Tidying data: extras

Packages

```
library(tidyverse)
```

The pig feed data again

```
my_url <- "http://ritsokiguess.site/STAC32/pigs1.txt"
pigs <- read_table(my_url)
pigs
```

```
# A tibble: 5 x 5
    pig feed1 feed2 feed3 feed4
  <dbl> <dbl> <dbl> <dbl> <dbl>
1     1  60.8  68.7  92.6  87.9
2     2  57    67.7  92.1  84.2
3     3  65    74    90.2  83.1
4     4  58.6  66.3  96.5  85.7
5     5  61.7  69.8  99.1  90.3
```

Make longer (as before)

```
pigs %>% pivot_longer(-pig, names_to="feed",
                      values_to="weight") -> pigs_longer
pigs_longer
```

```
# A tibble: 20 x 3
     pig feed  weight
   <dbl> <chr>  <dbl>
 1     1 feed1   60.8
 2     1 feed2   68.7
 3     1 feed3   92.6
 4     1 feed4   87.9
 5     2 feed1   57
 6     2 feed2   67.7
 7     2 feed3   92.1
 8     2 feed4   84.2
 9     3 feed1   65
10     3 feed2   74
11     3 feed3   90.2
12     3 feed4   83.1
13     4 feed1   58.6
14     4 feed2   66.3
15     4 feed3   96.5
16     4 feed4   85.7
17     5 feed1   61.7
18     5 feed2   69.8
19     5 feed3   99.1
20     5 feed4   90.3
```

Make wider two ways 1/2

`pivot_wider` is inverse of `pivot_longer`:

```
pigs_longer %>%
  pivot_wider(names_from=feed, values_from=weight)
```

```
# A tibble: 5 x 5
    pig feed1 feed2 feed3 feed4
  <dbl> <dbl> <dbl> <dbl> <dbl>
1     1  60.8  68.7  92.6  87.9
2     2  57    67.7  92.1  84.2
3     3  65    74    90.2  83.1
4     4  58.6  66.3  96.5  85.7
5     5  61.7  69.8  99.1  90.3
```

we are back where we started.

Make wider 2/2

Or

```
pigs_longer %>%
  pivot_wider(names_from=pig, values_from=weight)
```

```
# A tibble: 4 x 6
  feed    `1`   `2`   `3`   `4`   `5`
  <chr> <dbl> <dbl> <dbl> <dbl> <dbl>
1 feed1  60.8  57    65    58.6  61.7
2 feed2  68.7  67.7  74    66.3  69.8
3 feed3  92.6  92.1  90.2  96.5  99.1
4 feed4  87.9  84.2  83.1  85.7  90.3
```

Disease presence and absence at two locations

Frequencies of plants observed with and without disease at two locations:

```
Species     Disease present         Disease absent
       Location X Location Y  Location X Location Y
A            44         12            38         10
B            28         22            20         18
```

This has two rows of headers, so I rewrote the data file:

```
Species  present_x present_y    absent_x  absent_y
A            44         12            38         10
B            28         22            20         18
```

Read into data frame called `prevalence`.

```
# A tibble: 2 x 5
  Species present_x present_y absent_x absent_y
  <chr>       <dbl>     <dbl>    <dbl>    <dbl>
1 A              44        12       38       10
2 B              28        22       20       18
```

Lengthen and separate

```
prevalence %>%
  pivot_longer(-Species, names_to = "column",
               values_to = "freq") %>%
  separate(column, into = c("disease", "location"))
```

```
# A tibble: 8 x 4
  Species disease location  freq
  <chr>   <chr>   <chr>    <dbl>
1 A       present x           44
2 A       present y           12
3 A       absent  x           38
4 A       absent  y           10
5 B       present x           28
6 B       present y           22
7 B       absent  x           20
8 B       absent  y           18
```

Making longer, the better way

```
prevalence %>%
  pivot_longer(-Species, names_to=c("disease", "location"),
               names_sep="_", values_to="frequency") %>%
  arrange(Species, location, disease) -> prevalence_longer
prevalence_longer
```

```
# A tibble: 8 x 4
  Species disease location frequency
  <chr>   <chr>   <chr>        <dbl>
1 A       absent  x               38
2 A       present x               44
3 A       absent  y               10
4 A       present y               12
5 B       absent  x               20
6 B       present x               28
7 B       absent  y               18
8 B       present y               22
```

