DATA624-HW1-TimeSeries

FPP-Hyndman exercises 2.1, 2.2, 2.3, 2.6

Michael Y.

2/09/2020

Contents

```r
library(fpp2)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: forecast
```

```
## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## Registered S3 methods overwritten by 'forecast':
##   method             from
##   fitted.fracdiff    fracdiff
##   residuals.fracdiff fracdiff
```

```
## Loading required package: fma
```

```
## Loading required package: expsmooth
```

Homework 1 - Time Series

Please submit exercises 2.1, 2.2, 2.3 and 2.6 from the Hyndman online Forecasting book.
Please submit both your Rpubs link as well as attach the .rmd file with your code.

2.1 Use the help function to explore what the series `gold`, `woolyrnq` and `gas` represent.

```r
### Gold
help(gold)
```

```
## starting httpd help server ... done
```

```r
gold.title <- "Daily morning gold prices in US dollars. 1 January 1985 - 31 March 1989."

### Woolyrnq
help(woolyrnq)
woolyrnq.title <- "Quarterly production of woollen yarn in Australia: tonnes. Mar 1965 - Sep 1994."

### Gas
help(gas)
gas.title <- "Australian monthly gas production: 1956-1995."
```

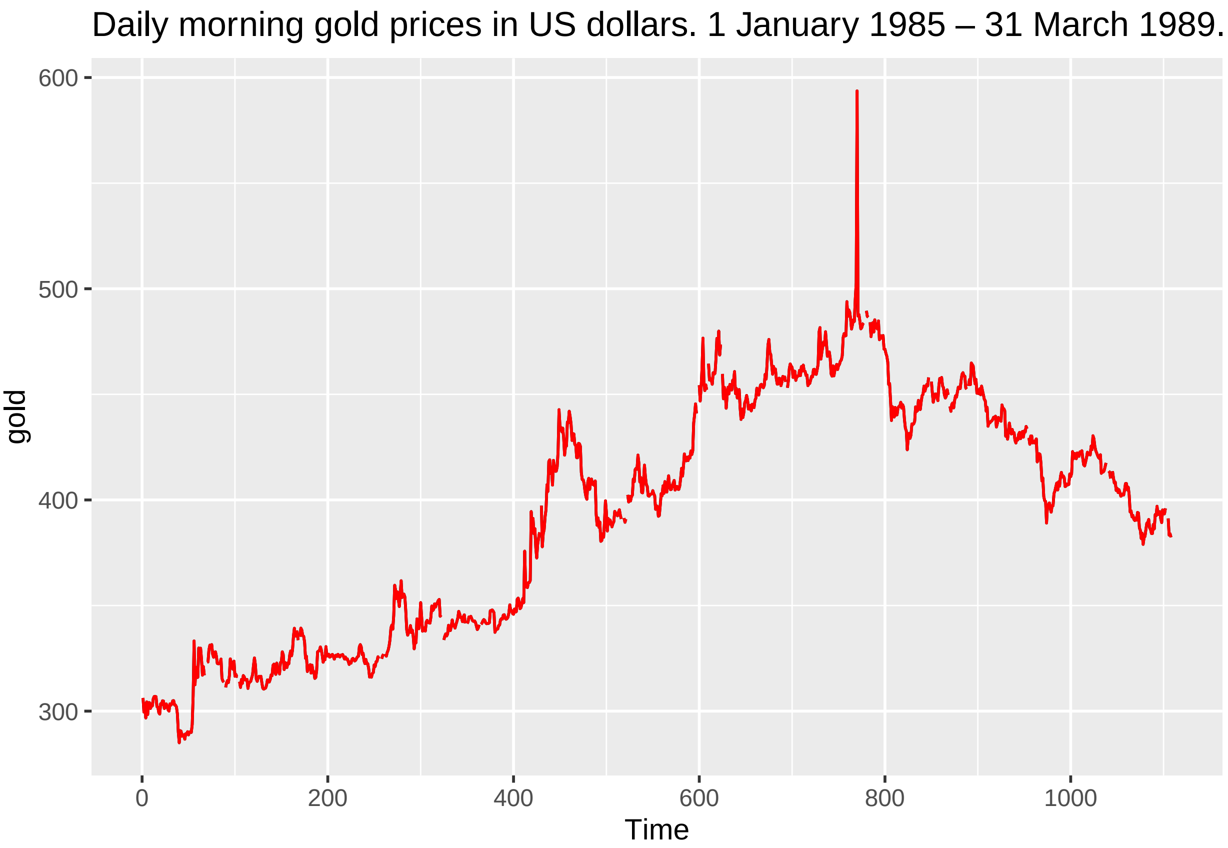
The `gold` series represents Daily morning gold prices in US dollars. 1 January 1985 – 31 March 1989.

The `woolyrnq` series represents Quarterly production of woollen yarn in Australia: tonnes. Mar 1965 – Sep 1994.

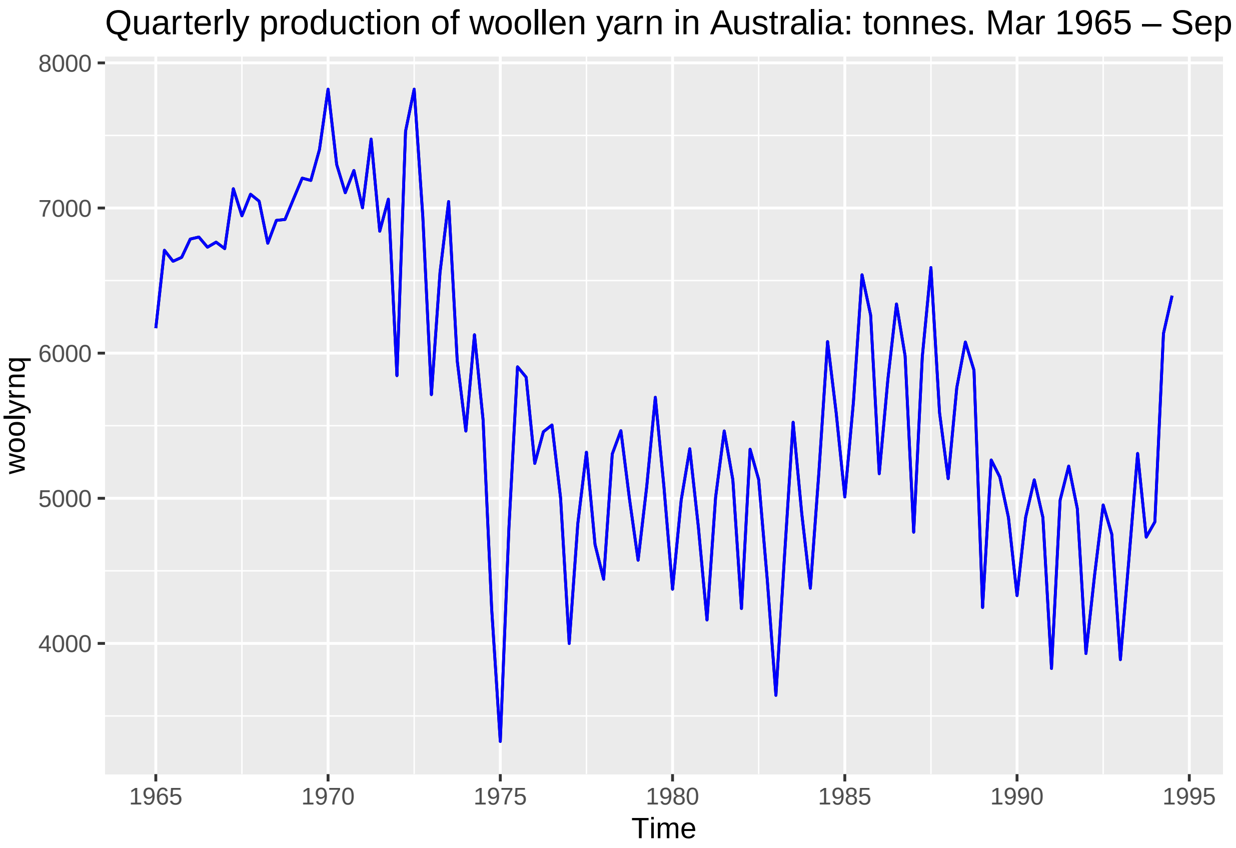
The `gas` series represents Australian monthly gas production: 1956–1995.

a. Use `autoplot()` to plot each of these in separate plots.

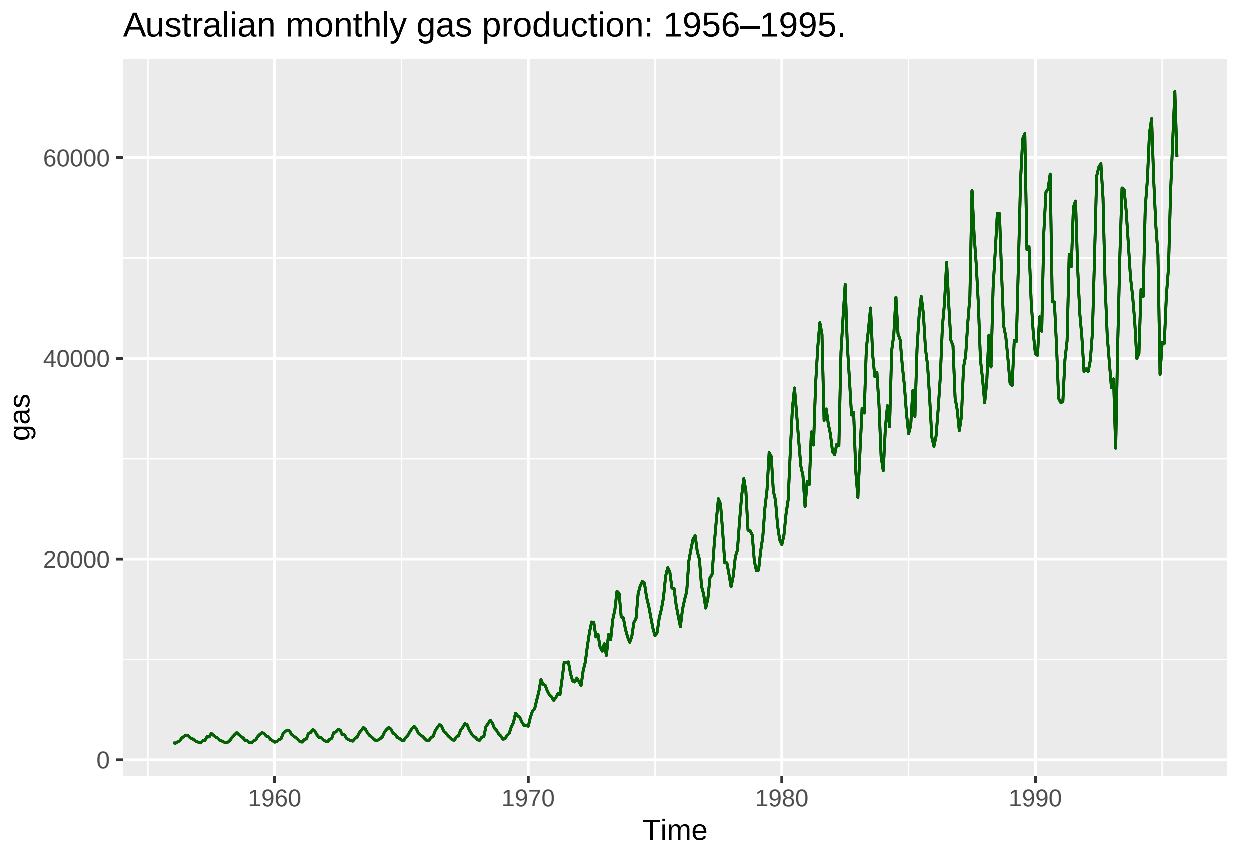
```
autoplot(gold) + ggtitle(gold.title) + geom_line(color="red")
```



```
autoplot(woolyrnq) + ggtitle(woolyrnq.title) + geom_line(color="blue")
```



```
autoplot(gas) + ggtitle(gas.title) + geom_line(color="darkgreen")
```



b. What is the frequency of each series? Hint: apply the frequency() function.

```
## Frequency - gold
(gold.freq <- frequency(gold))
```

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

```
## Frequency - woolyrnq
(woolyrnq.freq <- frequency(woolyrnq))
```

```
## [1] 4
```

```
## Frequency - gas
(gas.freq <- frequency(gas))
```

```
## [1] 12
```

The series for `gold` is stored with frequency = 1, which suggests “Annual”, but the frequency is actually daily(Weekday), i.e, 5 days per week, or 260 observations per year.

The frequency for `woolyrnq` is 4, which corresponds to Quarterly.

The frequency for `gas` is 12, which corresponds to Monthly.

c. Use which.max() to spot the outlier in the gold series. Which observation was it?

```
(gold.max.obs <- which.max(gold))
```

```
## [1] 770
```

```
(gold.max.val <- gold[gold.max.obs])
```

```
## [1] 593.7
```

The outlier is observation number 770, for which the associated price is 593.7 .

The data includes `NA` values for holidays such as New Years Day, Good Friday, Easter Monday, Christmas, Boxing Day, and the Bank Holiday which occur on Mondays at the beginning and end of May, and at the end of August.

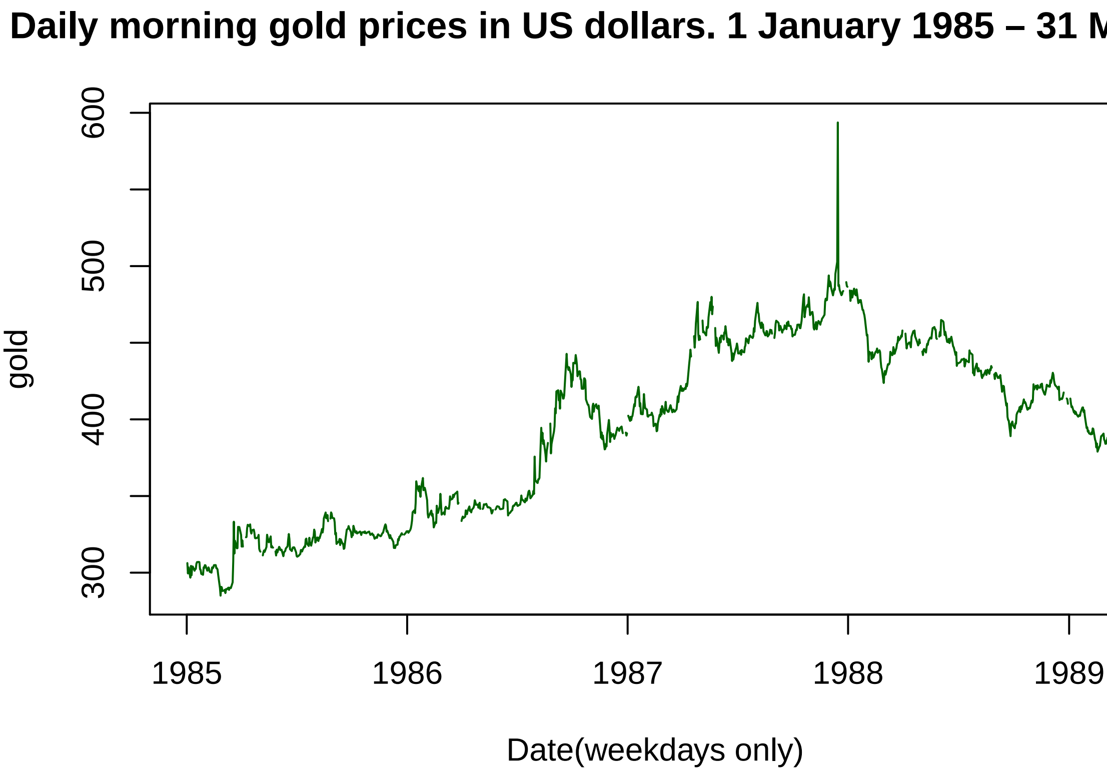
The problem with using “ts” as the data structure for weekday-only data is that Saturdays and Sundays have been omitted, but `ts` requires data to be equally spaced.

