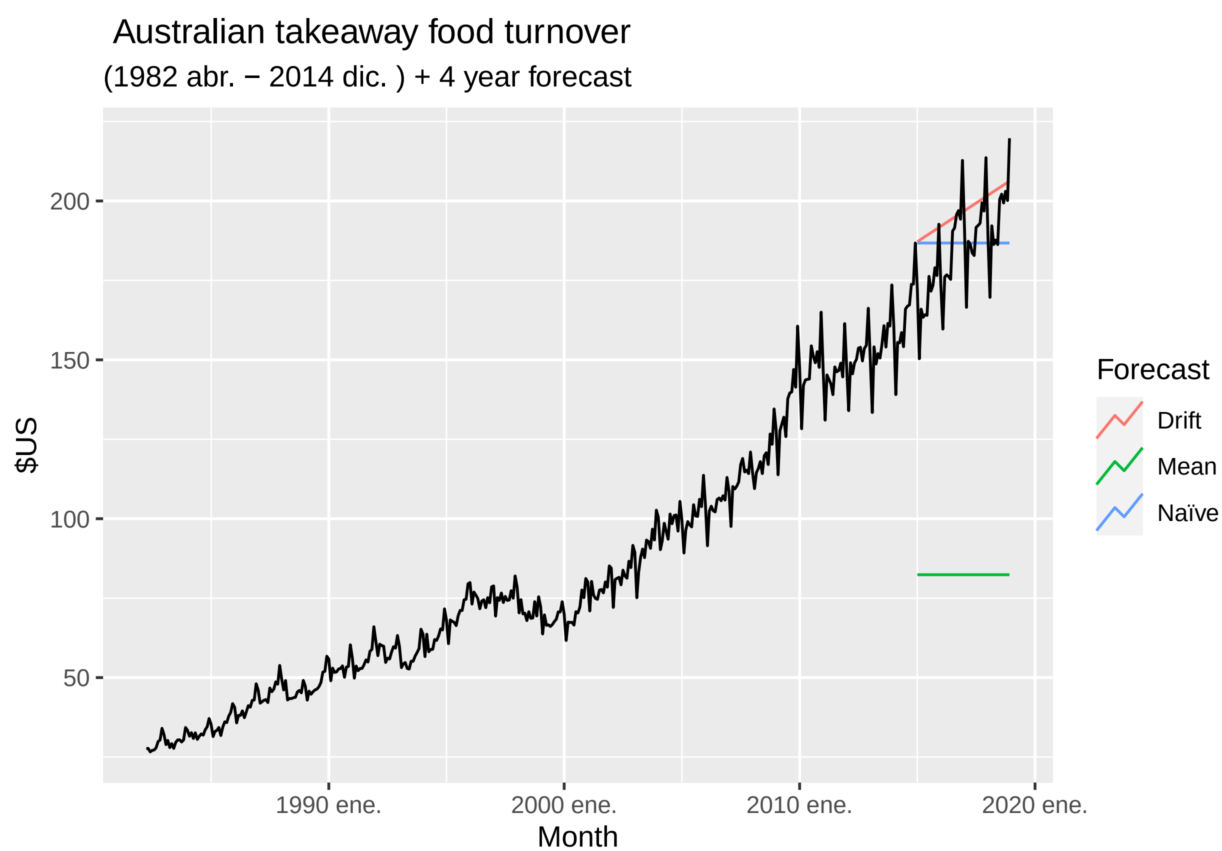
```
## # A tsibble: 393 x 2 [1M]
##         Month AVGTurnover
##         <mth>       <dbl>
##   1 1982 abr.        27.7
##   2 1982 may.        27.7
##   3 1982 jun.        26.6
##   4 1982 jul.        27.1
##   5 1982 ago.        27.2
##   6 1982 sep.        27.9
##   7 1982 oct.        29.9
##   8 1982 nov.        30.4
##   9 1982 dic.        34.0
##  10 1983 ene.        32.1
## # ... with 383 more rows
```

b) Fit all the appropriate benchmark methods to the training set and forecast the periods covered by the test set.

```
# Fit the models
retail_fit <- train %>%
  model(
    Mean = MEAN(AVGTurnover),
    `Naïve` = NAIVE(AVGTurnover),
    Drift = RW(AVGTurnover ~ drift())
  )

# Generate forecasts for 14 quarters
retail_fc <- retail_fit %>% forecast(h = 48) # 4*12

#Plot
retail_fc %>%
  autoplot(test_set,level = NULL) +
  labs(y = "$US",
       title = " Australian takeaway food turnover",
       subtitle = "(1982 abr. - 2014 dic.    ) + 4 year forecast") +
  guides(colour = guide_legend(title = "Forecast"))
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## font width unknown for character 0x9
```



c) Compute the accuracy of your forecasts. Which method does best?

```
accuracy(retail_fc, test_set)
```

```
## # A tibble: 3 x 10
##   .model .type     ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>  <chr>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Drift  Test  -11.5   16.2  13.4 -6.71  7.58  2.54  2.44 0.408
```

```
## 2 Mean   Test  103.    104.   103.   55.2  55.2  19.6  15.7  0.613
## 3 Naïve  Test   -1.55  14.8   12.0 -1.49  6.66  2.29  2.24 0.613
```

Naïve method achieve better metrics than the other methods

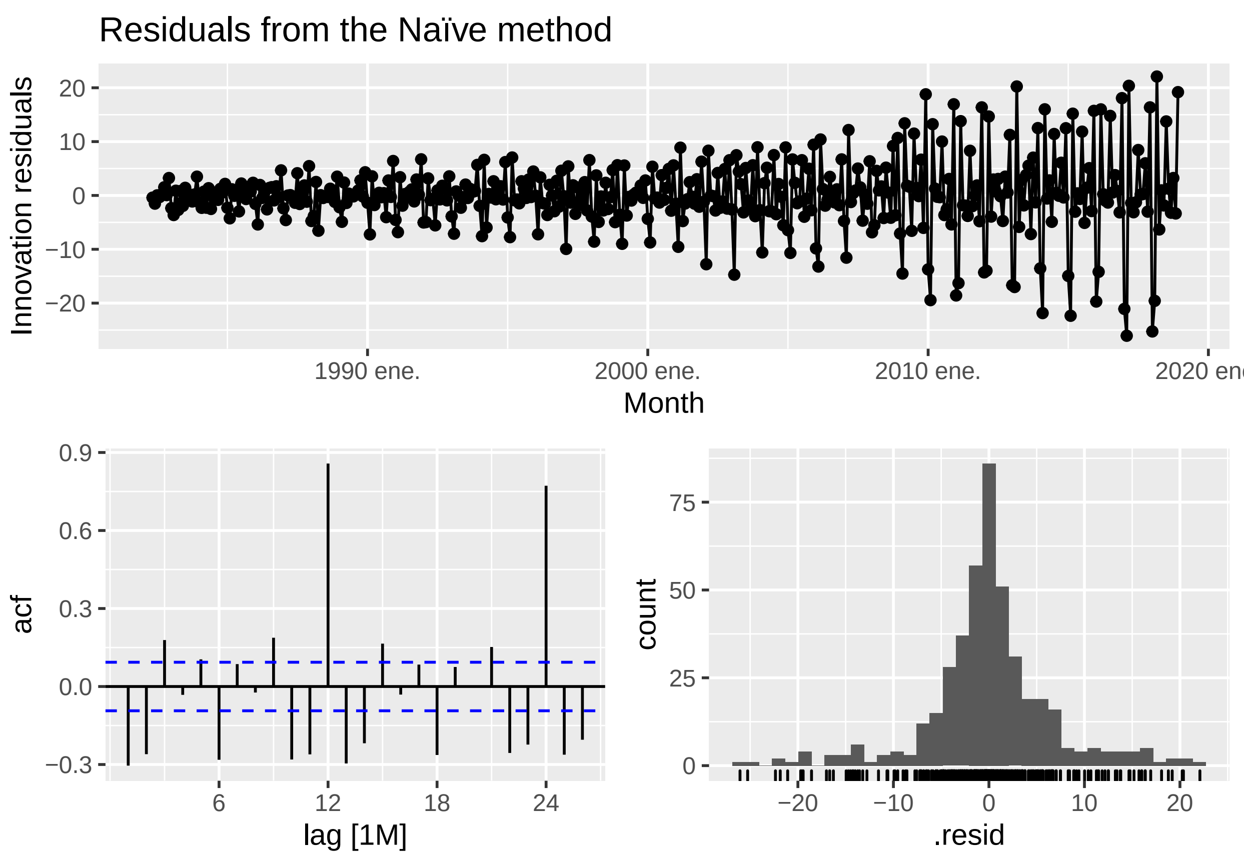
d) Do the residuals from the best method resemble white noise?

```
test_set %>%
  model(Drift = RW(AVGTurnover ~ drift())) %>%
  gg_tsresiduals() +
  labs(title = "Residuals from the Naïve method")
```

```
## 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).
```



