MONASH University

ETC5521: Exploratory Data Analysis

Exploring data having a space and time context

Lecturer: *Di Cook*

ETC5521.Clayton-x@monash.edu

Week 9 - Session 1

Time series analysis is what you do after all the interesting stuff has been done!

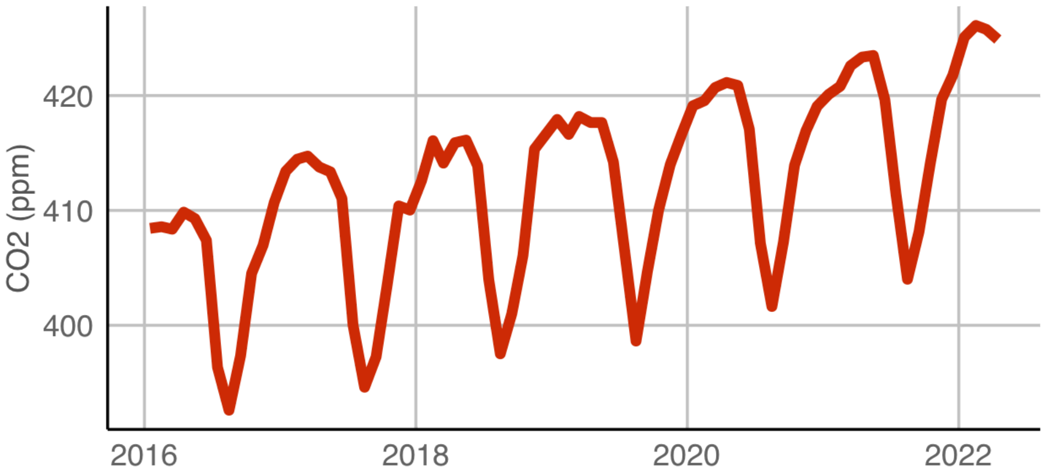
Heike Hofmann, 2005

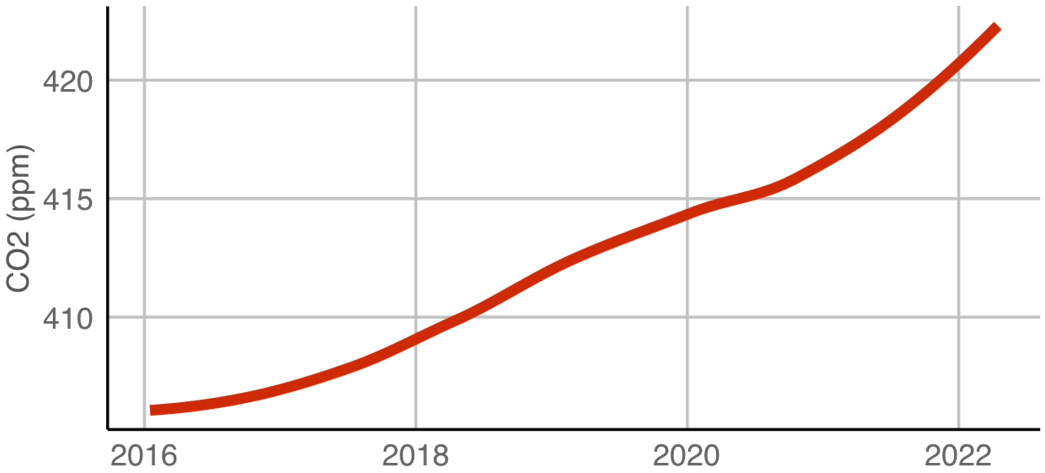
What is temporal data?

🕰️ Melbourne pedestrian sensor data

Sensor	Date_Time	Date	Time	Count
Birrarung Marr	*2015-02-14 22:00:00*	2015-02-14	22	7081
Birrarung Marr	*2015-02-21 21:00:00*	2015-02-21	21	8363
Birrarung Marr	*2015-02-21 22:00:00*	2015-02-21	22	9658
Birrarung Marr	*2015-02-21 23:00:00*	2015-02-21	23	10121
Birrarung Marr	*2015-02-22 00:00:00*	2015-02-22	0	8441
Birrarung Marr	*2015-03-07 20:00:00*	2015-03-07	20	7144
Birrarung Marr	*2015-03-07 21:00:00*	2015-03-07	21	7238
Birrarung Marr	*2015-03-08 13:00:00*	2015-03-08	13	7092
Birrarung Marr	*2015-03-08 14:00:00*	2015-03-08	14	7031
Birrarung Marr	*2015-03-08 15:00:00*	2015-03-08	15	6951
Birrarung Marr	*2015-03-08 16:00:00*	2015-03-08	16	7167

What is temporal data?