There are 1108 items in the `gold` data set, of which 34 are `NA` .

This corresponds to the number of weekdays from 1/2/1985 through 3/31/1989, inclusive. (January 1, 1985 has been omitted from the dataset, otherwise the first observatious would be `NA` .)

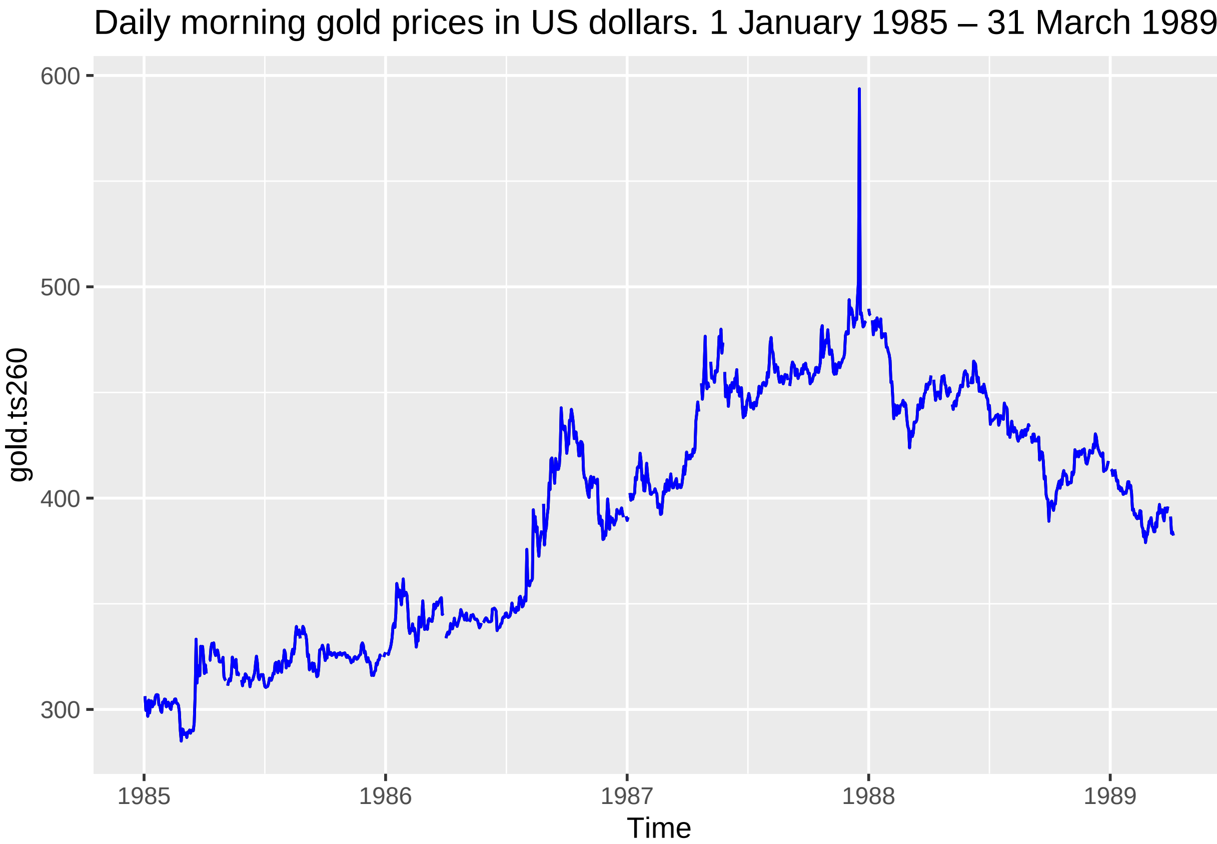
```
gold.days     <- seq( as.Date("1985/1/2", format="%Y/%m/%d"),
                      as.Date("1989/3/31", format="%Y/%m/%d"),"days")
gold.weekdays <- gold.days[ ! weekdays(gold.days) %in% c("Saturday", "Sunday") ]
plot(x=gold.weekdays,y=gold,type="l",col="darkgreen",
     xlab="Date(weekdays only)",main=gold.title)
```

We can create a sequence of dates which correspond to the observations, and make a plot which better labels the dates:



```
gold.ts260 <- ts(gold,start=c(1985,2),frequency = 260)
autoplot(gold.ts260) + ggtitle(gold.title) + geom_line(color="blue")
```

We can get a similar graph by putting the data into a `ts` object while specifying 260 observations per year:



Of course, doing this properly (with `ts`) would require expanding the dataset to include `NA` values for every Saturday and Sunday during the above time period, and would require 1551 rather than 1108 "observations".

```
gold.df        <- data.frame(gold.weekdays,gold)
gold.df[gold.max.obs,]
```

We can put the dates and prices into a data frame, and we can observe the date associated with the outlier:

```
##     gold.weekdays  gold
## 770    1987-12-15 593.7
```

2.2. Download the file `tute1.csv` from the book website, open it in Excel (or some other spreadsheet application), and review its contents. You should find four columns of information.

Columns *B* through *D* each contain a quarterly series, labelled `Sales`, `AdBudget` and `GDP`.

- `Sales` contains the quarterly sales for a small company over the period 1981-2005.
- `AdBudget` is the advertising budget and
- `GDP` is the gross domestic product.

All series have been adjusted for inflation.

a. You can read the data into R with the following script:

```
tute1 <- read.csv("tute1.csv", header=TRUE)
View(tute1)

## we can load from web, below
##tute11 <- readr::read_csv("https://otexts.com/fpp3/extrafiles/tute1.csv")
##View(tute11)
```

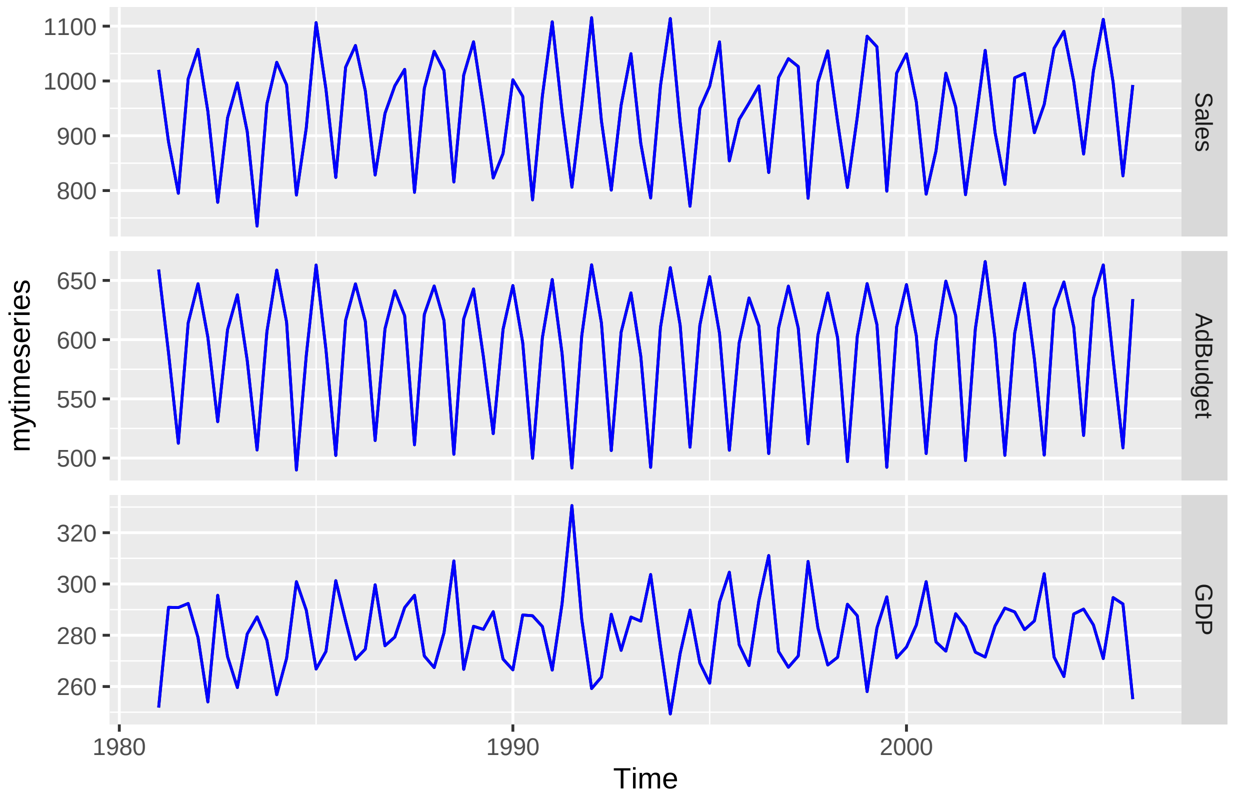
b. Convert the data to time series

```
mytimeseries <- ts(tute1[,-1], start=1981, frequency=4)
```

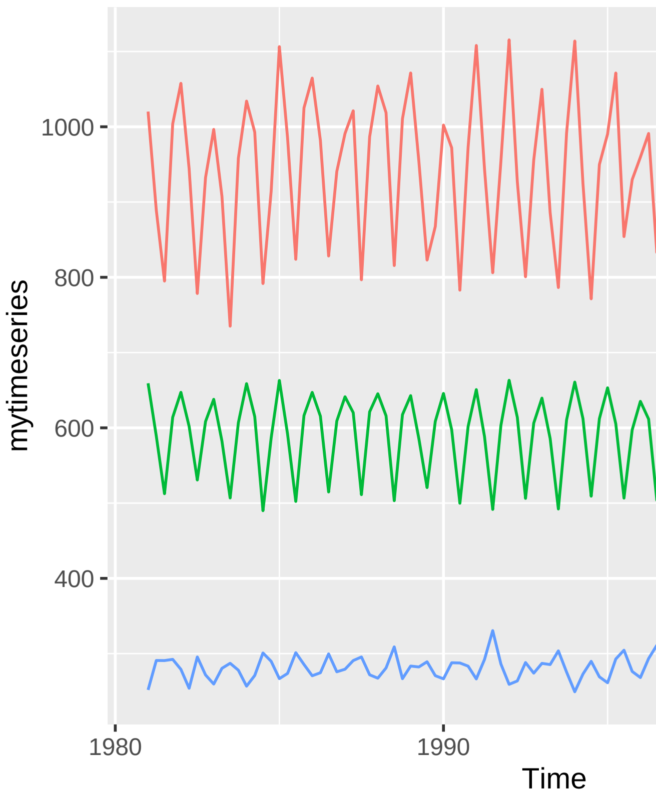
(The [,-1] removes the first column which contains the quarters as we don't need them now.)

c. Construct time series plots of each of the three series

```
autoplot(mytimeseries, facets=TRUE) + geom_line(color=c("blue"))
```



```
autoplot(mytimeseries, facets=F)
```



Check what happens when you don't include `facets=TRUE`.

If omitting "facets=TRUE", then all three series are plotted on the same axes. In this case because the values of the data do not intersect each other, the results can still be clearly observed. However, for other data sets with data of very different magnitudes, the result could be a graph which may not allow for easy viewing of the smallest data set.

2.3. Download some monthly Australian retail data from the book website. These represent retail sales in various categories for different Australian states, and are stored in a MS-Excel file.

a. You can read the data into R with the following script:

```
#### readxl does not read straight from URL without local download
####retaildata <- readxl::read_excel("https://otexts.com/fpp2/extrafiles/retail.xlsx", skip=1)
retaildata <- readxl::read_excel("retail.xlsx", skip=1)
View(retaildata)
```

The second argument (`skip=1`) is required because the Excel sheet has two header rows.

b. Select one of the time series as follows (but replace the column name with your own chosen column):

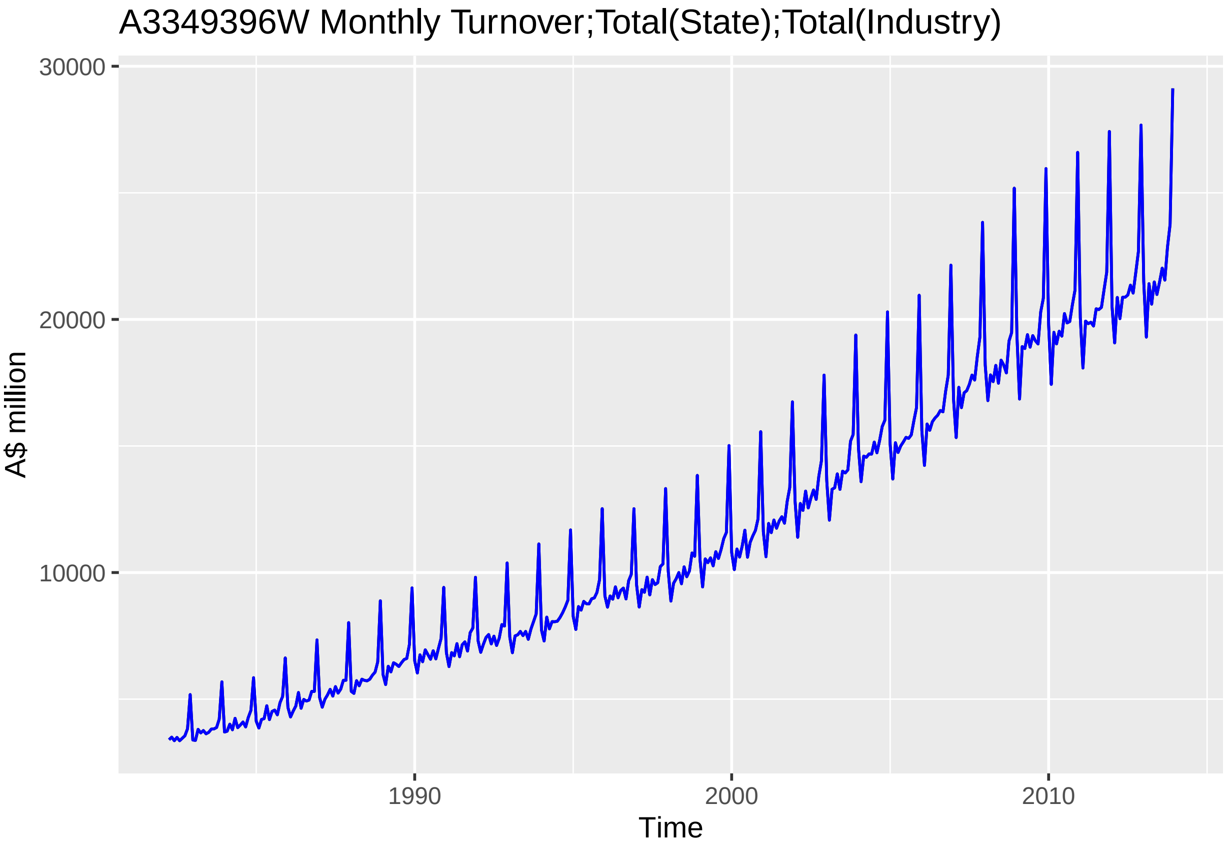
```
mycode <- "A3349396W"
mytitle <-  "Monthly Turnover;Total(State);Total(Industry)"
mymain <- paste(mycode,mytitle)
myts <- ts(retaildata[,"A3349396W"],
  frequency=12, start=c(1982,4))
```

