ETC5521: Exploratory Data Analysis

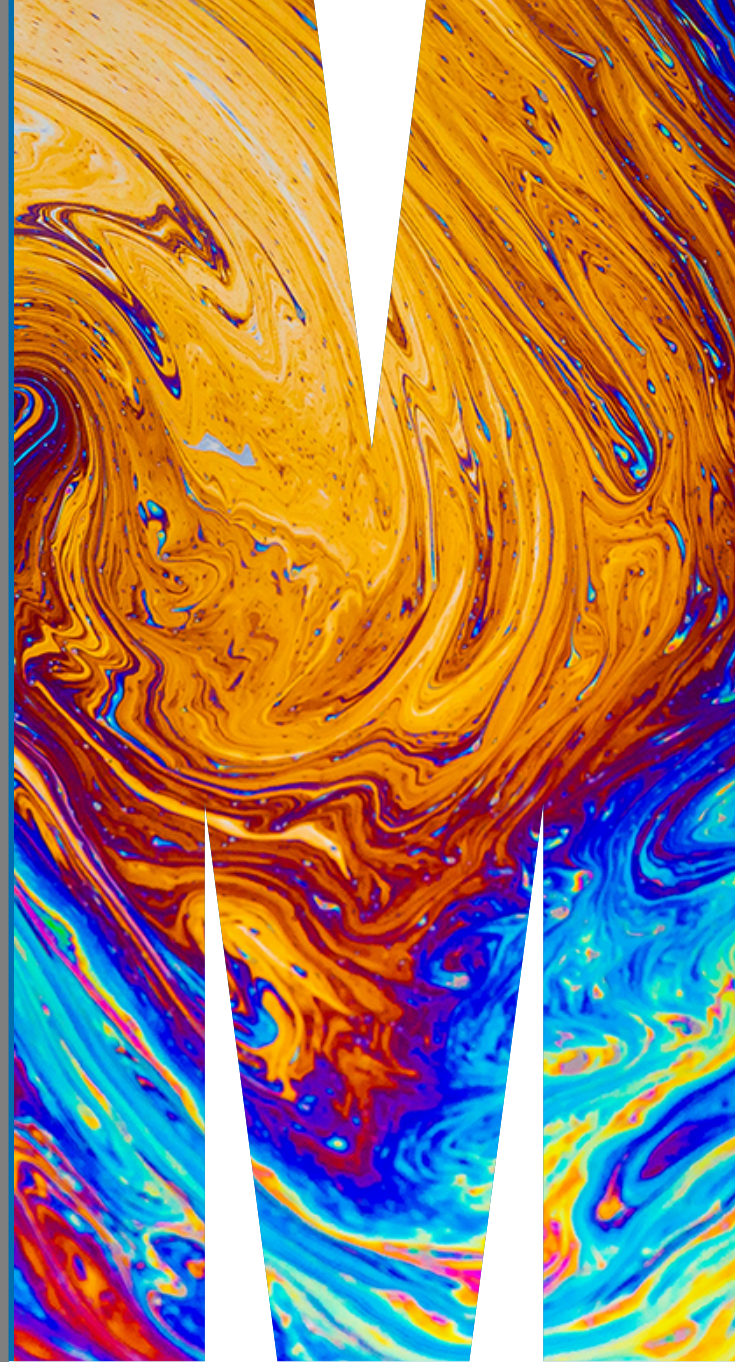
Exploring data having a space and time context

Lecturer: *Di Cook*

Department of Econometrics and Business Statistics

ETC5521.Clayton-x@monash.edu

Week 9 - Session 2



Outline

- missing values
- longitudinal data

Working with missings

Checking counting and filling missings in time

```
set.seed(328)
harvest <- tsibble(
  year = c(2010, 2011, 2013, 2011, 2012, 2013),
  fruit = rep(c("kiwi", "cherry"), each = 3),
  kilo = sample(1:10, size = 6),
  key = fruit, index = year
)
harvest
```

```
## # A tsibble: 6 x 3 [1Y]
## # Key:       fruit [2]
##    year fruit   kilo
##   <dbl> <chr>  <int>
## 1  2011 cherry     2
## 2  2012 cherry     7
## 3  2013 cherry     1
## 4  2010 kiwi       6
## 5  2011 kiwi       5
## 6  2013 kiwi       8
```

```
has_gaps(harvest, .full = TRUE)
```

```
## # A tibble: 2 x 2
##   fruit  .gaps
##   <chr>  <lgl>
## 1 cherry TRUE
## 2 kiwi   TRUE
```

Both levels of the key have missings.

Can you see the gaps in time?

Checking counting and filling missings in time

```
set.seed(328)
harvest <- tsibble(
  year = c(2010, 2011, 2013, 2011, 2012, 2013),
  fruit = rep(c("kiwi", "cherry"), each = 3),
  kilo = sample(1:10, size = 6),
  key = fruit, index = year
)
harvest
```

```
## # A tsibble: 6 x 3 [1Y]
## # Key:       fruit [2]
##    year fruit   kilo
##   <dbl> <chr>  <int>
## 1  2011 cherry     2
## 2  2012 cherry     7
## 3  2013 cherry     1
## 4  2010 kiwi       6
## 5  2011 kiwi       5
## 6  2013 kiwi       8
```

```
count_gaps(harvest,  .full=TRUE)
```

```
## # A tibble: 2 x 4
##   fruit  .from   .to    .n
##   <chr>  <dbl> <dbl> <int>
## 1 cherry  2010  2010     1
## 2 kiwi    2012  2012     1
```

One missing in each level, although it is a different year.

Notice how `tsibble` handles this summary so neatly.

Checking counting and filling missings in time

```
set.seed(328)
harvest <- tsibble(
  year = c(2010, 2011, 2013, 2011, 2012, 2013),
  fruit = rep(c("kiwi", "cherry"), each = 3),
  kilo = sample(1:10, size = 6),
  key = fruit, index = year
)
harvest
```

```
## # A tsibble: 6 x 3 [1Y]
## # Key:       fruit [2]
##    year fruit   kilo
##   <dbl> <chr>  <int>
## 1  2011 cherry     2
## 2  2012 cherry     7
## 3  2013 cherry     1
## 4  2010 kiwi       6
## 5  2011 kiwi       5
## 6  2013 kiwi       8
```

```
harvest <- fill_gaps(harvest, .full=TRUE)
harvest
```

```
## # A tsibble: 8 x 3 [1Y]
## # Key:       fruit [2]
##    year fruit   kilo
##   <dbl> <chr>  <int>
## 1  2010 cherry    NA
## 2  2011 cherry     2
## 3  2012 cherry     7
## 4  2013 cherry     1
## 5  2010 kiwi       6
## 6  2011 kiwi       5
## 7  2012 kiwi      NA
## 8  2013 kiwi       8
```

Make the implicit missing values explicit.

Checking counting and filling missings in time

```r
set.seed(328)
harvest <- tsibble(
  year = c(2010, 2011, 2013, 2011, 2012, 2013),
  fruit = rep(c("kiwi", "cherry"), each = 3),
  kilo = sample(1:10, size = 6),
  key = fruit, index = year
)
harvest
```

```r
harvest_nomiss <- harvest %>%
  group_by(fruit) %>%
  mutate(kilo = na_interpolation(kilo)) %>%
  ungroup()
harvest_nomiss
```

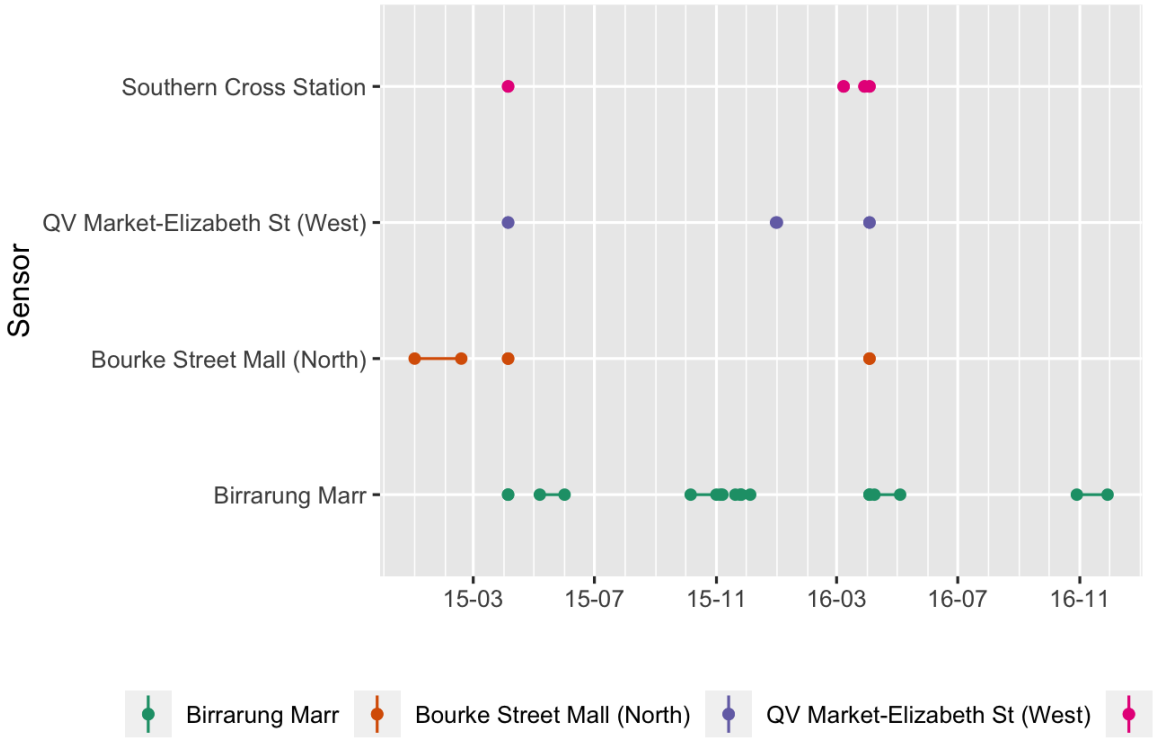
```
## # A tsibble: 8 x 3 [1Y]
## # Key:       fruit [2]
##    year fruit   kilo
##   <dbl> <chr>  <dbl>
## 1  2010 cherry   2
## 2  2011 cherry   2
## 3  2012 cherry   7
## 4  2013 cherry   1
## 5  2010 kiwi     6
## 6  2011 kiwi     5
## 7  2012 kiwi     6.5
## 8  2013 kiwi     8
```

