workshop.nectric.com.au/tidyfc2024

Time Series Analysis & Forecasting Using R

2. Time series graphics

tsibble feasts fable

Outline

1. Workshop data
2. Time plots
3. Lab Session 3
4. Seasonal plots
5. Lab Session 4
6. Seasonal or cyclic?
7. Lag plots and autocorrelation
8. Lab Session 4
9. White noise
10. Lab Session 5

Outline

1 Workshop data
2 Time plots
3 Lab Session 3
4 Seasonal plots
5 Lab Session 4
6 Seasonal or cyclic?
7 Lag plots and autocorrelation
8 Lab Session 4
9 White noise
10 Lab Session 5

Workshop data

https://workshop.nectric.com.au/tidyfc2024/labs.zip

Download this ZIP to access all the tidied data.

Open the project by double-clicking ‘tidyfc-exercises.Rproj’.

> **Alternatively…**
>
> ```
> usethis::use_course("https://workshop.nectric.com.au/tidyfc2024/labs.zip")
> ```

Outline

1 Workshop data
2 Time plots
3 Lab Session 3
4 Seasonal plots
5 Lab Session 4
6 Seasonal or cyclic?
7 Lag plots and autocorrelation
8 Lab Session 4
9 White noise
10 Lab Session 5

Time plots

Time plots are the simplest and most common visualisation for time series data (you've certainly seen these before!).

For this we put time on the x-axis and plot the measurements on the y-axis. We can make this plot easily with ggplot2, or with the `data |> autoplot(y)` helper function.

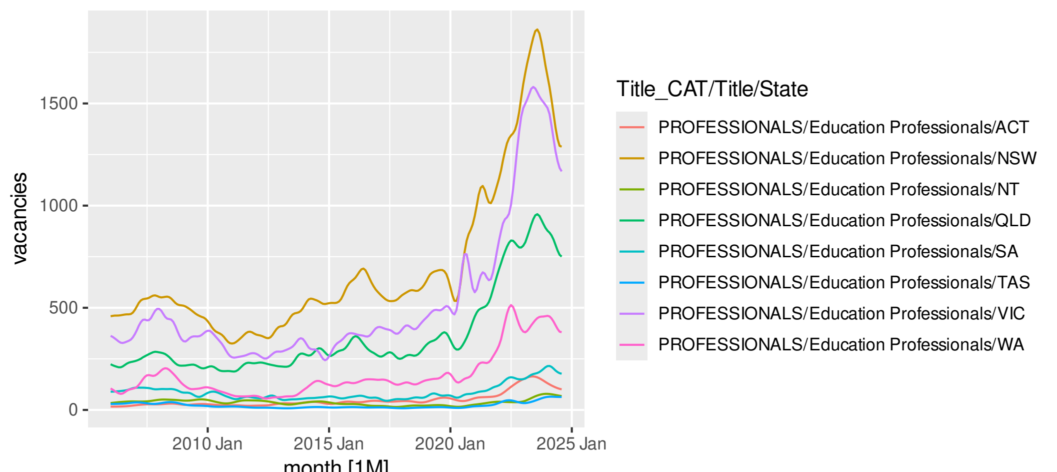
> **Overplotting beware**
>
> When working with many time series it's easy to over plot. Filter or aggregate the data before plotting.

Recall the internet vacancies dataset

```
library(readxl)
anzsco_categories <- read_excel("data/Internet Vacancies, ANZSCO2 Occupations, State
  filter(Level == 2) |>
  distinct(ANZSCO_CODE, Title)
read_excel("data/Internet Vacancies, ANZSCO2 Occupations, States and Territories - A
  # Tidy into a long form
  pivot_longer(matches("\\d{5}"), names_to = "month", values_to = "vacancies") |>
  mutate(month = yearmonth(as.Date(as.integer(month), origin = "1900-01-01"))) |>
  # Remove aggregates
  filter(Level == 3, State != "AUST") |>
  # Add level 2 category information
  mutate(ANZSCO_CODE_CAT = substr(ANZSCO_CODE, 1, 1)) |>
  left_join(anzsco_categories, by = c("ANZSCO_CODE_CAT" = "ANZSCO_CODE"), suffix = c
  select(ANZSCO_CODE, Title_CAT, Title, State, month, vacancies) |>
  # Convert to a tsibble
  as_tsibble(
    key = c(Title_CAT, Title, State),
    index = month
  ) -> internet_vacancies
```

Time plots

```
internet_vacancies |>
  filter(Title == "Education Professionals") |>
  autoplot(vacancies)
```



Time plots

Time plots help show the main changes in the data over time.

Here we can look for:

- Trend
- Seasonality
- Cycles
- Outliers

Time plots

Time plots help show the main changes in the data over time.

Here we can look for:

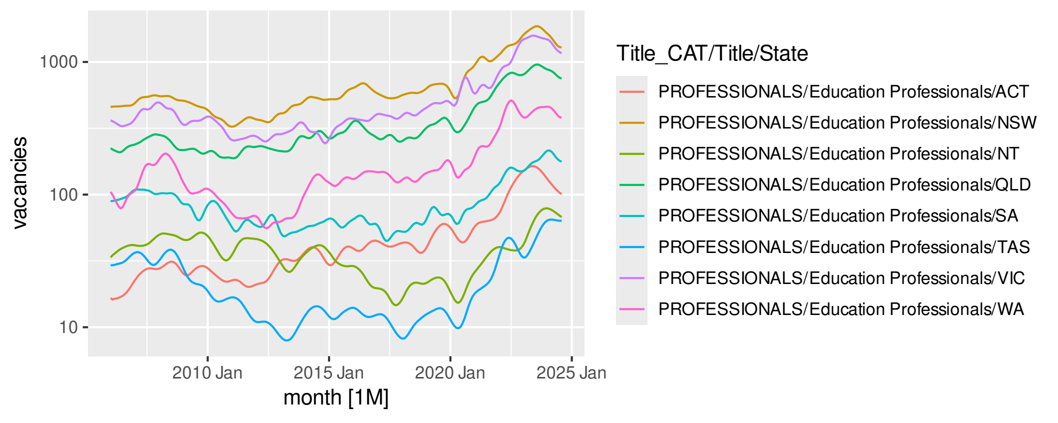
- Trend
- Seasonality
- Cycles
- Outliers

Story of time

Discuss overall patterns across time and highlight specific points in time which are interesting.

Time plots

```r
internet_vacancies |>
  filter(Title == "Education Professionals") |>
  autoplot(vacancies) +
  scale_y_log10()
```



Outline

1. Workshop data
2. Time plots
3. Lab Session 3
4. Seasonal plots
5. Lab Session 4
6. Seasonal or cyclic?
7. Lag plots and autocorrelation
8. Lab Session 4
9. White noise
10. Lab Session 5

Lab Session 3

1. Create time plots of the total school students and staff.

   Hint: You'll need to aggregate the data first.

2. Create time plots of the total students and staff by state.
3. Use ggplot2 to create a time plot from scratch, complete with labels.

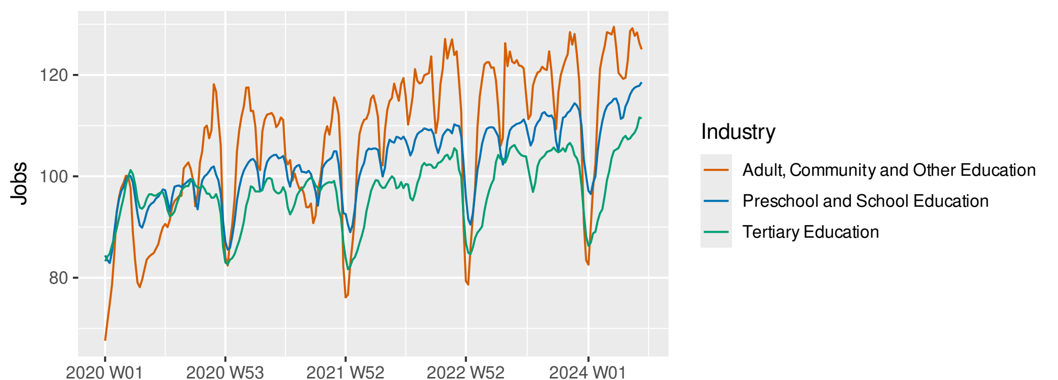
> **Finished early?**
>
> Try combining the student and staff datasets to create a time plot which directly compares the number of students and staff.