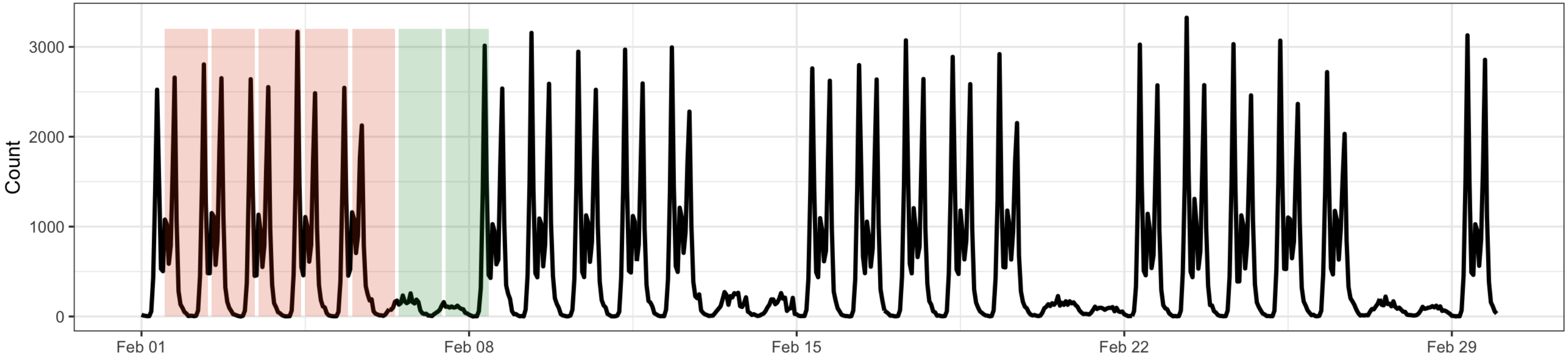


- Temporal data has date/time/ordering index variable, call it time.
- A time variable has special structure:
  - it can have *cyclical* patterns, eg seasonality (summer, winter), an over in cricket
  - the cyclical patterns can be *nested*, eg postcode within state, over within innings
- Measurements are also NOT independent - yesterday may influence today.
- It still likely has non-cyclical patterns, trends and associations with other variables, eg temperature increasing over time, over is bowled by Elise Perry or Sophie Molineaux

Case study 1 Melbourne pedestrian traffic

learn R

Pedestrian counts at Southern Cross in Feb 2016

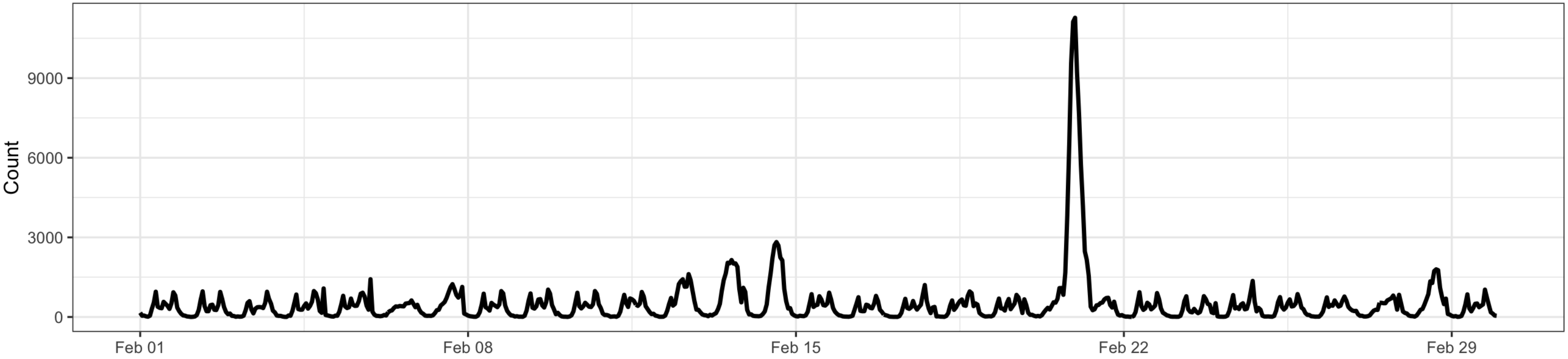


This is interesting!

Case study 1 Melbourne pedestrian traffic

🖼️ learn R

Pedestrian counts at Birrarung Marr in Feb 2016

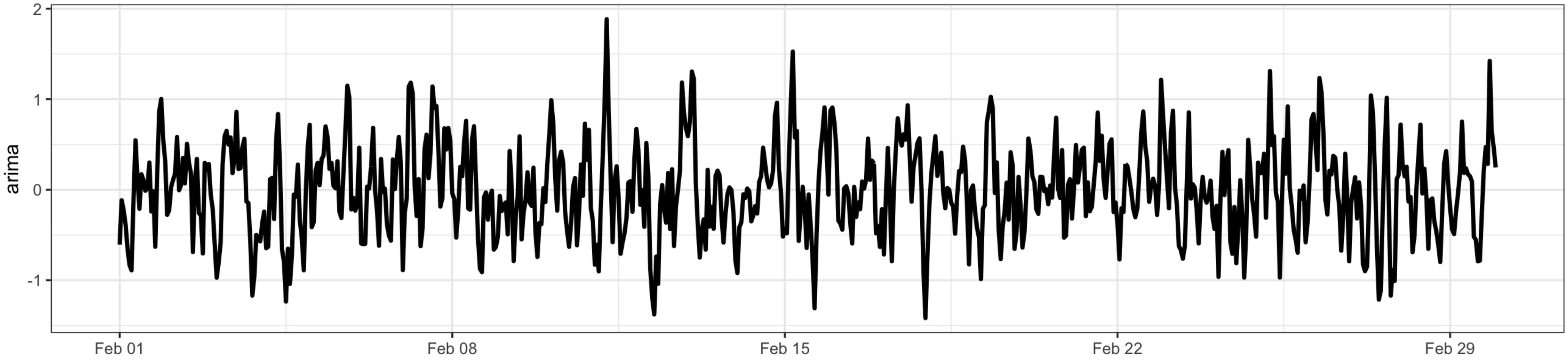


This is interesting!

Case study 1 Melbourne pedestrian traffic

learn R

What does Heike mean?



This is a little bit boring! It is important for fitting a model that accounts for dependencies between measurements, though.

Exploratory analysis of temporal data is interested in extracting the trend and general patterns.

What is exploratory analysis of time series?

Exploratory analysis of time series investigates trends, patterns, cyclical, nested cyclical, temporal outliers, and temporal dependence.

For the pedestrian sensor data this is:

- work day vs holiday pattern
- daily patterns
- weather and season related changes
- event related patterns

`tsibble`: temporal object in R

The tsibble package provides a data infrastructure for tidy temporal data with wrangling tools. Adapting the tidy data principles, tsibble is a data- and model-oriented object. In tsibble:

- Index is a variable with inherent ordering from past to present.
- Key is a set of variables that define observational units over time.
- Each observation should be uniquely identified by index and key.
- Each observational unit should be measured at a common interval, if regularly spaced.