Case study 3 Melbourne pedestrian traffic Part 1/5

```
has_gaps(pedestrian, .full = TRUE)
```

```
## # A tibble: 4 x 2
##   Sensor                       .gaps
##   <chr>                        <lgl>
## 1 Birrarung Marr               TRUE
## 2 Bourke Street Mall (North)   TRUE
## 3 QV Market-Elizabeth St (West) TRUE
## 4 Southern Cross Station       TRUE
```

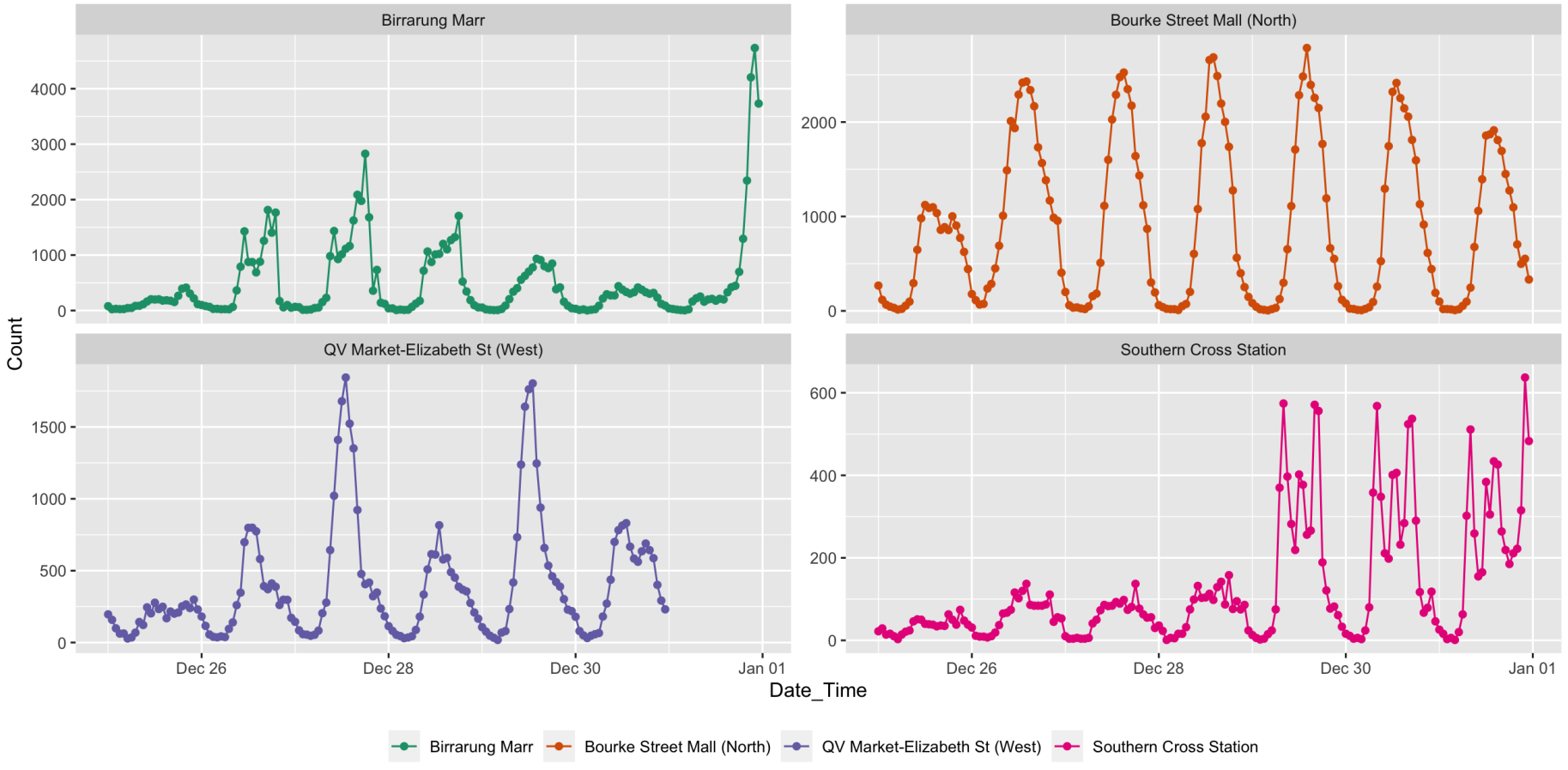
```
ped_gaps <- pedestrian %>%
  count_gaps(.full = TRUE)
```



What happens in April, for there to be missing on all sensors?

Case study 3 Melbourne pedestrian traffic Part 2/5

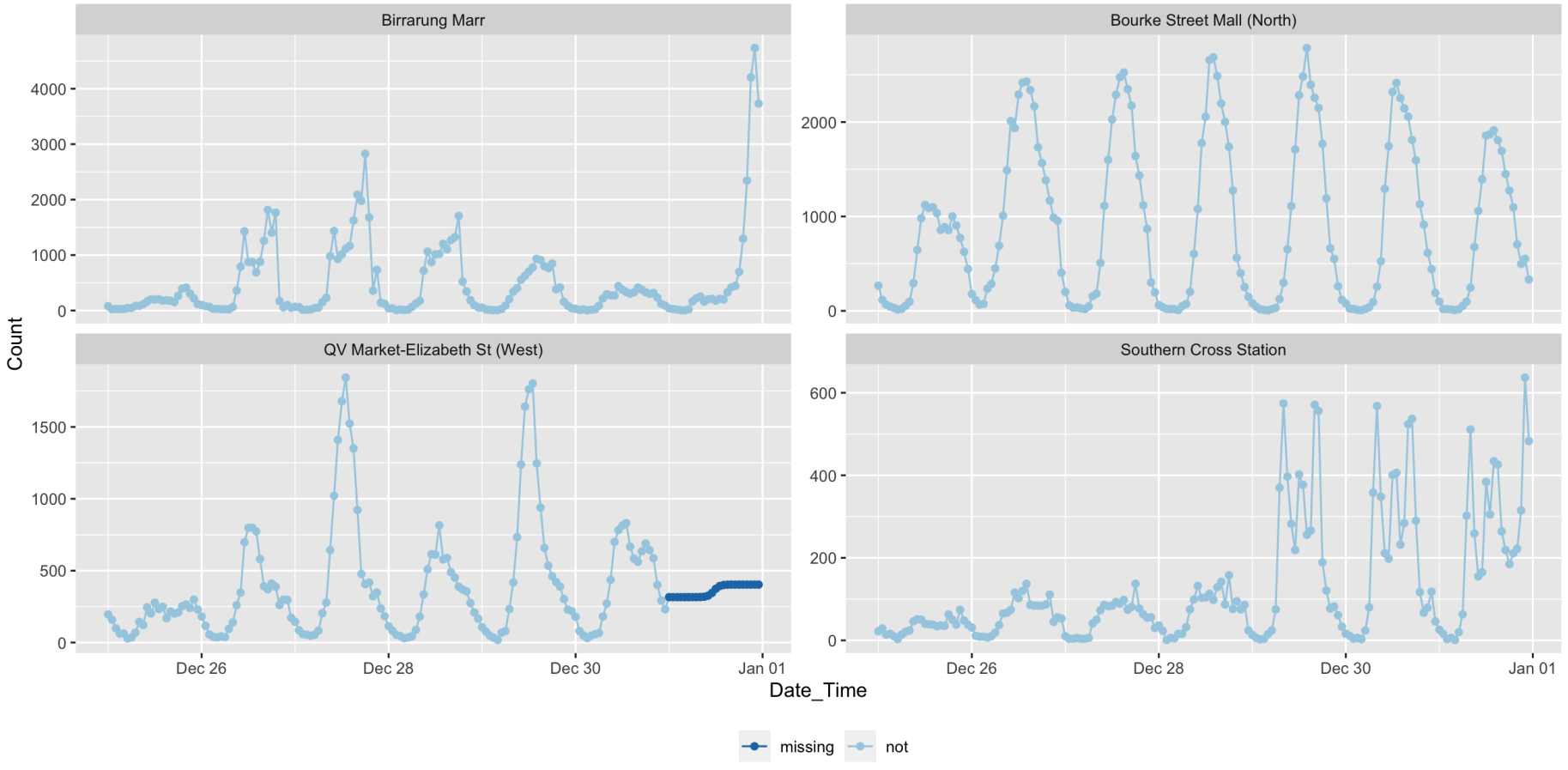
R



Missings at the end of the year at QV market.

