MONASH University

MONASH
BUSINESS
SCHOOL

# High dimensional time series analysis

tsibble feasts Fable

## 1. Tidy time series tsibbles

robjhyndman.com/hdtsa

# Outline

1. Time series data and tsibbles
2. Working with `tsibble` objects
3. Create a tsibble from a csv
4. Time plots

# Outline

1 Time series data and tsibbles

2 Working with `tsibble` objects

3 Create a tsibble from a csv

4 Time plots

# Tidyverts packages

tidyverts.org

# Time series data

- Four-yearly Olympic winning times
- Annual Google profits
- Quarterly Australian beer production
- Monthly rainfall
- Weekly retail sales
- Daily IBM stock prices
- Hourly electricity demand
- 5-minute freeway traffic counts
- Time-stamped stock transaction data

# Class packages

```r
# Data manipulation and plotting functions
library(tidyverse)
# Time series manipulation
library(tsibble)
# Forecasting functions
library(fable)
# Time series graphics and statistics
library(feasts)
# Tidy time series data
library(tsibbledata)
```

# Class packages

```r
# Data manipulation and plotting functions
library(tidyverse)
# Time series manipulation
library(tsibble)
# Forecasting functions
library(fable)
# Time series graphics and statistics
library(feasts)
# Tidy time series data
library(tsibbledata)
```

```r
# Almost all of the above
library(fpp3)
```

# tsibble objects

```
global_economy
```

```
## # A tsibble: 15,150 x 6 [1Y]
## # Key:       Country [263]
##     Year Country             GDP Imports Exports Population
##    <dbl> <fct>             <dbl>   <dbl>   <dbl>      <dbl>
##  1  1960 Afghanistan  537777811.    7.02    4.13    8996351
##  2  1961 Afghanistan  548888896.    8.10    4.45    9166764
##  3  1962 Afghanistan  546666678.    9.35    4.88    9345868
##  4  1963 Afghanistan  751111191.   16.9     9.17    9533954
##  5  1964 Afghanistan  800000044.   18.1     8.89    9731361
##  6  1965 Afghanistan 1006666638.   21.4    11.3     9938414
##  7  1966 Afghanistan 1399999967.   18.6     8.57   10152331
##  8  1967 Afghanistan 1673333418.   14.2     6.77   10372630
##  9  1968 Afghanistan 1373333367.   15.2     8.90   10604346
## 10  1969 Afghanistan 1408888922.   15.0    10.1    10854428
## # ... with 15,140 more rows
```

# tsibble objects

```
global_economy
```

```
## # A tsibble: 15,150 x 6 [1Y]
## # Key:       Country [263]
##     Year Country             GDP Imports Exports Population
##    Index <fct>             <dbl>   <dbl>   <dbl>      <dbl>
##  1  1960 Afghanistan  537777811.    7.02    4.13    8996351
##  2  1961 Afghanistan  548888896.    8.10    4.45    9166764
##  3  1962 Afghanistan  546666678.    9.35    4.88    9345868
##  4  1963 Afghanistan  751111191.   16.9     9.17    9533954
##  5  1964 Afghanistan  800000044.   18.1     8.89    9731361
##  6  1965 Afghanistan 1006666638.   21.4    11.3     9938414
##  7  1966 Afghanistan 1399999967.   18.6     8.57   10152331
##  8  1967 Afghanistan 1673333418.   14.2     6.77   10372630
##  9  1968 Afghanistan 1373333367.   15.2     8.90   10604346
## 10  1969 Afghanistan 1408888922.   15.0    10.1    10854428
## # ... with 15,140 more rows
```

# tsibble objects

```
global_economy
```

```
## # A tsibble: 15,150 x 6 [1Y]
## # Key:       Country [263]
##     Year Country               GDP Imports Exports Population
##     Index Key                <dbl>   <dbl>   <dbl>      <dbl>
##  1  1960 Afghanistan  537777811.    7.02    4.13    8996351
##  2  1961 Afghanistan  548888896.    8.10    4.45    9166764
##  3  1962 Afghanistan  546666678.    9.35    4.88    9345868
##  4  1963 Afghanistan  751111191.   16.9     9.17    9533954
##  5  1964 Afghanistan  800000044.   18.1     8.89    9731361
##  6  1965 Afghanistan 1006666638.   21.4    11.3     9938414
##  7  1966 Afghanistan 1399999967.   18.6     8.57   10152331
##  8  1967 Afghanistan 1673333418.   14.2     6.77   10372630
##  9  1968 Afghanistan 1373333367.   15.2     8.90   10604346
## 10  1969 Afghanistan 1408888922.   15.0    10.1    10854428
## # ... with 15,140 more rows
```