Regular vs irregular

The Melbourne pedestrian sensor data has a regular period. Counts are provided for every hour, at numerous locations.

```
## # A tsibble: 66,037 x 5 [1h] <Australia/Melbourne>
## # Key:       Sensor [4]
##    Sensor         Date_Time           Date        Time
##    <chr>          <dttm>              <date>     <int>
##  1 Birrarung Marr 2015-01-01 00:00:00 2015-01-01     0
##  2 Birrarung Marr 2015-01-01 01:00:00 2015-01-01     1
##  3 Birrarung Marr 2015-01-01 02:00:00 2015-01-01     2
##  4 Birrarung Marr 2015-01-01 03:00:00 2015-01-01     3
##  5 Birrarung Marr 2015-01-01 04:00:00 2015-01-01     4
##  6 Birrarung Marr 2015-01-01 05:00:00 2015-01-01     5
##  7 Birrarung Marr 2015-01-01 06:00:00 2015-01-01     6
##  8 Birrarung Marr 2015-01-01 07:00:00 2015-01-01     7
```

In contrast, the US flights data, below, is irregular.

```
## # A tsibble: 336,776 x 20 [!] <UTC>
## # Key:       origin, dest, carrier, tailnum [52,807]
##     year month   day dep_time sched_dep_time dep_delay
##    <int> <int> <int>    <int>          <int>     <dbl>
##  1  2013     1    30     2224           2000       144
##  2  2013     2    17     2012           2010         2
##  3  2013     2    26     2356           2000       236
##  4  2013     3    13     1958           2005        -7
##  5  2013     5    16     2214           2000       134
##  6  2013     5    30     2045           2000        45
##  7  2013     9    11     2254           2159        55
##  8  2013     9    12       NA           2159        NA
```

question discussion

Is pedestrian traffic regular, really?

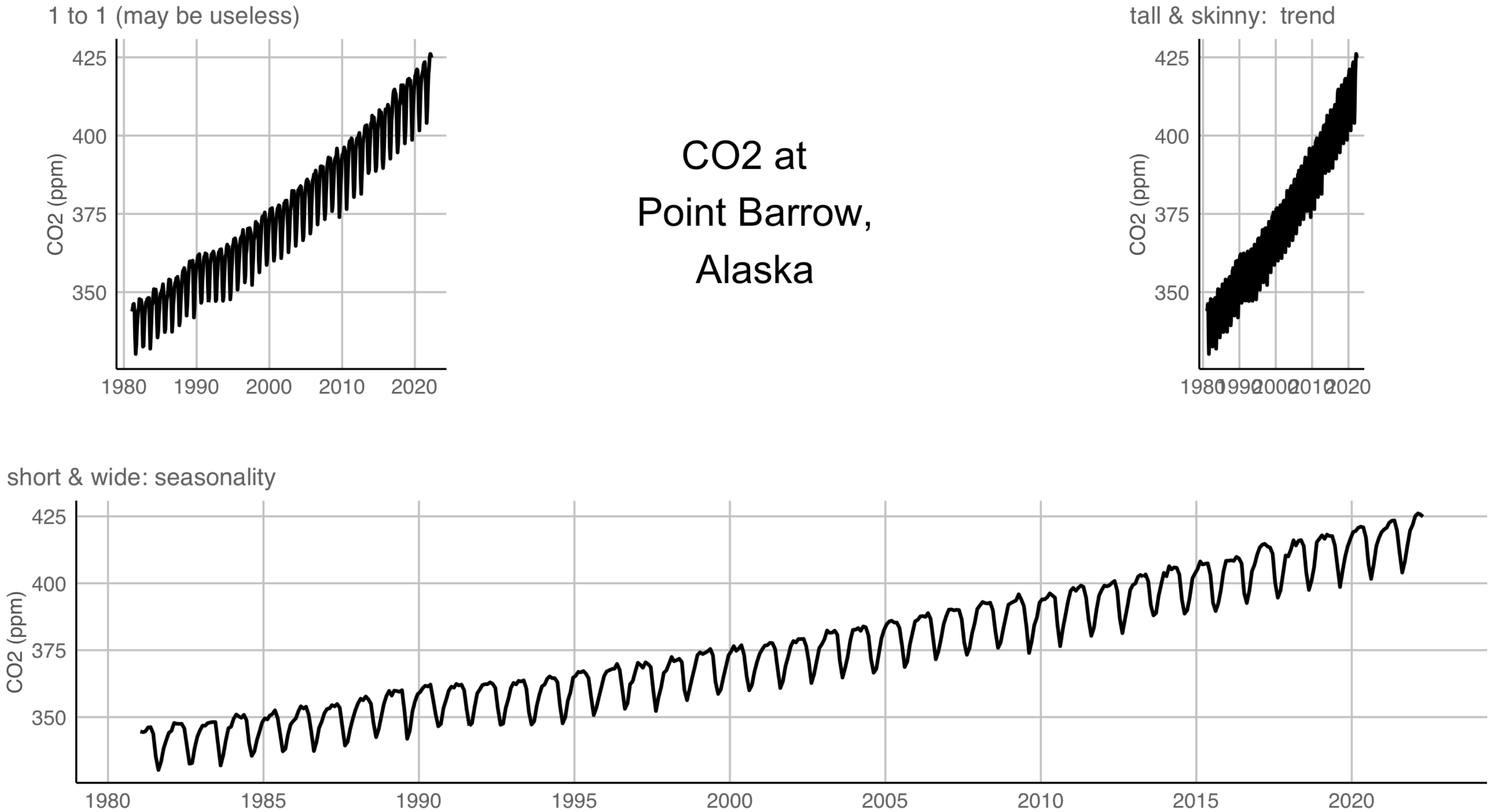
Let's make some plots

Plotting temporal data

- lines: connecting sequential time points indicates the temporal dependence is important
- aspect ratio: wide or tall? Cleveland, McGill, McGill (1988) argue the average line slope in a line chart should be 45 degrees, which is called banking to 45 degrees. But this is refuted in Talbot, Gerth, Hanrahan (2012) that the conclusion was based on a flawed study. Nevertheless, aspect ratio is an inescapable skill for designing effective plots. For time series, typically a wide aspect ratio is good.
- conventions:
  - time on the horizontal axis,
  - ordering of elements like week day, month.