Case study 3 Melbourne pedestrian traffic Part 3/5

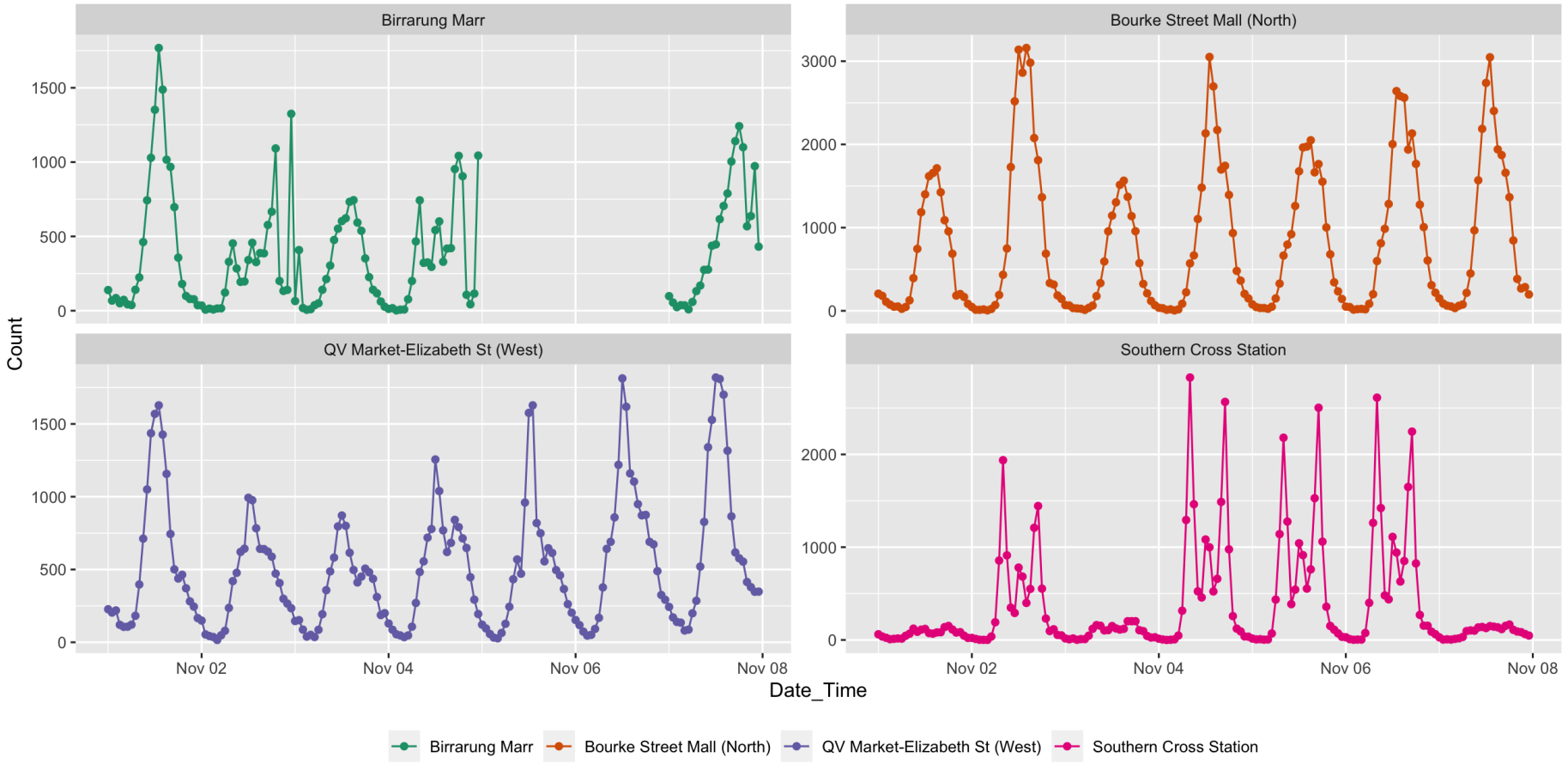
R



Imputed by moving average. (Would be better imputation if we used hour and type of day.)

Case study 3 Melbourne pedestrian traffic Part 4/5

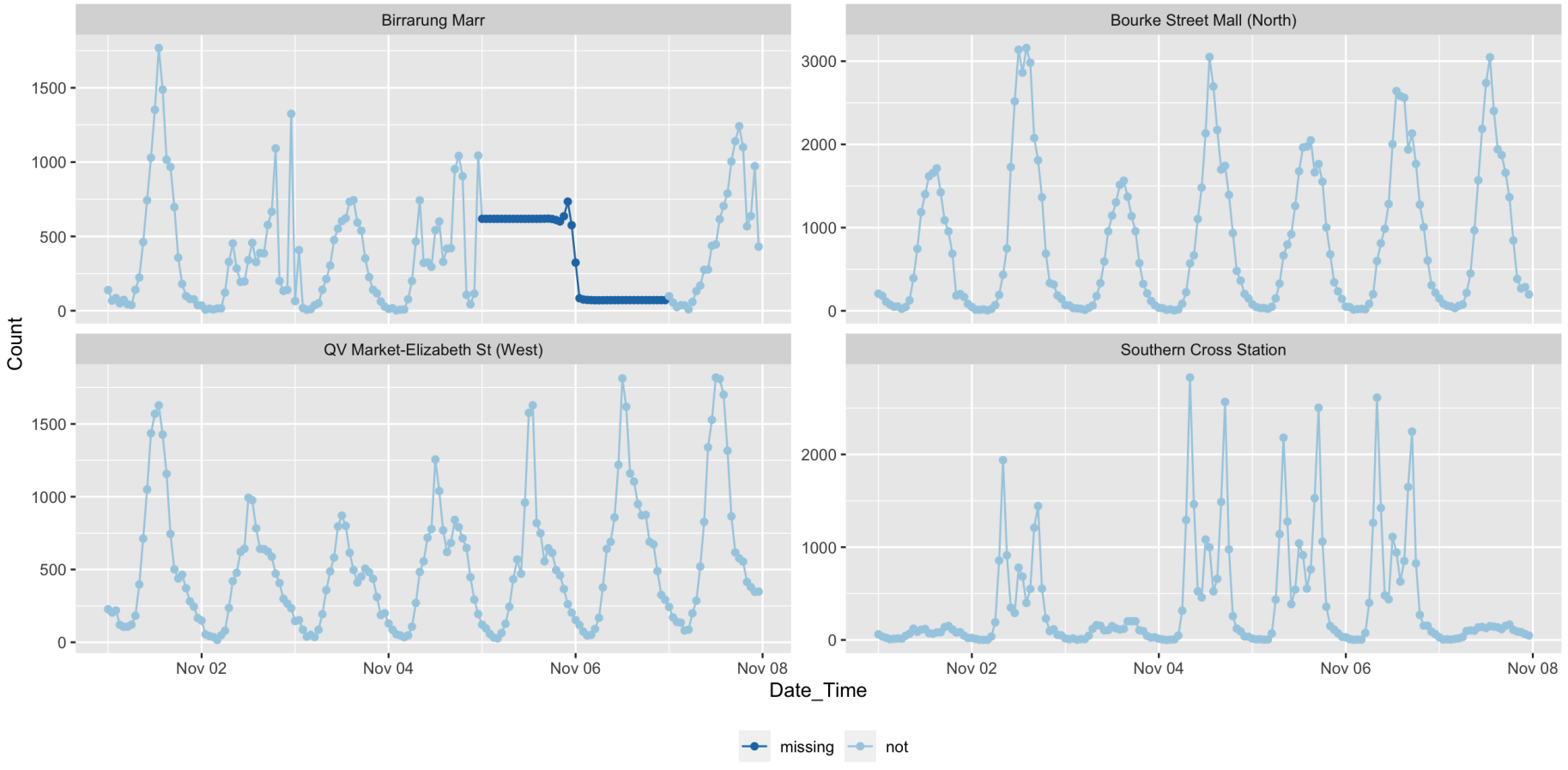
R



Missings in November at Birrarung Marr.