Outline

1 Workshop data
2 Time plots
3 Lab Session 3
4 Seasonal plots
5 Lab Session 4
6 Seasonal or cyclic?
7 Lag plots and autocorrelation
8 Lab Session 4
9 White noise
10 Lab Session 5

Seasonal plots

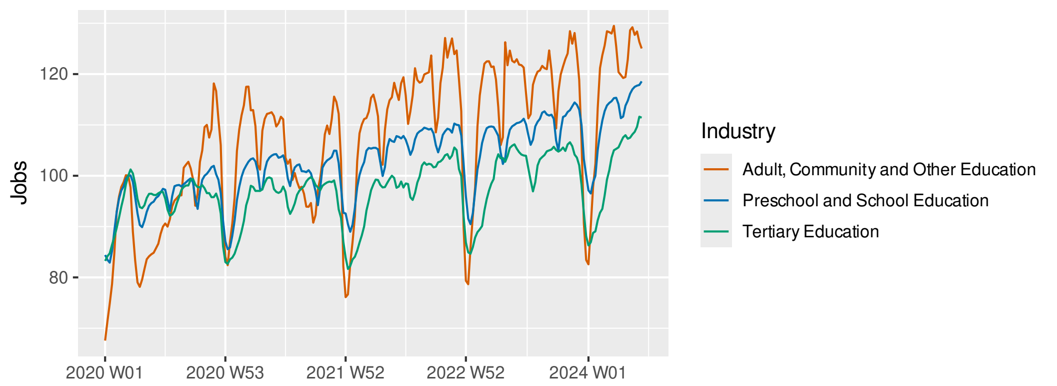
```
payroll_education <- readabs::read_payrolls("subindustry_jobs") |>
  filter(industry_division == "P-Education & training") |>
  transmute(Industry = industry_subdivision, Week = yearweek(date), Jobs = value) |>
  as_tsibble(index = Week, key = Industry)
payroll_education |>
  autoplot(Jobs)
```



Seasonal plots

Ups and downs (peaks and troughs)

When is the seasonal maximum and minimum?

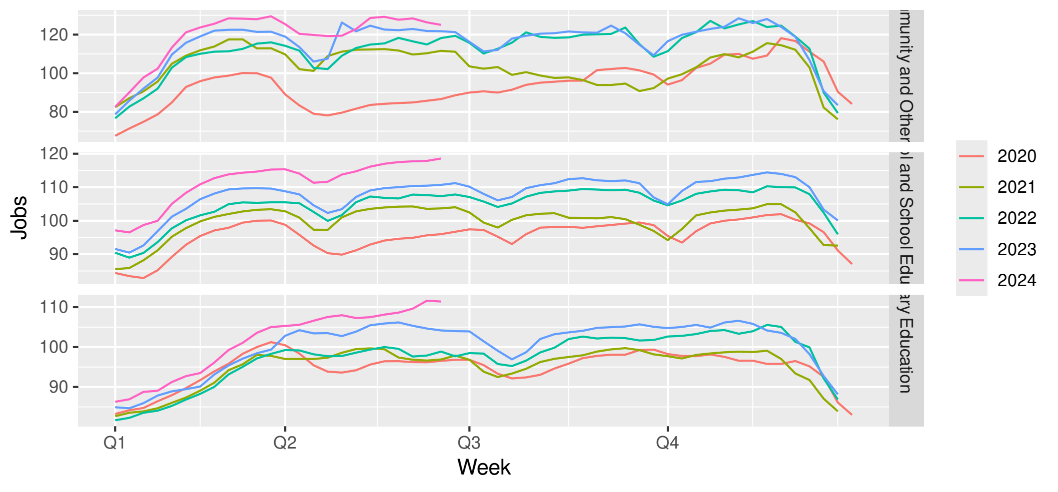


Seasonal plots

- Data plotted against the individual “seasons” in which the data were observed. (In this case a “season” is a month.)
- Something like a time plot except that the data from each season are overlapped.
- Enables the underlying seasonal pattern to be seen more clearly, and also allows any substantial departures from the seasonal pattern to be easily identified.
- In R: `gg_season()`

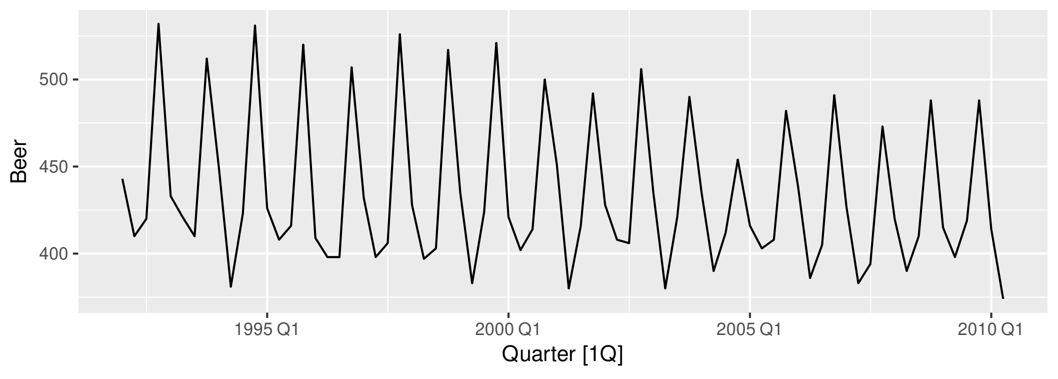
Seasonal plots

```r
payroll_education |>
  gg_season(Jobs)
```



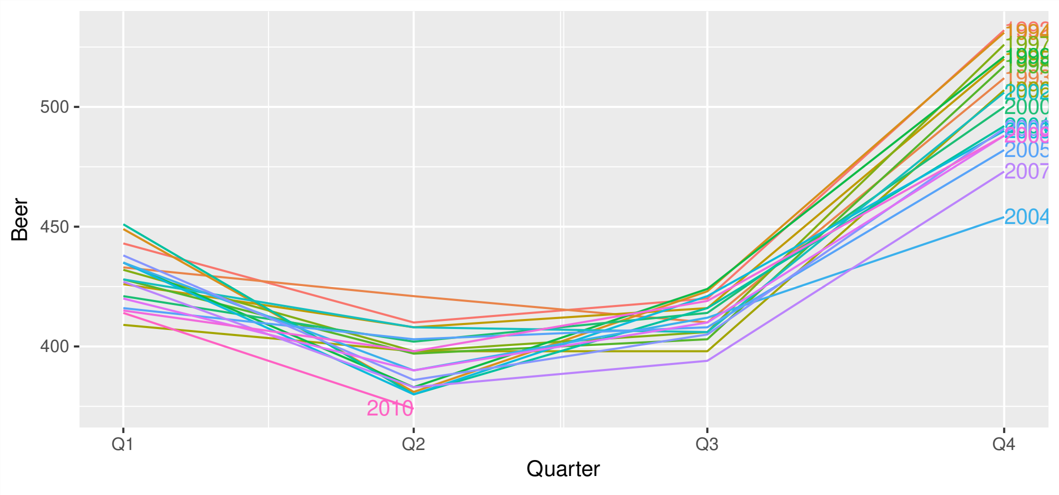
Quarterly Australian Beer Production

```r
beer <- aus_production |>
  select(Quarter, Beer) |>
  filter(year(Quarter) >= 1992)
beer |> autoplot(Beer)
```



Quarterly Australian Beer Production

```r
beer |> gg_season(Beer, labels = "right")
```