# tsibble objects

```r
global_economy
```

```
## # A tsibble: 15,150 x 6 [1Y]
## # Key:       Country [263]
##     Year Country             GDP Imports Exports Population
##     Index Key                Measured variables
##  1  1960 Afghanistan  537777811.    7.02    4.13    8996351
##  2  1961 Afghanistan  548888896.    8.10    4.45    9166764
##  3  1962 Afghanistan  546666678.    9.35    4.88    9345868
##  4  1963 Afghanistan  751111191.   16.9     9.17    9533954
##  5  1964 Afghanistan  800000044.   18.1     8.89    9731361
##  6  1965 Afghanistan 1006666638.   21.4    11.3     9938414
##  7  1966 Afghanistan 1399999967.   18.6     8.57   10152331
##  8  1967 Afghanistan 1673333418.   14.2     6.77   10372630
##  9  1968 Afghanistan 1373333367.   15.2     8.90   10604346
## 10  1969 Afghanistan 1408888922.   15.0    10.1    10854428
## # ... with 15,140 more rows
```

# tsibble objects

```
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 SA    Business  135.
##  2 1998 Q2 Adelaide SA    Business  110.
##  3 1998 Q3 Adelaide SA    Business  166.
##  4 1998 Q4 Adelaide SA    Business  127.
##  5 1999 Q1 Adelaide SA    Business  137.
##  6 1999 Q2 Adelaide SA    Business  200.
##  7 1999 Q3 Adelaide SA    Business  169.
##  8 1999 Q4 Adelaide SA    Business  134.
##  9 2000 Q1 Adelaide SA    Business  154.
## 10 2000 Q2 Adelaide SA    Business  169.
## # ... with 24,310 more rows
```

# tsibble objects

```
tourism
```

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

# tsibble objects

```
tourism
```

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

# tsibble objects

```
tourism
```

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

# tsibble objects

```
tourism
```

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

Index

Keys

Measure

Domestic visitor nights in thousands by state/region and purpose.

# `tsibble` objects

- A `tsibble` allows storage and manipulation of time series in R.
- It contains:
  - Measured variable(s): numbers of interest
  - An index: time information about the observation
  - Key variable(s): optional unique identifiers for each series
- It works with tidyverse functions.

# The tsibble index

## Example

```
library(tsibble)
y <- tsibble(year = 2012:2016,
  y = c(123,39,78,52,110), index = year)
y
```

```
## # A tsibble: 5 x 2 [1Y]
##    year     y
##   <int> <dbl>
## 1  2012   123
## 2  2013    39
## 3  2014    78
## 4  2015    52
## 5  2016   110
```

# The `tsibble` index

For observations more frequent than once per year, we need to use a time class function on the index.

```
z
```

```
## # A tibble: 5 x 2
##   Month    Observation
##   <chr>          <dbl>
## 1 2019 Jan          50
## 2 2019 Feb          23
## 3 2019 Mar          34
## 4 2019 Apr          30
## 5 2019 May          25
```

# The tsibble index

For observations more frequent than once per year, we need to use a time class function on the index.

```
z %>%
  mutate(Month = yearmonth(Month)) %>%
  as_tsibble(index = Month)
```

```
## # A tsibble: 5 x 2 [1M]
##      Month Observation
##      <mth>       <dbl>
## 1 2019 Jan          50
## 2 2019 Feb          23
## 3 2019 Mar          34
## 4 2019 Apr          30
## 5 2019 May          25
```

# The `tsibble` index

Common time index variables can be created with these functions:

| Frequency | Function |
|---|---|
| Annual | `start:end` |
| Quarterly | `yearquarter()` |
| Monthly | `yearmonth()` |
| Weekly | `yearweek()` |
| Daily | `as_date()`, `ymd()` |
| Sub-daily | `as_datetime()` |

# Outline

1 Time series data and tsibbles

2 Working with `tsibble` objects

3 Create a tsibble from a csv

4 Time plots

# Australian Pharmaceutical Benefits Scheme

# Australian Pharmaceutical Benefits Scheme

The **Pharmaceutical Benefits Scheme** (PBS) is the Australian government drugs subsidy scheme.

# Australian Pharmaceutical Benefits Scheme

The **Pharmaceutical Benefits Scheme** (PBS) is the Australian government drugs subsidy scheme.