Aspect ratio

learn R



Case study 2 nycflights13 Part 1/7

```r
library(nycflights13)
```

What is a useful time element to use, in order to study traffic over time?

Hour, 15 minutes, day, month?

Possibly, all of these.

Let's start with hourly.

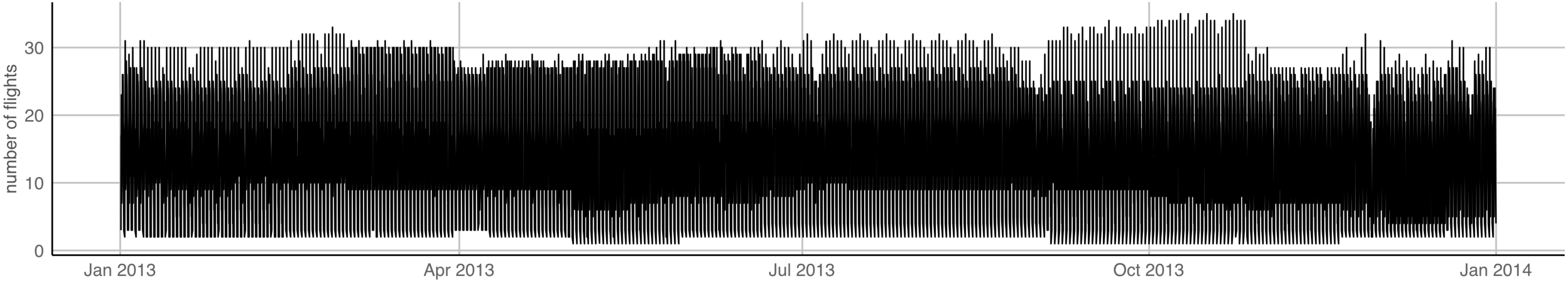
```r
flights_hourly <- flights %>%
  group_by(time_hour, origin) %>%
  summarise(count = n(),
    dep_delay = mean(dep_delay,
                     na.rm = TRUE)) %>%
  ungroup() %>%
  as_tsibble(index = time_hour,
             key = origin)
flights_hourly
```

```
## # A tsibble: 19,486 x 4 [1h] <America/New_York>
## # Key:       origin [3]
##    time_hour           origin count dep_delay
##    <dttm>              <chr>  <int>     <dbl>
##  1 2013-01-01 05:00:00 EWR        2     -1
```

Case study 2 nycflights13 Part 2/7

IDA: Pick one airport, and examine the hourly number of flights.

```
flights_hourly %>%
  filter(origin == "JFK") %>%
  ggplot(aes(x=time_hour, y=count)) +
  geom_line() +
  xlab("") + ylab("number of flights")
```

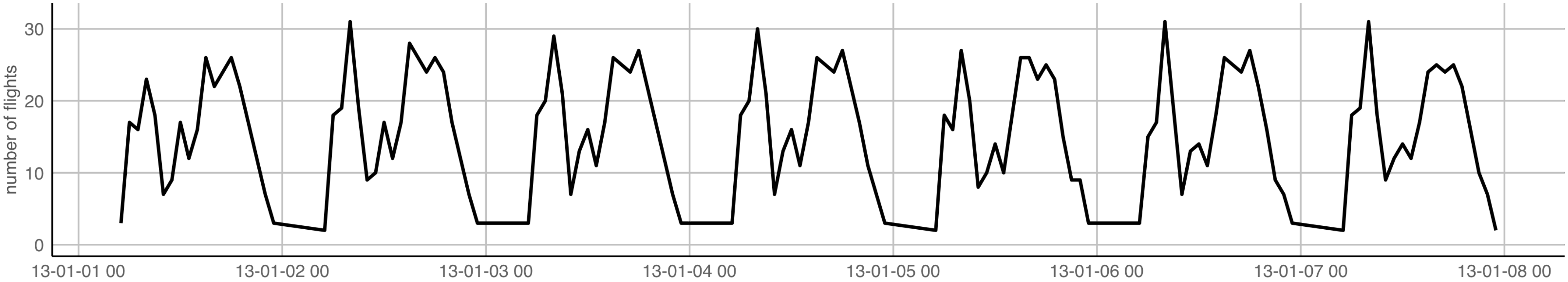


No, that's too much information, too much time. There's no overall trend. Not an interesting plot.