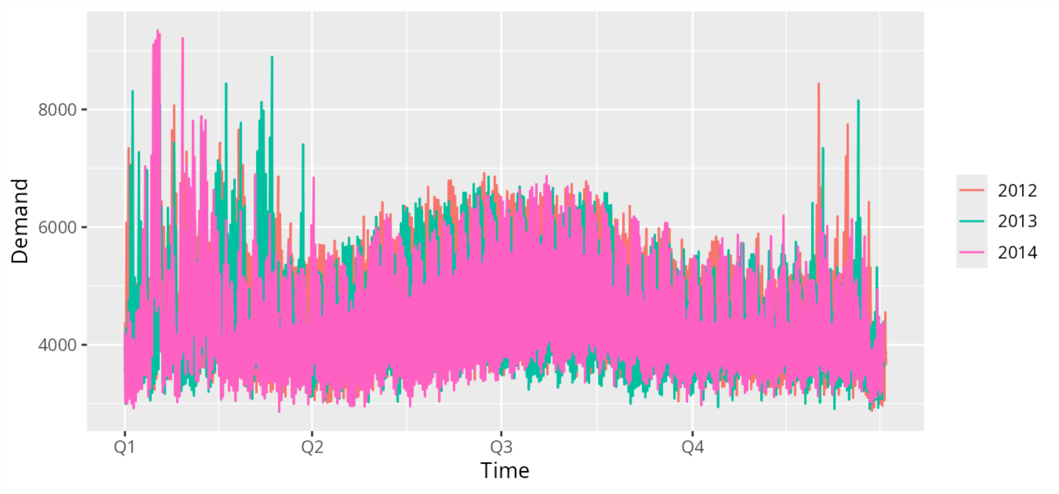
Multiple seasonal periods

```
vic_elec
```

```
# A tsibble: 52,608 x 5 [30m] <Australia/Melbourne>
   Time                Demand Temperature Date       Holiday
   <dttm>               <dbl>       <dbl> <date>     <lgl>
 1 2012-01-01 00:00:00  4383.        21.4 2012-01-01 TRUE
 2 2012-01-01 00:30:00  4263.        21.0 2012-01-01 TRUE
 3 2012-01-01 01:00:00  4049.        20.7 2012-01-01 TRUE
 4 2012-01-01 01:30:00  3878.        20.6 2012-01-01 TRUE
 5 2012-01-01 02:00:00  4036.        20.4 2012-01-01 TRUE
 6 2012-01-01 02:30:00  3866.        20.2 2012-01-01 TRUE
 7 2012-01-01 03:00:00  3694.        20.1 2012-01-01 TRUE
 8 2012-01-01 03:30:00  3562.        19.6 2012-01-01 TRUE
 9 2012-01-01 04:00:00  3433.        19.1 2012-01-01 TRUE
10 2012-01-01 04:30:00  3359.        19.0 2012-01-01 TRUE
# i 52,598 more rows
```

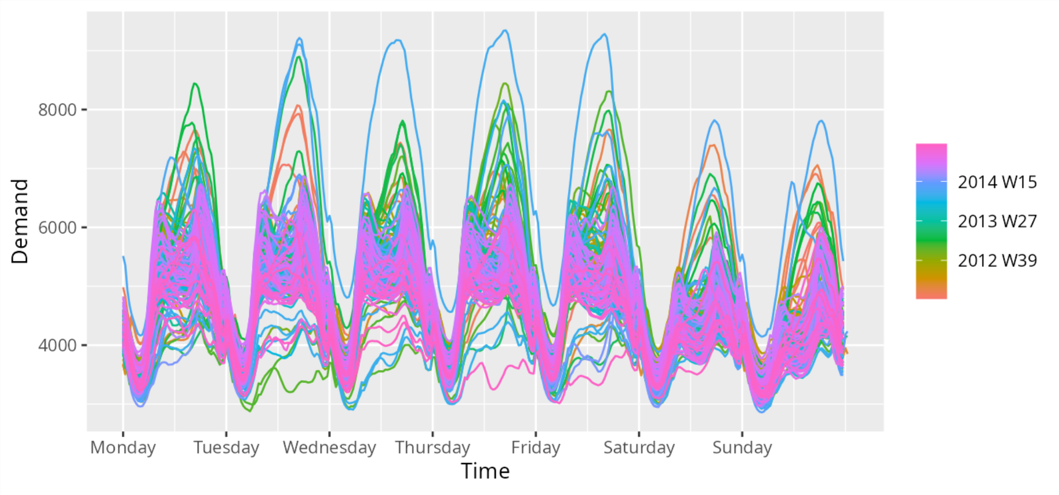
Multiple seasonal periods

```
vic_elec |> gg_season(Demand)
```



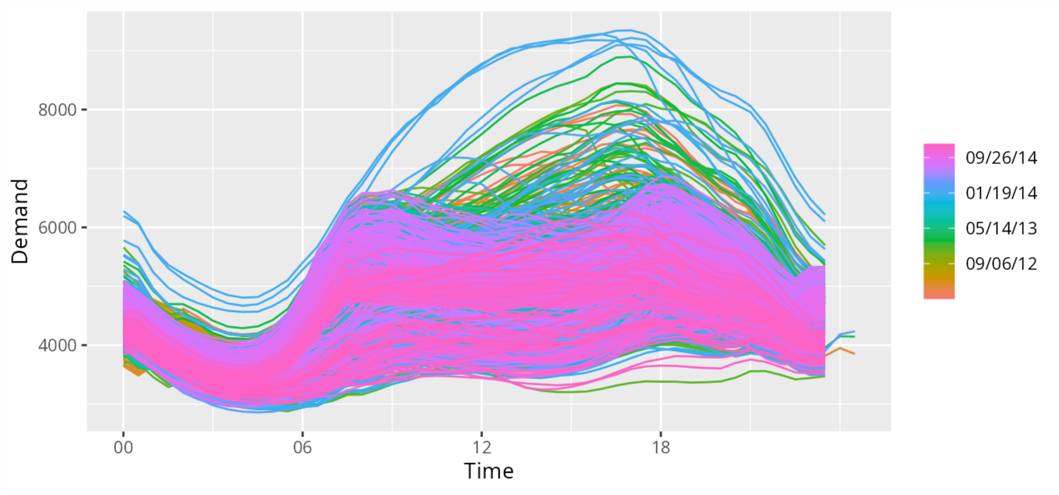
Multiple seasonal periods

```
vic_elec |> gg_season(Demand, period = "week")
```



Multiple seasonal periods

```
vic_elec |> gg_season(Demand, period = "day")
```

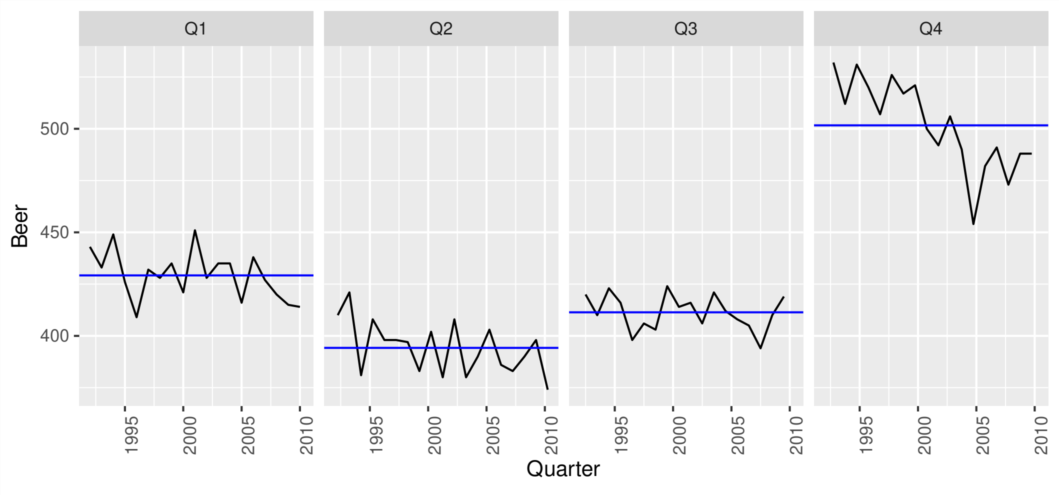


Seasonal subseries plots

- Data for each season collected together in time plot as separate time series.
- Enables the underlying seasonal pattern to be seen clearly, and changes in seasonality over time to be visualized.
- In R: `gg_subseries()`

Quarterly Australian Beer Production

```
beer |> gg_subseries(Beer)
```

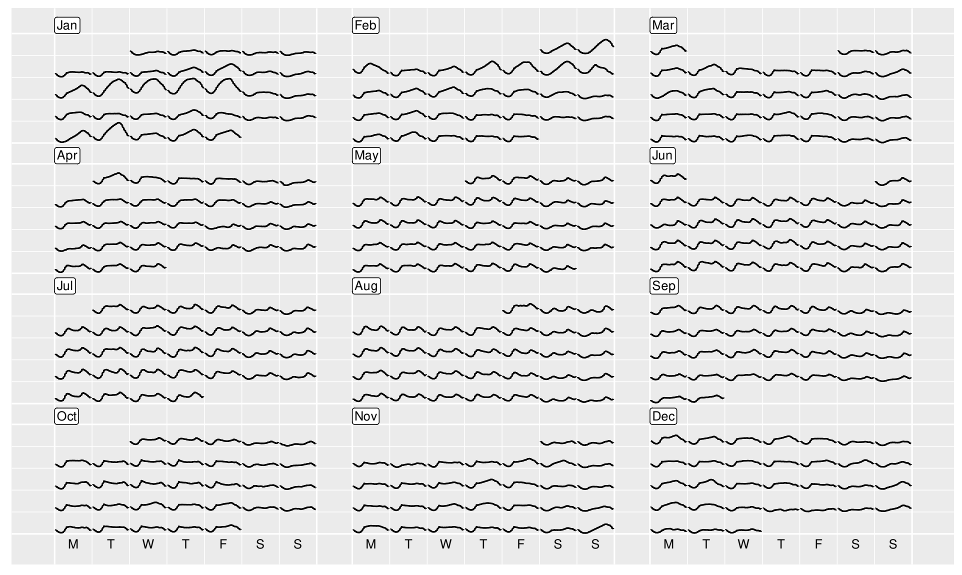


Calendar plots

```r
library(sugrrants)
vic_elec |>
  filter(year(Date) == 2014) |>
  mutate(Hour = hour(Time)) |>
  frame_calendar(x = Hour, y = Demand, date = Date, nrow = 4) |>
  ggplot(aes(x = .Hour, y = .Demand, group = Date)) +
  geom_line() -> p1
prettify(p1,
  size = 3,
  label.padding = unit(0.15, "lines")
)
```

- `frame_calendar()` makes a compact calendar plot
- `facet_calendar()` provides an easier ggplot2 integration.