The rightmost column: A3349396W: “Turnover;Total(State);Total(Industry)”

c. Explore your chosen retail time series using the following functions:

`autoplot()`, `ggseasonplot()`, `ggsubseriesplot()`, `gglagplot()`, `ggAcf()`

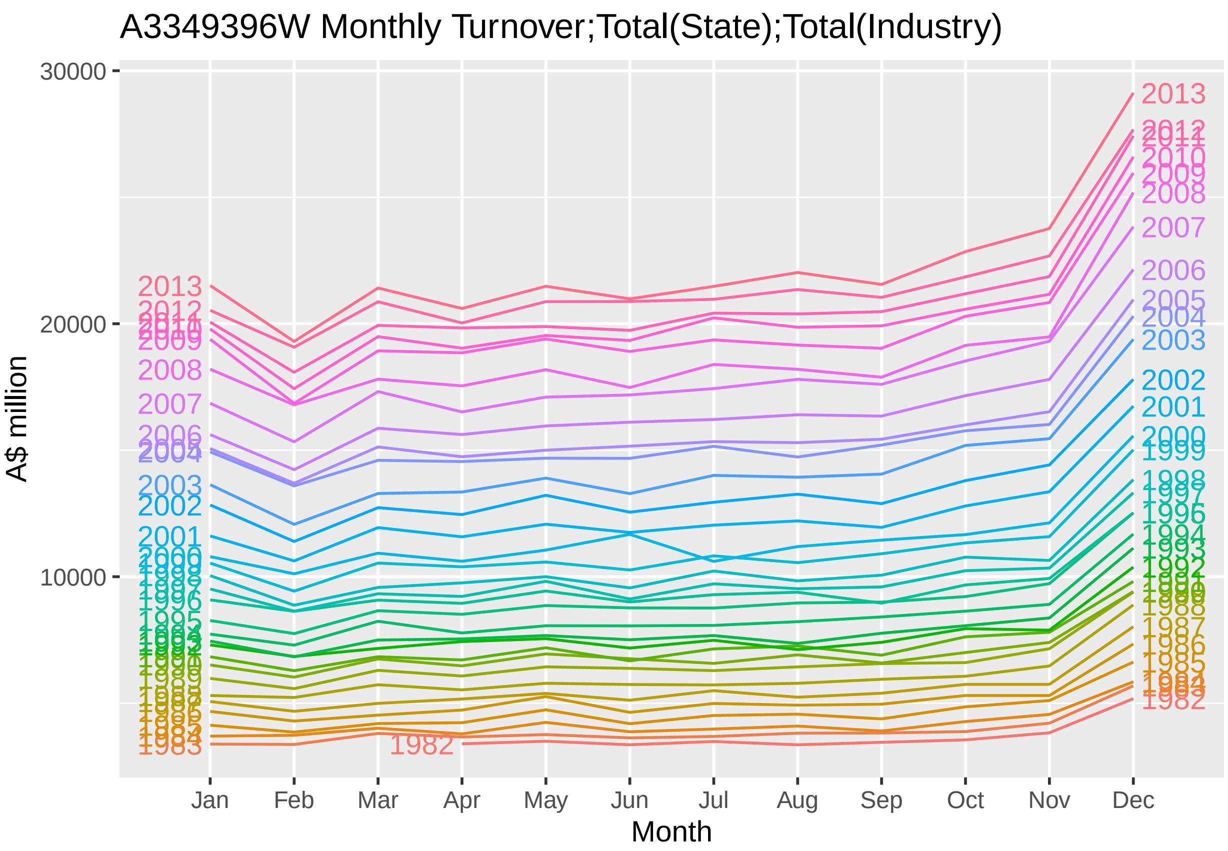
```
autoplot(object=myts)+
  geom_line(color=c("blue"))+
  ylab("A$ million") +
  ggtitle(mymain)
```



Autoplot

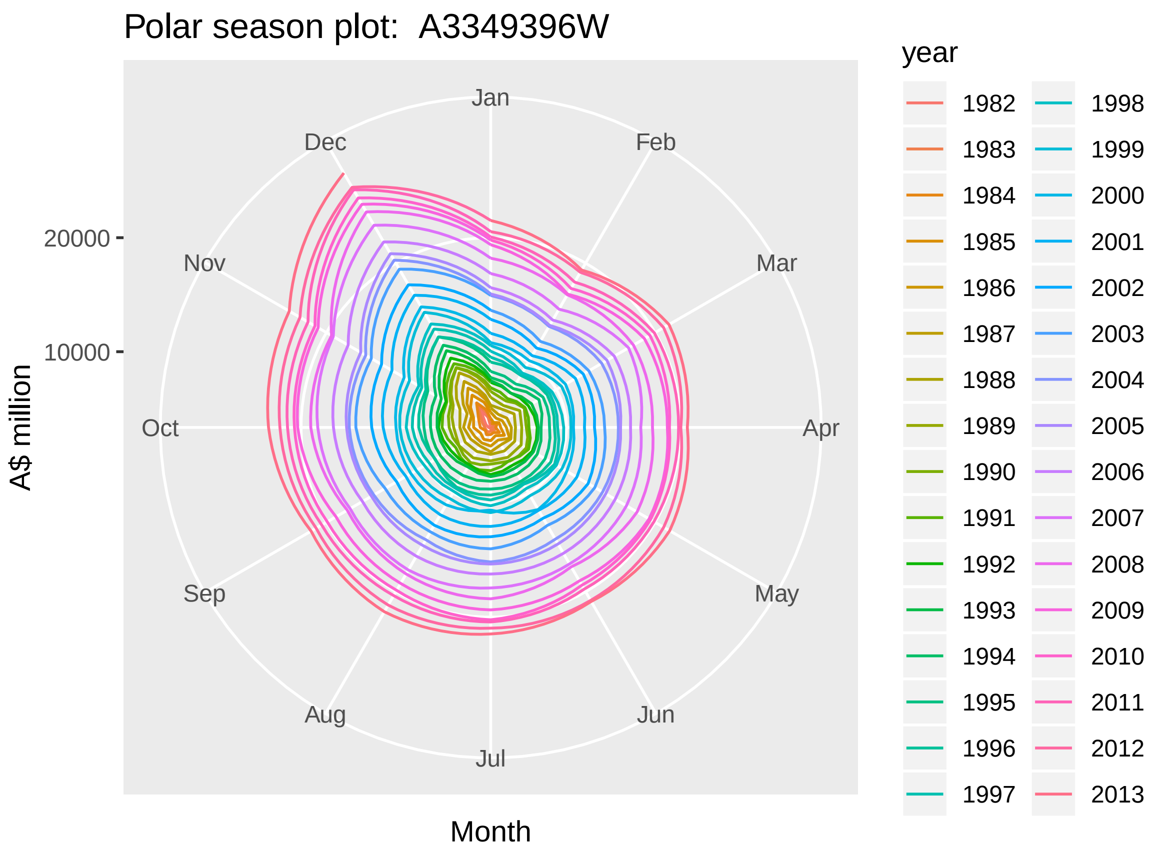
The above plot shows very strong annual seasonality in addition to an upward YOY (Year-over-year) trend.

```
ggseasonplot(x=myts, year.labels=TRUE, year.labels.left=TRUE) +
  ylab("A$ million") +
  ggtitle(mymain)
```



ggseasonplot

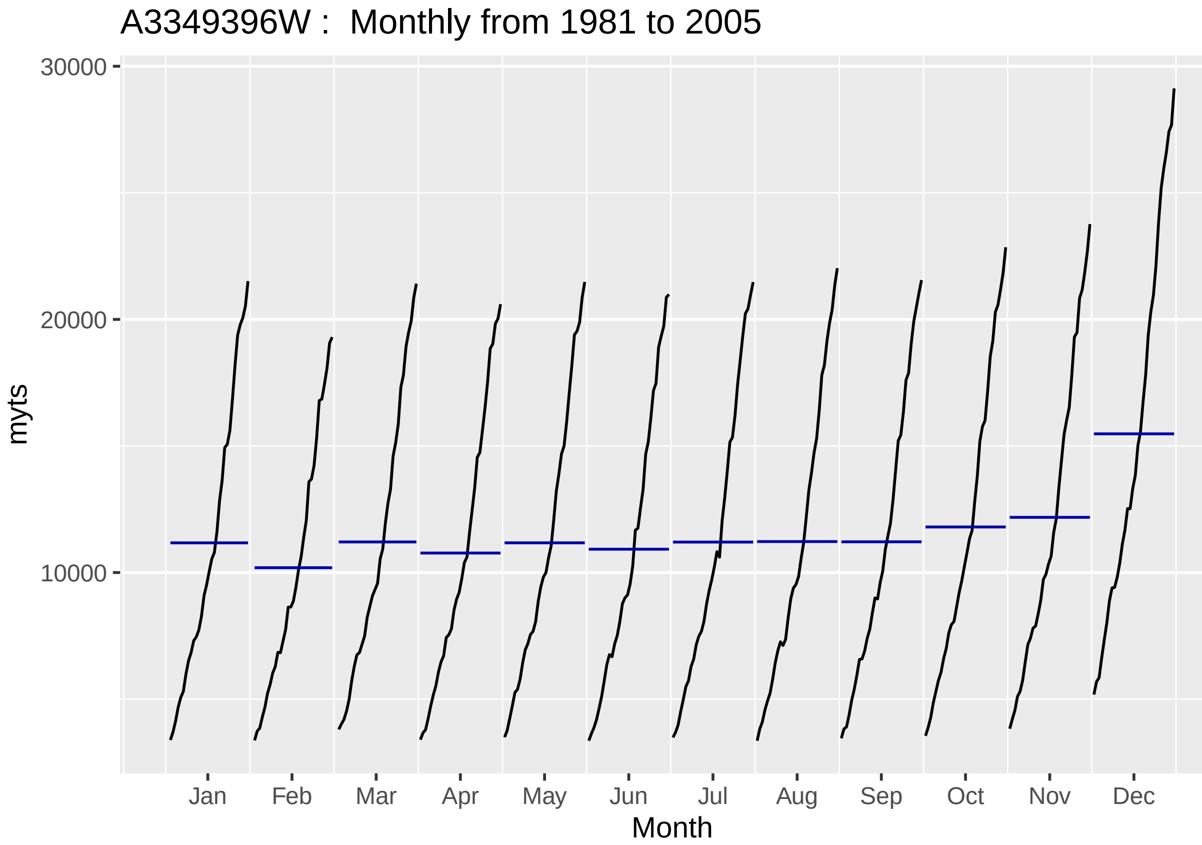
```
ggseasonplot(x=myts, polar=TRUE) +
  ylab("A$ million") +
  ggtitle(paste("Polar season plot: ",mycode))
```



The season plot shows a sharp increase each December – likely due to added holiday-season sales.

Additionally, the trend indicates is a consistent year-over-year increase

```
ggsubseriesmain <- paste(mycode, ": " , "Monthly from 1981 to 2005")
ggsubseriesplot(x=myts) +
  ggtitle(ggsubseriesmain)
```

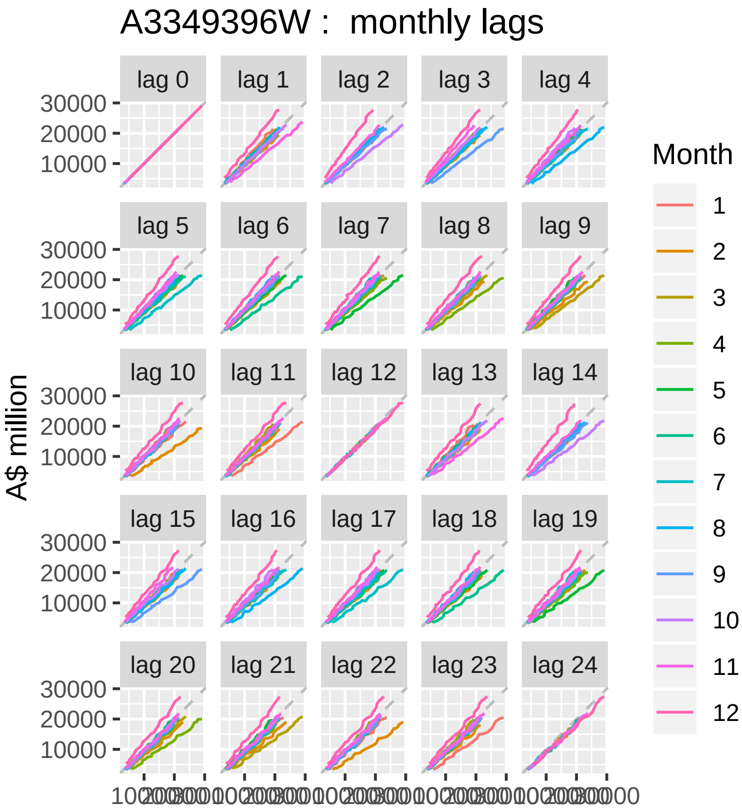


ggsubseriesplot

The month-by-month graphs indicate a significant year-over-year trend.

Additionally, the sharp jump in December reflects seasonality, likely driven by holiday sales.