Case study 2 nycflights13 Part 3/7

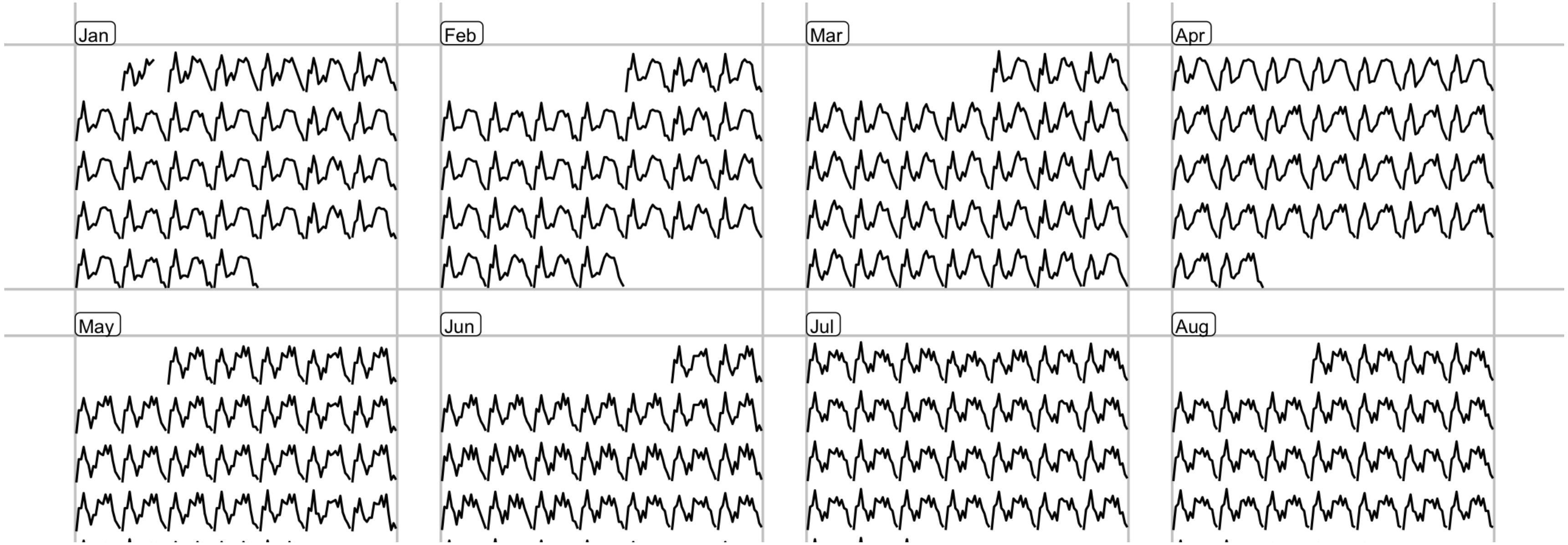
IDA: Reduce the time frame to check for periodicities

```
flights_hourly %>%
  filter(origin == "JFK",
         time_hour < ymd("2013-01-08")) %>%
  ggplot(aes(x=time_hour, y=count)) +
    geom_line(size=1.1) +
    scale_x_datetime("",
                     date_breaks = "1 day",
                     date_labels = "%y-%m-%d %H",
                     date_minor_breaks = "6 hours") +
```