Calendar plots



Outline

1 Workshop data
2 Time plots
3 Lab Session 3
4 Seasonal plots
5 Lab Session 4
6 Seasonal or cyclic?
7 Lag plots and autocorrelation
8 Lab Session 4
9 White noise
10 Lab Session 5

Lab Session 4

1. Look at the monthly labour force of 15-24 year olds by State/Territory and educational attendance.

   Data is sourced from the ABS 6202.0 Table 16.

   The code to prepare this data is in `student_labour.R`.

2. Use `autoplot()`, `gg_season()` and `gg_subseries()` to explore the data.

   Look at different aggregations of the data, for example total persons by attendance.

3. What do you learn?

Outline

1. Workshop data
2. Time plots
3. Lab Session 3
4. Seasonal plots
5. Lab Session 4
6. Seasonal or cyclic?
7. Lag plots and autocorrelation
8. Lab Session 4
9. White noise
10. Lab Session 5

Time series patterns

Trend pattern exists when there is a long-term increase or decrease in the data.

Seasonal pattern exists when a series is influenced by seasonal factors (e.g., the quarter of the year, the month, or day of the week).

Cyclic pattern exists when data exhibit rises and falls that are *not of fixed period* (duration usually of at least 2 years).

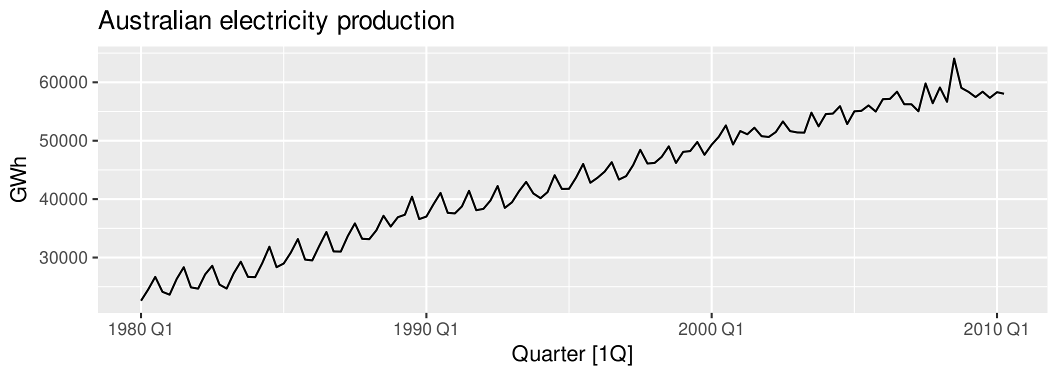
Time series components

Differences between seasonal and cyclic patterns:

- seasonal pattern constant length; cyclic pattern variable length
- average length of cycle longer than length of seasonal pattern
- magnitude of cycle more variable than magnitude of seasonal pattern

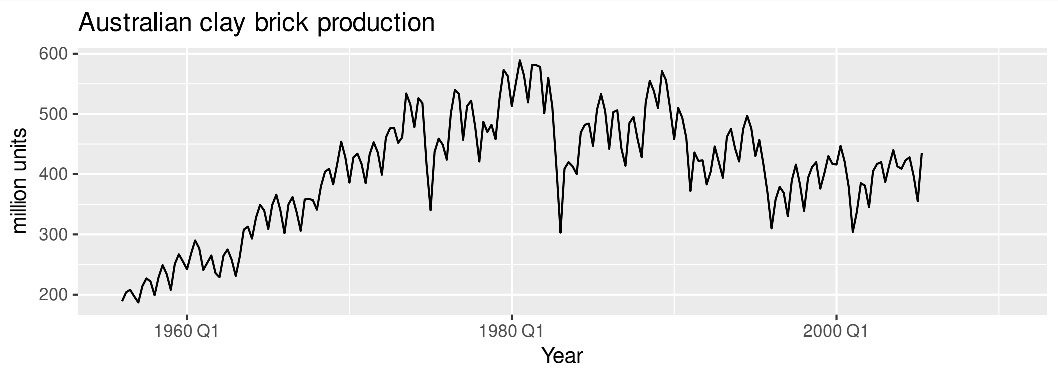
Time series patterns

```r
aus_production |>
  filter(year(Quarter) >= 1980) |>
  autoplot(Electricity) +
  labs(y = "GWh", title = "Australian electricity production")
```



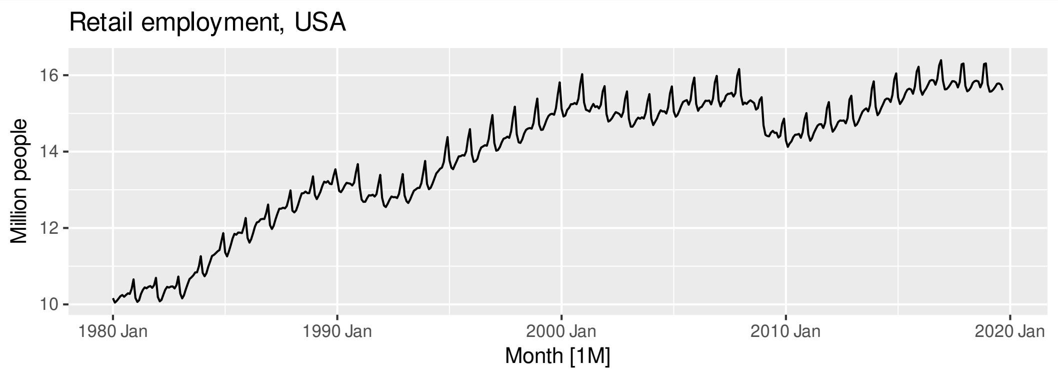
Time series patterns

```r
aus_production |>
  autoplot(Bricks) +
  labs(title = "Australian clay brick production",
       x = "Year", y = "million units")
```



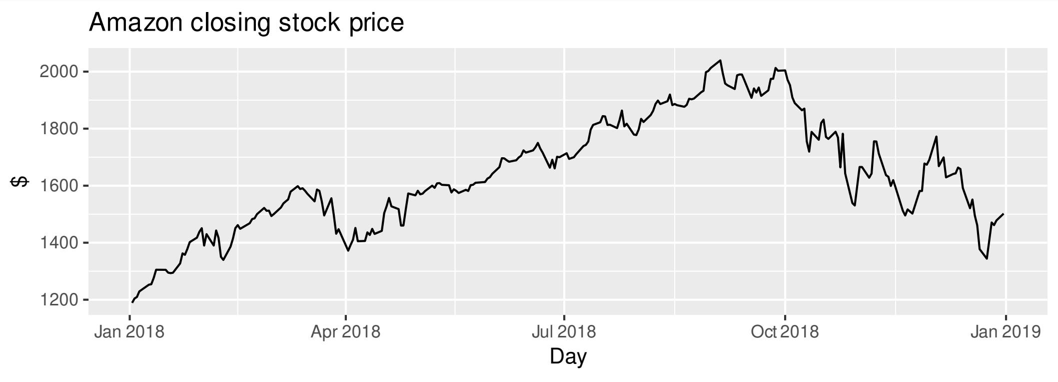
Time series patterns

```r
us_employment |>
  filter(Title == "Retail Trade", year(Month) >= 1980) |>
  autoplot(Employed / 1e3) +
  labs(title = "Retail employment, USA", y = "Million people")
```