Case study 3 Melbourne pedestrian traffic Part 5/5

R



Imputed by moving average. Its difficult to do well at imputation with the irregular patterns at this location.

Longitudinal data

Information from the same individuals, recorded at multiple points in time.

Usually irregular, and not easy to regularise. Lots more short series.

Longitudinal data has the same properties as time series, but generally different objectives for the analysis.

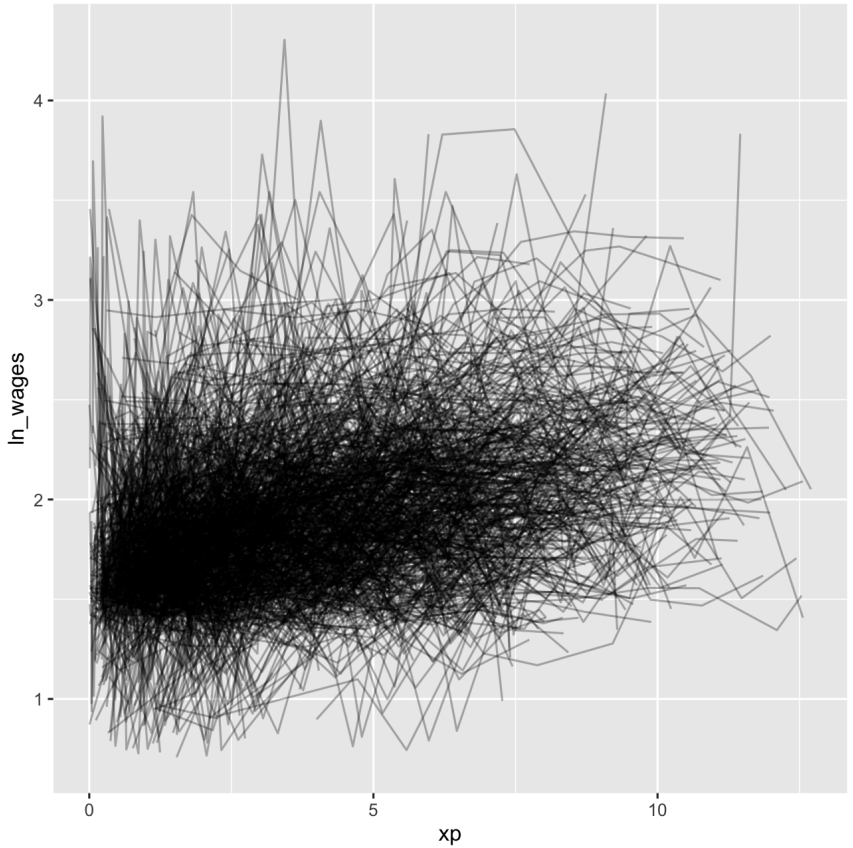
In the `brolgar` package methods build on the `tsibble` data object.

```
## # A tsibble: 6,402 x 9 [!]
## # Key:       id [888]
##       id ln_wages    xp   ged xp_since_ged black hispanic high_grade unemploy_rate
##    <int>    <dbl> <dbl> <int>        <dbl> <int>    <int>      <int>          <dbl>
##  1    31     1.49 0.015     1        0.015     0        1          8           3.21
##  2    31     1.43 0.715     1        0.715     0        1          8           3.21
##  3    31     1.47 1.73      1        1.73      0        1          8           3.21
##  4    31     1.75 2.77      1        2.77      0        1          8           3.3
##  5    31     1.93 3.93      1        3.93      0        1          8           2.89
##  6    31     1.71 4.95      1        4.95      0        1          8           2.49
##  7    31     2.09 5.96      1        5.96      0        1          8           2.6
##  8    31     2.13 6.98      1        6.98      0        1          8           4.8
##  9    36     1.98 0.315     1        0.315     0        0          9           4.89
## 10    36     1.80 0.983     1        0.983     0        0          9           7.4
## # … with 6,392 more rows
```

Case study 4 Wages Part 1/15

```
wages %>%
  ggplot(aes(x = xp,
             y = ln_wages,
             group = id)) +
  geom_line(alpha=0.3)
```

Log(wages) of 888 individuals, measured at various times in their employment (workforce experience).