In this case we can see that the residuals increase over time, although they are grouped by the 0 value, and they resemble a normal distribution. From the ACF of the residuals we can see that there is correlation in multiple values, which means that the model hasn't capture all the information, this is because the naïve model doesn't capture the seasonality of the data.

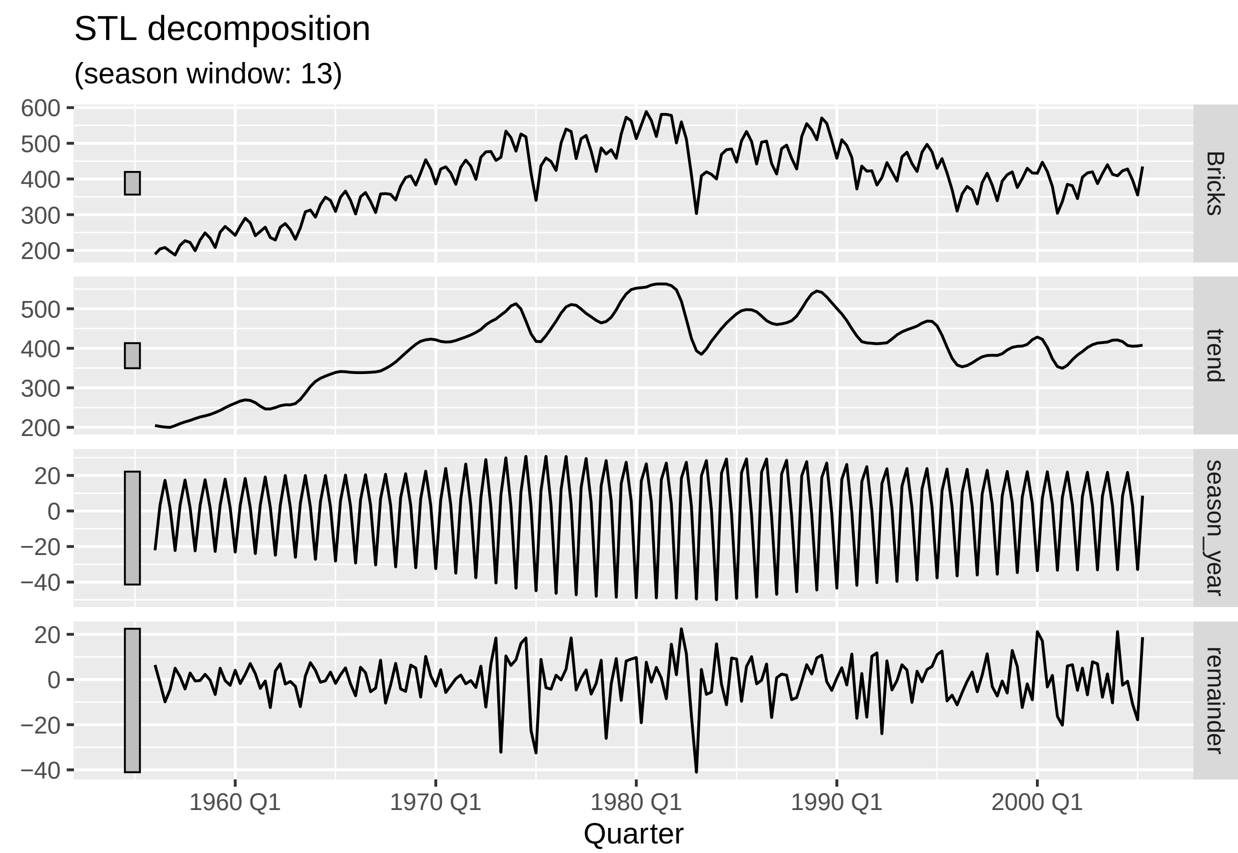
5.11 exercises 11

```r
bricks_aus_production   <- aus_production  %>%
  select(Quarter, Bricks) %>%
  drop_na()
bricks_aus_production
```

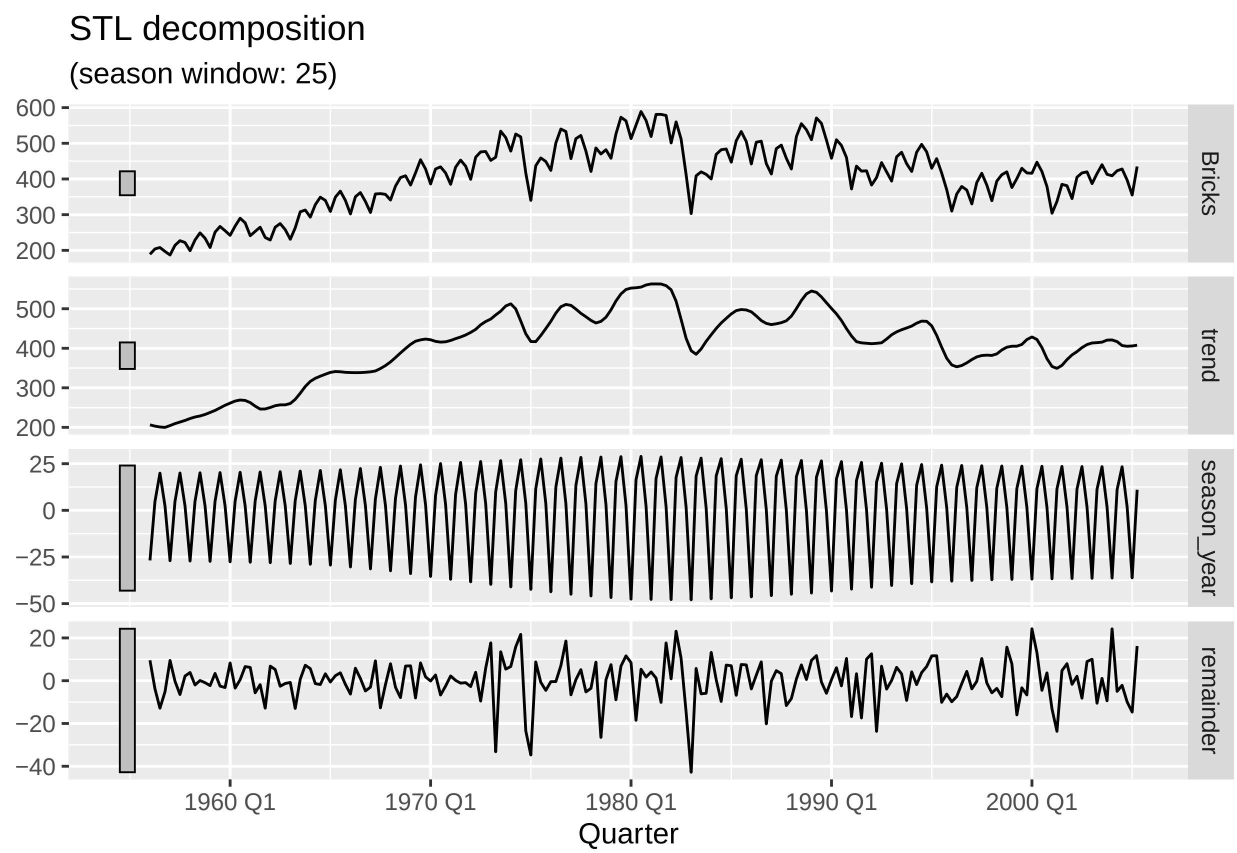
```
## # A tsibble: 198 x 2 [1Q]
##    Quarter Bricks
##      <qtr>  <dbl>
##  1 1956 Q1    189
##  2 1956 Q2    204
##  3 1956 Q3    208
##  4 1956 Q4    197
##  5 1957 Q1    187
##  6 1957 Q2    214
##  7 1957 Q3    227
##  8 1957 Q4    222
##  9 1958 Q1    199
## 10 1958 Q2    229
## # ... with 188 more rows
```

a) Use an STL decomposition to calculate the trend-cycle and seasonal indices. (Experiment with having fixed or changing seasonality.)