Case study 2 nycflights13 Part 4/7

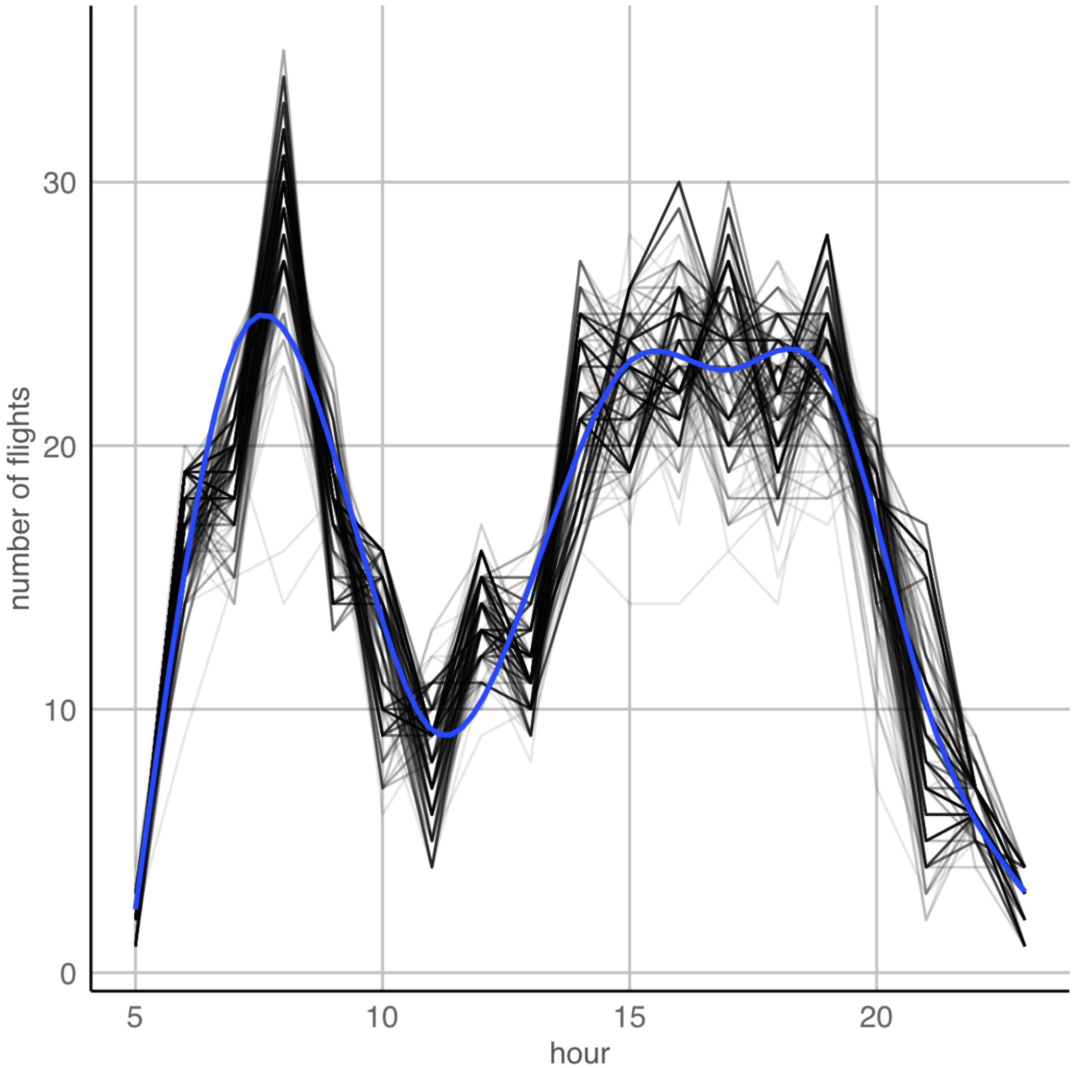
learn R



Case study 2 nycflights13 Part 5/7

```
flights_hourly %>%
  filter(origin == "JFK") %>%
  mutate(month = month(time_hour),
         hour = hour(time_hour),
         date = as.Date(time_hour)) %>%
  ggplot(aes(x=hour, y=count)) +
    geom_line(aes(group=date),
              alpha = 0.1) +
    geom_smooth(se = FALSE) +
    xlab("hour") +
  ylab("number of flights")
```

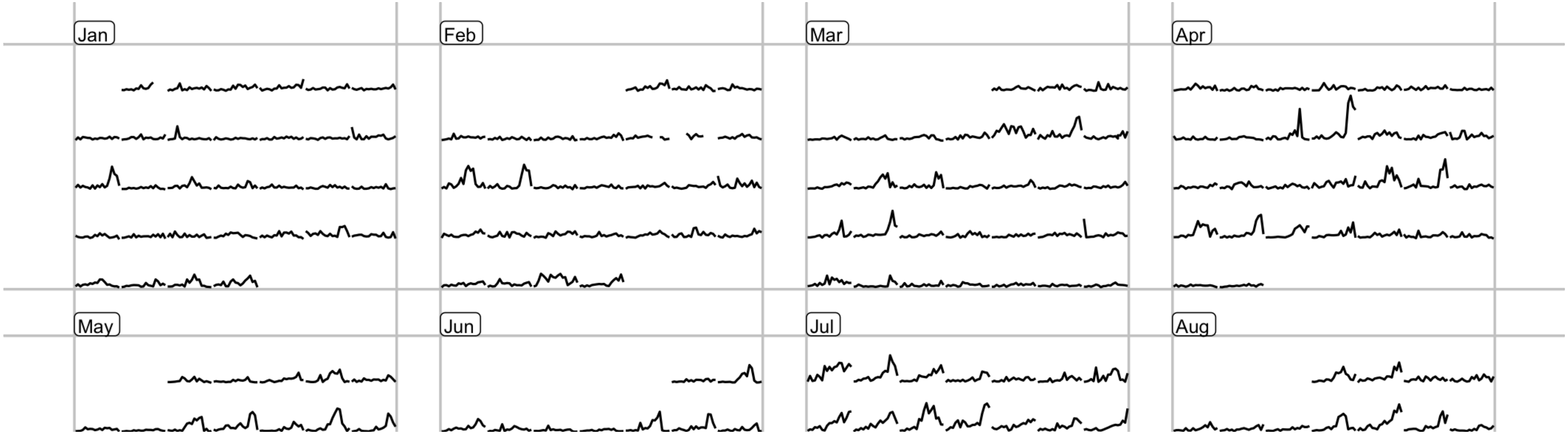
This data has a very regular. The volume per hour is very similar from one day to the next. Why is it so regular?



Examine departure delays

Case study 2 nycflights13 Part 6/7

learn R

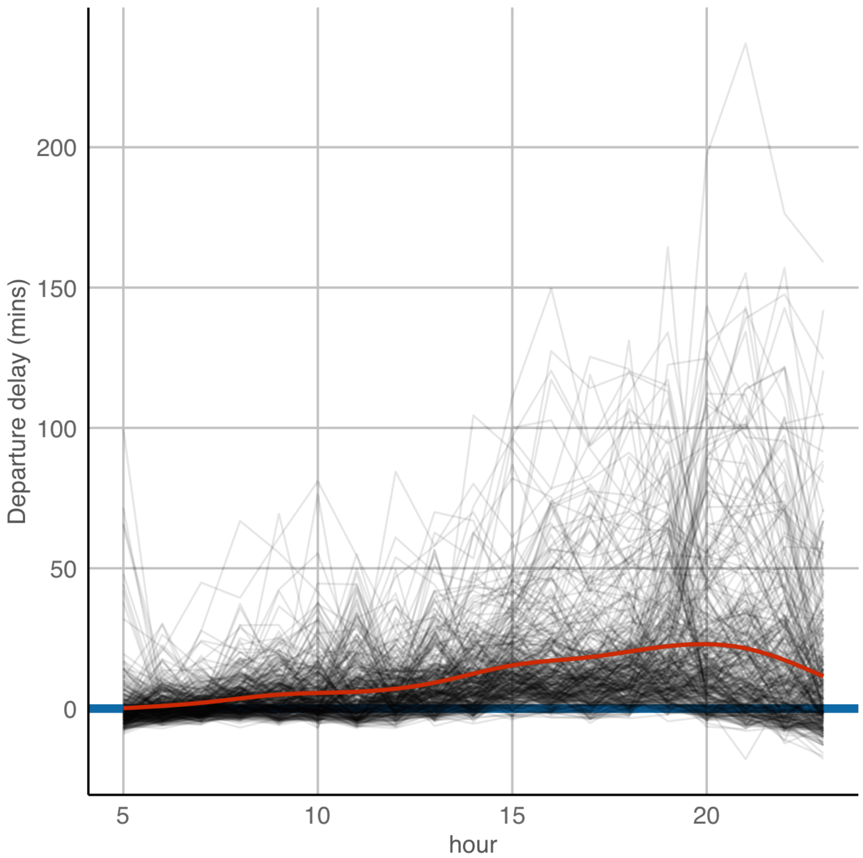


Case study 2 nycflights13 Part 7/7

Days in comparison to each other.

```
flights_hourly %>%
  filter(origin == "JFK") %>%
  mutate(month = month(time_hour),
         hour = hour(time_hour),
         date = as.Date(time_hour)) %>%
  ggplot(aes(x=hour, y=dep_delay)) +
    geom_hline(yintercept=0,
               colour="#027EB6", size=2) +
    geom_line(aes(group=date), alpha = 0.1) +
    geom_smooth(se=FALSE, colour="#D93F00") +
    xlab("hour") + ylab("Departure delay (mins)")
```

- A lot of day to day variability - modeling and forecasting delays will need other information like weather.
- Delays worsen, on average, later in the day.
- Interestingly, a lot of flights depart a few minutes early, especially later in the day.