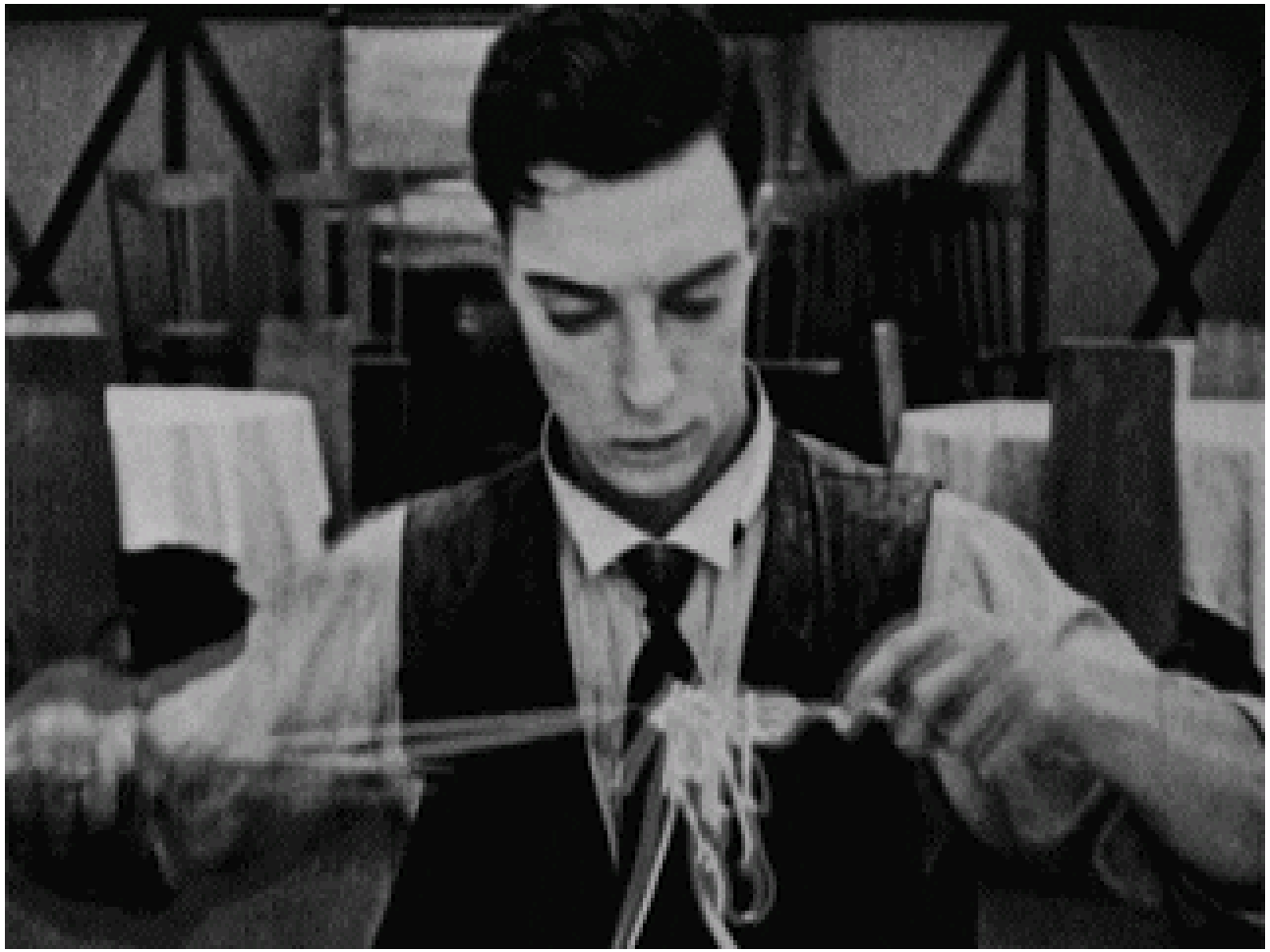
from a spaghetti mess

Source: giphy

to controlled spaghetti handling

Source: giphy

to perfection



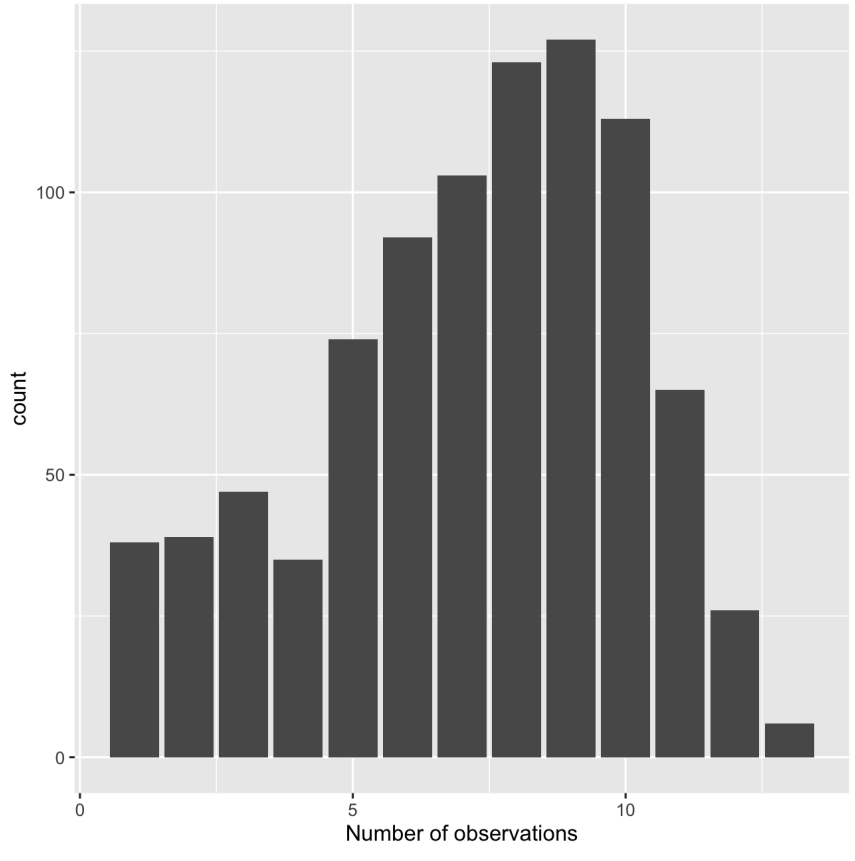
Source: giphy

Case study 4 Wages Part 2/15

Using features, compute the number of measurements for each subject

```
wages %>%
  features(ln_wages, n_obs) %>%
  ggplot(aes(x = n_obs)) +
  geom_bar() +
  xlab("Number of observations")
```

Different number of observations per person!



Case study 4 Wages Part 3/15

It can be important to filter on this, to remove subjects with little information

```
wages <- wages %>% add_n_obs()
wages %>%
  filter(n_obs > 3) %>%
  select(id, ln_wages, xp, n_obs)
```

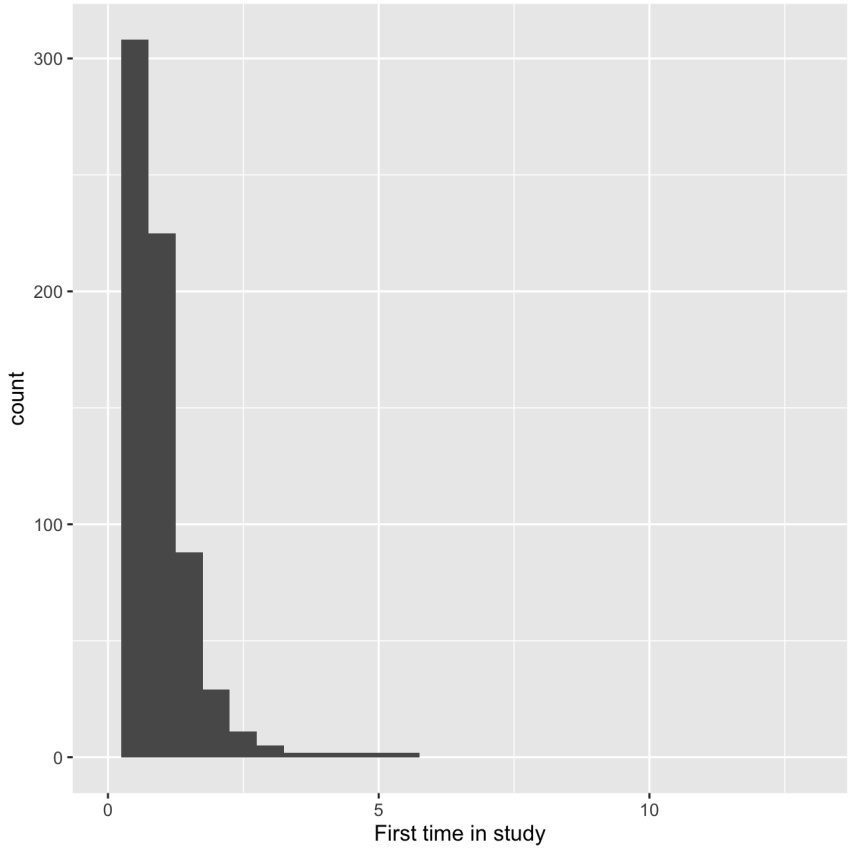
```
## # A tsibble: 6,145 x 4 [!]
## # Key:       id [764]
##       id ln_wages    xp n_obs
##    <int>    <dbl> <dbl> <int>
##  1    31     1.49 0.015     8
##  2    31     1.43 0.715     8
##  3    31     1.47 1.73      8
##  4    31     1.75 2.77      8
##  5    31     1.93 3.93      8
##  6    31     1.71 4.95      8
##  7    31     2.09 5.96      8
##  8    31     2.13 6.98      8
##  9    36     1.98 0.315    10
## 10    36     1.80 0.983    10
## # … with 6,135 more rows
```

Case study 4 Wages Part 4/15

Using features to extract minimum time

```
wages %>%
  features(xp, list(min = min)) %>%
  ggplot(aes(x = min)) +
  geom_histogram(binwidth=0.5) +
  xlim(c(0, 13)) +
  xlab("First time in study")
```

Subjects start in the study at different employment experience times



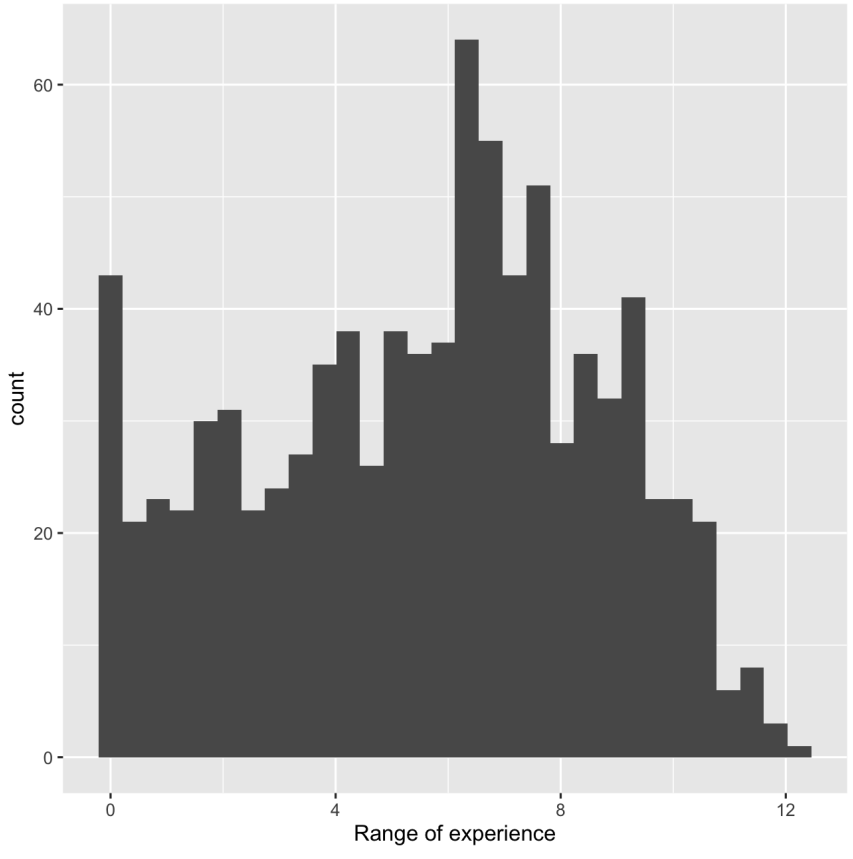
Case study 4 Wages Part 5/15

Using features to extract range of time index

```
wages_xp_range <- wages %>%
  features(xp, feat_ranges)

ggplot(wages_xp_range,
       aes(x = range_diff)) +
  geom_histogram() +
  xlab("Range of experience")
```

There's a range of workforce experience.