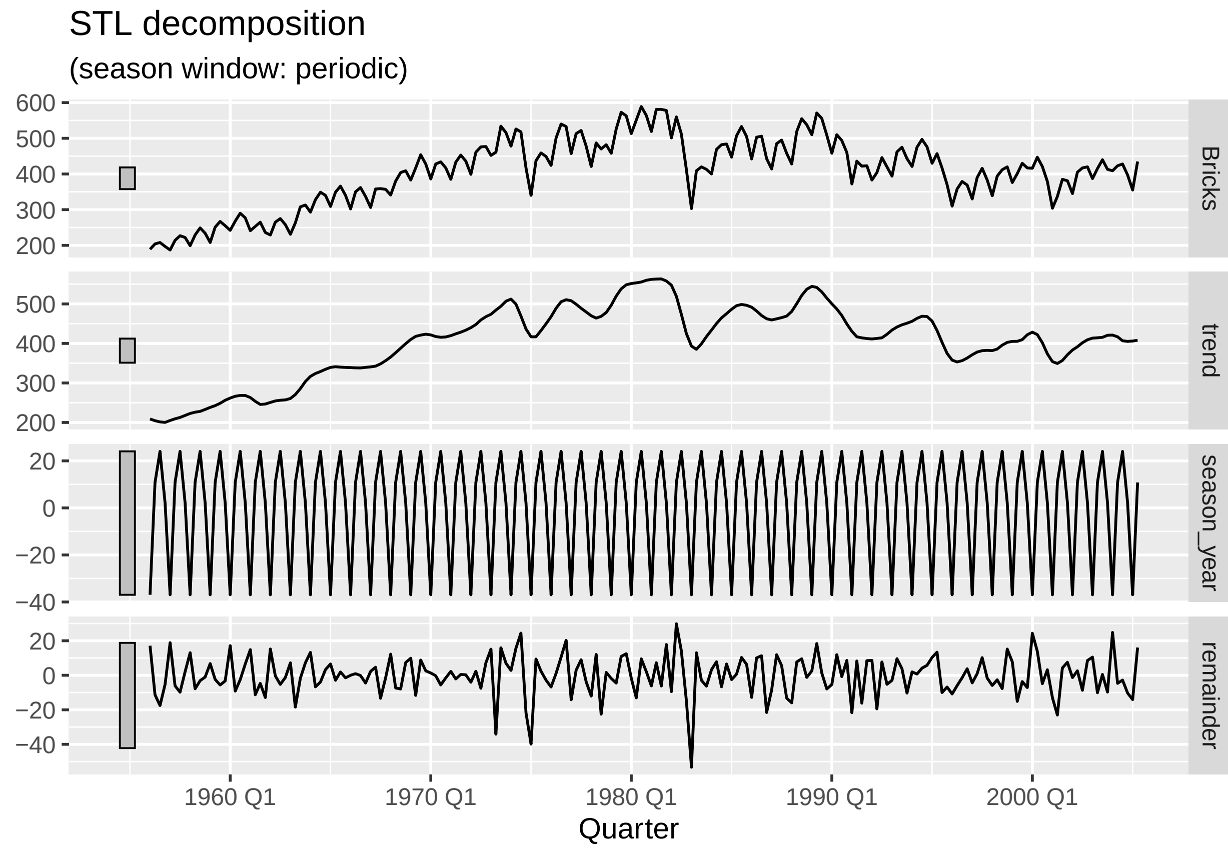
```r
dcmp <- bricks_aus_production %>%
  model(
    STL(Bricks ~ season(window = 13) #"periodic"
    )) %>%
  components()

autoplot(dcmp) +
  labs(title = "STL decomposition ",
       subtitle = "(season window: 13)")
```




```
dcmp <- bricks_aus_production %>%
  model(
    STL(Bricks ~ season(window = 25) #"periodic"
    )) %>%
  components()

autoplot(dcmp) +
  labs(title = "STL decomposition ",
       subtitle = "(season window: 25)")
```

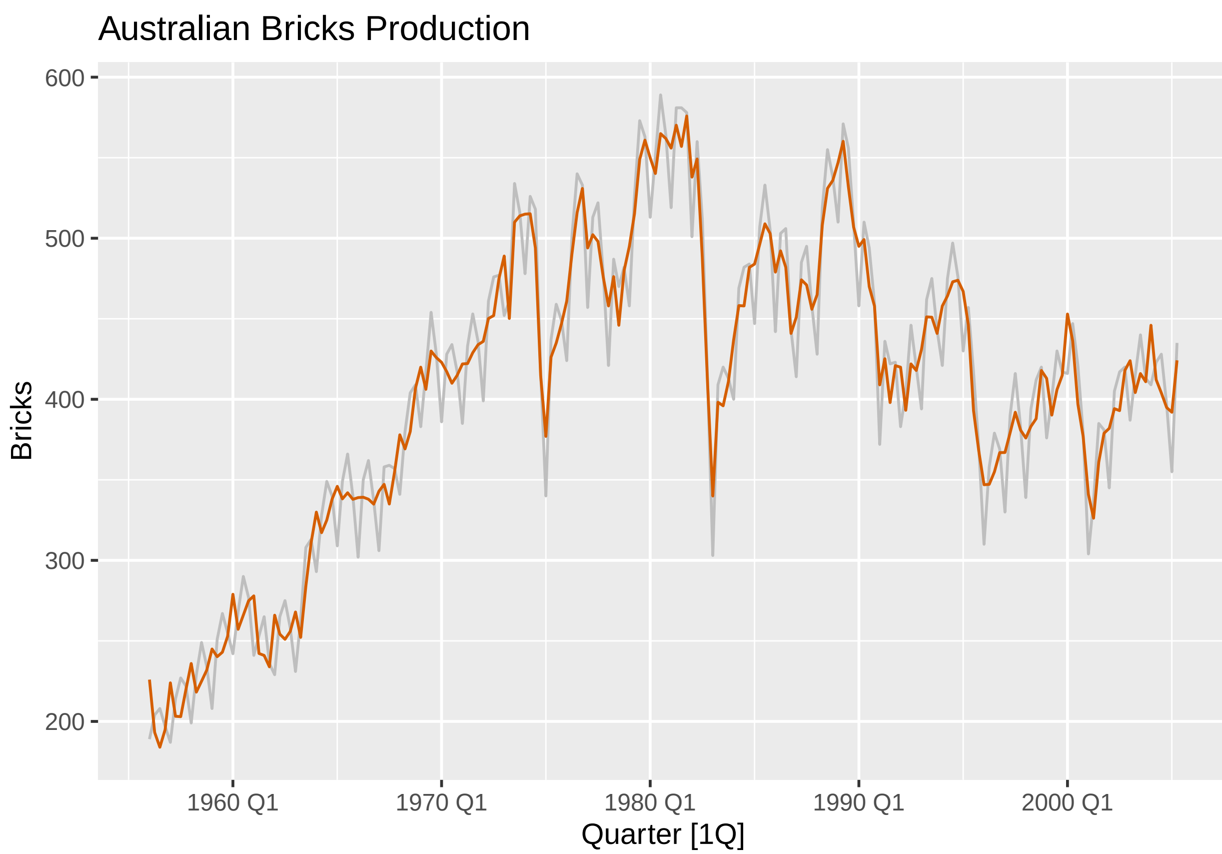



```
dcmp <- bricks_aus_production %>%
  model(
    STL(Bricks ~ season(window = "periodic") #, robust = TRUE
  )) %>%
  components()

autoplot(dcmp) +
  labs(title = "STL decomposition ",
       subtitle = "(season window: periodic)")
```