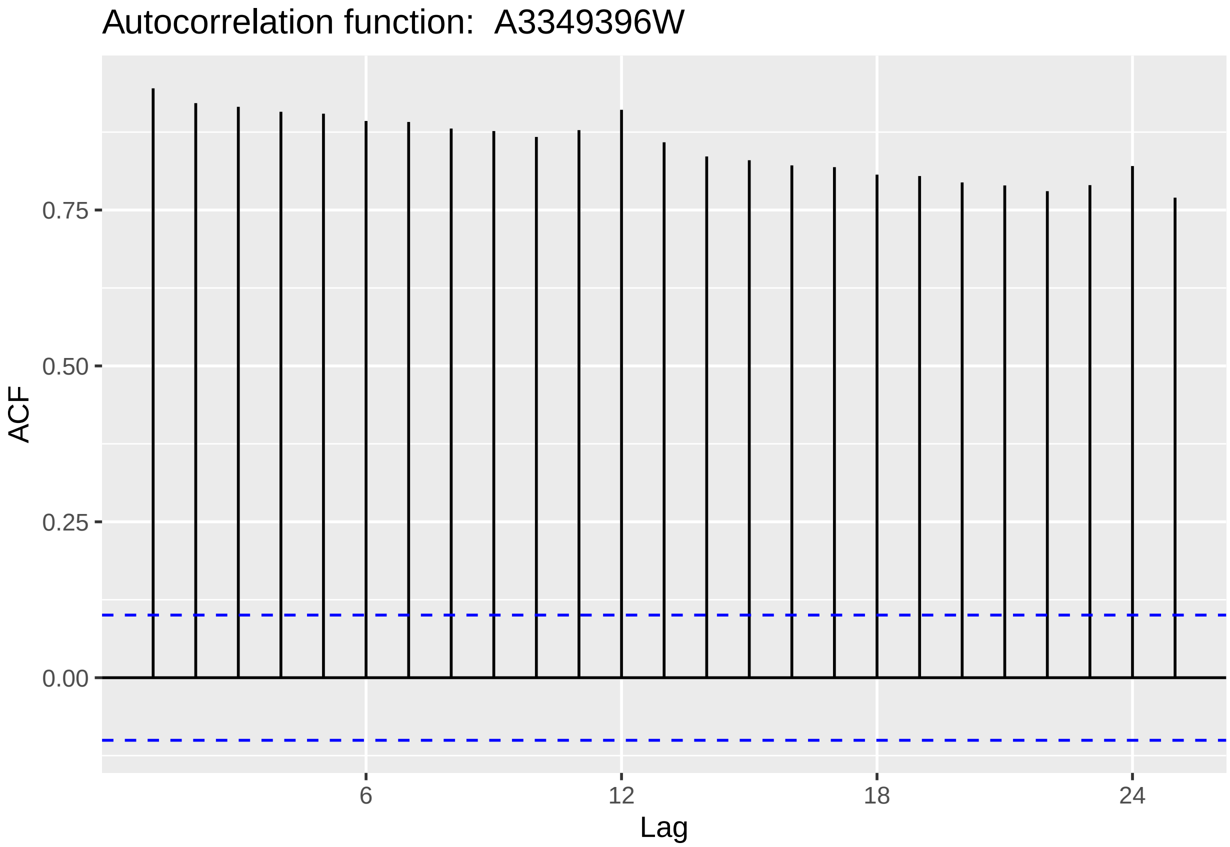
```r
gglagmain <- paste(mycode, ": ", "monthly lags")
gglagplot(x=myts,set.lags = 0:24)+
  ylab("A$ million") +
  ggtitle(gglagmain)
```



gglagplot

The diagonal alignment on the lag12 and lag24 graphs confirm a sharp annual seasonality.

```
ggAcf(x=myts)+ggtitle(paste('Autocorrelation function: ', mycode))
```



ggAcf

The extremely high autocorrelation across lags indicate a strong trend where the sales in one month are, generally, closely correlated to those in the preceding month. The upward spikes in lags 12 and 24 reflect the seasonality, where each month's sales are more highly correlated to the sales of the same month rather than to the months (other than the mose recent 1 or 2 months.)

Can you spot any seasonality, cyclicity and trend? What do you learn about the series?

The series exhibits a strong upward trend and annual seasonality. No cyclicality is detected.

2.6 Use the following graphics functions: `autoplot()`, `ggseasonplot()`, `ggsubseriesplot()`, `gglagplot()`, `ggAcf()`

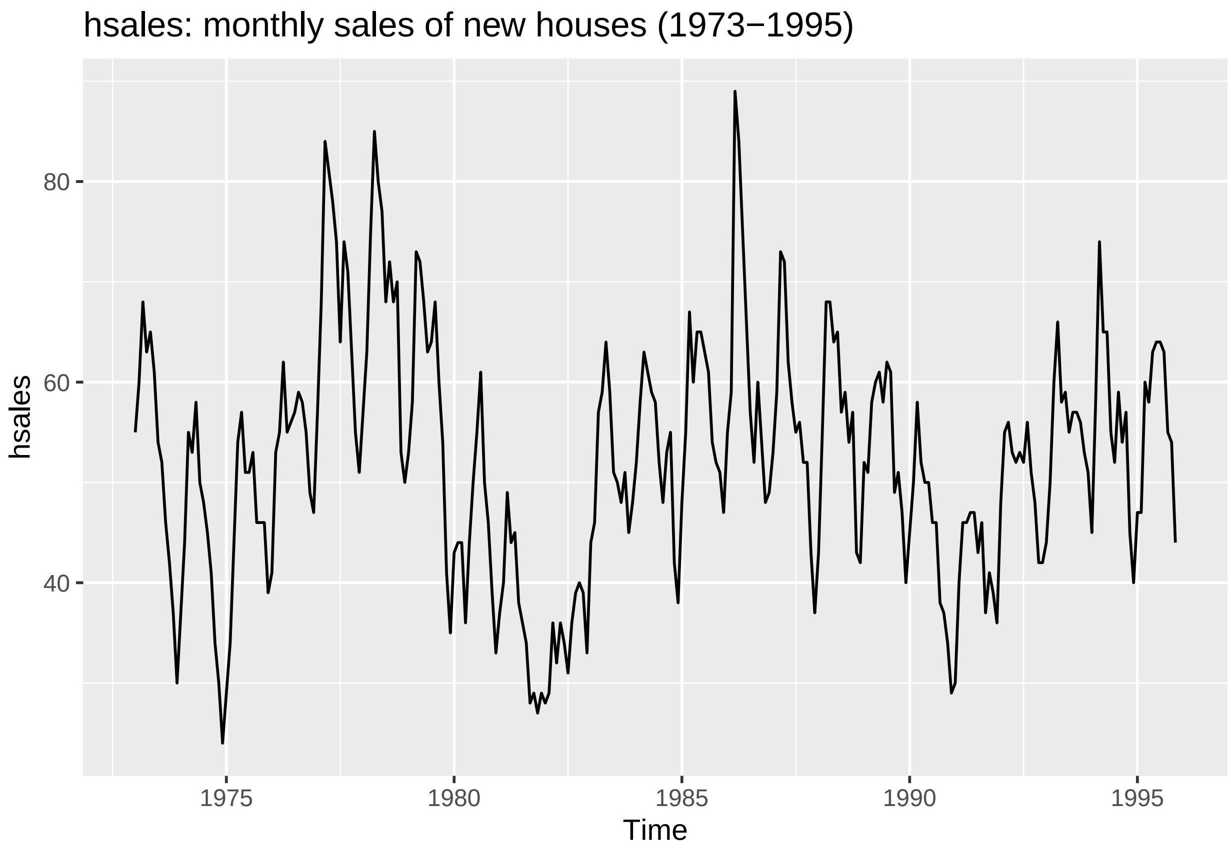
and explore features from the following time series: `hsales`, `usdeaths`, `bricksq`, `sunspotarea`, `gasoline`.

Can you spot any seasonality, cyclicity and trend?

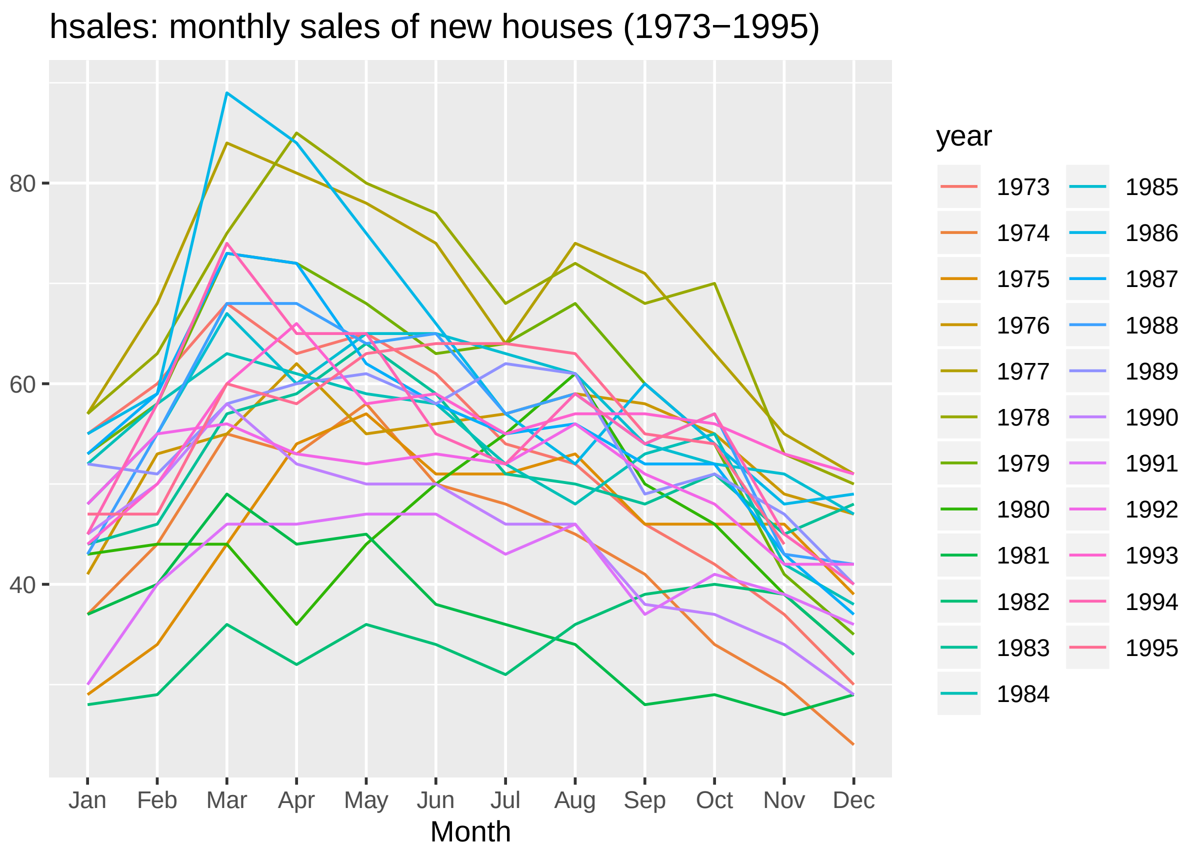
What do you learn about the series?

2.6a: `hsales` - Monthly sales of new one-family houses sold in the USA since 1973.

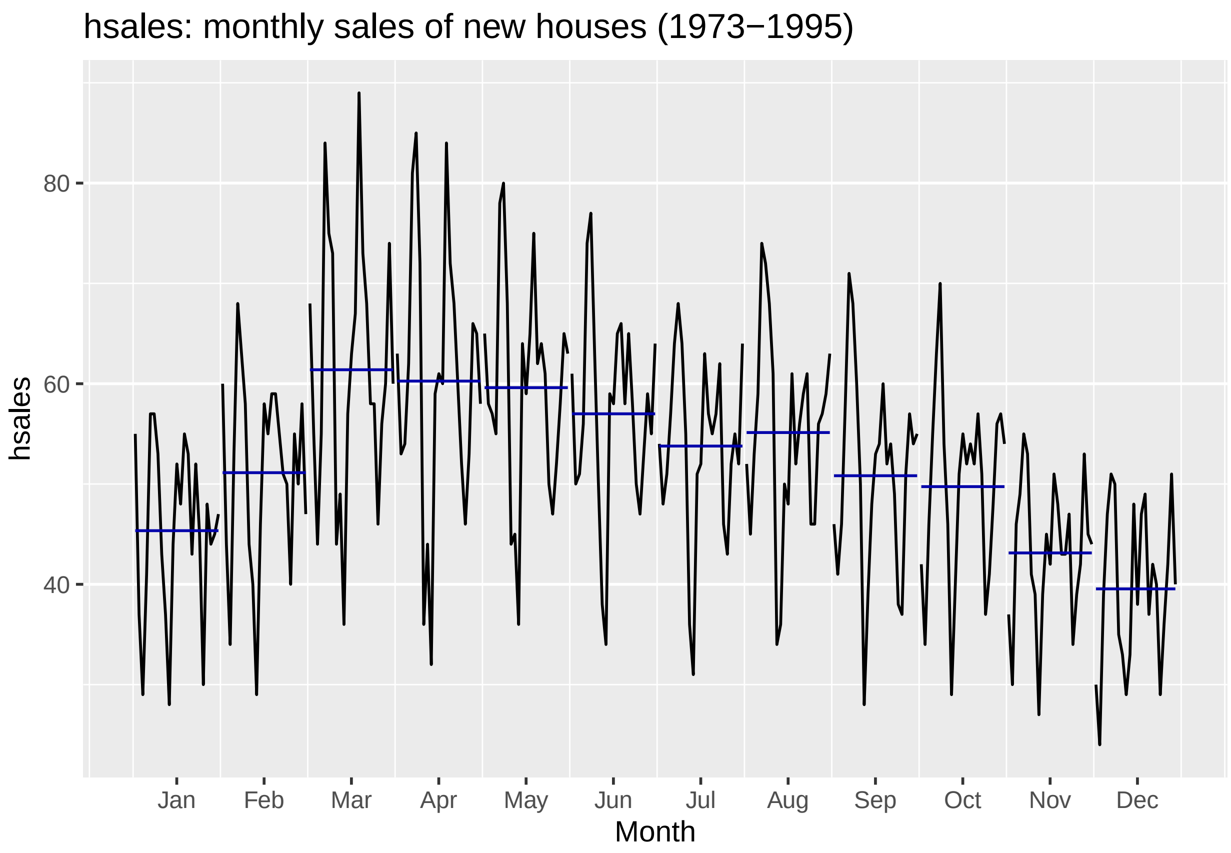
```
mytitle <- "hsales: monthly sales of new houses (1973-1995)"
autoplot(hsales) + ggtitle(mytitle)
```



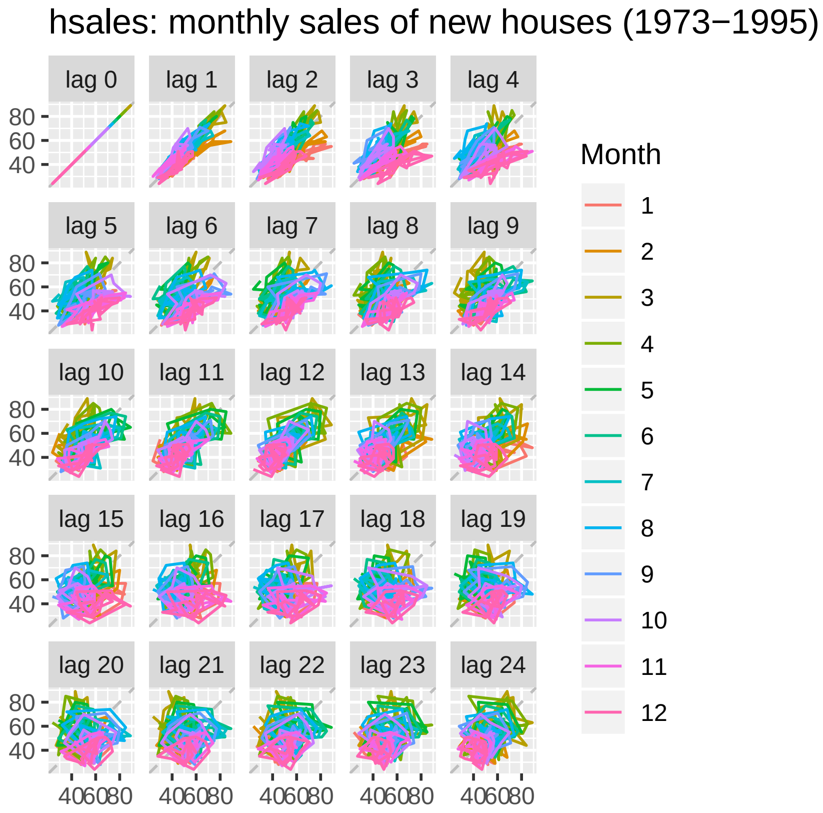
```
ggseasonplot(hsales) + ggtitle(mytitle)
```