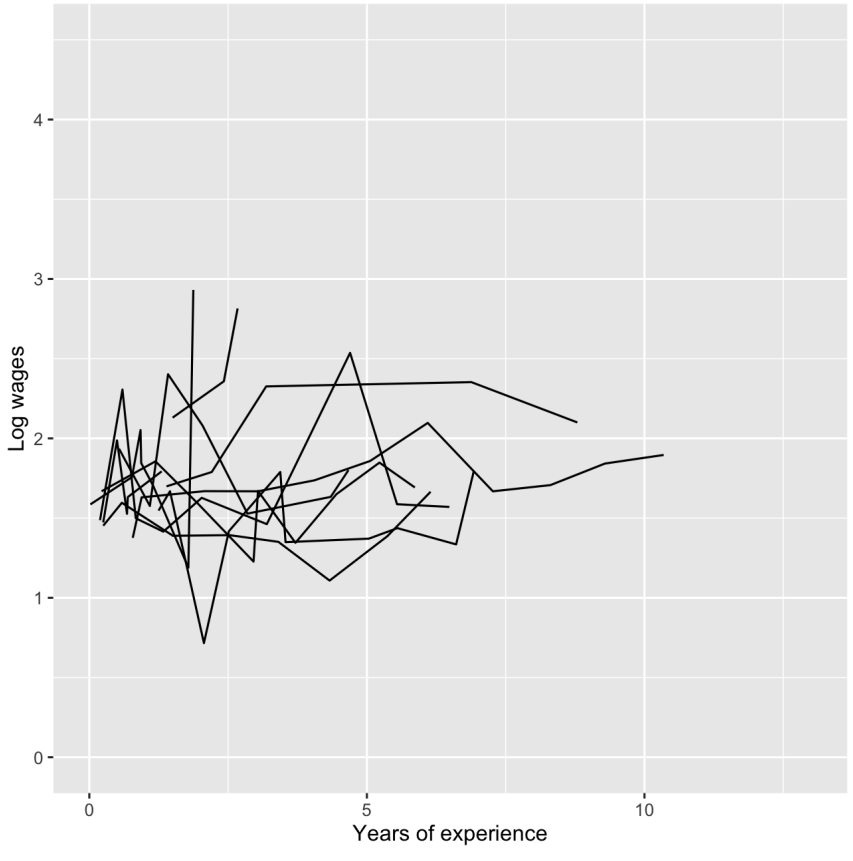


Small spoonfuls of spaghetti

Sample some individuals

```
wages %>%
  sample_n_keys(size = 10) %>%
  ggplot(aes(x = xp,
             y = ln_wages,
             group = id)) +
  geom_line() +
  xlim(c(0,13)) + ylim(c(0, 4.5)) +
  xlab("Years of experience") +
  ylab("Log wages")
```

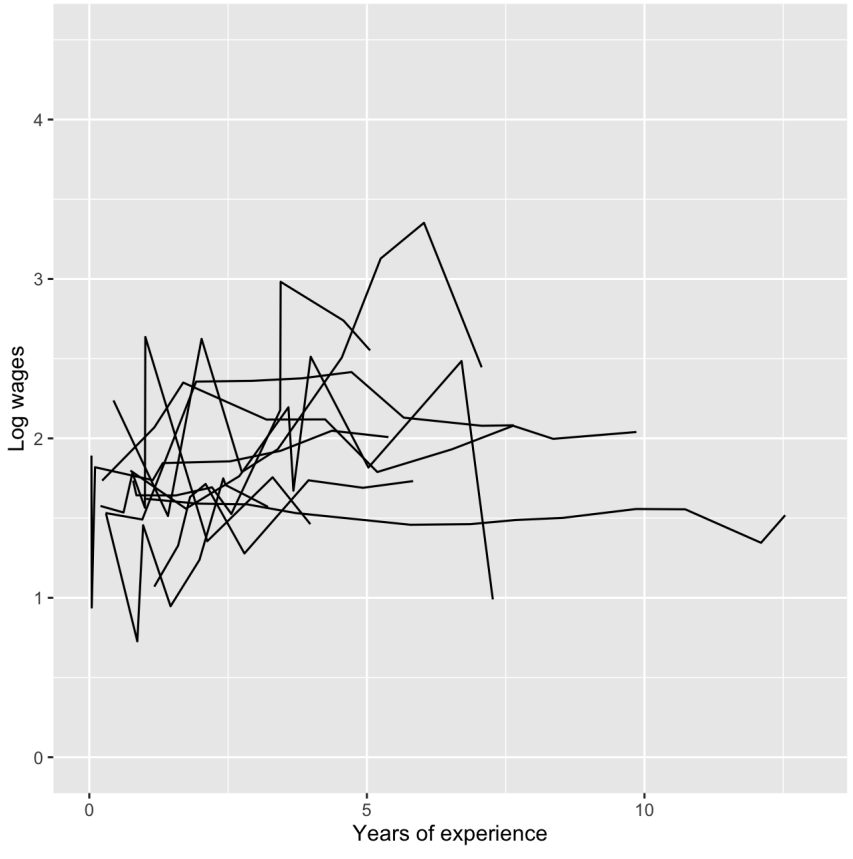
Wages conversion 0.5 = $1.65; 4.5 = $90



Take a spoonful of different lengths

Sample experienced individuals

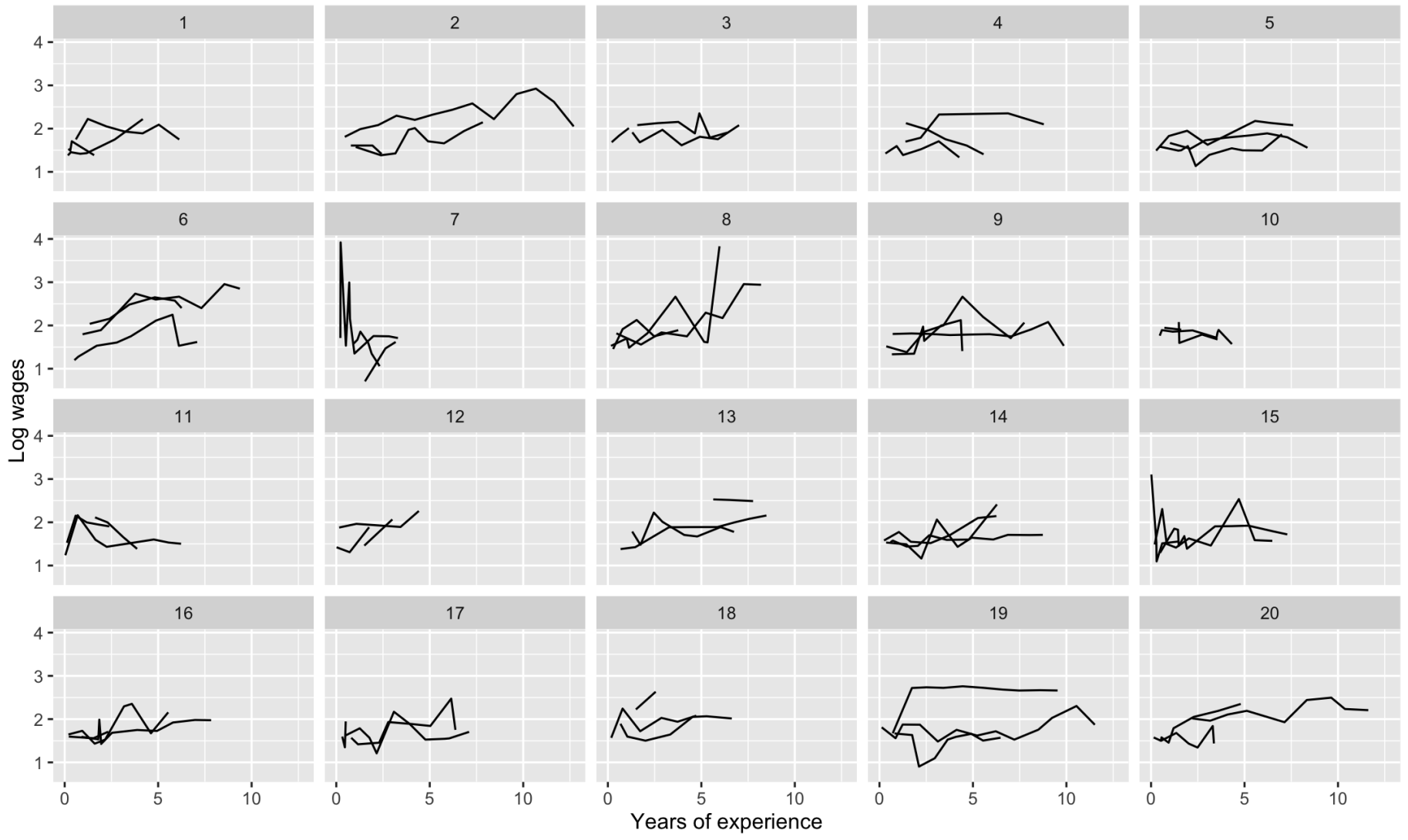
```
wages %>%
  add_n_obs() %>%
  filter(n_obs > 7) %>%
  sample_n_keys(size = 10) %>%
  ggplot(aes(x = xp,
             y = ln_wages,
             group = id)) +
  geom_line()  +
  xlim(c(0,13)) + ylim(c(0, 4.5)) +
  xlab("Years of experience") +
  ylab("Log wages")
```



Wages conversion 0.5 = $1.65; 4.5 = $90

Case study 4 Wages Part 8/15

info R



Special features

Remember scagnostics?