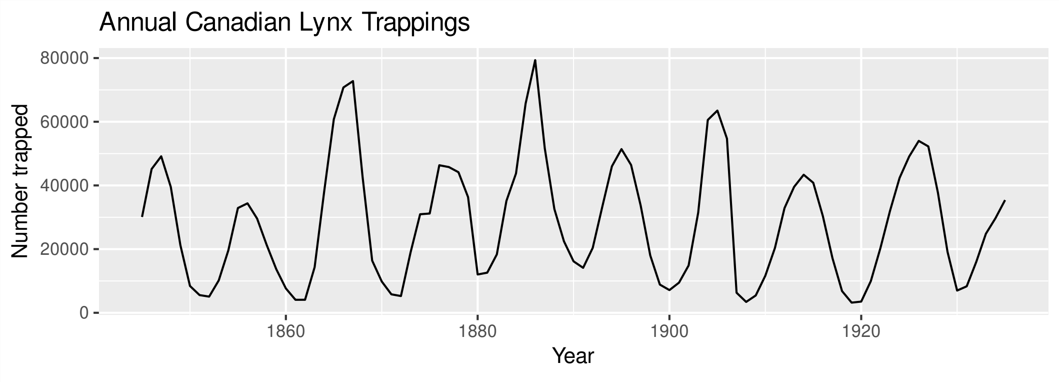
Time series patterns

```
gafa_stock |>
  filter(Symbol == "AMZN", year(Date) >= 2018) |>
  autoplot(Close) +
  labs(title = "Amazon closing stock price", x = "Day", y = "$")
```



Time series patterns

```r
pelt |>
  autoplot(Lynx) +
  labs(title = "Annual Canadian Lynx Trappings",
       x = "Year", y = "Number trapped")
```



Seasonal or cyclic?

Differences between seasonal and cyclic patterns:

- seasonal pattern constant length; cyclic pattern variable length
- average length of cycle longer than length of seasonal pattern
- magnitude of cycle more variable than magnitude of seasonal pattern

Seasonal or cyclic?

Differences between seasonal and cyclic patterns:

- seasonal pattern constant length; cyclic pattern variable length
- average length of cycle longer than length of seasonal pattern
- magnitude of cycle more variable than magnitude of seasonal pattern

The timing of peaks and troughs is predictable with seasonal data, but unpredictable in the long term with cyclic data.

Outline

1 Workshop data
2 Time plots
3 Lab Session 3
4 Seasonal plots
5 Lab Session 4
6 Seasonal or cyclic?
7 Lag plots and autocorrelation
8 Lab Session 4
9 White noise
10 Lab Session 5

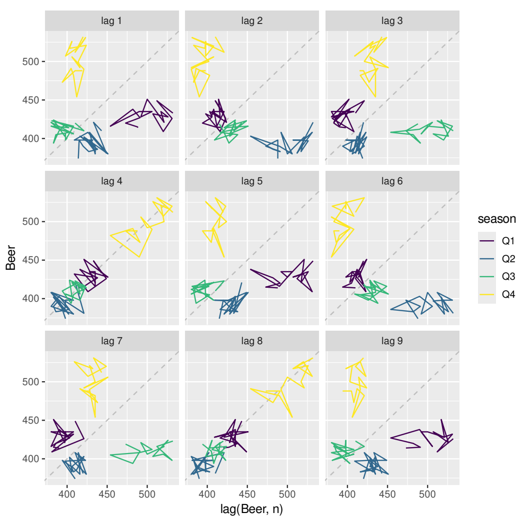
Example: Beer production

```r
new_production <- aus_production |>
  filter(year(Quarter) >= 1992)
new_production
```

```
# A tsibble: 74 x 7 [1Q]
   Quarter  Beer Tobacco Bricks Cement Electricity   Gas
     <qtr> <dbl>   <dbl>  <dbl>  <dbl>       <dbl> <dbl>
 1 1992 Q1   443    5777    383   1289       38332   117
 2 1992 Q2   410    5853    404   1501       39774   151
 3 1992 Q3   420    6416    446   1539       42246   175
 4 1992 Q4   532    5825    420   1568       38498   129
 5 1993 Q1   433    5724    394   1450       39460   116
 6 1993 Q2   421    6036    462   1668       41356   149
 7 1993 Q3   410    6570    475   1648       42949   163
 8 1993 Q4   512    5675    443   1863       40974   138
 9 1994 Q1   449    5311    421   1468       40162   127
10 1994 Q2   381    5717    475   1755       41199   159
# i 64 more rows
```

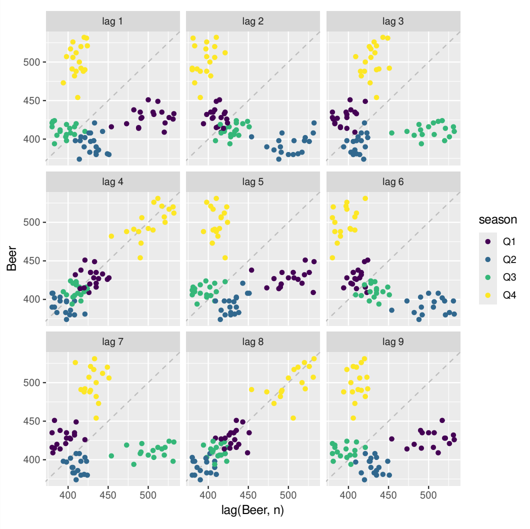
Example: Beer production

```
new_production |> gg_lag(Beer)
```



Example: Beer production

```
new_production |> gg_lag(Beer, geom = "point")
```



Lagged scatterplots

- Each graph shows y_t plotted against y_{t-k} for different values of k.
- The autocorrelations are the correlations associated with these scatterplots.
- ACF (autocorrelation function):
  - $r_1 = \text{Correlation}(y_t, y_{t-1})$
  - $r_2 = \text{Correlation}(y_t, y_{t-2})$
  - $r_3 = \text{Correlation}(y_t, y_{t-3})$
  - etc.
- If there is **seasonality**, the ACF at the seasonal lag (e.g., 12 for monthly data) will be **large and positive**.

Autocorrelation

Results for first 9 lags for beer data:

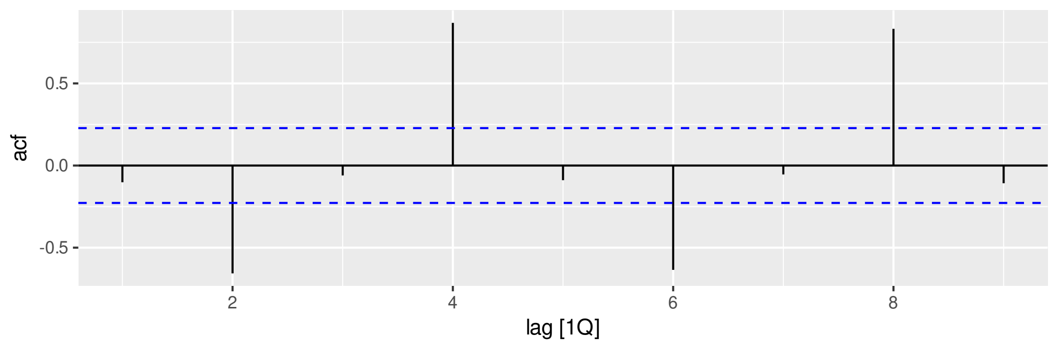
```
new_production |> ACF(Beer, lag_max = 9)
```

```
# A tsibble: 9 x 2 [1Q]
        lag      acf
  <cf_lag>    <dbl>
1        1Q -0.102
2        2Q -0.657
3        3Q -0.0603
4        4Q  0.869
5        5Q -0.0892
6        6Q -0.635
7        7Q -0.0542
8        8Q  0.832
9        9Q -0.108
```

Autocorrelation

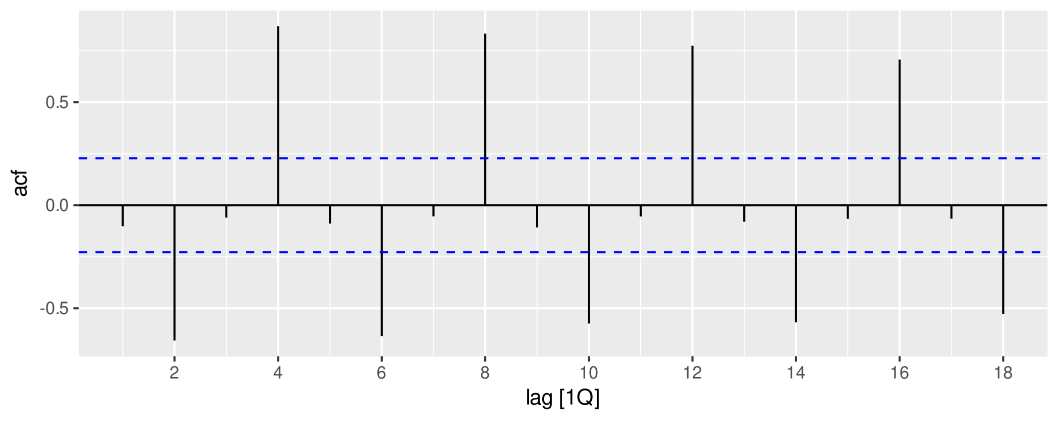
Results for first 9 lags for beer data:

```
new_production |>
  ACF(Beer, lag_max = 9) |>
  autoplot()
```