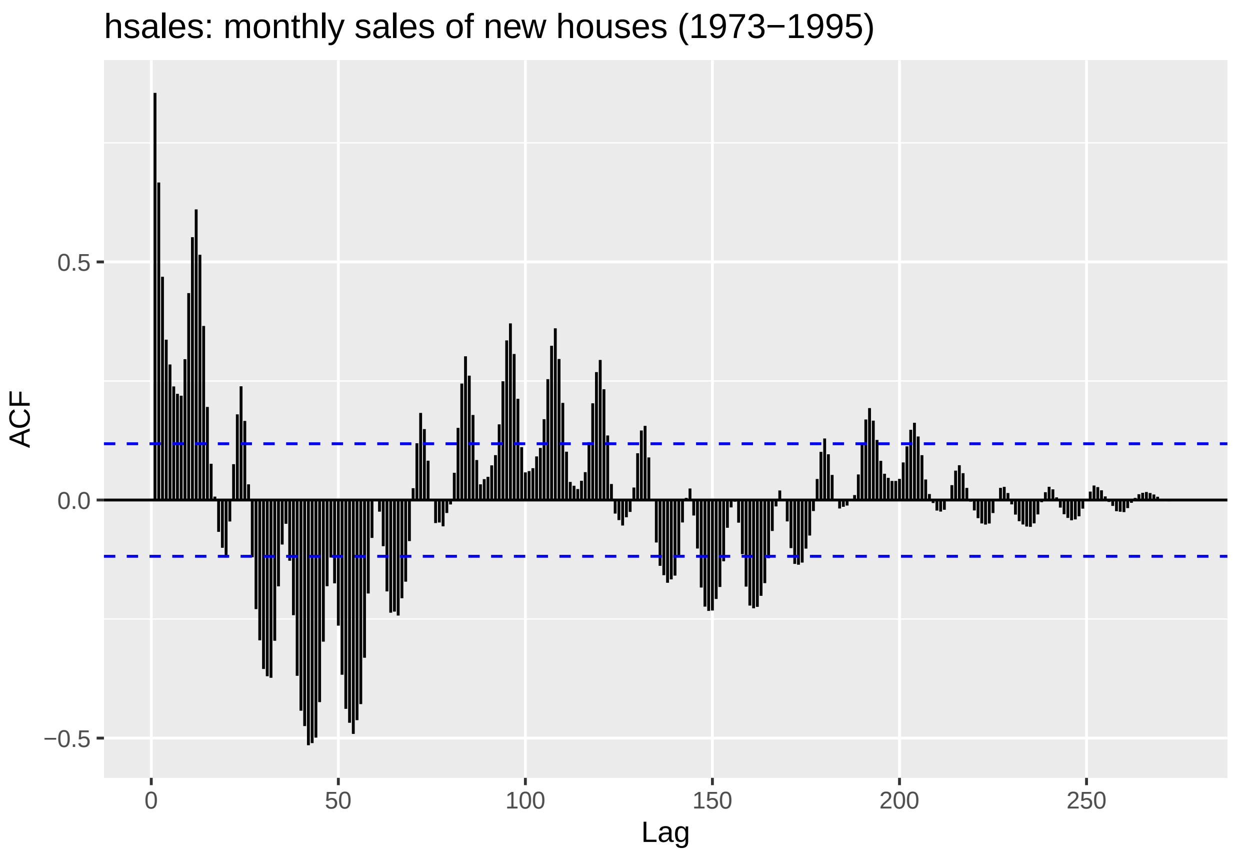
```
ggsubseriesplot(hsales) + ggtitle(mytitle)
```



```r
gglagplot(hsales,set.lags = 0:24) + ggtitle(mytitle)
```



```
ggAcf(hsales, lag.max = 275) + ggtitle(mytitle)
```



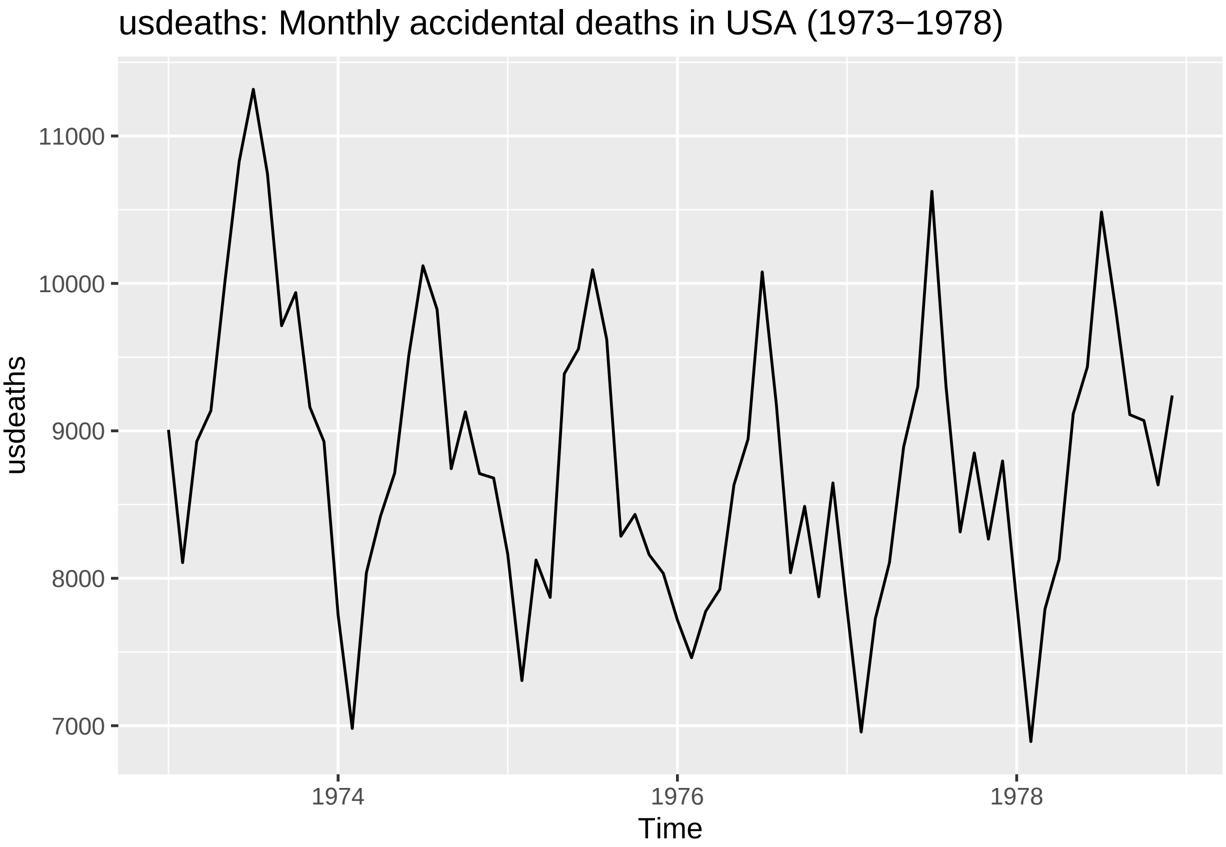
The plots exhibit seasonality, with highest sales in the Spring of each year (March, April, May.)

Cyclicality is also evident, with a cycle of about 8 years.

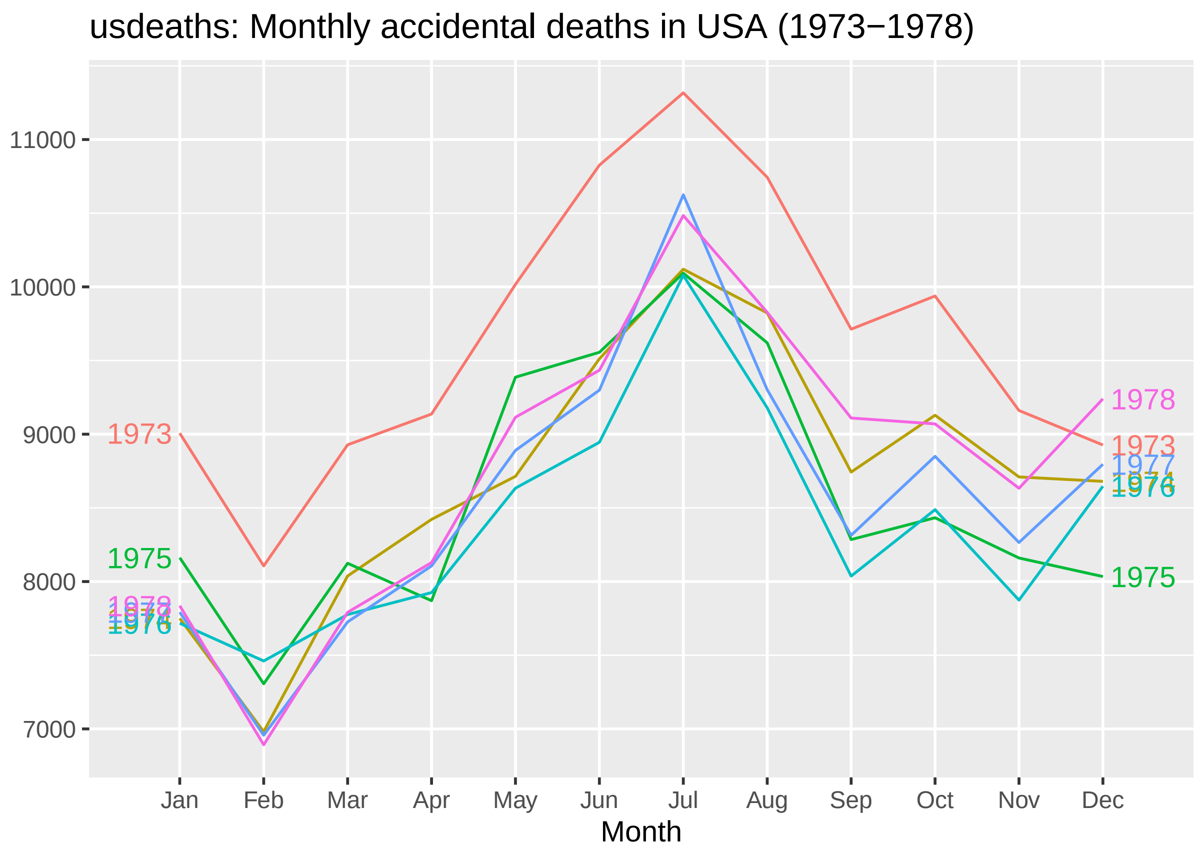
The plots do not show any long-term trend.

2.6b: `usdeaths` - Monthly accidental deaths in USA (1973-1978).

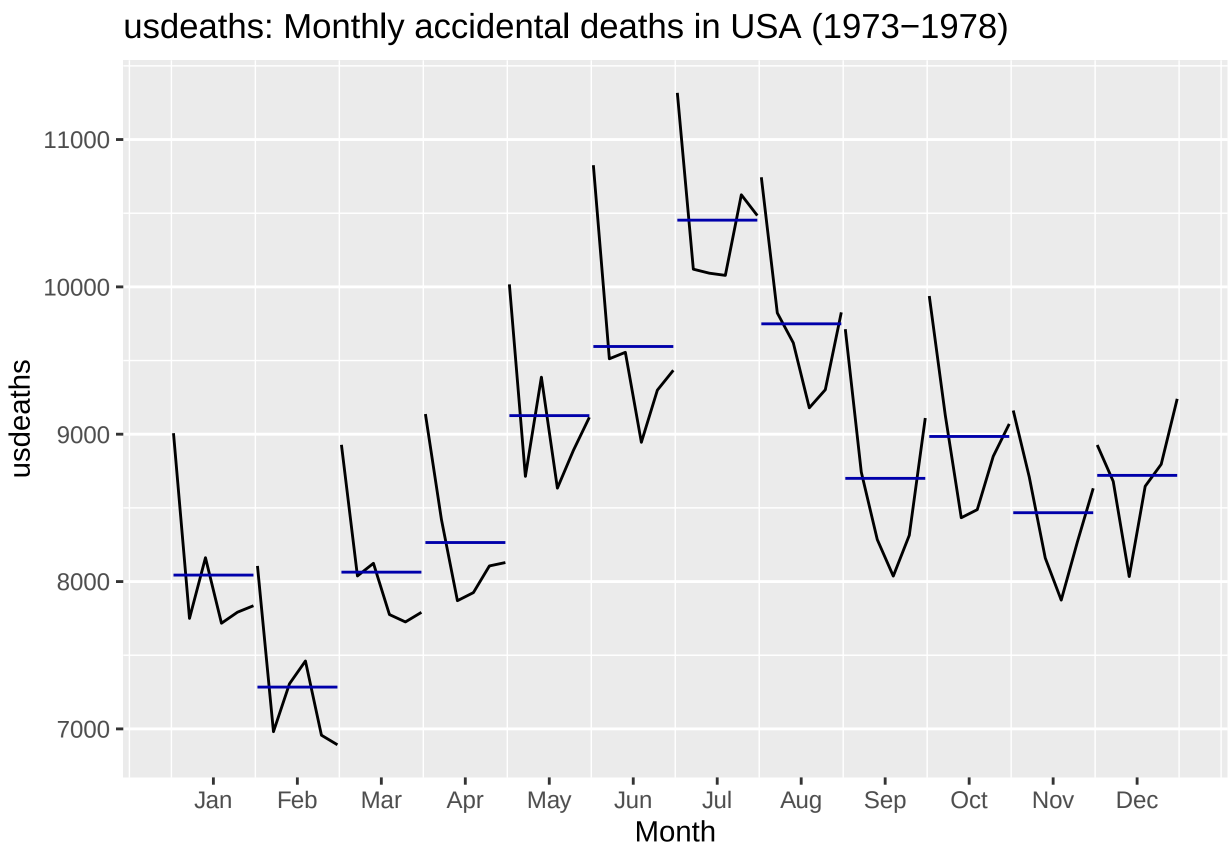
```
mytitle <- "usdeaths: Monthly accidental deaths in USA (1973-1978)"
autoplot(usdeaths) + ggtitle(mytitle)
```



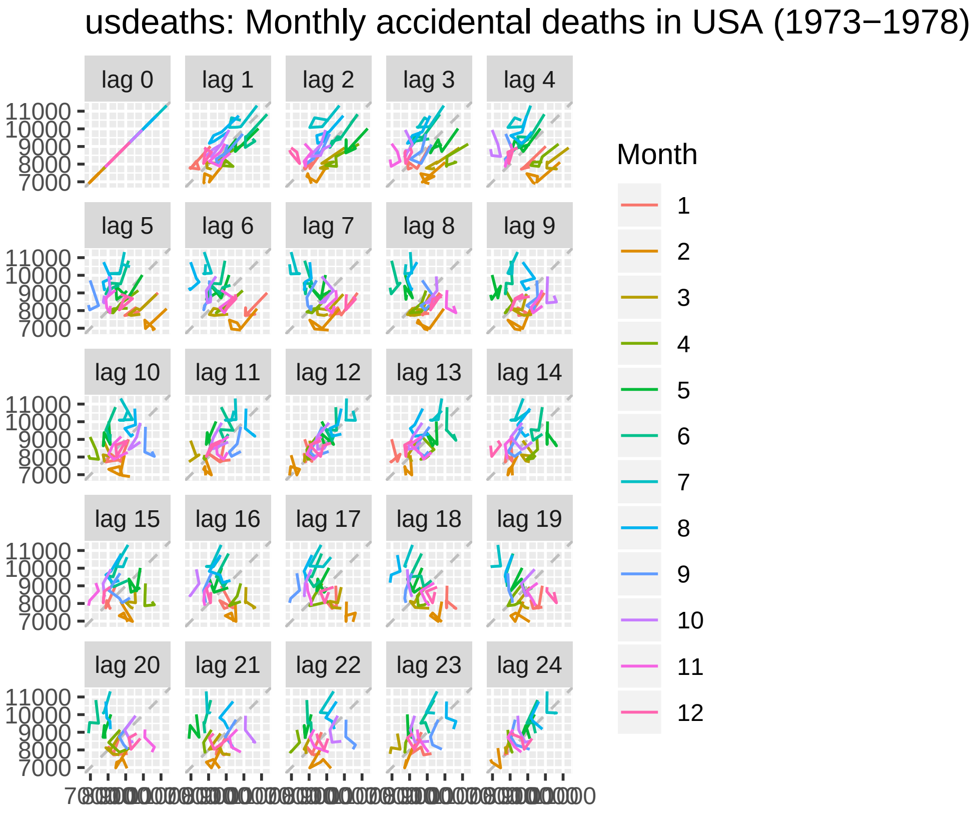
```
ggseasonplot(usdeaths, year.labels=TRUE, year.labels.left=TRUE) + ggtitle(mytitle)
```