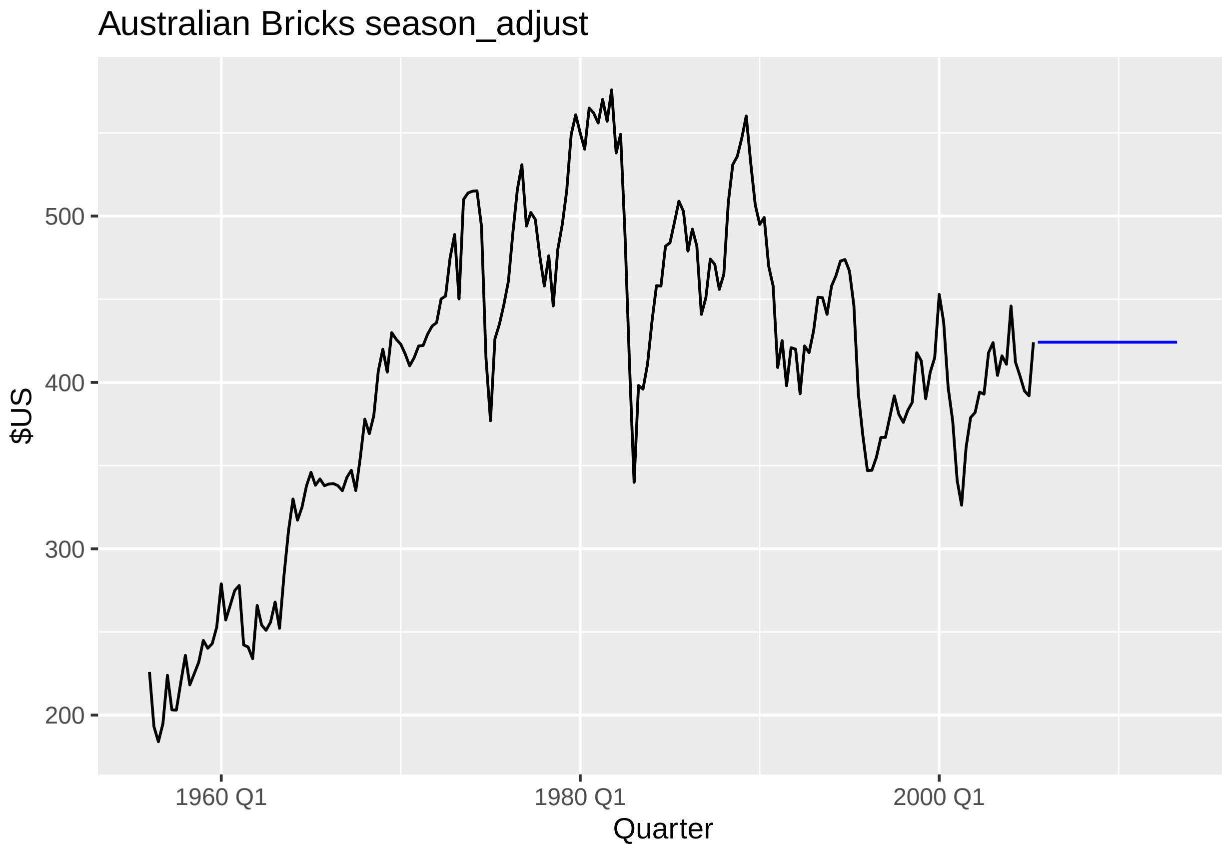
b) Compute and plot the seasonally adjusted data.

```
dcmp %>%
  as_tsibble() %>%
  autoplot(Bricks, colour="gray") +
  geom_line(aes(y=season_adjust), colour = "#D55E00") +
  labs(title = "Australian Bricks Production")
```



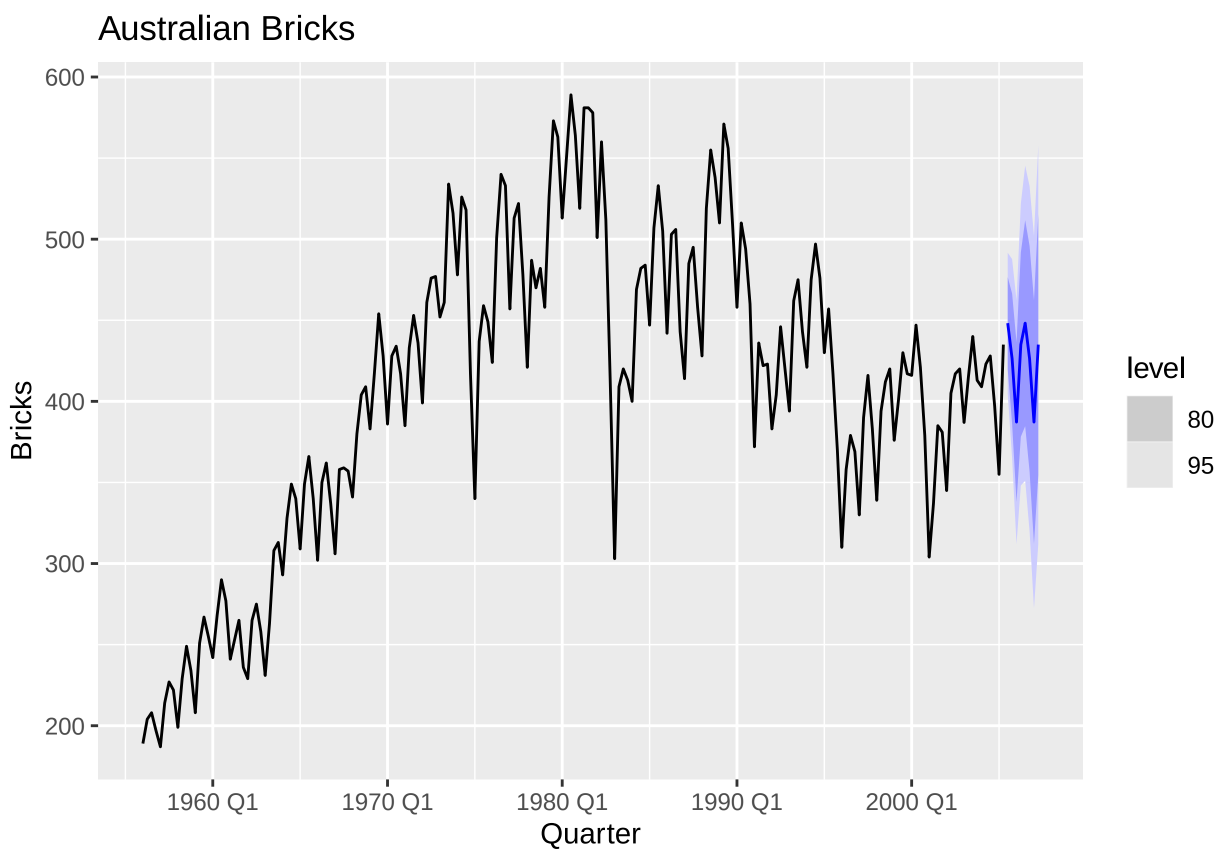
c) Use a naïve method to produce forecasts of the seasonally adjusted data.

```
dcmp2 <- dcmp %>%
  select(-.model)

bricks_fit <- dcmp2 %>%
  model(NAIVE(season_adjust))


# Generate forecasts for 32 quarters
bricks_fc <- bricks_fit %>% forecast(h = 32) # 4*12


#Plot
bricks_fc %>%
  autoplot(dcmp2,level = NULL) +
  labs(y = "$US",
       title = "Australian Bricks season_adjust") +
  guides(colour = guide_legend(title = "Forecast"))
```



d) Use decomposition_model() to reseasonalise the results, giving forecasts for the original data.