- Many drugs bought from pharmacies are subsidised to allow more equitable access to modern drugs.
- The cost to government is determined by the number and types of drugs purchased. Currently nearly 1% of GDP.
- The total cost is budgeted based on forecasts of drug usage.
- Costs are disaggregated by drug type (ATC1 x15 / ATC2 84), concession category (x2) and patient type (x2), giving 84 $\times$ 2 $\times$ 2 = 336 time series.

# Working with `tsibble` objects

```
PBS
```

```
## # A tsibble: 65,219 x 9 [1M]
## # Key:       Concession, Type, ATC1, ATC2 [336]
##          Month Concession Type  ATC1  ATC1_desc ATC2
##          <mth> <chr>      <chr> <chr> <chr>     <chr>
##  1   1991 Jul Concessio~ Co-p~ A     Alimenta~ A01
##  2   1991 Aug Concessio~ Co-p~ A     Alimenta~ A01
##  3   1991 Sep Concessio~ Co-p~ A     Alimenta~ A01
##  4   1991 Oct Concessio~ Co-p~ A     Alimenta~ A01
##  5   1991 Nov Concessio~ Co-p~ A     Alimenta~ A01
##  6   1991 Dec Concessio~ Co-p~ A     Alimenta~ A01
##  7   1992 Jan Concessio~ Co-p~ A     Alimenta~ A01
##  8   1992 Feb Concessio~ Co-p~ A     Alimenta~ A01
##  9   1992 Mar Concessio~ Co-p~ A     Alimenta~ A01
## 10   1992 Apr Concessio~ Co-p~ A     Alimenta~ A01
## # ... with 65,209 more rows, and 3 more variables:
## #   ATC2_desc <chr>, Scripts <dbl>, Cost <dbl>
```

# Working with `tsibble` objects

We can use the `filter()` function to select rows.

```
PBS %>%
  filter(ATC2=="A10")
```

```
## # A tsibble: 816 x 9 [1M]
## # Key:       Concession, Type, ATC1, ATC2 [4]
##         Month Concession Type  ATC1  ATC1_desc ATC2
##         <mth> <chr>      <chr> <chr> <chr>     <chr>
##  1   1991 Jul Concessio~ Co-p~ A     Alimenta~ A10
##  2   1991 Aug Concessio~ Co-p~ A     Alimenta~ A10
##  3   1991 Sep Concessio~ Co-p~ A     Alimenta~ A10
##  4   1991 Oct Concessio~ Co-p~ A     Alimenta~ A10
##  5   1991 Nov Concessio~ Co-p~ A     Alimenta~ A10
##  6   1991 Dec Concessio~ Co-p~ A     Alimenta~ A10
##  7   1992 Jan Concessio~ Co-p~ A     Alimenta~ A10
##  8   1992 Feb Concessio~ Co-p~ A     Alimenta~ A10
##  9   1992 Mar Concessio~ Co-p~ A     Alimenta~ A10
## 10   1992 Apr Concessio~ Co-p~ A     Alimenta~ A10
## #     with 806 more rows, and 3 more variables:
```

# Working with `tsibble` objects

We can use the `select()` function to select columns.

```r
PBS %>%
  filter(ATC2=="A10") %>%
  select(Cost)
```

```
Selecting index: "Month"
Error: The result is not a valid tsibble.
Do you need as_tibble() to work with data frame?
```

# Working with `tsibble` objects

We can use the `select()` function to select columns.

```
PBS %>%
  filter(ATC2=="A10") %>%
  select(Month, Concession, Type, Cost)
```

```
## # A tsibble: 816 x 4 [1M]
## # Key:       Concession, Type [4]
##       Month Concession   Type            Cost
##       <mth> <chr>        <chr>          <dbl>
##  1 1991 Jul Concessional Co-payments 2092878
##  2 1991 Aug Concessional Co-payments 1795733
##  3 1991 Sep Concessional Co-payments 1777231
##  4 1991 Oct Concessional Co-payments 1848507
##  5 1991 Nov Concessional Co-payments 1686458
##  6 1991 Dec Concessional Co-payments 1843079
##  7 1992 Jan Concessional Co-payments 1564702
##  8 1992 Feb Concessional Co-payments 1732508
##  9 1992 Mar Concessional Co-payments 2046102
## 10 1992 Apr Concessional Co-payments 2225977
```