Summary: Melting time

```
##  [1] "year"           "month"          "day"            "dep_time"      
##  [5] "sched_dep_time" "dep_delay"      "arr_time"       "sched_arr_time"
##  [9] "arr_delay"      "carrier"        "flight"         "tailnum"       
## [13] "origin"         "dest"           "air_time"       "distance"      
## [17] "hour"           "minute"         "time_hour"
```

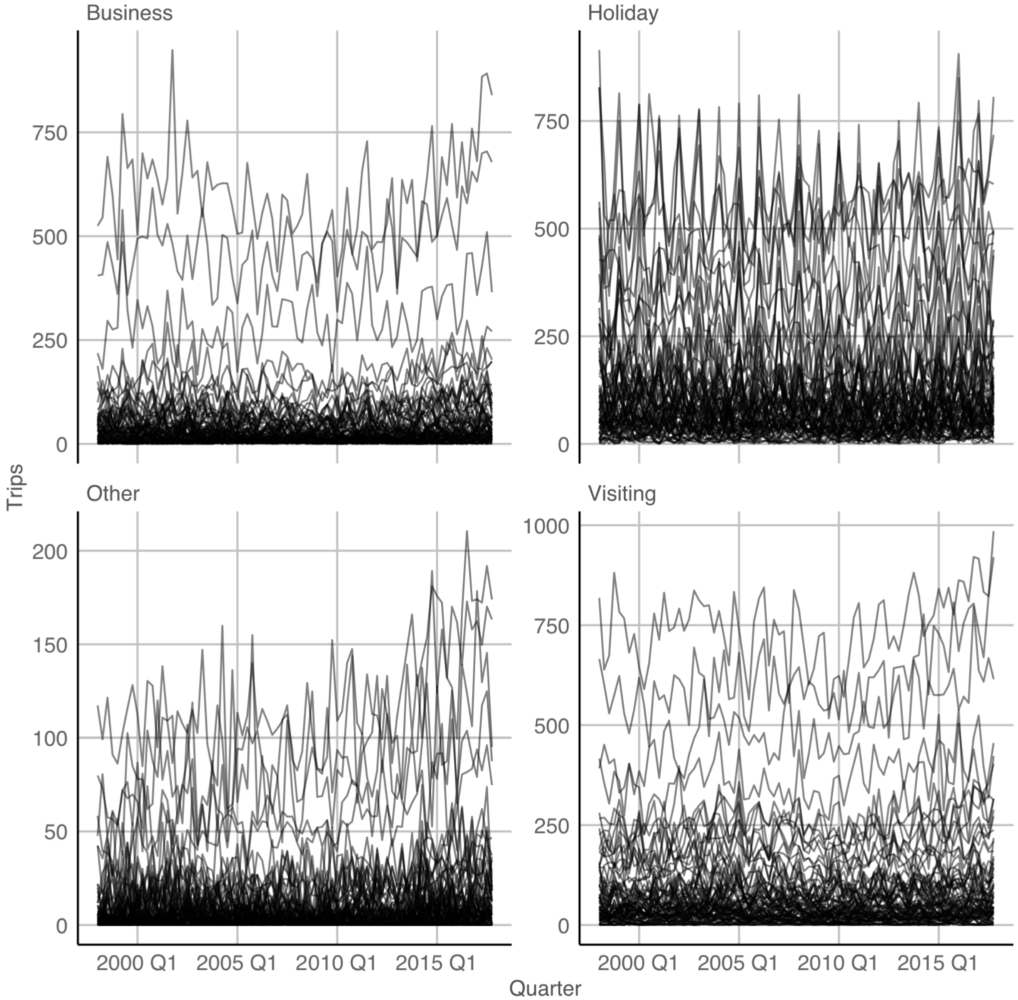
- The structure of the `flights` table is very handy. Date-time has already been melted into: `year`, `month`, `day`, `hour`, `minute`.
- There are also several possible key variables: `origin`, `carrier`, `tailnum`.

Why isn't `dest` considered a key variable? Why not have `air_time` as a key variable?