ACF

```
new_production |>
  ACF(Beer) |>
  autoplot()
```



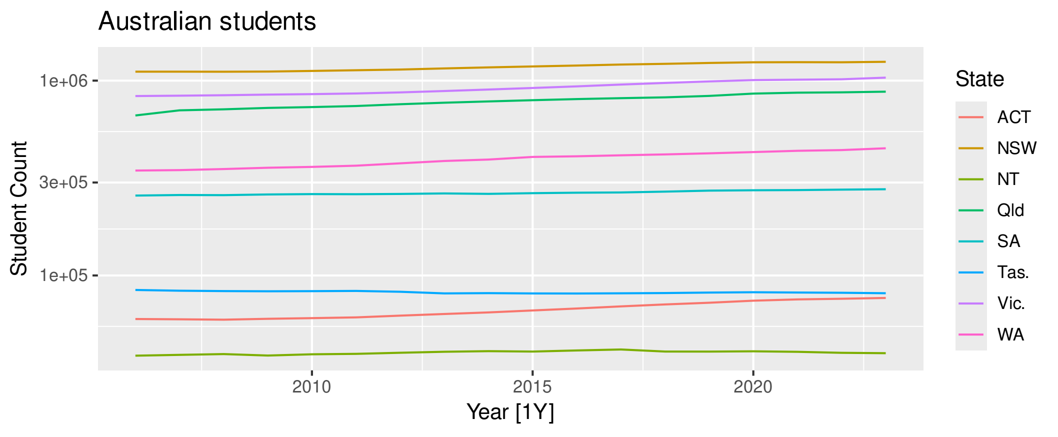
Australian student enrolments

```r
students <- readxl::read_excel("data/schools/Table 42b Number of Full-time and Part-
  # Group by Year and all the character variables
  group_by(Year, across(where(is.character), identity)) |>
  # Add up the duplicate rows
  summarise(across(ends_with("count"), sum), .groups = "drop") |>
  # Convert to a tsibble
  as_tsibble(
    key = where(is.character),
    index = Year
  )
```

```
# A tsibble: 81,307 x 13 [1Y]
# Key:       State/Territory, Affiliation (Gov/Non-gov), Affiliation
#   (Gov/Cath/Ind), Sex, Aboriginal and Torres Strait Islander Status,
#   School Level, National Report on Schooling (ANR) School Level, Year
#   (Grade), Age [6,575]
    Year `State/Territory` Affiliation (Gov/Non-go~1 Affiliation (Gov/Cat~2
   <dbl> <chr>             <chr>                     <chr>
 1  2006 a NSW             a Government              a Government
```

Australian student enrolments

```
students |> autoplot(Count) +
  labs(y = "Student Count", title = "Australian students") +
  scale_y_log10()
```



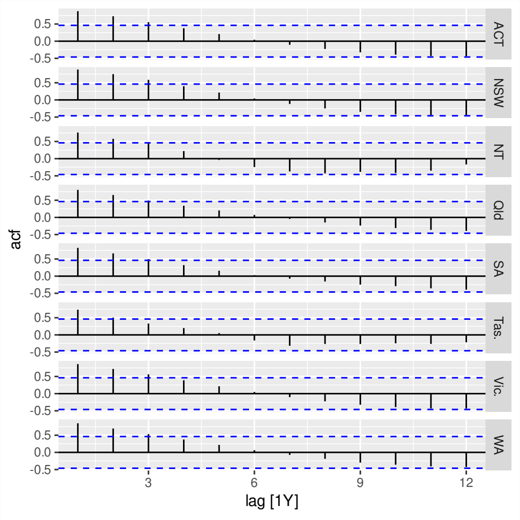
Australian holidays

```
students |> ACF(Count)
```

```
# A tsibble: 96 x 3 [1Y]
# Key:       State [8]
   State      lag     acf
   <chr> <cf_lag>   <dbl>
 1 ACT         1Y  0.875
 2 ACT         2Y  0.727
 3 ACT         3Y  0.555
 4 ACT         4Y  0.377
 5 ACT         5Y  0.207
 6 ACT         6Y  0.0415
 7 ACT         7Y -0.104
 8 ACT         8Y -0.226
 9 ACT         9Y -0.324
10 ACT        10Y -0.392
# i 86 more rows
```

Australian holidays

```
students |> ACF(Count) |> autoplot()
```

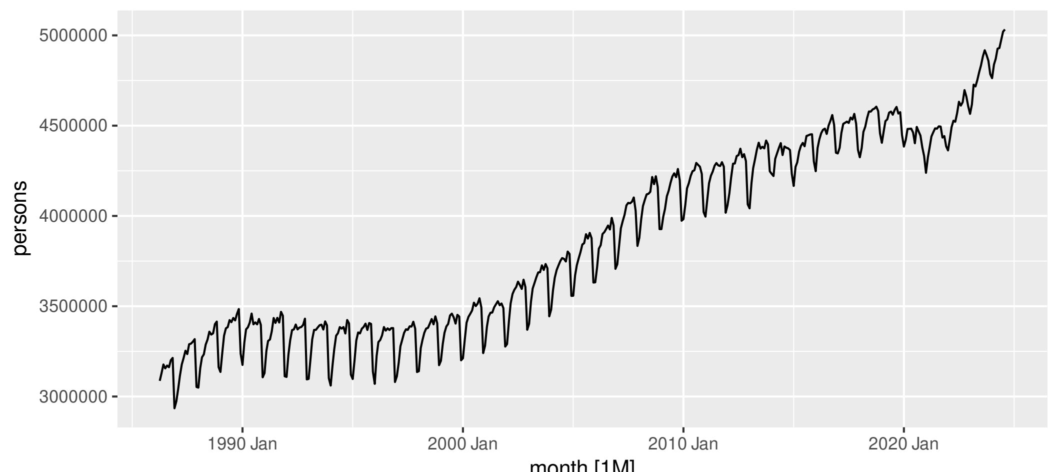


Trend and seasonality in ACF plots

- When data have a trend, the autocorrelations for small lags tend to be large and positive.
- When data are seasonal, the autocorrelations will be larger at the seasonal lags (i.e., at multiples of the seasonal frequency)
- When data are trended and seasonal, you see a combination of these effects.

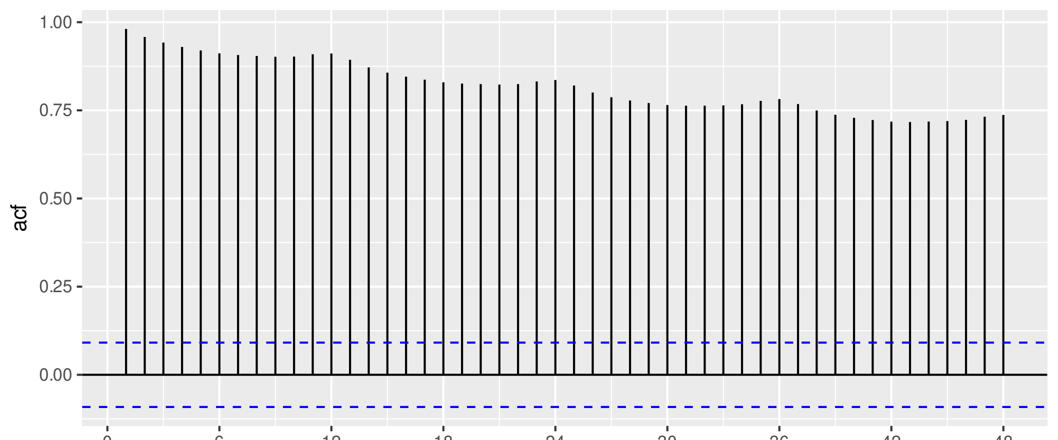
Youth labour force

```
student_labour |>
  summarise(persons = sum(persons)) |>
  autoplot(persons)
```