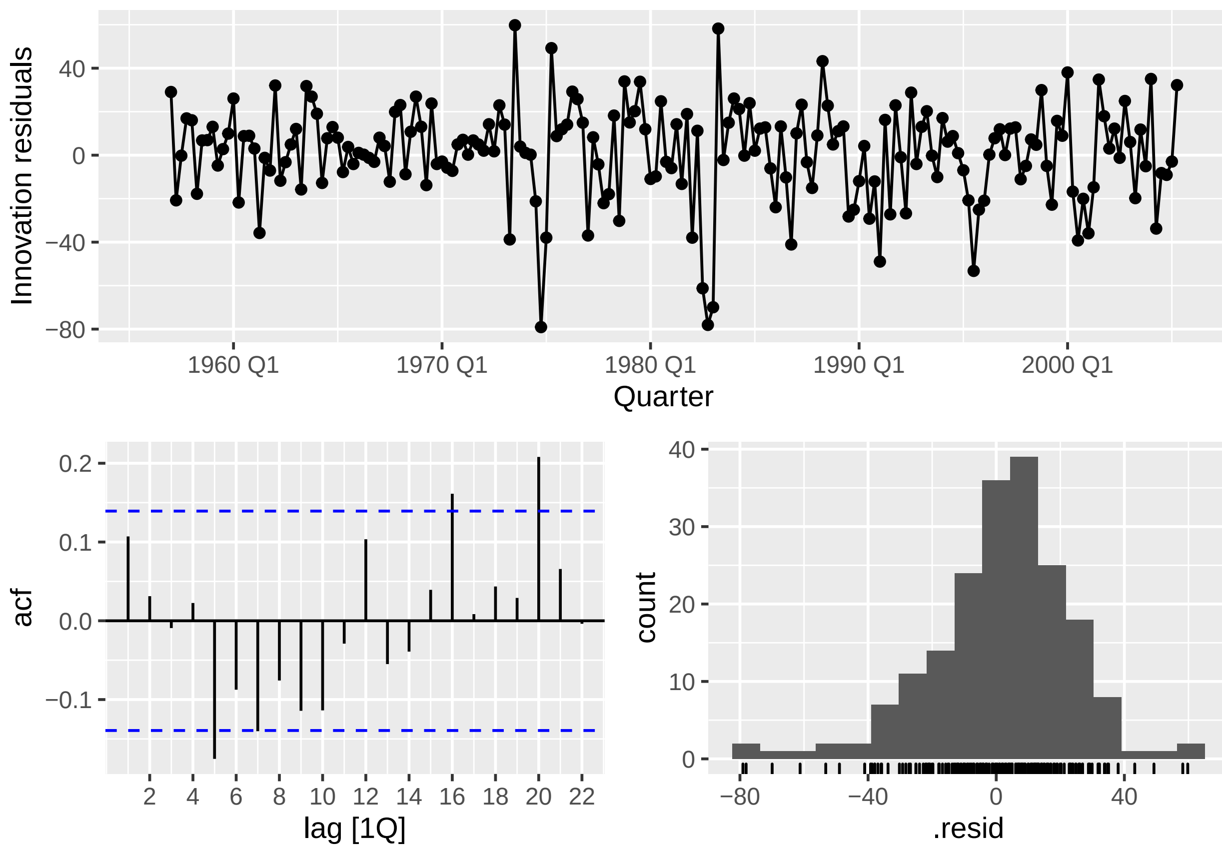
```
fit_dcmp <- bricks_aus_production %>%
  model(stlf = decomposition_model(
    STL(Bricks ~ season(window = "periodic")), #, robust = TRUE
    NAIVE(season_adjust)
  ))
fit_dcmp %>%
  forecast() %>%
  autoplot(bricks_aus_production)+
  labs(title = "Australian Bricks")
```



e) Do the residuals look uncorrelated?

```
fit_dcmp %>% gg_tsresiduals()
```

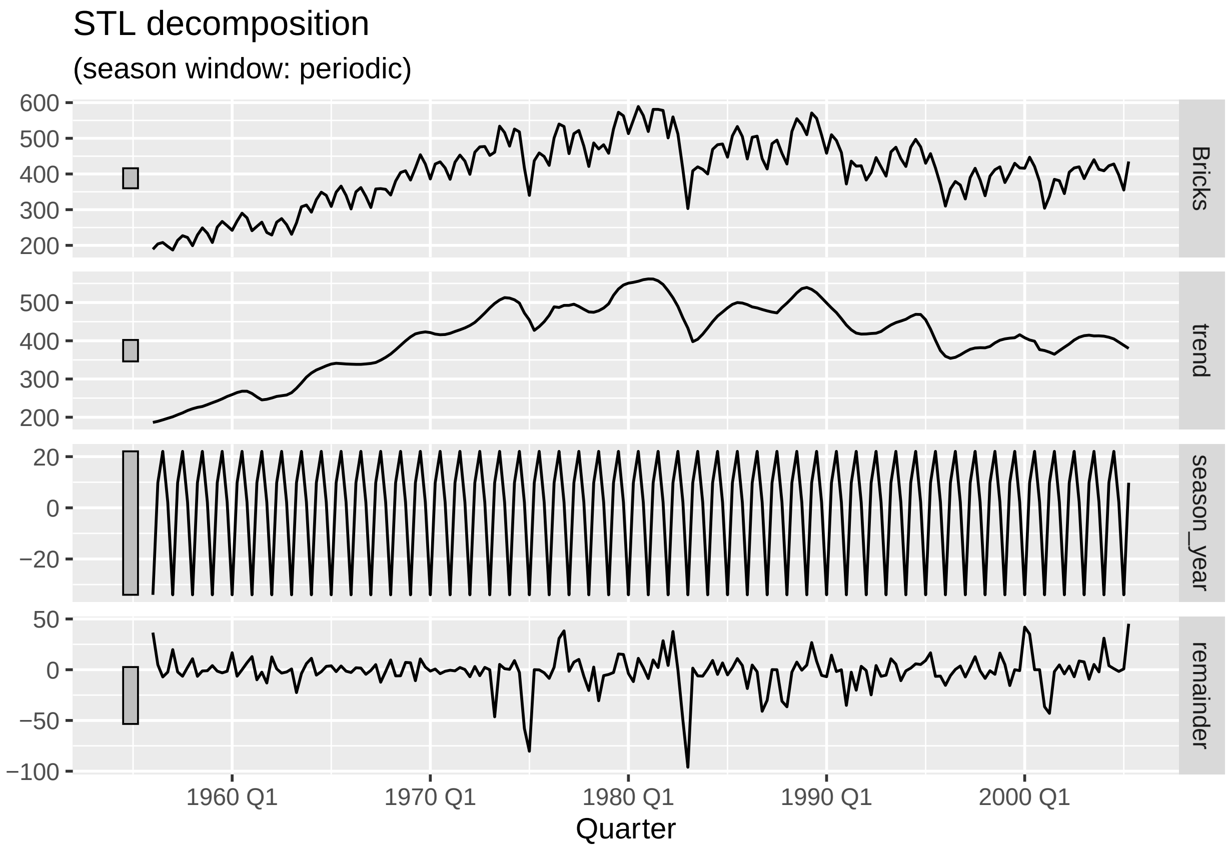
```
## Warning: Removed 4 row(s) containing missing values (geom_path).

## Warning: Removed 4 rows containing missing values (geom_point).

## Warning: Removed 4 rows containing non-finite values (stat_bin).
```