# Working with tsibble objects

We can use the summarise() function to summarise over keys.

```
PBS %>%
  filter(ATC2=="A10") %>%
  select(Month, Concession, Type, Cost) %>%
  summarise(TotalC = sum(Cost))
```

```
## # A tsibble: 204 x 2 [1M]
##        Month  TotalC
##        <mth>   <dbl>
##   1 1991 Jul 3526591
##   2 1991 Aug 3180891
##   3 1991 Sep 3252221
##   4 1991 Oct 3611003
##   5 1991 Nov 3565869
##   6 1991 Dec 4306371
##   7 1992 Jan 5088335
##   8 1992 Feb 2814520
##   9 1992 Mar 2985811
##  10 1992 Apr 3204780
```

# Working with `tsibble` objects

We can use the `mutate()` function to create new variables.

```
PBS %>%
  filter(ATC2=="A10") %>%
  select(Month, Concession, Type, Cost) %>%
  summarise(TotalC = sum(Cost)) %>%
  mutate(Cost = TotalC/1e6)
```

```
## # A tsibble: 204 x 3 [1M]
##       Month  TotalC  Cost
##       <mth>   <dbl> <dbl>
##  1 1991 Jul 3526591  3.53
##  2 1991 Aug 3180891  3.18
##  3 1991 Sep 3252221  3.25
##  4 1991 Oct 3611003  3.61
##  5 1991 Nov 3565869  3.57
##  6 1991 Dec 4306371  4.31
##  7 1992 Jan 5088335  5.09
##  8 1992 Feb 2814520  2.81
##  9 1992 Mar 2985811  2.99
```

# Working with `tsibble` objects

We can use the `mutate()` function to create new variables.

```
PBS %>%
  filter(ATC2=="A10") %>%
  select(Month, Concession, Type, Cost) %>%
  summarise(TotalC = sum(Cost)) %>%
  mutate(Cost = TotalC/1e6) -> a10
```

```
## # A tsibble: 204 x 3 [1M]
##        Month  TotalC  Cost
##        <mth>   <dbl> <dbl>
##   1 1991 Jul 3526591  3.53
##   2 1991 Aug 3180891  3.18
##   3 1991 Sep 3252221  3.25
##   4 1991 Oct 3611003  3.61
##   5 1991 Nov 3565869  3.57
##   6 1991 Dec 4306371  4.31
##   7 1992 Jan 5088335  5.09
##   8 1992 Feb 2814520  2.81
##   9 1992 Mar 2985811  2.99
```

# Outline

1. Time series data and tsibbles
2. Working with `tsibble` objects
3. Create a tsibble from a csv
4. Time plots

# Create a tsibble from a csv

| date | state | gender | legal | indigenous | count |
|---|---|---|---|---|---:|
| 2005-03-01 | ACT | Female | Remanded | ATSI | 0 |
| 2005-03-01 | ACT | Female | Remanded | Other | 2 |
| 2005-03-01 | ACT | Female | Sentenced | ATSI | 0 |
| 2005-03-01 | ACT | Female | Sentenced | Other | 0 |
| 2005-03-01 | ACT | Male | Remanded | ATSI | 7 |
| 2005-03-01 | ACT | Male | Remanded | Other | 58 |
| 2005-03-01 | ACT | Male | Sentenced | ATSI | 0 |
| 2005-03-01 | ACT | Male | Sentenced | Other | 0 |
| 2005-03-01 | NSW | Female | Remanded | ATSI | 51 |
| 2005-03-01 | NSW | Female | Remanded | Other | 131 |
| 2005-03-01 | NSW | Female | Sentenced | ATSI | 0 |
| 2005-03-01 | NSW | Female | Sentenced | Other | 0 |
| 2005-03-01 | NSW | Male | Remanded | ATSI | 255 |

# Read a csv file and convert to a tsibble

```
prison <- readr::read_csv("prison_population.csv") %>%
  mutate(Quarter = yearquarter(date)) %>%
  select(-date) %>%
  as_tsibble(index=Quarter,
    key=c(state, gender, legal, indigenous))
```

```
prison
```

```
## # A tsibble: 3,072 x 6 [1Q]
## # Key:       state, gender, legal, indigenous [64]
##    state gender legal    indigenous count Quarter
##    <chr> <chr>  <chr>    <chr>      <dbl>   <qtr>
##  1 ACT   Female Remanded ATSI           0 2005 Q1
##  2 ACT   Female Remanded ATSI           1 2005 Q2
##  3 ACT   Female Remanded ATSI           0 2005 Q3
##  4 ACT   Female Remanded ATSI           0 2005 Q4
##  5 ACT   Female Remanded ATSI           1 2006 Q1
```