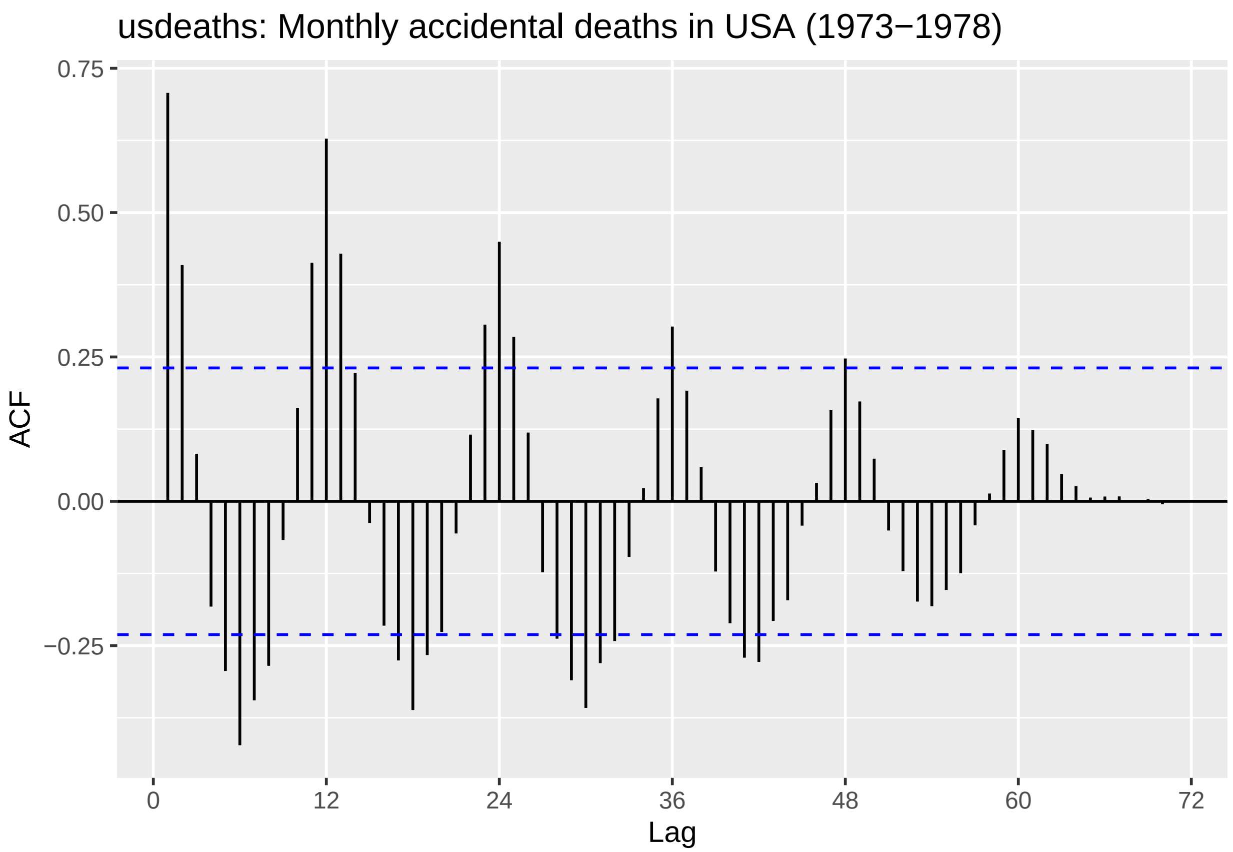
```r
ggsubseriesplot(usdeaths) + ggtitle(mytitle)
```



```
gglagplot(usdeaths,set.lags = 0:24) + ggtitle(mytitle)
```



```
ggAcf(usdeaths, lag.max = 72) + ggtitle(mytitle)
```

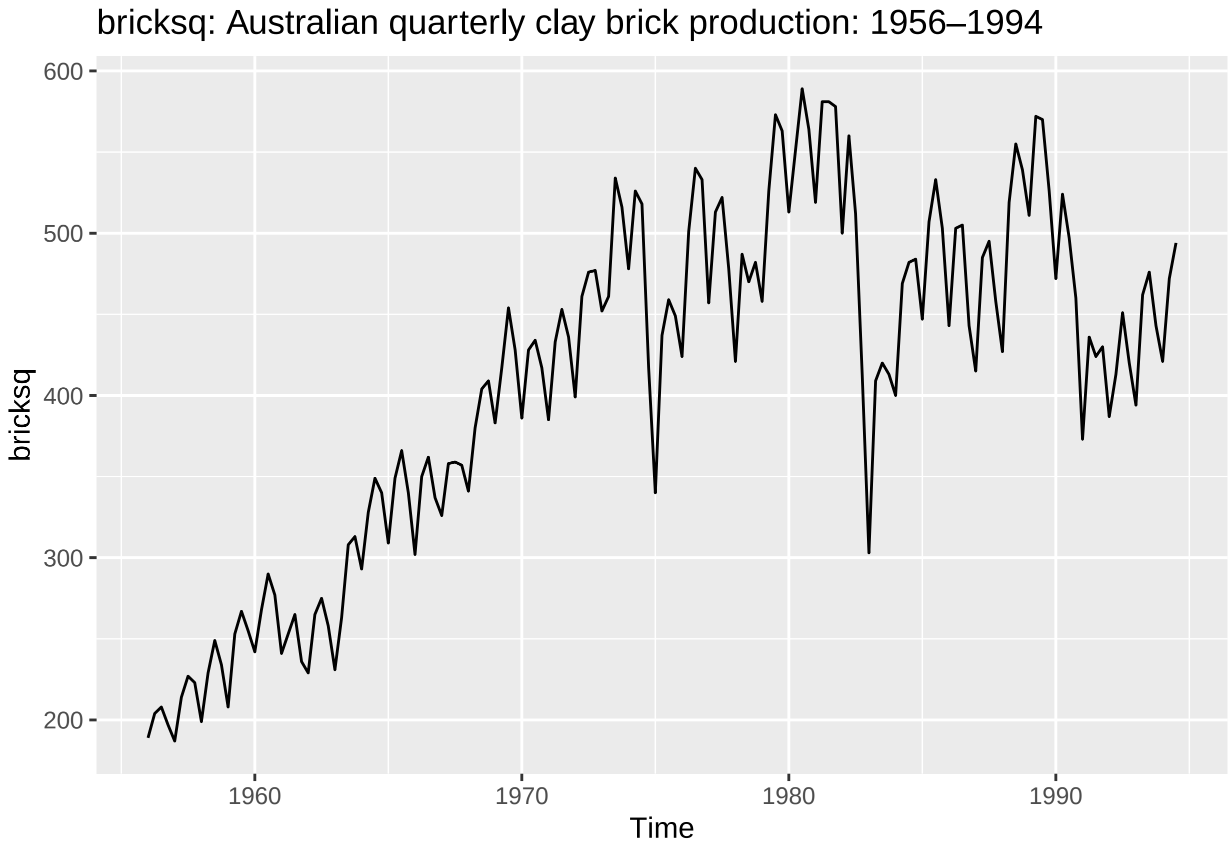


The data exhibits seasonality, with more accidental deaths occurring in the summertime (e.g., July) and fewer occuring in the winter (e.g., February.)

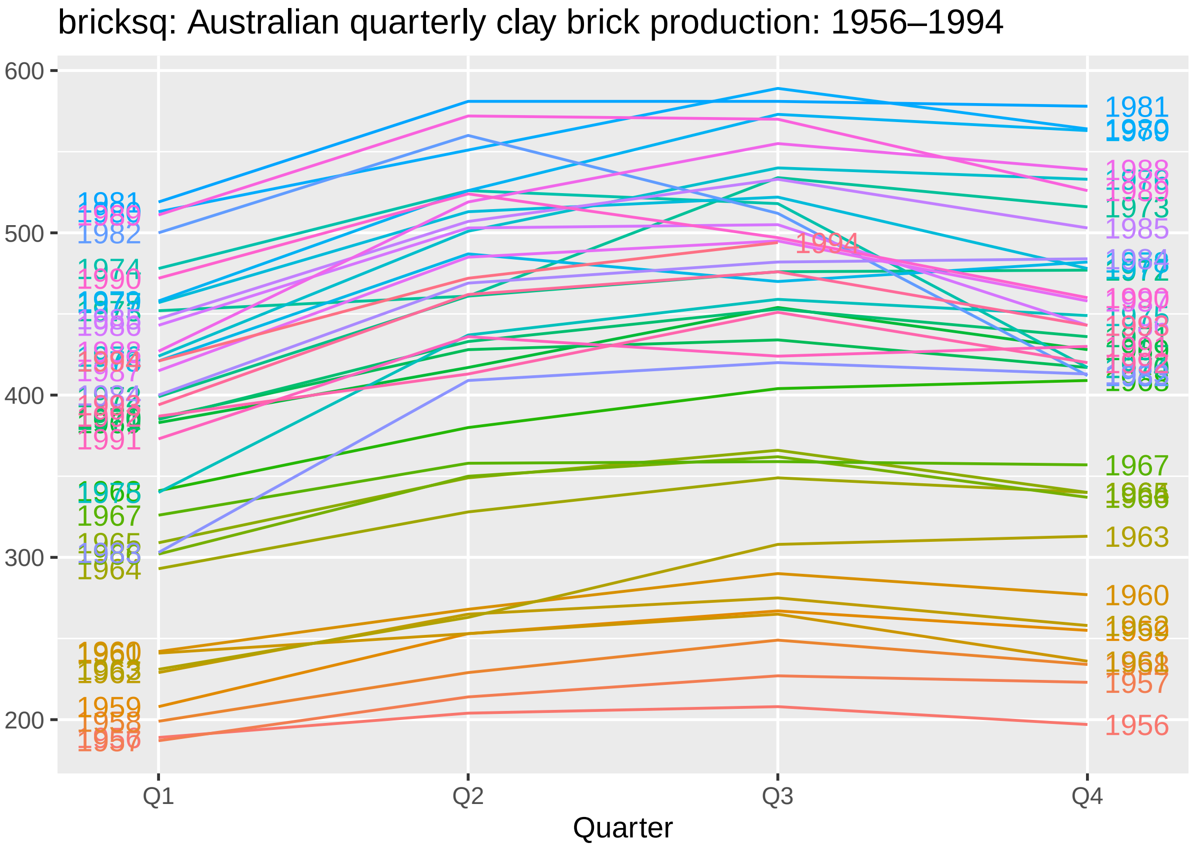
The data does not exhibit any long-term trend nor does it display cyclicality.

2.6c: `bricksq` - Australian quarterly clay brick production: 1956–1994.

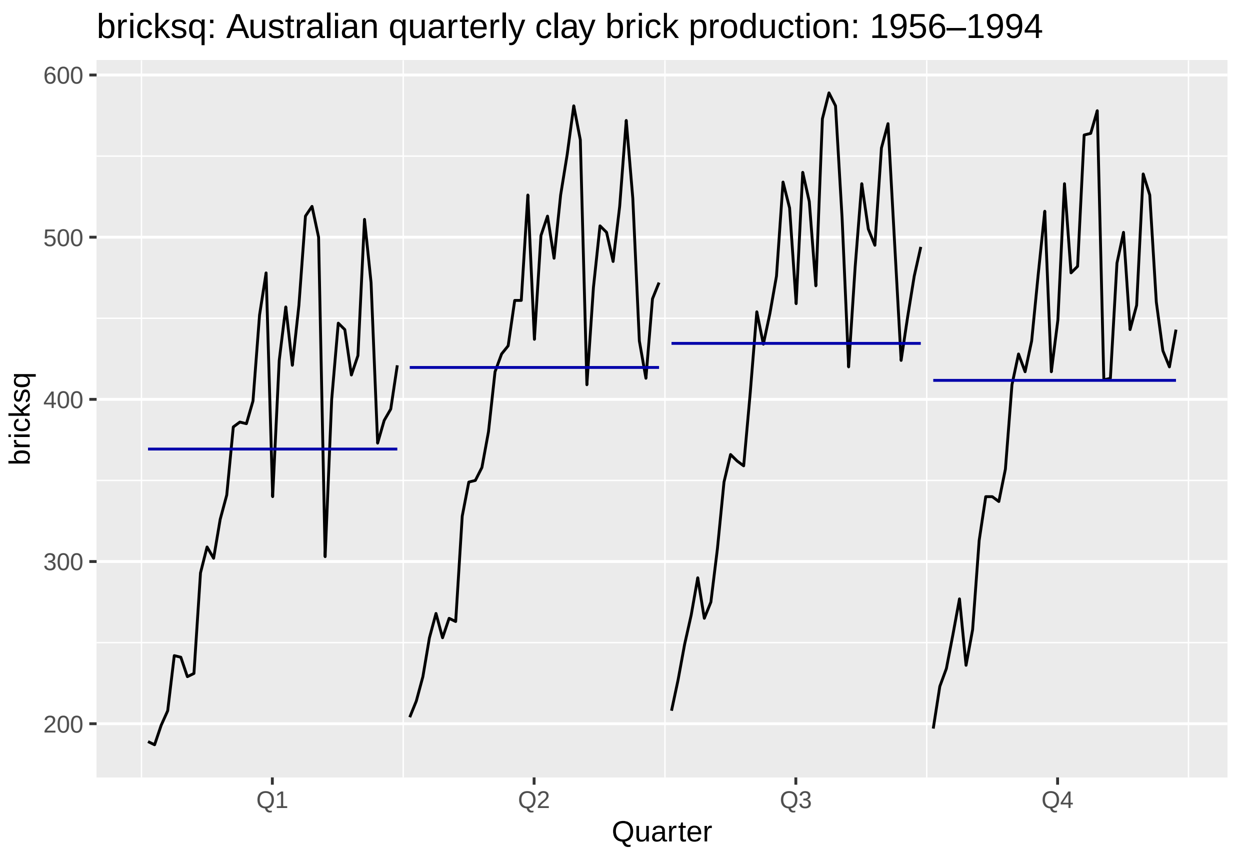
```
mytitle <- "bricksq: Australian quarterly clay brick production: 1956-1994"
autoplot(bricksq) + ggtitle(mytitle)
```