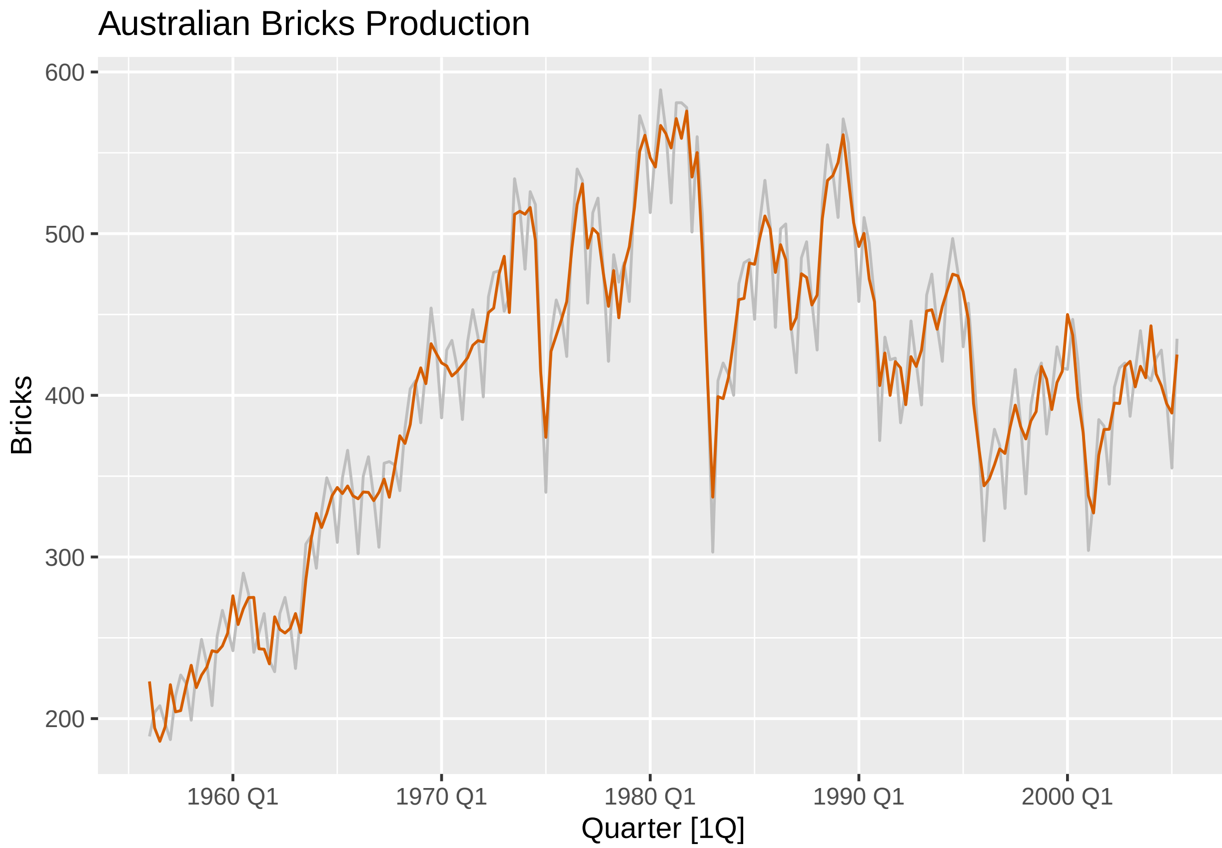
There are values lags for which the residuals are correlated, meaning that there is still information remaining on the residuals.

f) Repeat with a robust STL decomposition. Does it make much difference?

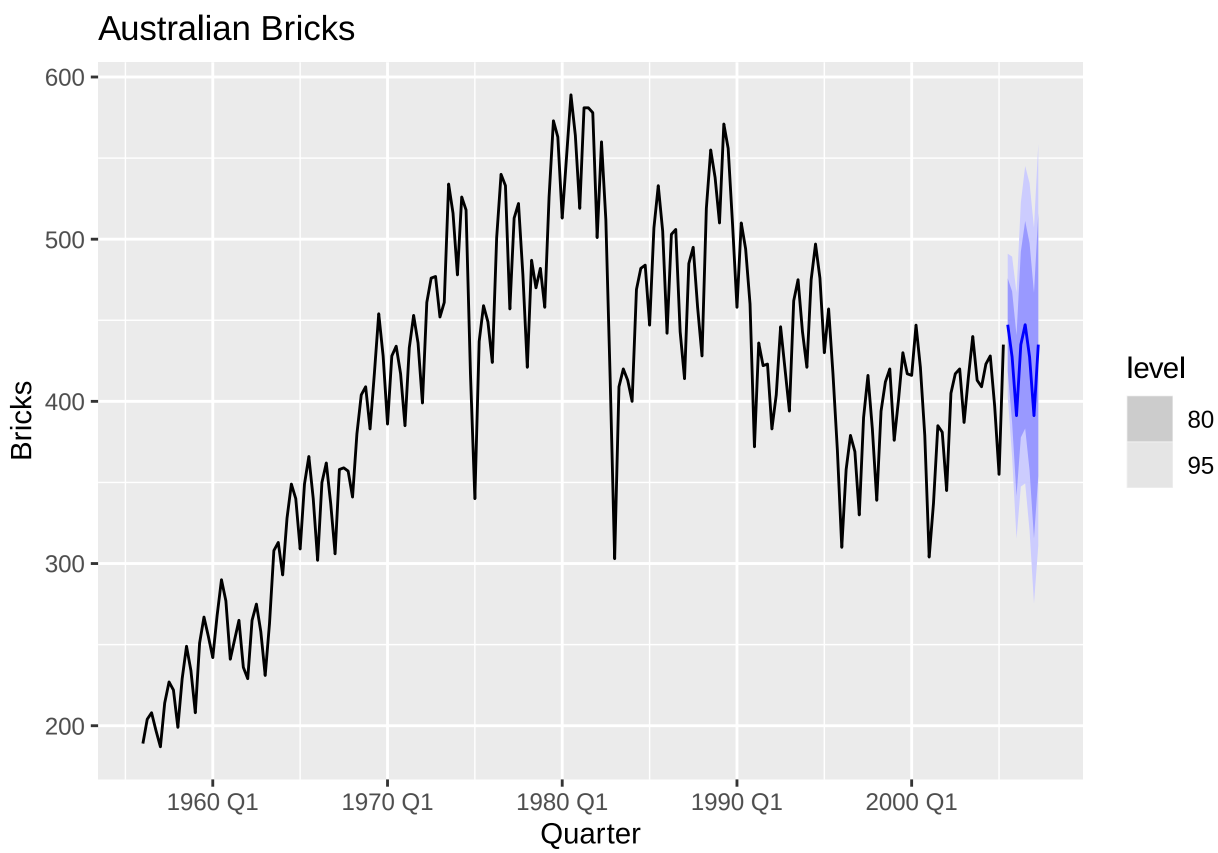
```
dcmp <- bricks_aus_production %>%
  model(
    STL(Bricks ~ season(window = "periodic"), #"periodic"
    robust = TRUE)) %>%
  components()

autoplot(dcmp) +
  labs(title = "STL decomposition ",
       subtitle = "(season window: periodic)")
```




```
dcmp %>%
  as_tsibble() %>%
  autoplot(Bricks, colour="gray") +
  geom_line(aes(y=season_adjust), colour = "#D55E00") +
  labs(title = "Australian Bricks Production")
```