Compute longnostics for each subject

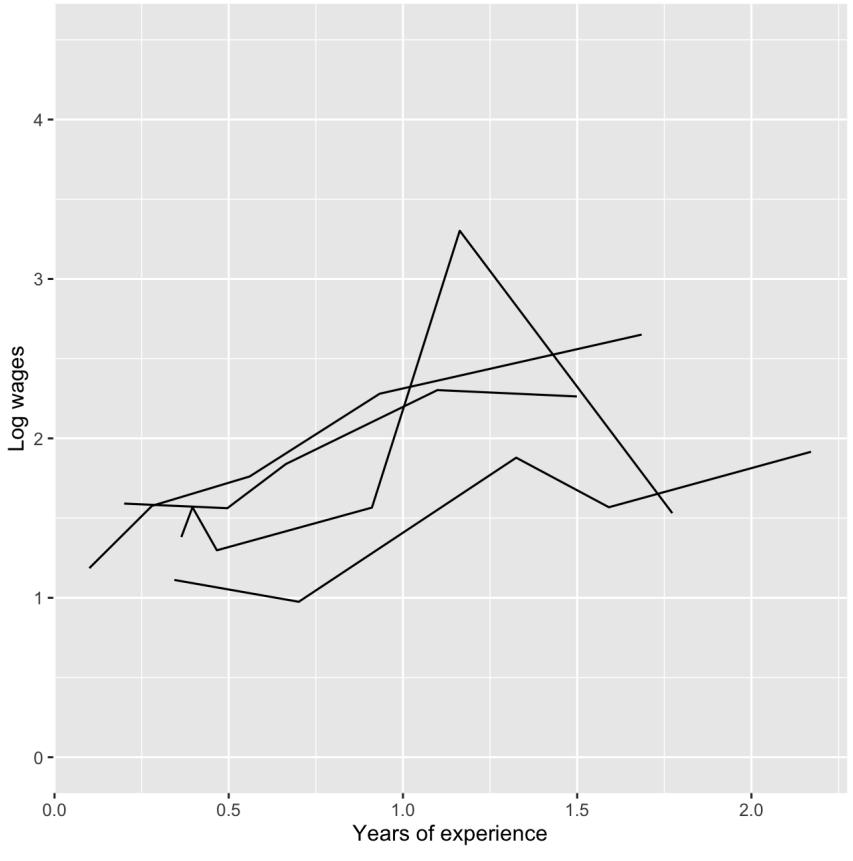
- Slope, intercept from simple linear model
- Variance, standard deviation
- Jumps, differences

For large collections of time series, take a look at the `feasts` package, which has a long list of time series features (tignostics) to calculate.

increasing

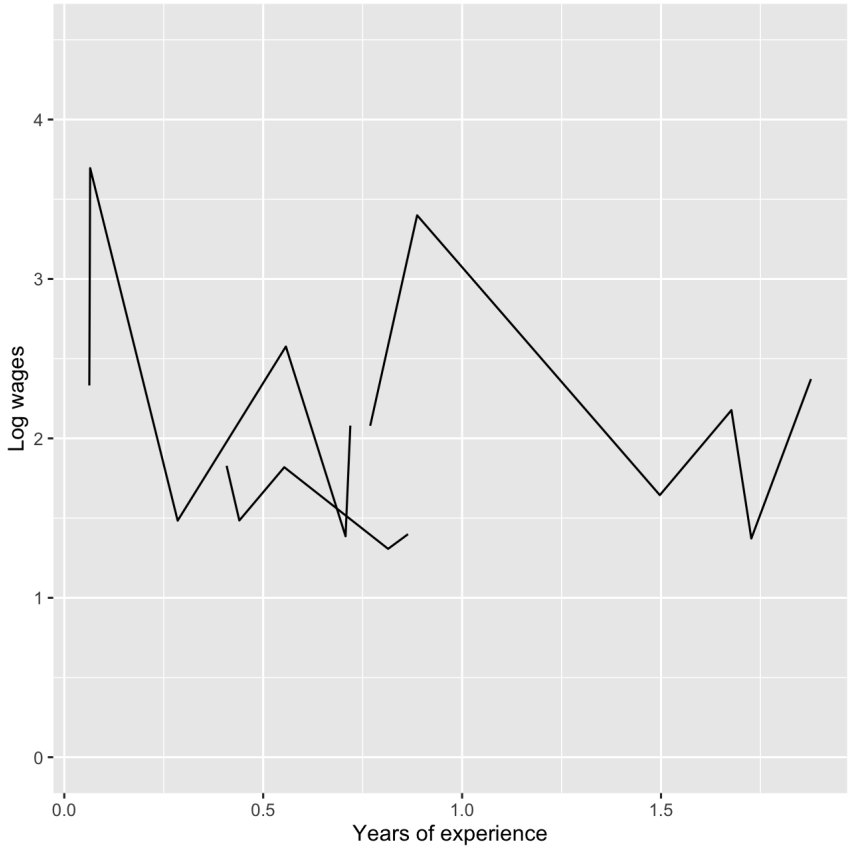
```
wages_slope <- wages %>%
  add_n_obs() %>%
  filter(n_obs > 4) %>%
  add_key_slope(ln_wages ~ xp) %>%
  as_tsibble(key = id, index = xp)

wages_slope %>%
  filter(.slope_xp > 0.4) %>%
  ggplot(aes(x = xp,
             y = ln_wages,
             group = id)) +
  geom_line() +
  ylim(c(0, 4.5)) +
  xlab("Years of experience") +
  ylab("Log wages")
```



decreasing

```
wages_slope %>%
  filter(.slope_xp < (-0.7)) %>%
  ggplot(aes(x = xp,
             y = ln_wages,
             group = id)) +
  geom_line() +
  ylim(c(0, 4.5)) +
  xlab("Years of experience") +
  ylab("Log wages")
```



Summarising individuals

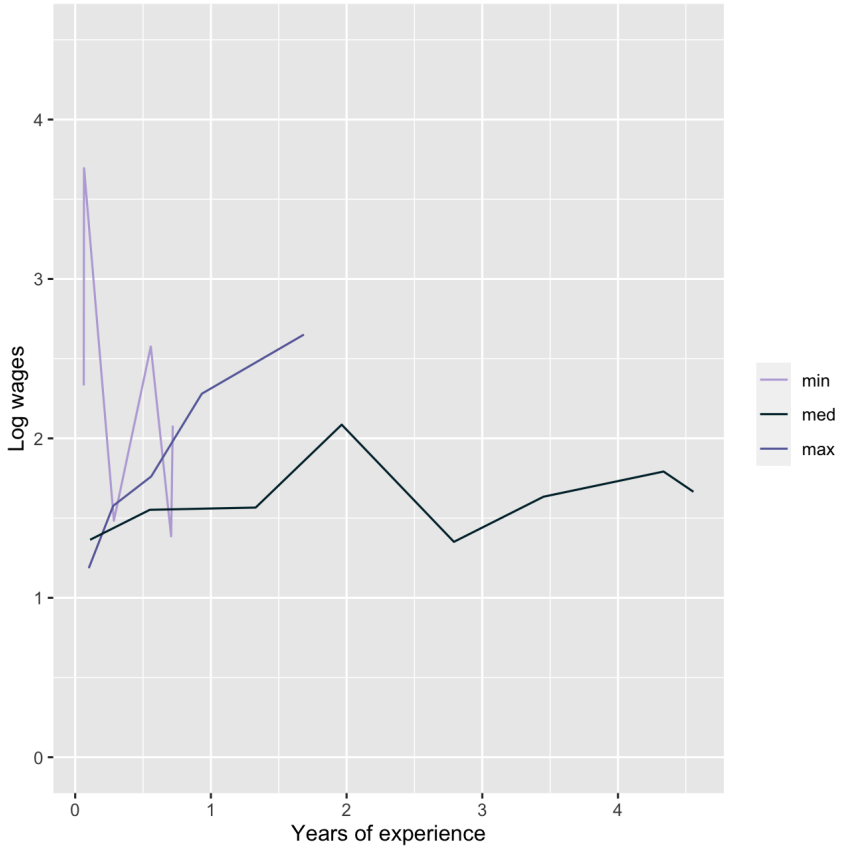
A different style of five number summary

Who is average? Who is different?