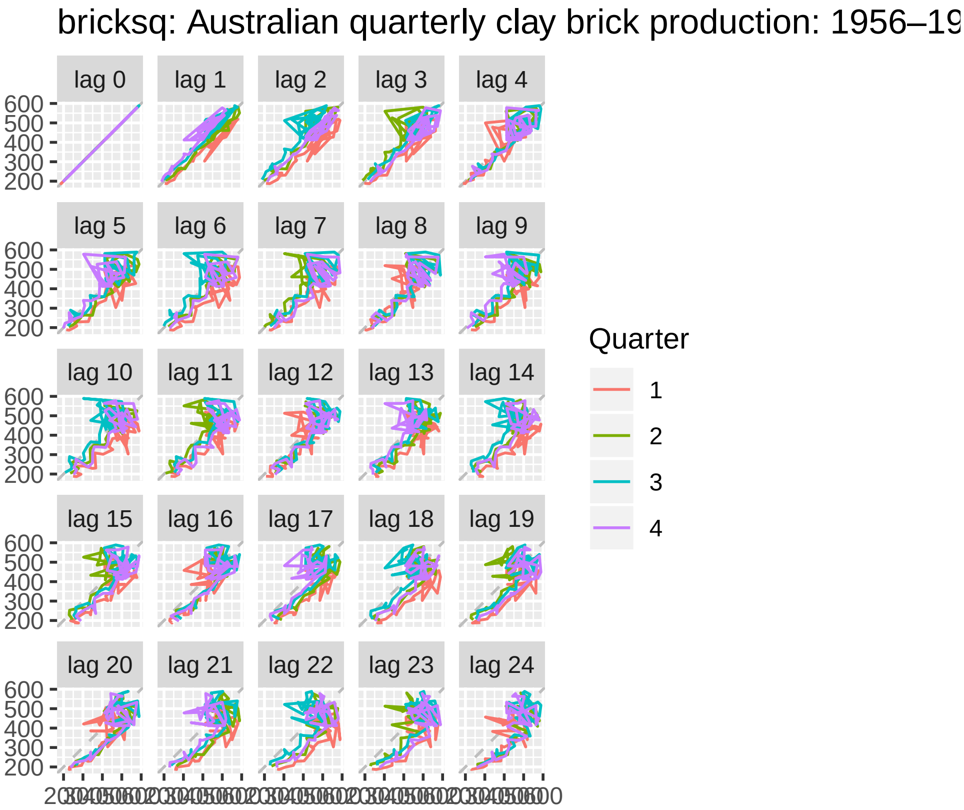
```
ggseasonplot(bricksq, year.labels=TRUE, year.labels.left=TRUE) + ggtitle(mytitle)
```



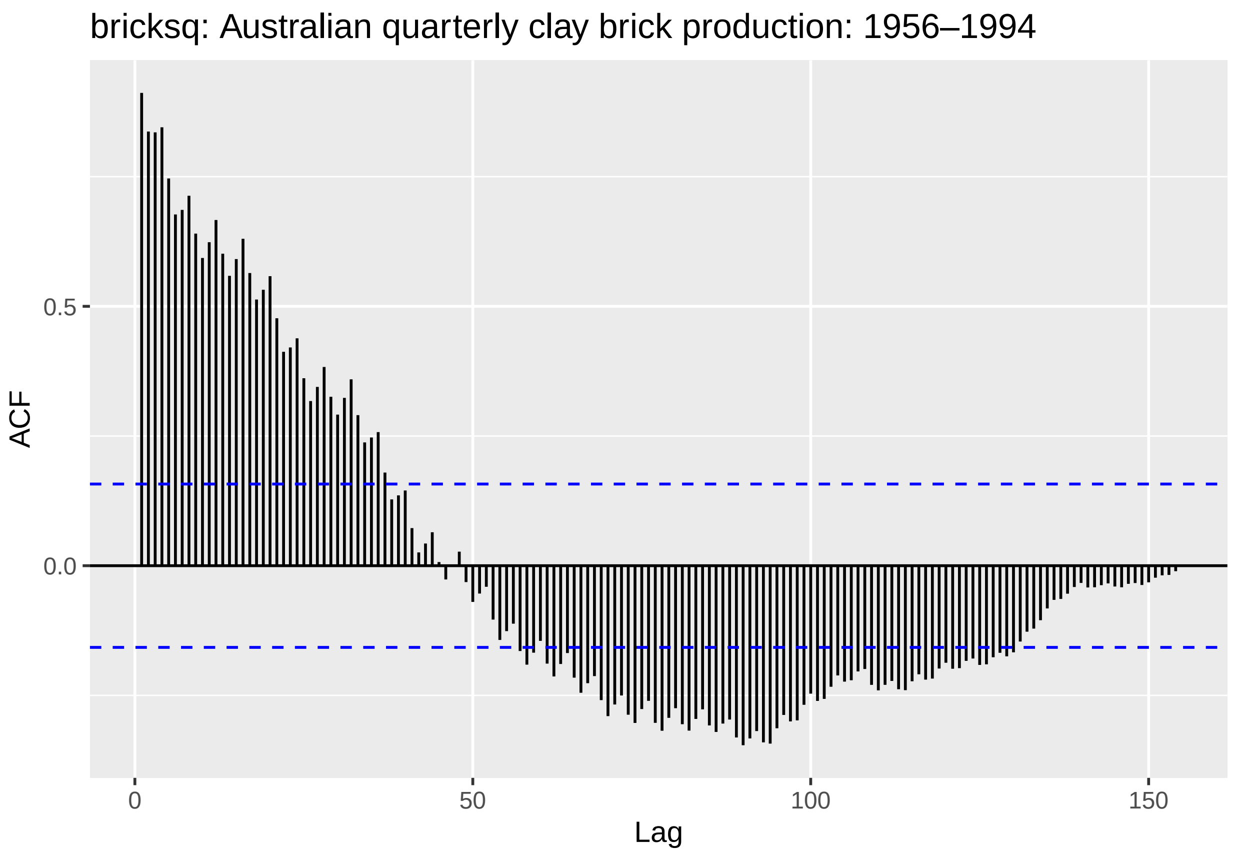
```r
ggsubseriesplot(bricksq) + ggtitle(mytitle)
```



```
gglagplot(bricksq,set.lags = 0:24) + ggtitle(mytitle)
```



```
ggAcf(bricksq, lag.max = 155) + ggtitle(mytitle)
```

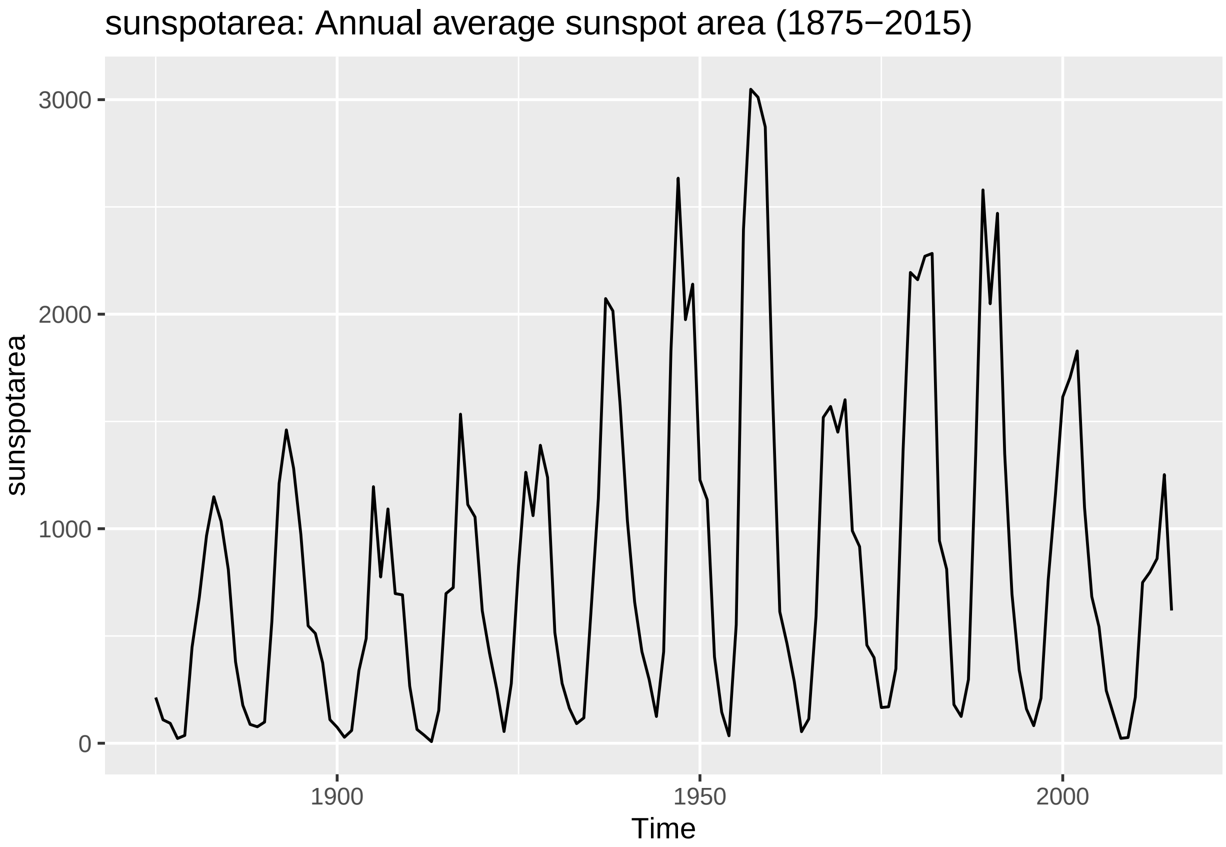


Seasonality is evident with higher production of bricks in Q3, which includes the Australian winter months (July-August-September) and lower production in Q1, which includes the Australian summer, perhaps because of vacation holidays.

The data exhibits a clear upward trend from 1956-1974, after which time there is a clear "regime shift", when the data becomes cyclical instead of trending upwards, with dips in 1975, 1983, and 1991 indicating an eight-year cycle following such change.

2.6d: sunspotarea - Annual average sunspot area (1875-2015).

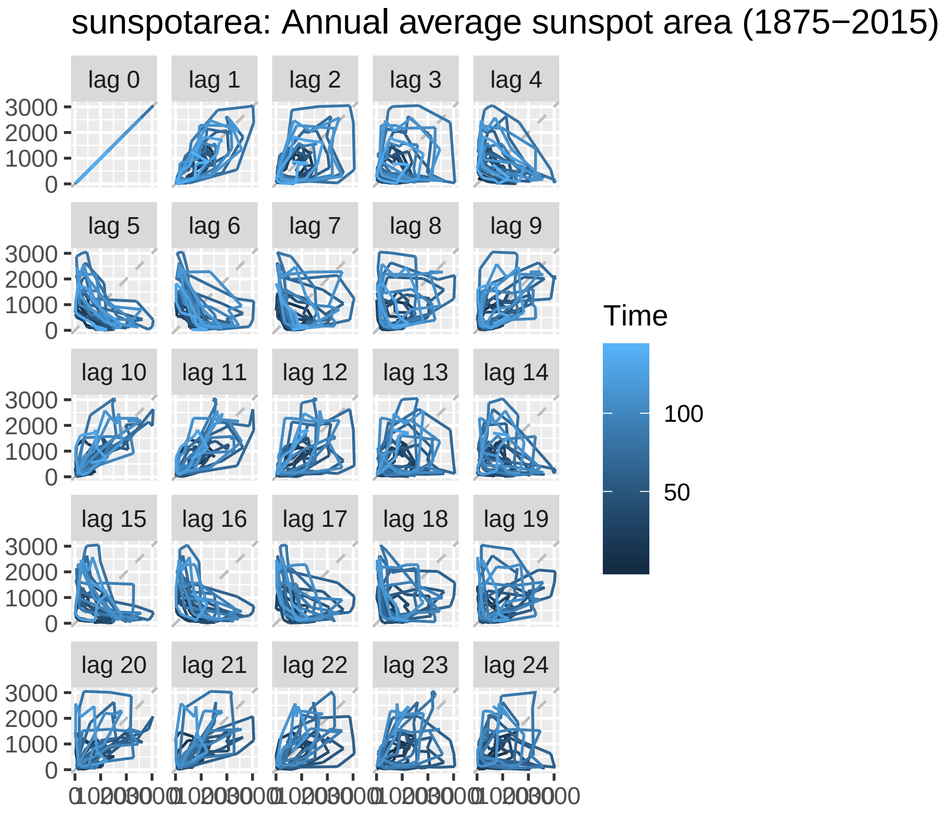
```
mytitle <- "sunspotarea: Annual average sunspot area (1875-2015)"
autoplot(sunspotarea) + ggtitle(mytitle)
```



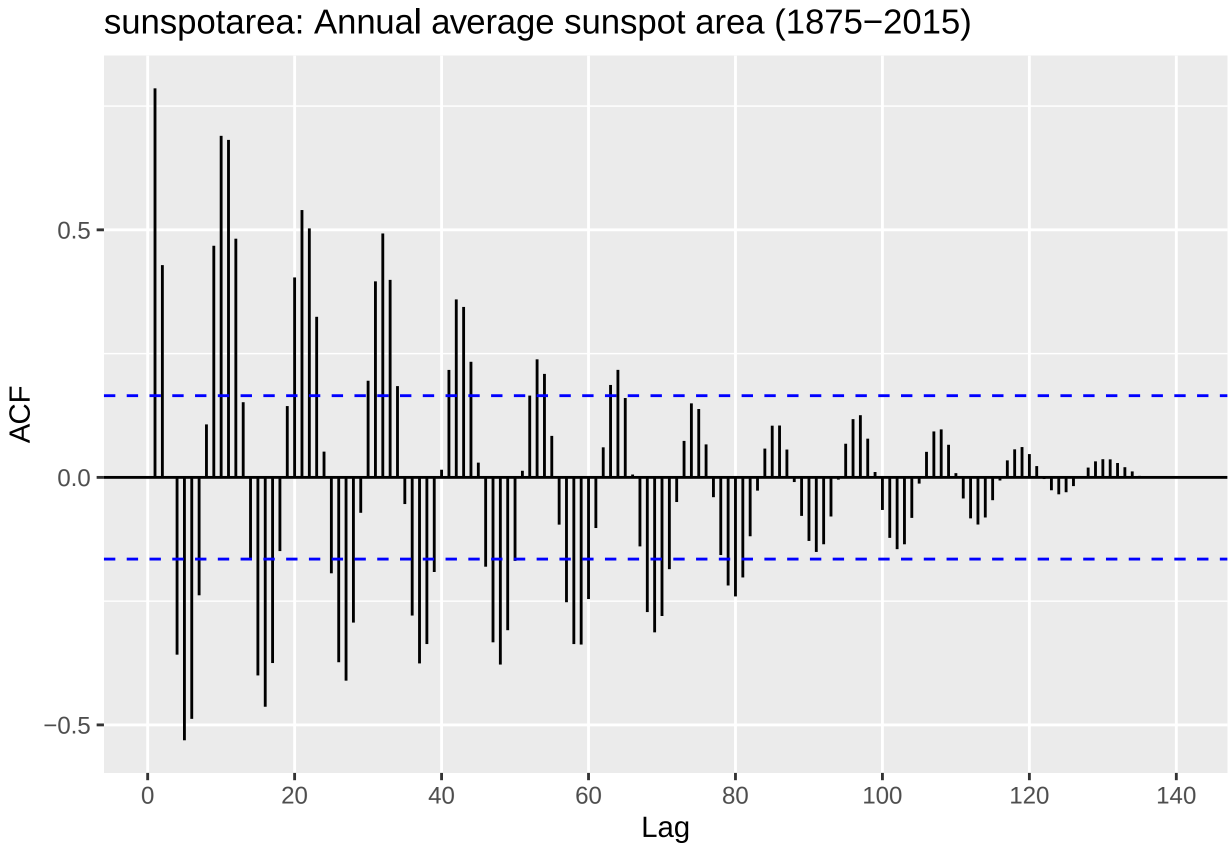
```
##ggseasonplot(sunspotarea, year.labels=TRUE, year.labels.left=TRUE) + ggtitle(mytitle)
## Error in ggseasonplot(sunspotarea, year.labels = TRUE, year.labels.left = TRUE) :
##  Data are not seasonal

##ggsubseriesplot(sunspotarea) + ggtitle(mytitle)
## Error in ggsubseriesplot(sunspotarea) : Data are not seasonal

gglagplot(sunspotarea,set.lags = 0:24) + ggtitle(mytitle)
```



```
ggAcf(sunspotarea, lag.max = 141) + ggtitle(mytitle)
```