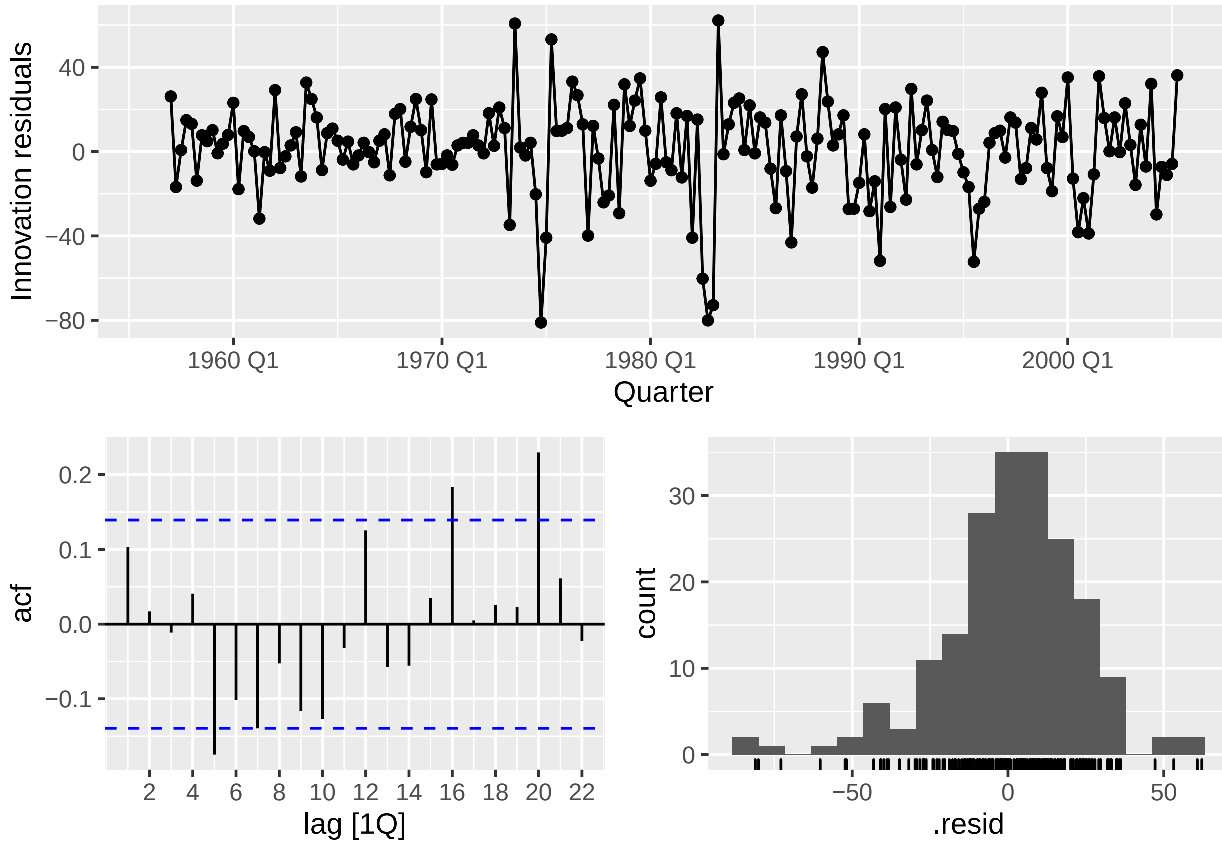



```
dcmp2 <- dcmp %>%
  select(-.model)

bricks_fit <- dcmp2 %>%
  model(NAIVE(season_adjust))


# Generate forecasts for 32 quarters
bricks_fc <- bricks_fit %>% forecast(h = 32) # 4*12

fit_dcmp <- bricks_aus_production %>%
  model(stlf = decomposition_model(
    STL(Bricks ~ season(window = "periodic"), robust = TRUE), #, robust = TRUE
    NAIVE(season_adjust)
  ))
fit_dcmp %>%
  forecast() %>%
  autoplot(bricks_aus_production)+
  labs(title = "Australian Bricks")
```




```
fit_dcmp %>% gg_tsresiduals()
```

```
## Warning: Removed 4 row(s) containing missing values (geom_path).

## Warning: Removed 4 rows containing missing values (geom_point).

## Warning: Removed 4 rows containing non-finite values (stat_bin).
```



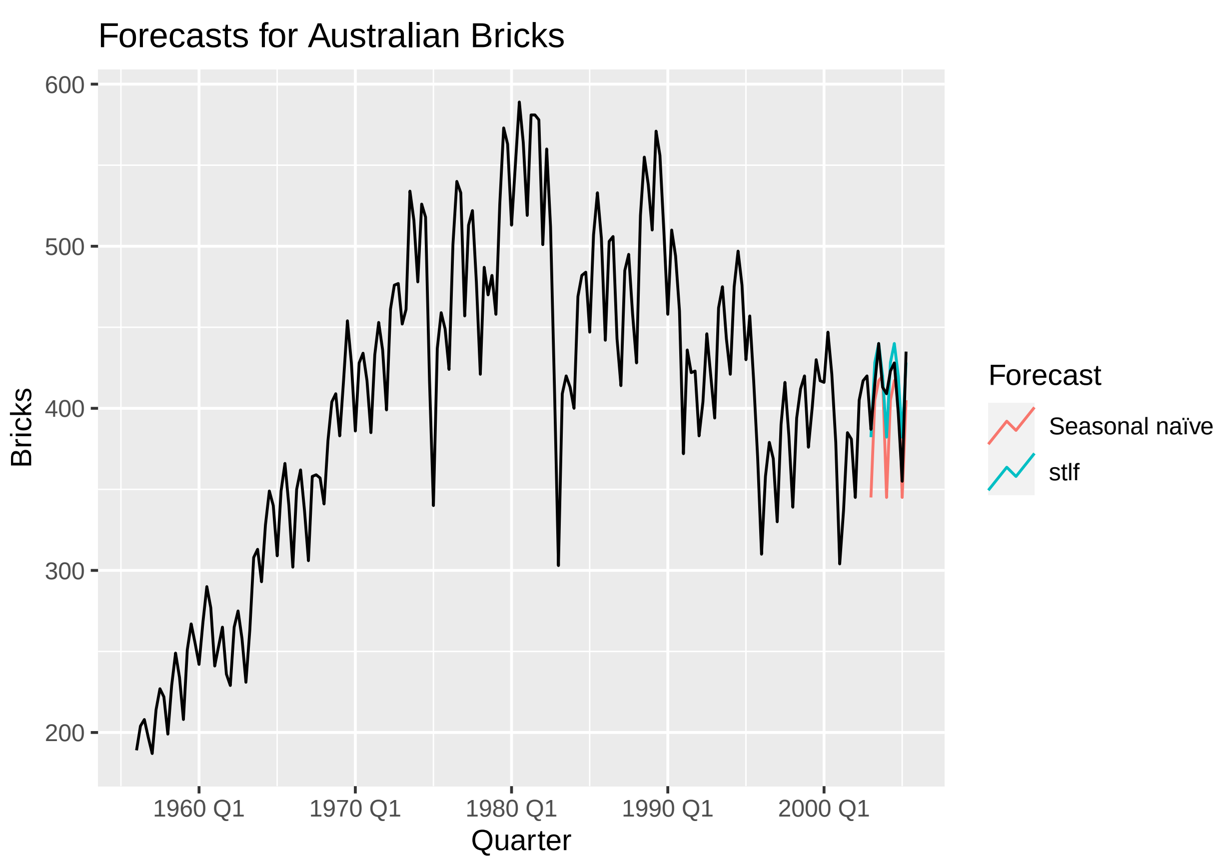
There looks like there isn't much difference, But we can see that the residuals are less spread in the Robust STL than in the normal one, which means that is has capture more information.