Making wider, different ways

```
prevalence_longer %>%
  pivot_wider(names_from=c(Species, location), values_from=frequency)
```

```
# A tibble: 2 x 5
  disease   A_x   A_y   B_x   B_y
  <chr>   <dbl> <dbl> <dbl> <dbl>
1 absent     38    10    20    18
2 present    44    12    28    22
```

```
prevalence_longer %>%
  pivot_wider(names_from=location, values_from=frequency)
```

```
# A tibble: 4 x 4
  Species disease     x     y
  <chr>   <chr>   <dbl> <dbl>
1 A       absent     38    10
2 A       present    44    12
3 B       absent     20    18
4 B       present    28    22
```

Interlude

```
pigs_longer
```

```
# A tibble: 20 x 3
     pig feed  weight
   <dbl> <chr>  <dbl>
 1     1 feed1   60.8
 2     1 feed2   68.7
 3     1 feed3   92.6
 4     1 feed4   87.9
 5     2 feed1   57
 6     2 feed2   67.7
 7     2 feed3   92.1
 8     2 feed4   84.2
 9     3 feed1   65
```

```
10      3 feed2   74
11      3 feed3   90.2
12      3 feed4   83.1
13      4 feed1   58.6
14      4 feed2   66.3
15      4 feed3   96.5
16      4 feed4   85.7
17      5 feed1   61.7
18      5 feed2   69.8
19      5 feed3   99.1
20      5 feed4   90.3
```

```r
pigs_longer %>%
  group_by(feed) %>%
  summarize(weight_mean=mean(weight))
```

```
# A tibble: 4 x 2
  feed  weight_mean
  <chr>       <dbl>
1 feed1        60.6
2 feed2        69.3
3 feed3        94.1
4 feed4        86.2
```

What if summary is more than one number?

eg. quartiles:

```r
pigs_longer %>%
  group_by(feed) %>%
  summarize(r=quantile(weight, c(0.25, 0.75)))
```

```
# A tibble: 8 x 2
# Groups:   feed [4]
  feed      r
  <chr> <dbl>
1 feed1  58.6
2 feed1  61.7
3 feed2  67.7
4 feed2  69.8
```

```
5 feed3  92.1
6 feed3  96.5
7 feed4  84.2
8 feed4  87.9
```

this also works

```
pigs_longer %>%
  group_by(feed) %>%
  summarize(r=list(quantile(weight, c(0.25, 0.75)))) %>%
  unnest(r)
```

```
# A tibble: 8 x 2
  feed      r
  <chr> <dbl>
1 feed1  58.6
2 feed1  61.7
3 feed2  67.7
4 feed2  69.8
5 feed3  92.1
6 feed3  96.5
7 feed4  84.2
8 feed4  87.9
```

or, even better, use `enframe`:

```
quantile(pigs_longer$weight, c(0.25, 0.75))
```

```
   25%    75%
65.975 90.225
```

```
enframe(quantile(pigs_longer$weight, c(0.25, 0.75)))
```

```
# A tibble: 2 x 2
  name  value
  <chr> <dbl>
1 25%    66.0
2 75%    90.2
```

A nice look

Run this one line at a time to see how it works:

```
pigs_longer %>%
  group_by(feed) %>%
  summarize(r=list(enframe(quantile(weight, c(0.25, 0.75))))) %>%
  unnest(r) %>%
  pivot_wider(names_from=name, values_from=value)
```

```
# A tibble: 4 x 3
  feed   `25%` `75%`
  <chr> <dbl> <dbl>
1 feed1  58.6  61.7
2 feed2  67.7  69.8
3 feed3  92.1  96.5
4 feed4  84.2  87.9
```

A hairy one

18 people receive one of three treatments. At 3 different times (pre, post, followup) two variables `y` and `z` are measured on each person:

```
# A tibble: 18 x 8
   id        treatment pre_y post_y  fu_y pre_z post_z  fu_z
   <chr>     <chr>     <dbl>  <dbl> <dbl> <dbl>  <dbl> <dbl>
 1 A.1       A             3     13     9     0      0     9
 2 A.2       A             0     14    10     6      6     3
 3 A.3       A             4      6    17     8      2     6
 4 A.4       A             7      7    13     7      6     4
 5 A.5       A             3     12    11     6     12     6
 6 A.6       A            10     14     8    13      3     8
 7 B.1       B             9     11    17     8     11    27
 8 B.2       B             4     16    13     9      3    26
 9 B.3       B             8     10     9    12      0    18
10 B.4       B             5      9    13     3      0    14
11 B.5       B             0     15    11     3      0    25
12 B.6       B             4     11    14     4      2     9
13 Control.1 Control      10     12    15     4      3     7
14 Control.2 Control       2      8    12     8      7    20
15 Control.3 Control       4      9    10     2      0    10
```

```
16 Control.4 Control      10      8     8     5      8    14
17 Control.5 Control      11     11    11     1      0    11
18 Control.6 Control       1      5    15     8      9    10
```