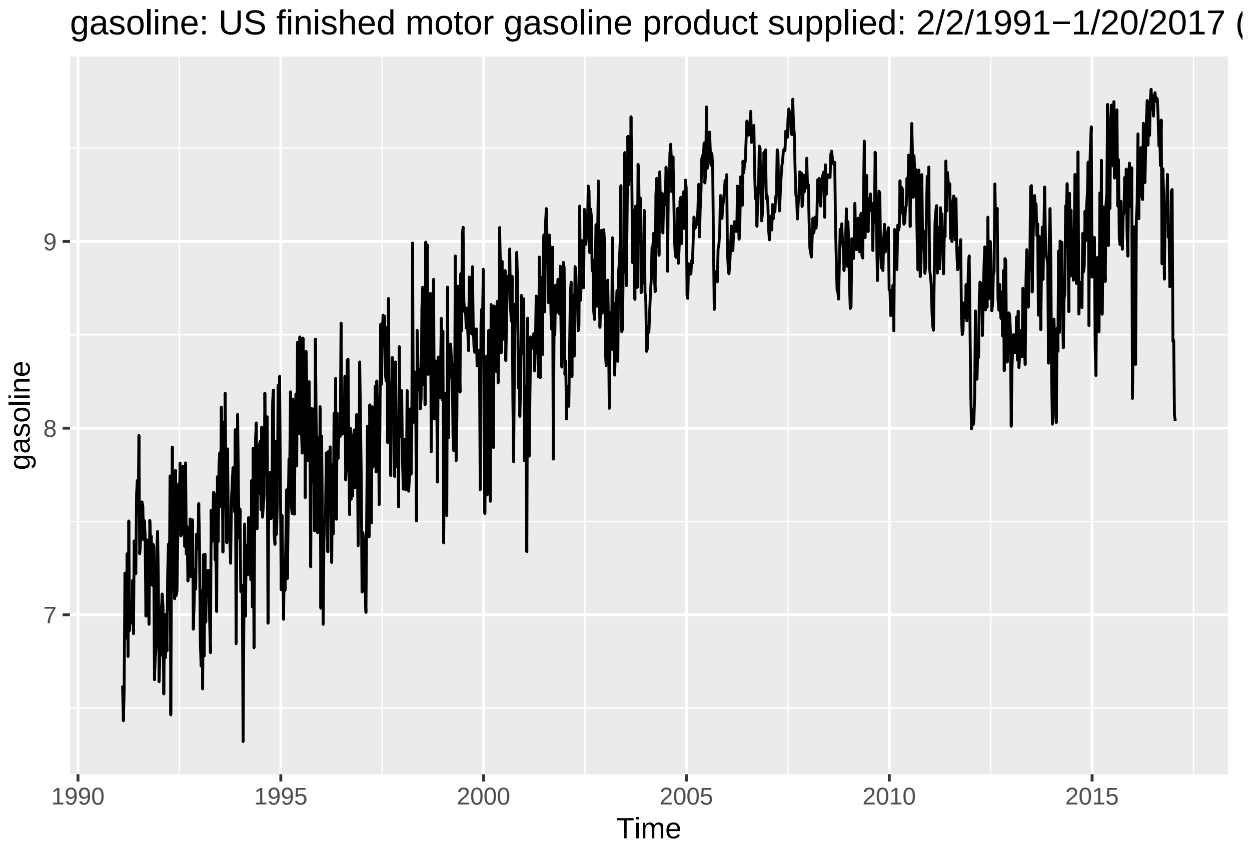


Because the data is annual, it cannot exhibit seasonality, thus the `ggseasonplot` and `ggsubseriesplot` functions return errors.

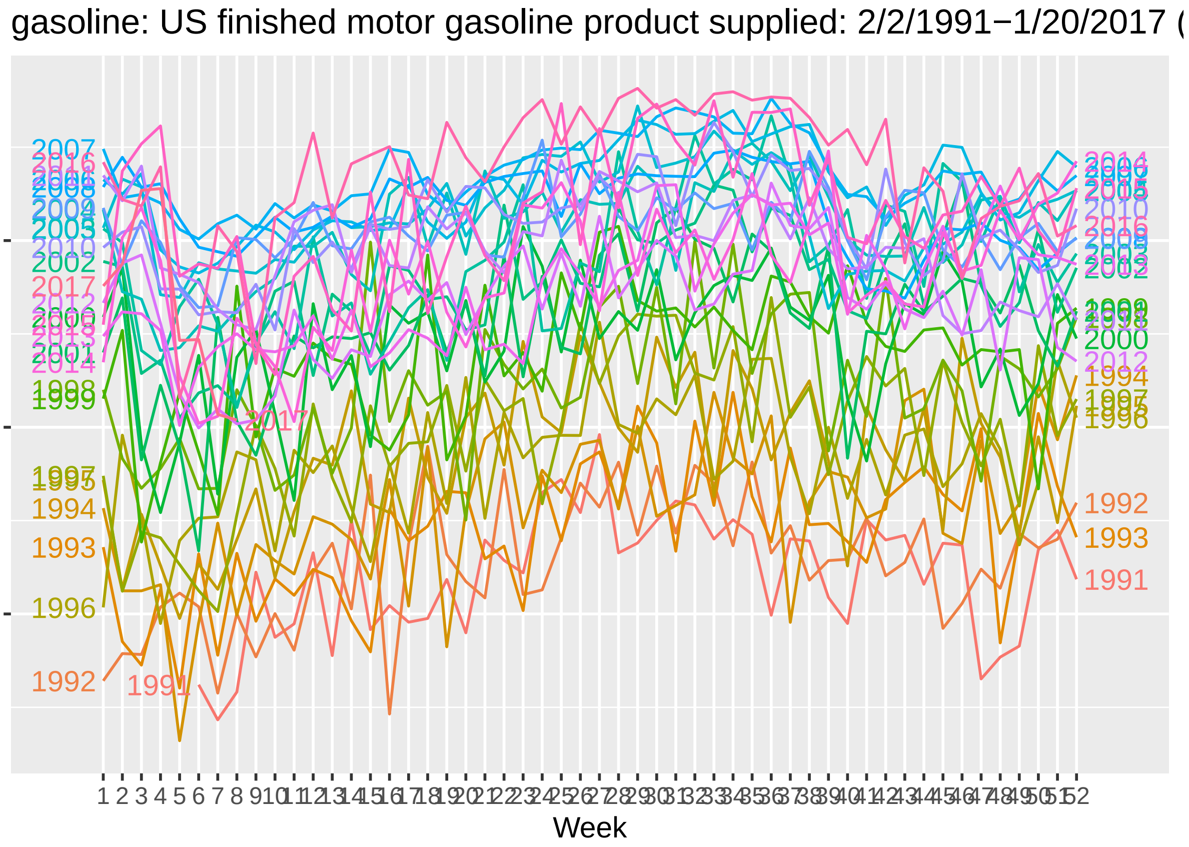
The data is cyclical, with the sunspot cycle running about 11 years in duration.

2.6e: `gasoline` - US finished motor gasoline product supplied: 2/2/1991-1/20/2017 (weekly).

```
mytitle <- "gasoline: US finished motor gasoline product supplied: 2/2/1991-1/20/2017 (weekly)"
autoplot(gasoline) + ggtitle(mytitle)
```

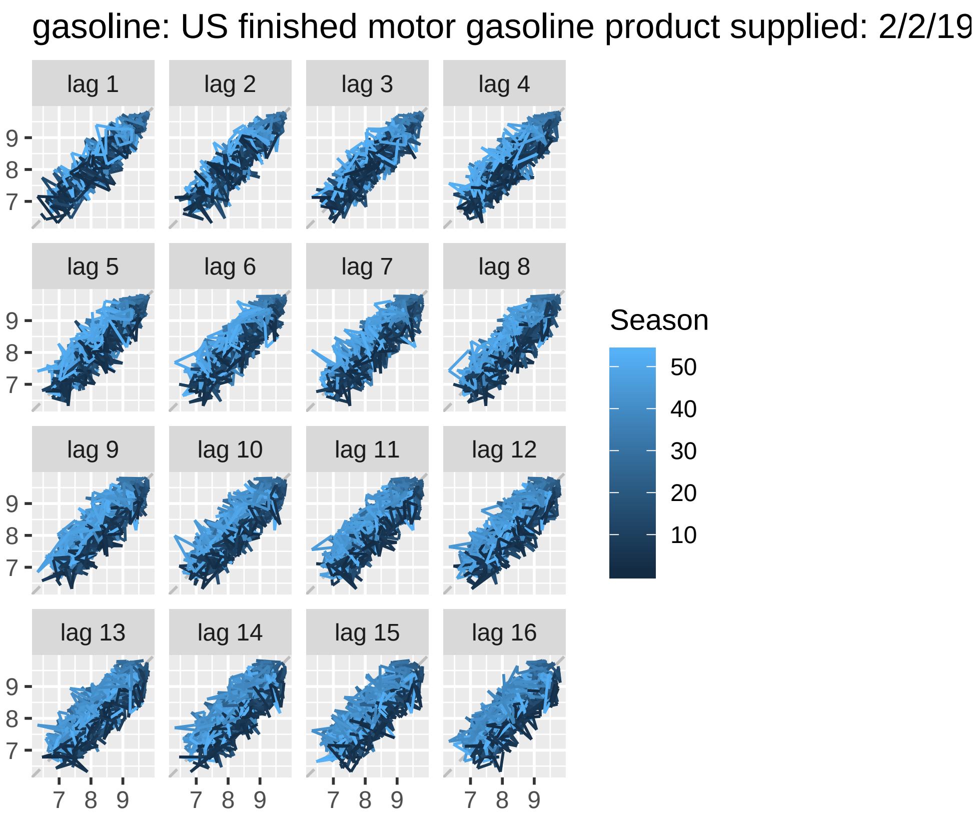


```
ggseasonplot(gasoline, year.labels=TRUE, year.labels.left=TRUE) + ggtitle(mytitle)
```

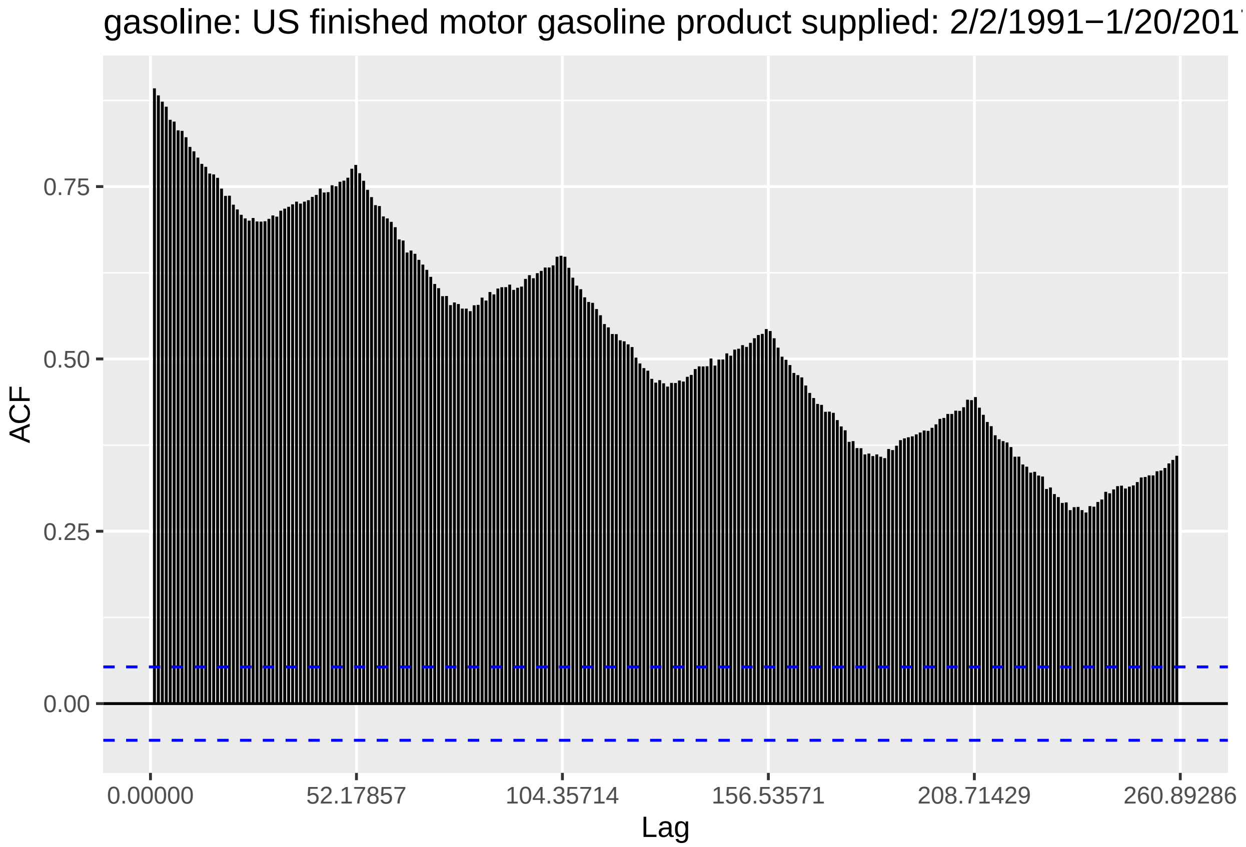


```
## ggsubseriesplot(gasoline) + ggtitle(mytitle)
## Error in ggsubseriesplot(gasoline) : Each season requires at least 2 observations.
## This may be caused from specifying a time-series with non-integer frequency.

gglagplot(gasoline) + ggtitle(mytitle)
```



```
ggAcf(gasoline, lag.max = 260) + ggtitle(mytitle)
```



The data exhibits an upward trend from 1991 up until 2007, when such increase stops.

Because of the way the weekly data is stored in the ts object, it appears that the software is unable to map the n^{th} week of each calendar year, as it easily does with monthly or quarterly data.

The attribute associated with the ts object indicates that the frequency is 52.1785714 . If adjustments were made, it might be possible to have the software treat the frequency as 52 rather than 52.1785714 .

However, "scallops" in the ACF plot indicate seasonality on a 52-week cycle.