g) Compare forecasts from decomposition_model() with those from SNAIVE(), using a test set comprising the last 2 years of data. Which is better?

```
bricks_train <-bricks_aus_production %>%
  filter(year(Quarter) <= 2002)
bricks_train
```

```
## # A tsibble: 188 x 2 [1Q]
##    Quarter Bricks
##      <qtr>  <dbl>
##  1 1956 Q1    189
##  2 1956 Q2    204
##  3 1956 Q3    208
##  4 1956 Q4    197
##  5 1957 Q1    187
##  6 1957 Q2    214
##  7 1957 Q3    227
##  8 1957 Q4    222
##  9 1958 Q1    199
## 10 1958 Q2    229
## # ... with 178 more rows
```

```r
bricks_fit <- bricks_train %>%
  model(
    `Seasonal naïve` = SNAIVE(Bricks),
    stlf = decomposition_model(
      STL(Bricks ~ season(window = "periodic"), robust = TRUE), #, robust = TRUE
      NAIVE(season_adjust))
  )

bricks_fc <- bricks_fit %>%
  forecast(h = 10)


bricks_fc %>%
  autoplot(
    bricks_aus_production,
    level = NULL
  ) +
  labs(title = "Forecasts for Australian Bricks") +
  guides(colour = guide_legend(title = "Forecast"))
```




```r
accuracy(bricks_fc, bricks_aus_production)
```

```
## # A tibble: 2 x 10
##   .model         .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE    ACF1
##   <chr>          <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl>
```

```
## 1 Seasonal naïve Test  17.8   29.2  23.8  4.32  5.82 0.656 0.592 -0.186
## 2 stlf           Test  -4.94  15.7  12.6 -1.31  3.17 0.347 0.318  0.0292
```

It looks like the decomposition_model achive better metrics that the SNAIVE method in all the metrics and in the graph

5.11 exercises 12

a) Extract data from the Gold Coast region using filter() and aggregate total overnight trips (sum over Purpose) using summarise(). Call this new dataset gc_tourism.

```
tourism
```

```
## # A tsibble: 24,320 x 5 [1Q]
## # Key:       Region, State, Purpose [304]
##    Quarter Region   State           Purpose  Trips
##      <qtr> <chr>    <chr>           <chr>    <dbl>
##  1 1998 Q1 Adelaide South Australia Business  135.
##  2 1998 Q2 Adelaide South Australia Business  110.
##  3 1998 Q3 Adelaide South Australia Business  166.
##  4 1998 Q4 Adelaide South Australia Business  127.
##  5 1999 Q1 Adelaide South Australia Business  137.
##  6 1999 Q2 Adelaide South Australia Business  200.
##  7 1999 Q3 Adelaide South Australia Business  169.
##  8 1999 Q4 Adelaide South Australia Business  134.
##  9 2000 Q1 Adelaide South Australia Business  154.
## 10 2000 Q2 Adelaide South Australia Business  169.
## # ... with 24,310 more rows
```

```
gc_tourism <- tourism %>%
  filter(Region == "Gold Coast") %>%
  #group_by(Purpose) %>%
  summarise(TotalTrips = sum(Trips)) #%>%
  #select(Quarter, Purpose, TotalTrips)
gc_tourism
```

```
## # A tsibble: 80 x 2 [1Q]
##    Quarter TotalTrips
##      <qtr>      <dbl>
##  1 1998 Q1       827.
##  2 1998 Q2       681.
##  3 1998 Q3       839.
##  4 1998 Q4       820.
##  5 1999 Q1       987.
##  6 1999 Q2       751.
##  7 1999 Q3       822.
##  8 1999 Q4       914.
##  9 2000 Q1       871.
## 10 2000 Q2       780.
## # ... with 70 more rows
```

b) Using slice() or filter(), create three training sets for this data excluding the last 1, 2 and 3 years. For example, gc_train_1 <- gc_tourism %>% slice(1:(n()-4)).

```
gc_train_1 <- gc_tourism %>% slice(1:(n()-4))
gc_train_1
```

```
## # A tsibble: 76 x 2 [1Q]
##    Quarter TotalTrips
##      <qtr>      <dbl>
##  1 1998 Q1       827.
##  2 1998 Q2       681.
##  3 1998 Q3       839.
##  4 1998 Q4       820.
##  5 1999 Q1       987.
##  6 1999 Q2       751.
##  7 1999 Q3       822.
##  8 1999 Q4       914.
##  9 2000 Q1       871.
## 10 2000 Q2       780.
## # ... with 66 more rows
```

```
gc_train_2 <- gc_tourism %>% slice(1:(n()-8))
gc_train_2
```

```
## # A tsibble: 72 x 2 [1Q]
##    Quarter TotalTrips
##      <qtr>      <dbl>
##  1 1998 Q1       827.
##  2 1998 Q2       681.
##  3 1998 Q3       839.
##  4 1998 Q4       820.
##  5 1999 Q1       987.
##  6 1999 Q2       751.
##  7 1999 Q3       822.
##  8 1999 Q4       914.
##  9 2000 Q1       871.
## 10 2000 Q2       780.
## # ... with 62 more rows
```

```
gc_train_3 <- gc_tourism %>% slice(1:(n()-12))
gc_train_3
```

```
## # A tsibble: 68 x 2 [1Q]
##    Quarter TotalTrips
##      <qtr>      <dbl>
##  1 1998 Q1       827.
##  2 1998 Q2       681.
##  3 1998 Q3       839.
##  4 1998 Q4       820.
##  5 1999 Q1       987.
##  6 1999 Q2       751.
##  7 1999 Q3       822.
```

```
##   8 1999 Q4       914.
##   9 2000 Q1       871.
## 10 2000 Q2       780.
## # ... with 58 more rows
```

```r
gc_test_1 <- gc_tourism %>% slice((n()-4+1):(n()))
gc_test_1
```

```
## # A tsibble: 4 x 2 [1Q]
##   Quarter TotalTrips
##     <qtr>      <dbl>
## 1 2017 Q1      1140.
## 2 2017 Q2       904.
## 3 2017 Q3      1053.
## 4 2017 Q4       908.
```

```r
gc_test_2 <- gc_tourism %>% slice((n()-8+1):(n()-4))
gc_test_2
```

```
## # A tsibble: 4 x 2 [1Q]
##   Quarter TotalTrips
##     <qtr>      <dbl>
## 1 2016 Q1       933.
## 2 2016 Q2       854.
## 3 2016 Q3       850.
## 4 2016 Q4      1066.
```

```r
gc_test_3 <- gc_tourism %>% slice((n()-12+1):(n()-8))
gc_test_3
```

```
## # A tsibble: 4 x 2 [1Q]
##   Quarter TotalTrips
##     <qtr>      <dbl>
## 1 2015 Q1       943.
## 2 2015 Q2       800.
## 3 2015 Q3       895.
## 4 2015 Q4      1017.
```

c)Compute one year of forecasts for each training set using the seasonal naïve (SNAIVE()) method. Call these gc_fc_1, gc_fc_2 and gc_fc_3, respectively.

```r
gc_fit_1 <- gc_train_1 %>%
  model(SNAIVE(TotalTrips))
# Generate forecasts for 4 quarters
gc_fc_1 <- gc_fit_1 %>% forecast(h = 4)


gc_fit_2 <- gc_train_2 %>%
  model(SNAIVE(TotalTrips))
# Generate forecasts for 4 quarters
gc_fc_2 <- gc_fit_2 %>% forecast(h = 4)
```

```r
gc_fit_3 <- gc_train_3 %>%
  model(SNAIVE(TotalTrips))
# Generate forecasts for 4 quarters
gc_fc_3 <- gc_fit_3 %>% forecast(h = 4)
```

d)Use accuracy() to compare the test set forecast accuracy using MAPE. Comment on these.

```r
accuracy(gc_fc_1, gc_test_1)
```

```
## # A tibble: 1 x 10
##   .model             .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE   ACF1
##   <chr>              <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 SNAIVE(TotalTrips) Test   75.1  167.  154.  6.36  15.1   NaN   NaN -0.410
```

```r
accuracy(gc_fc_2, gc_test_2)
```

```
## # A tibble: 1 x 10
##   .model             .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE   ACF1
##   <chr>              <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 SNAIVE(TotalTrips) Test   12.0  43.1  39.5  1.14  4.32   NaN   NaN -0.792
```

```r
accuracy(gc_fc_3, gc_test_3)
```

```
## # A tibble: 1 x 10
##   .model             .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>              <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 SNAIVE(TotalTrips) Test   35.8  91.4  83.9  3.56  9.07   NaN   NaN 0.239
```

If we take a look at the accuracy using MAPE, we can see that for the model 2 was the lowest value at 4.320729, follow by the model 3 at 9.067368, while model 1 got the highest value at 15.06055.