Attempt 1

```
repmes %>% pivot_longer(contains("_"),
                        names_to=c("time", "var"),
                        names_sep="_"
                         )
```

```
# A tibble: 108 x 5
   id    treatment time  var   value
   <chr> <chr>     <chr> <chr> <dbl>
 1 A.1   A         pre   y         3
 2 A.1   A         post  y        13
 3 A.1   A         fu    y         9
 4 A.1   A         pre   z         0
 5 A.1   A         post  z         0
 6 A.1   A         fu    z         9
 7 A.2   A         pre   y         0
 8 A.2   A         post  y        14
 9 A.2   A         fu    y        10
10 A.2   A         pre   z         6
# i 98 more rows
```

This is *too* long! We wanted a column called `y` and a column called `z`, but they have been pivoted-longer too.

Attempt 2

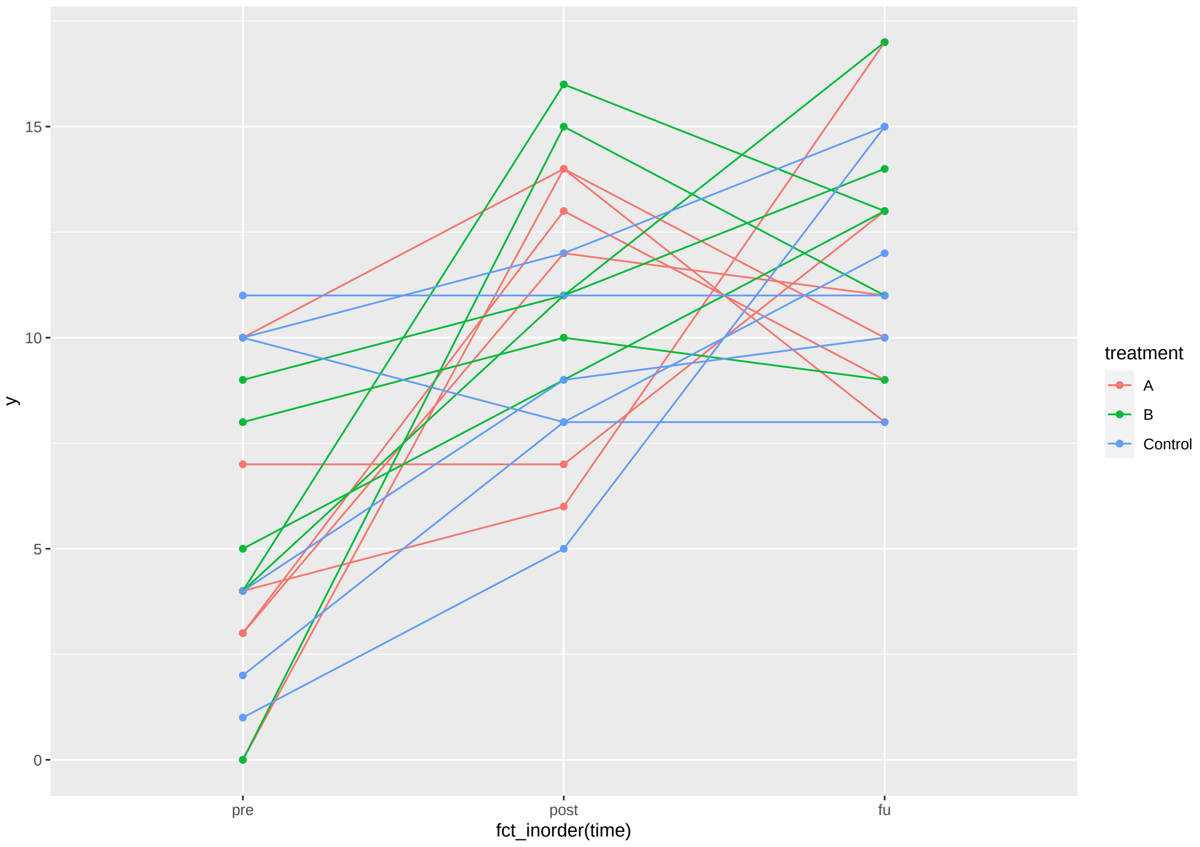
```
repmes %>% pivot_longer(contains("_"),
                        names_to=c("time", ".value"),
                        names_sep="_"
                        ) -> repmes3
repmes3
```

```
# A tibble: 54 x 5
   id    treatment time      y     z
   <chr> <chr>     <chr> <dbl> <dbl>
 1 A.1   A         pre       3     0
 2 A.1   A         post     13     0
 3 A.1   A         fu        9     9
 4 A.2   A         pre       0     6
 5 A.2   A         post     14     6
 6 A.2   A         fu       10     3
 7 A.3   A         pre       4     8
 8 A.3   A         post      6     2
 9 A.3   A         fu       17     6
10 A.4   A         pre       7     7
# i 44 more rows
```