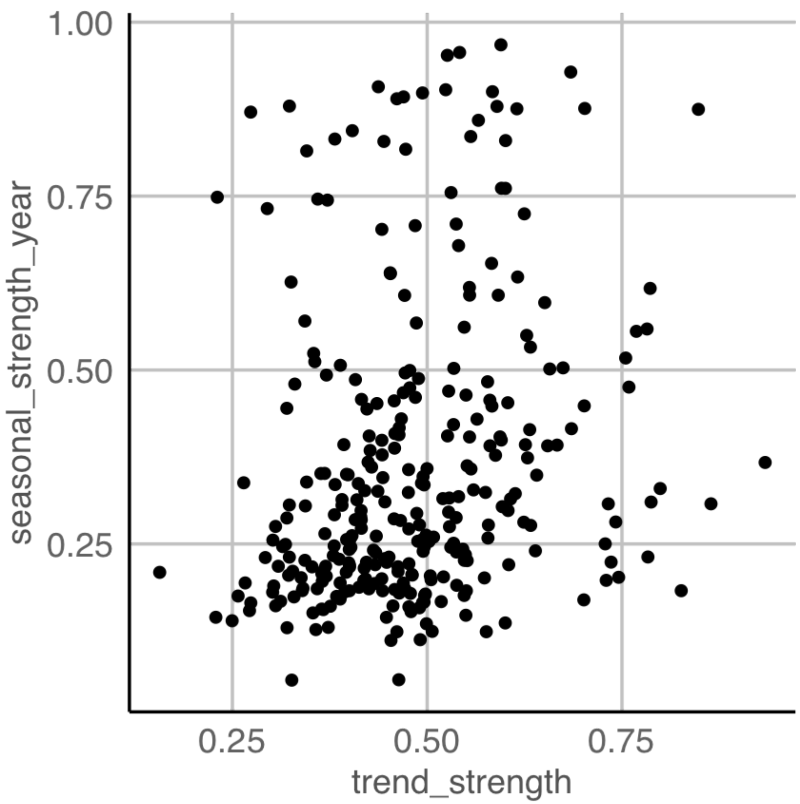
- Aggregate by temporal components, in different ways to explore different patterns of variables in relation to elements of time.

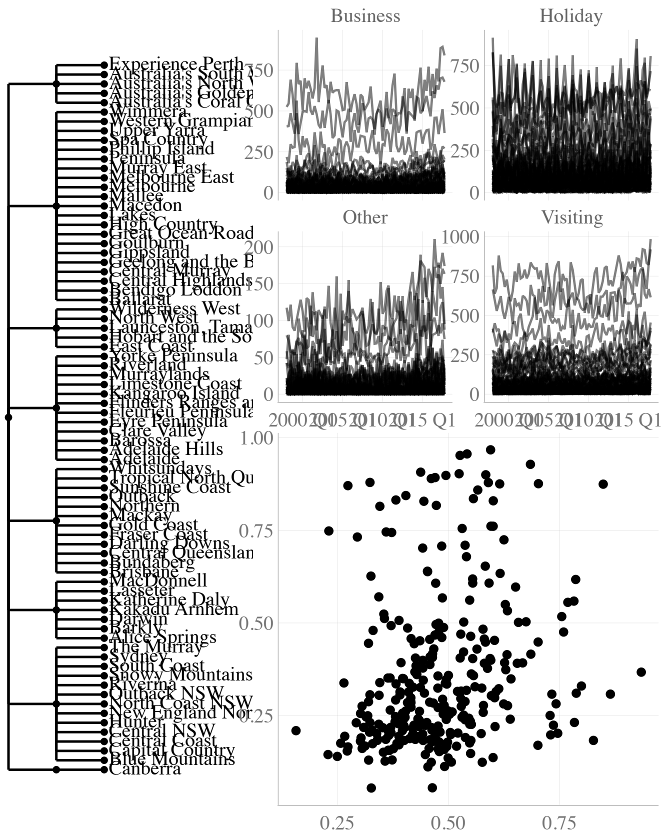
Interactive exploration with tsibbletalk

R



Remember scagnostics? These are examples of tignostics, time series diagnostics.






```
library(plotly)
subplot(p0,
  subplot(
    ggplotly(p1, tooltip = "Region", width = 700),
    ggplotly(p2, tooltip = "Region", width = 600),
    nrows = 2),
  widths = c(.4, .6)) %>%
  highlight(dynamic = FALSE)
```

Live demos

Interactive wrapping to explore periodicities

🔧 Your turn, cut and paste the code into your R console. Drag the scroll bar to wrap the series on itself.

```
p <- fill_gaps(pedestrian) %>%
  filter_index(~ "2015") %>%
  ggplot(aes(x = Date_Time, y = Count, colour = Sensor)) +
  geom_line(size = .2) +
  facet_wrap(~ Sensor, scales = "free_y") +
  theme(legend.position = "none")

library(shiny)
ui <- fluidPage(tsibbleWrapUI("tswrap"))
server <- function(input, output, session) {
  tsibbleWrapServer("tswrap", p, period = "1 day")
}
shinyApp(ui, server)
```

A step back in time

Some series that look periodic, are not. Try to patch the peaks

Annual numbers of lynx trappings for 1821–1934 in Canada. Almost 10 year cycle.

```
lynx_tsb <- as_tsibble(lynx) %>%
  rename(count = value)
pl <- ggplot(lynx_tsb,
  aes(x = index, y = count)) +
  geom_line(size = .2)

ui <- fluidPage(
  tsibbleWrapUI("tswrap"))
server <- function(input, output,
                   session) {
  tsibbleWrapServer("tswrap", pl,
       period = "10 year")
}
shinyApp(ui, server)
```

Monthly mean relative sunspot numbers from 1749 to 1983. Almost 10 year cycle.

```
sunspots_tsb <- as_tsibble(sunspots) %>%
  rename(count = value)
pl <- ggplot(sunspots_tsb,
  aes(x = index, y = count)) +
  geom_line(size = .2)

ui <- fluidPage(
  tsibbleWrapUI("tswrap"))
server <- function(input, output,
                   session) {
  tsibbleWrapServer("tswrap", pl,
          period = "10 year")
}
shinyApp(ui, server)
```

Resources and Acknowledgement

- The temporal data object tsibble
- Wang & Cook, Conversations in Time: Interactive Visualization to Explore Structured Temporal Data, The R Journal, 2020
- Data coding using `tidyverse` suite of R packages
- Slides constructed with `xaringan`, remark.js, `knitr`, and R Markdown.
- In Semester 3's ETC5550 expect to learn more about regular time series, which will include some exploration and some modeling

Lecturer: *Di Cook*

ETC5521.Clayton-x@monash.edu

Week 9 - Session 1