# Outline

1. Time series data and tsibbles
2. Working with `tsibble` objects
3. Create a tsibble from a csv
4. Time plots

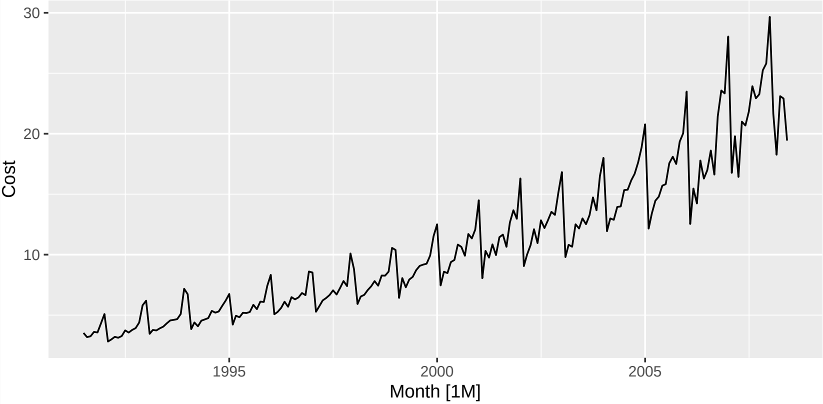
# Time plots

```
a10
```

```
## # A tsibble: 204 x 3 [1M]
##        Month  TotalC  Cost
##        <mth>   <dbl> <dbl>
##  1 1991 Jul 3526591  3.53
##  2 1991 Aug 3180891  3.18
##  3 1991 Sep 3252221  3.25
##  4 1991 Oct 3611003  3.61
##  5 1991 Nov 3565869  3.57
##  6 1991 Dec 4306371  4.31
##  7 1992 Jan 5088335  5.09
```

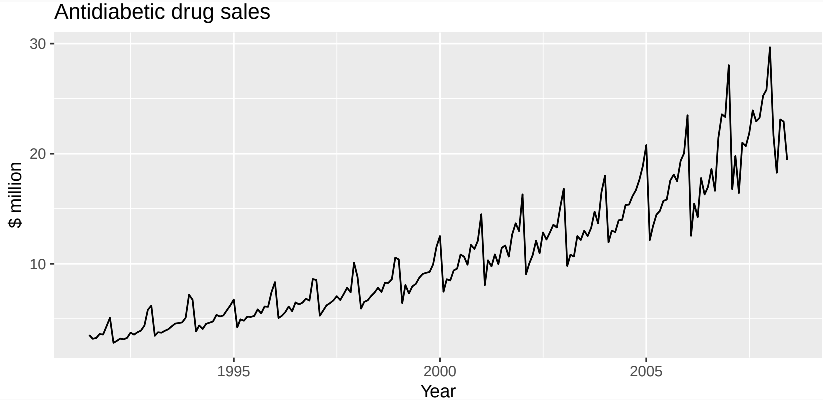
# Time plots

```
a10 %>% autoplot(Cost)
```



# Time plots

```r
a10 %>% autoplot(Cost) +
  ylab("$ million") + xlab("Year") +
  ggtitle("Antidiabetic drug sales")
```



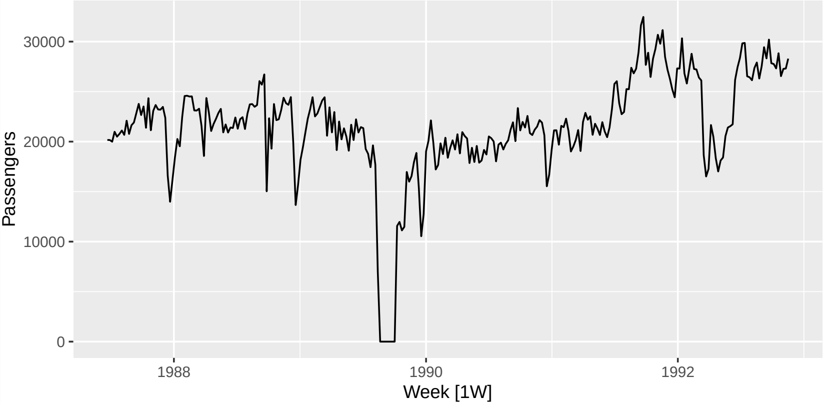
# Time plots

```
ansett
```

```
## # A tsibble: 7,407 x 4 [1W]
## # Key:       Airports, Class [30]
##         Week Airports Class    Passengers
##       <week> <chr>    <chr>         <dbl>
##   1 1989 W28 ADL-PER  Business        193
##   2 1989 W29 ADL-PER  Business        254
##   3 1989 W30 ADL-PER  Business        185
##   4 1989 W31 ADL-PER  Business        254
##   5 1989 W32 ADL-PER  Business        191
##   6 1989 W33 ADL-PER  Business        136
##   7 1989 W34 ADL-PER  Business          0
##   8 1989 W35 ADL-PER  Business          0
```