This has done what we wanted.

make a graph

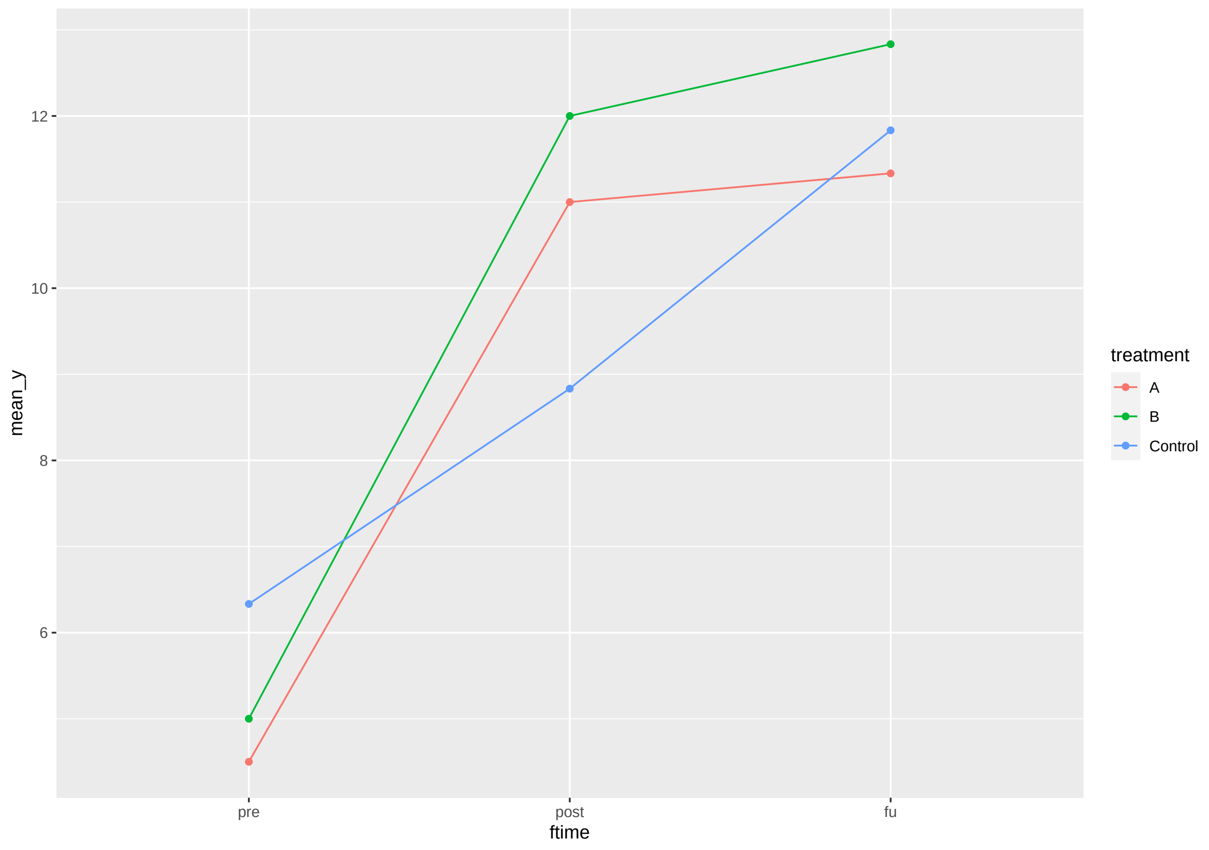
```
ggplot(repmes3, aes(x=fct_inorder(time), y=y,
                    colour=treatment, group=id)) +
  geom_point() + geom_line()
```



A so-called spaghetti plot. The three measurements for each person are joined by lines, and the lines are coloured by treatment.

or do the plot with means

```
repmes3 %>% group_by(treatment, ftime=fct_inorder(time)) %>%
  summarize(mean_y=mean(y)) %>%
  ggplot(aes(x=ftime, y=mean_y, colour=treatment,
             group=treatment)) +
    geom_point() + geom_line()
```



On average, the two real treatments go up and level off, but the control group is very different.