US retail trade employment

```r
student_labour |>
  summarise(persons = sum(persons)) |>
  ACF(persons, lag_max = 48) |>
  autoplot()
```



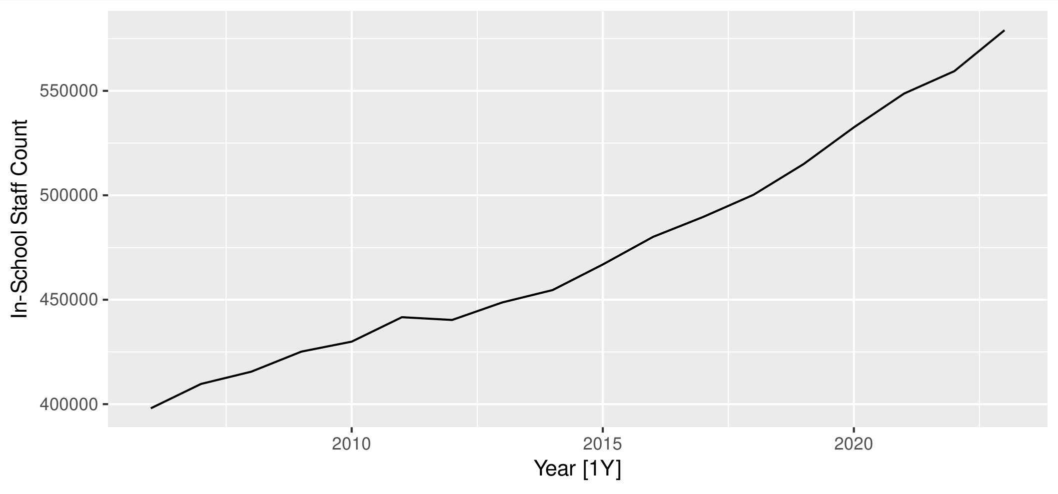
In-school staff

```
staff_total <- staff |>
  summarise(`In-School Staff Count` = sum(`In-School Staff Count`))
staff_total
```

```
# A tsibble: 18 x 2 [1Y]
    Year `In-School Staff Count`
   <dbl>                   <dbl>
 1  2006                  398003
 2  2007                  409678
 3  2008                  415541
 4  2009                  425166
 5  2010                  429933
 6  2011                  441631
 7  2012                  440313
 8  2013                  448711
 9  2014                  454615
10  2015                  466867
11  2016                  480077
```

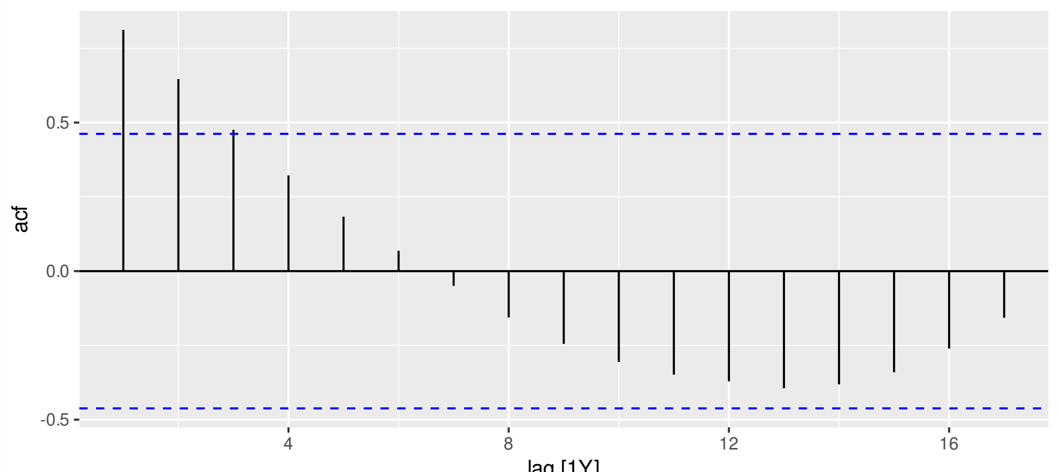
Google stock price

```
staff_total |> autoplot(`In-School Staff Count`)
```



Google stock price

```
staff_total |>
  ACF(`In-School Staff Count`, lag_max = 100) |>
  autoplot()
```



Outline

1 Workshop data
2 Time plots
3 Lab Session 3
4 Seasonal plots
5 Lab Session 4
6 Seasonal or cyclic?
7 Lag plots and autocorrelation
8 Lab Session 4
9 White noise
10 Lab Session 5

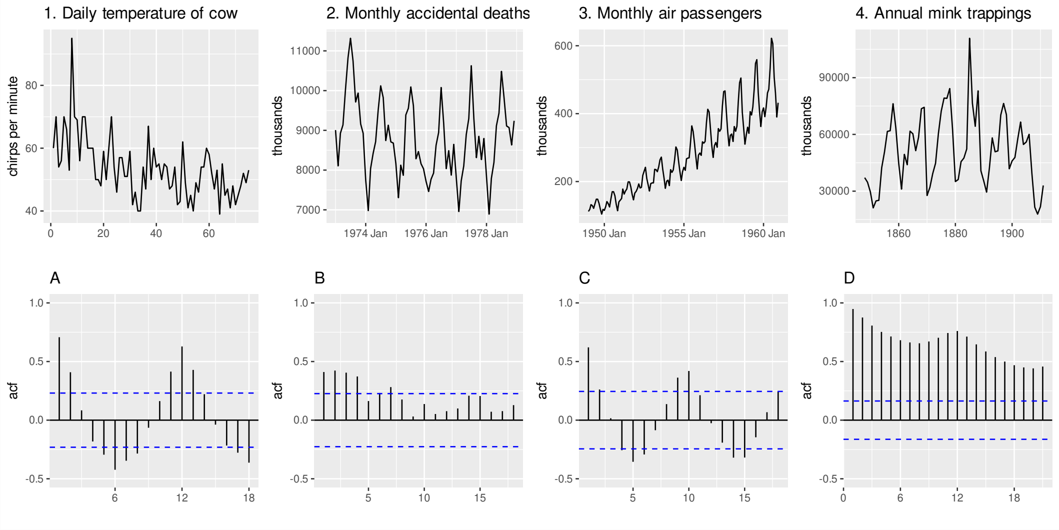
Lab Session 4

Explore the `ACF()` of the following time series:

1. Education payroll (`payroll_education`)
2. Total full-time students (`students`)
3. Total working students aged 15-24 in ACT (`student_labour`)

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

Which is which?

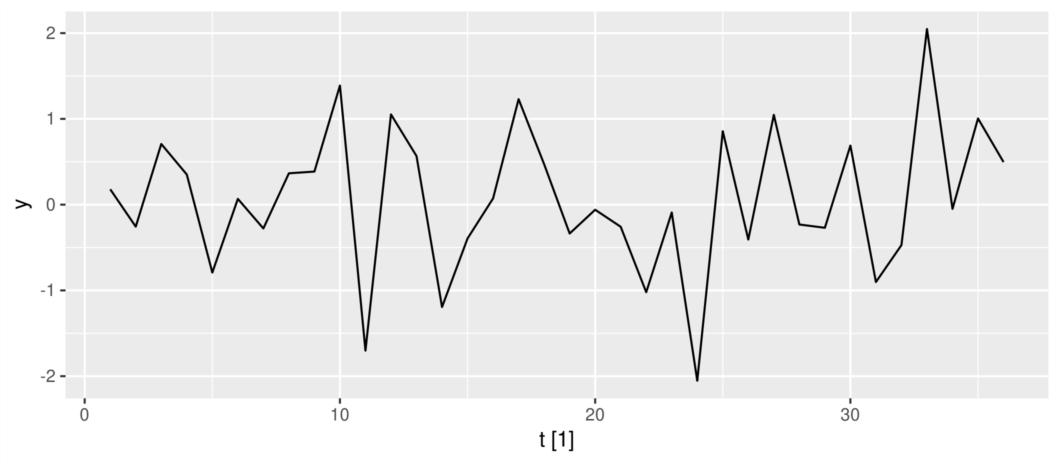


Outline

1. Workshop data
2. Time plots
3. Lab Session 3
4. Seasonal plots
5. Lab Session 4
6. Seasonal or cyclic?
7. Lag plots and autocorrelation
8. Lab Session 4
9. White noise
10. Lab Session 5