Find those individuals who are representative of the min, median, maximum, etc of growth, using `keys_near()`

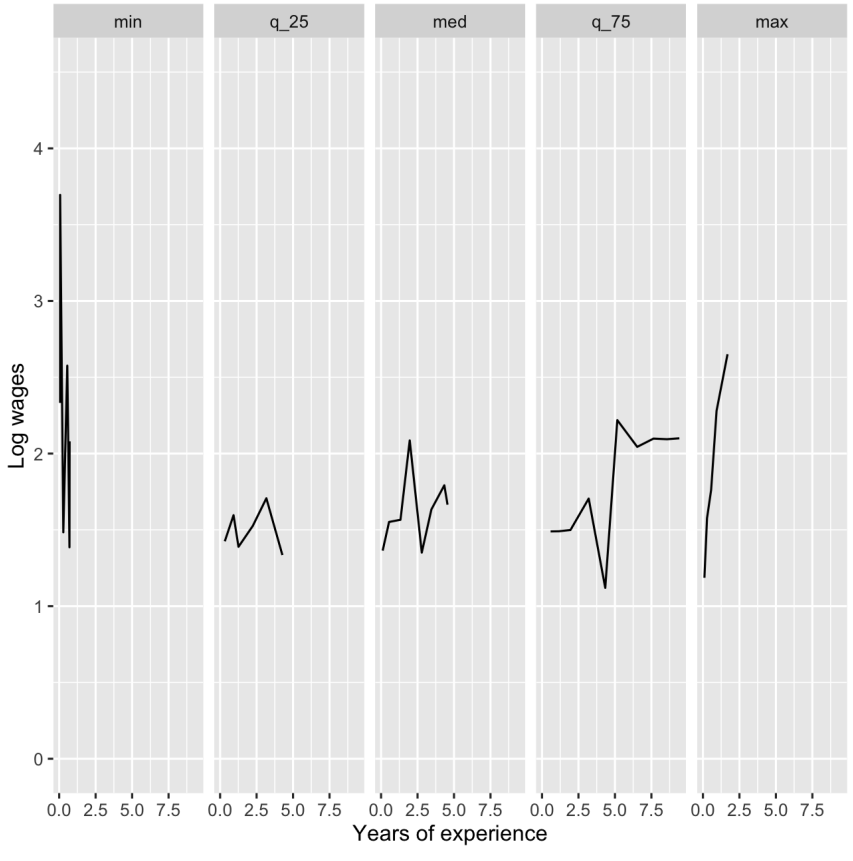
Case study 4 Wages Part 11/15

```
wages_threenum <- wages %>%
  add_n_obs() %>%
  filter(n_obs > 4) %>%
  key_slope(ln_wages ~ xp) %>%
  keys_near(key = id,
            var = .slope_xp,
            funs = l_three_num) %>%
  left_join(wages, by = "id") %>%
  as_tsibble(key = id, index = xp)
```



Case study 4 Wages Part 12/15

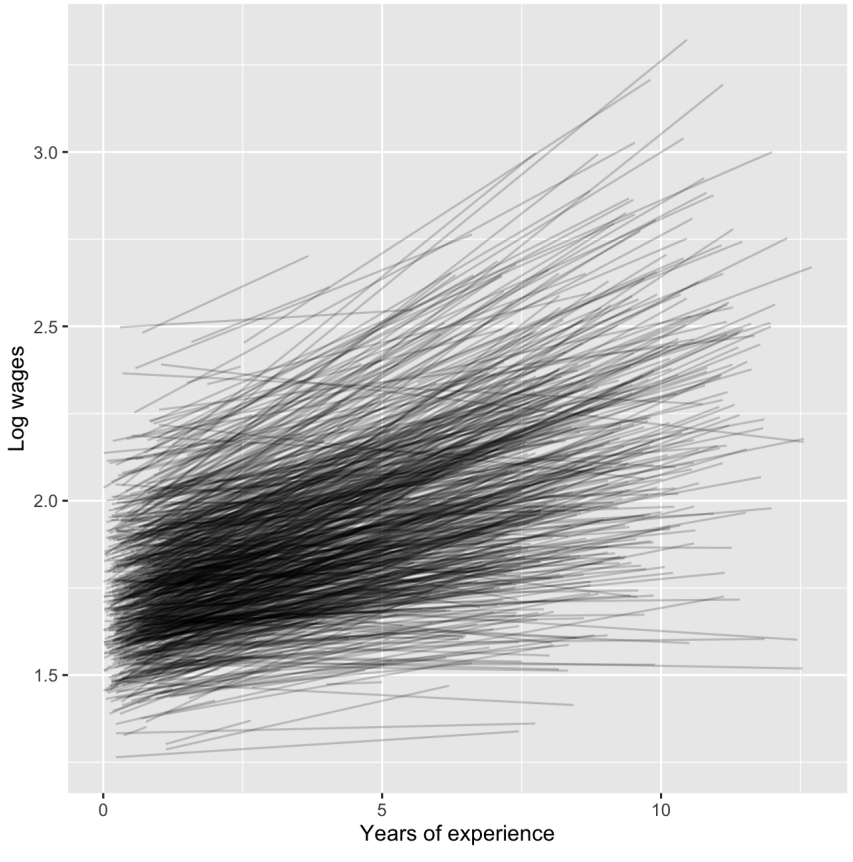
```
wages_fivenum <- wages %>%
  add_n_obs() %>%
  filter(n_obs > 4) %>%
  key_slope(ln_wages ~ xp) %>%
  keys_near(key = id,
            var = .slope_xp,
            funs = l_five_num) %>%
  left_join(wages, by = "id") %>%
  as_tsibble(key = id, index = xp)
```



Sculpting spaghetti

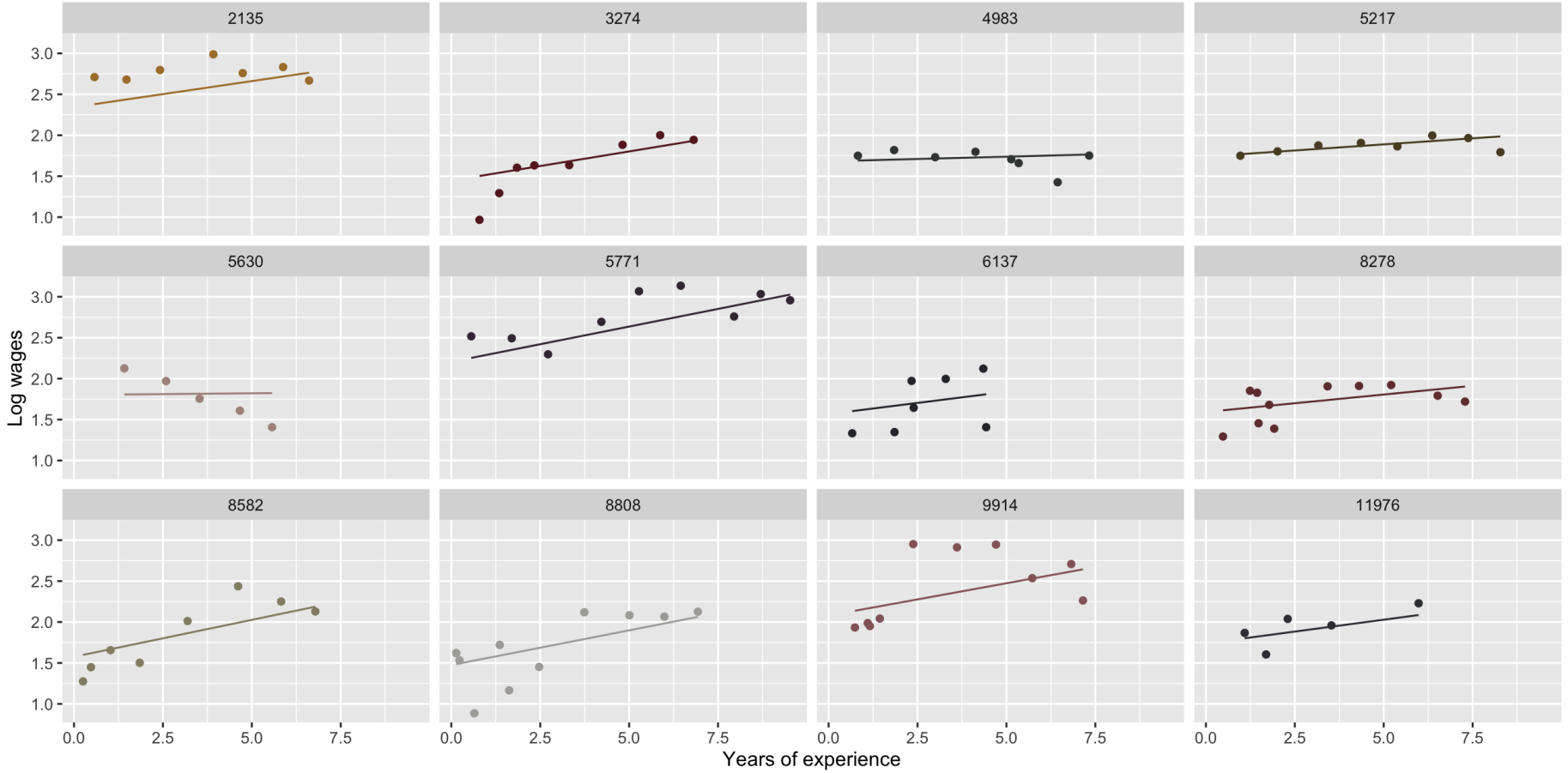
Mixed effects model, education as fixed effect, subject random effect using slope.

```
wages_fit_int <-
  lmer(ln_wages ~ xp + high_grade +
         (xp |id), data = wages)
wages_aug <- wages %>%
  add_predictions(wages_fit_int,
                  var = "pred_int") %>%
  add_residuals(wages_fit_int,
                var = "res_int")
```



R

Sample and show the data, too



Case study 4 Wages Part 15/15

- The individual wage experience is extremely varied
- Some individuals see a decline in their wages the longer they are in the workforce
- Most individuals generally see some (small) increase, on average

Exploratory analysis of this individual temporal patterns is really interesting!

That's it, for this lecture!

Lecturer: Di Cook

Department of Econometrics and Business Statistics

ETC5521.Clayton-x@monash.edu