# Time plots

```
ansett %>%
  filter(Airports=="MEL-SYD", Class=="Economy") %>%
  autoplot(Passengers)
```

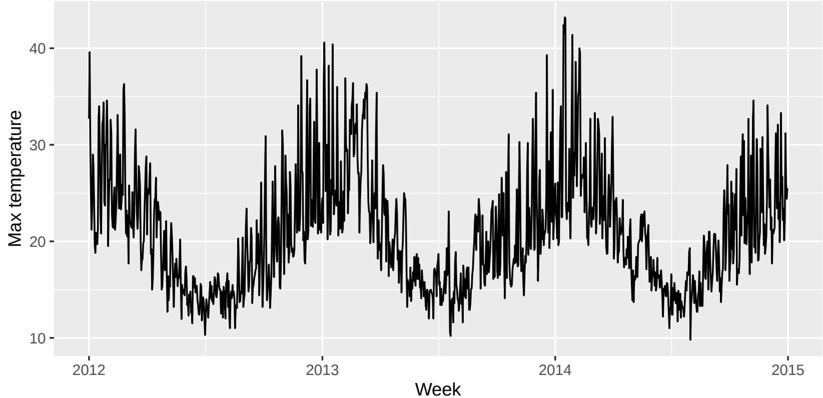


# Your turn

- Create plots of the following time series: Bricks from `aus_production`, Lynx from `pelt`, Google from `gafa_stock`
- Use `help()` to find out about the data in each series.
- For the last plot, modify the axis labels and title.

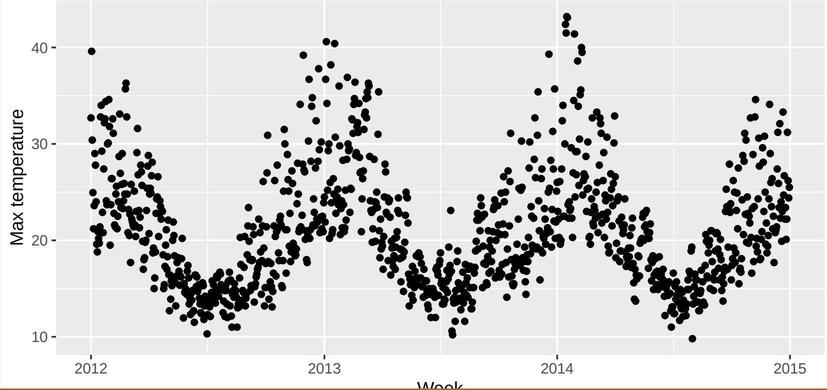
# Are time plots best?

```
maxtemp %>%
  autoplot(Temperature) +
  xlab("Week") + ylab("Max temperature")
```

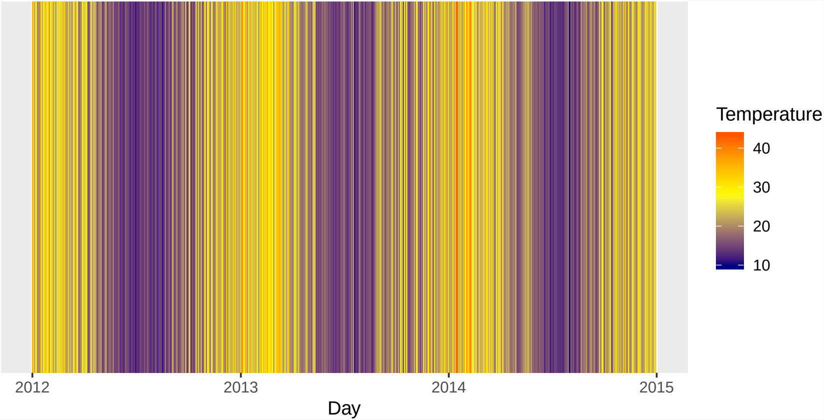


# Are time plots best?

```
maxtemp %>%
  ggplot(aes(x = Day, y = Temperature)) +
  geom_point() +
  xlab("Week") + ylab("Max temperature")
```



# Are time plots best?



# Are time plots best?