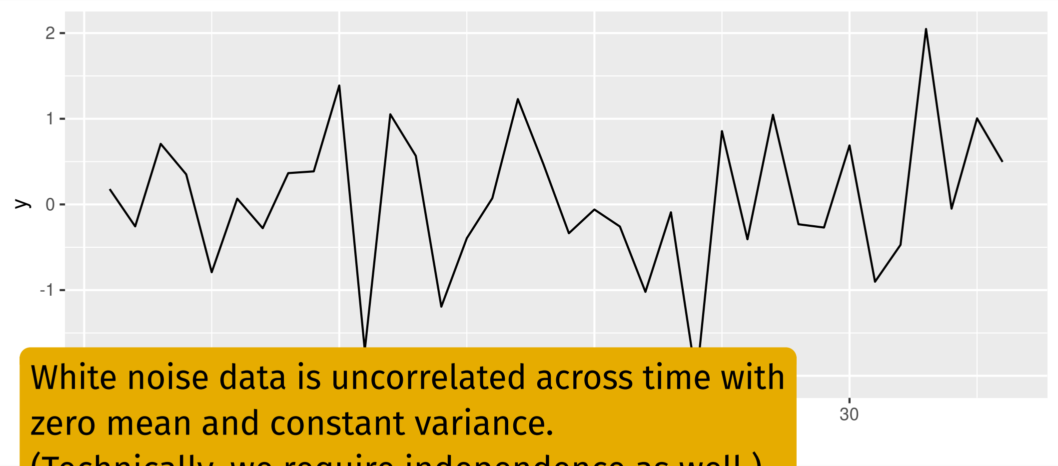
Example: White noise

```
wn <- tsibble(t = seq(36), y = rnorm(36), index = t)
wn |> autoplot(y)
```



Example: White noise

```r
wn <- tsibble(t = seq(36), y = rnorm(36), index = t)
wn |> autoplot(y)
```

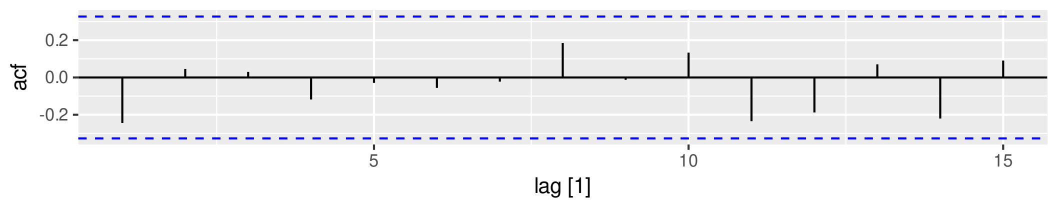


White noise data is uncorrelated across time with zero mean and constant variance.
(Technically, we require independence as well.)

Example: White noise

```
wn |> ACF(y)
```

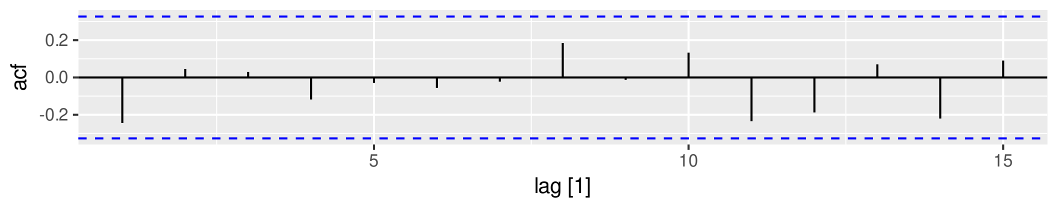
r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}
-0.244	0.046	0.030	-0.118	-0.029	-0.056	-0.022	0.185	-0.013	0.133



Example: White noise

```
wn |> ACF(y)
```

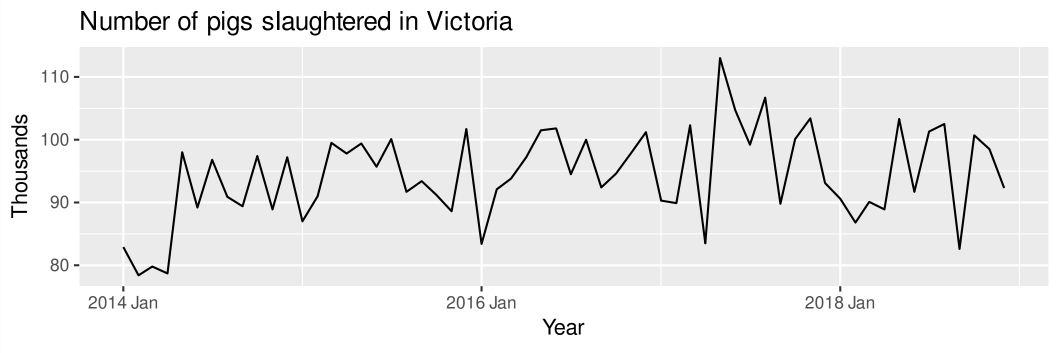
r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}
-0.244	0.046	0.030	-0.118	-0.029	-0.056	-0.022	0.185	-0.013	0.133



- Sample autocorrelations for white noise series.
- Expect each autocorrelation to be close to zero.
- Blue lines show 95% critical values.

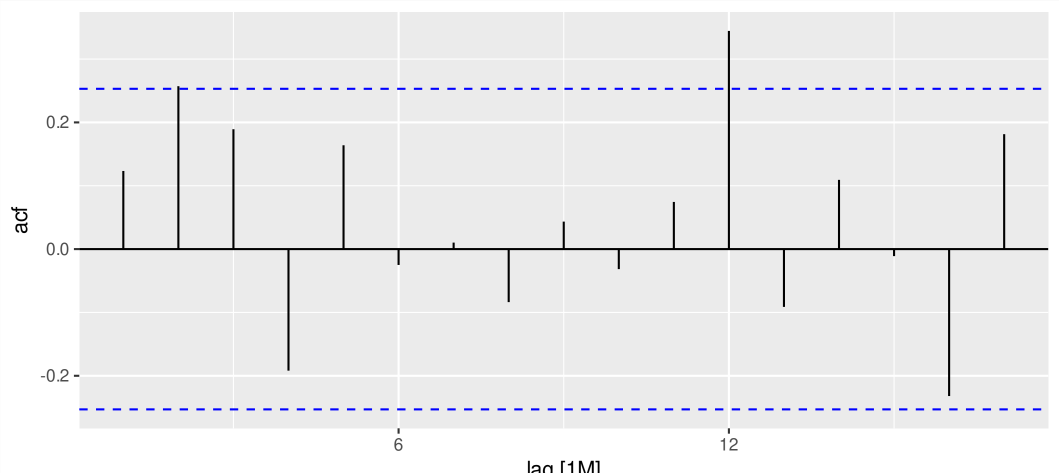
Example: Pigs slaughtered

```
pigs <- aus_livestock |>
  filter(State == "Victoria", Animal == "Pigs", year(Month) >= 2014)
pigs |> autoplot(Count / 1e3) +
  labs(x = "Year", y = "Thousands",
       title = "Number of pigs slaughtered in Victoria")
```



Example: Pigs slaughtered

```
pigs |>
  ACF(Count) |>
  autoplot()
```



Example: Pigs slaughtered

Monthly total number of pigs slaughtered in the state of Victoria, Australia, from January 2014 through December 2018 (Source: Australian Bureau of Statistics.)

Example: Pigs slaughtered

Monthly total number of pigs slaughtered in the state of Victoria, Australia, from January 2014 through December 2018 (Source: Australian Bureau of Statistics.)

- Difficult to detect pattern in time plot.
- ACF shows significant autocorrelation for lag 2 and 12.
- Indicate some slight seasonality.

Example: Pigs slaughtered

Monthly total number of pigs slaughtered in the state of Victoria, Australia, from January 2014 through December 2018 (Source: Australian Bureau of Statistics.)

- Difficult to detect pattern in time plot.
- ACF shows significant autocorrelation for lag 2 and 12.
- Indicate some slight seasonality.

These show the series is **not a white noise series**.

Outline

1 Workshop data
2 Time plots
3 Lab Session 3
4 Seasonal plots
5 Lab Session 4
6 Seasonal or cyclic?
7 Lag plots and autocorrelation
8 Lab Session 4
9 White noise
10 Lab Session 5

Lab Session 5

Plot the difference in ACT student enrolments, it can be done as follows:

```
students |>
  filter(`State/Territory` == "ACT") |>
  summarise(total = sum(`All Full-time and Part-time Student count`)) |>
  autoplot(difference(total))
```

Does `diff` look like white noise? Hint: Check the ACF