From Longer and Wider, We Stand on Guard for Thee

Ken Butler, Department of Computer and Mathematical Sciences, UTSC (Scarborough), butler@utsc.utoronto.ca, @kenbutler12

Packages

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------
```

```
## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(broom)
```

Pig feed

20 pigs are randomly assigned to one of four pig feeds, and the weight gain of each pig is measured:

pig	feed1	feed2	feed3	feed4
1	60.8	68.7	92.6	87.9
2	57.0	67.7	92.1	84.2
3	65.0	74.0	90.2	83.1
4	58.6	66.3	96.5	85.7
5	61.7	69.8	99.1	90.3

Say we want graphs of weight gain for each feed.

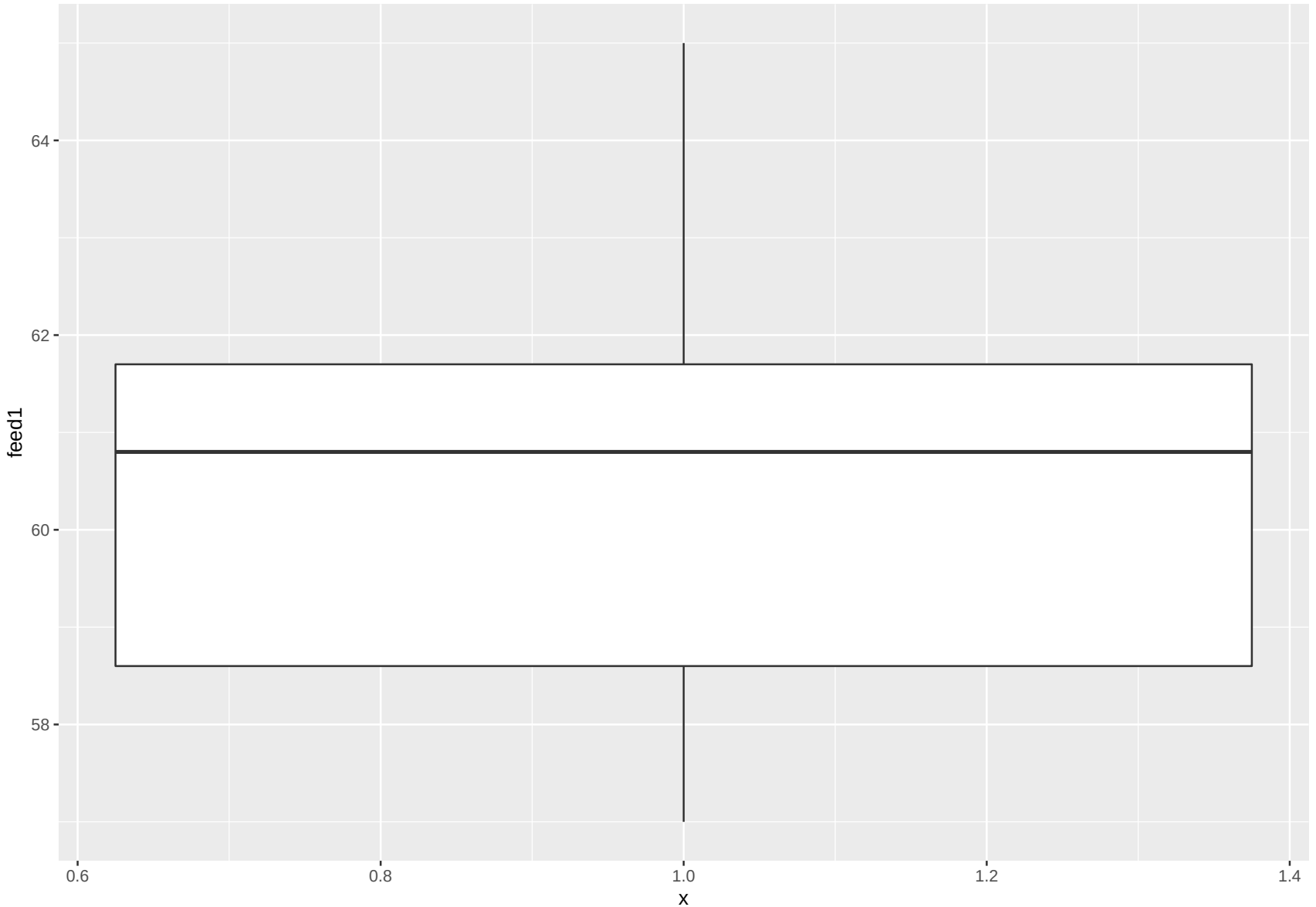
Read in:

```
pigs <- read_table("pigs1.txt")
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
```

and then we have to do this 4 times...

```
ggplot(pigs, aes(x=1, y=feed1)) + geom_boxplot()
```



The problem

- The data frame is the *wrong shape*.
- Need all the weight gains in *one* column, with another column saying what feed that weight gain was from
- Make data frame longer.
- Old tools:
  - `reshape`
  - `reshape2`
  - `gather` (from `tidyr`)
- New tool: `pivot_longer`

On the pig feed data

```
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
```

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

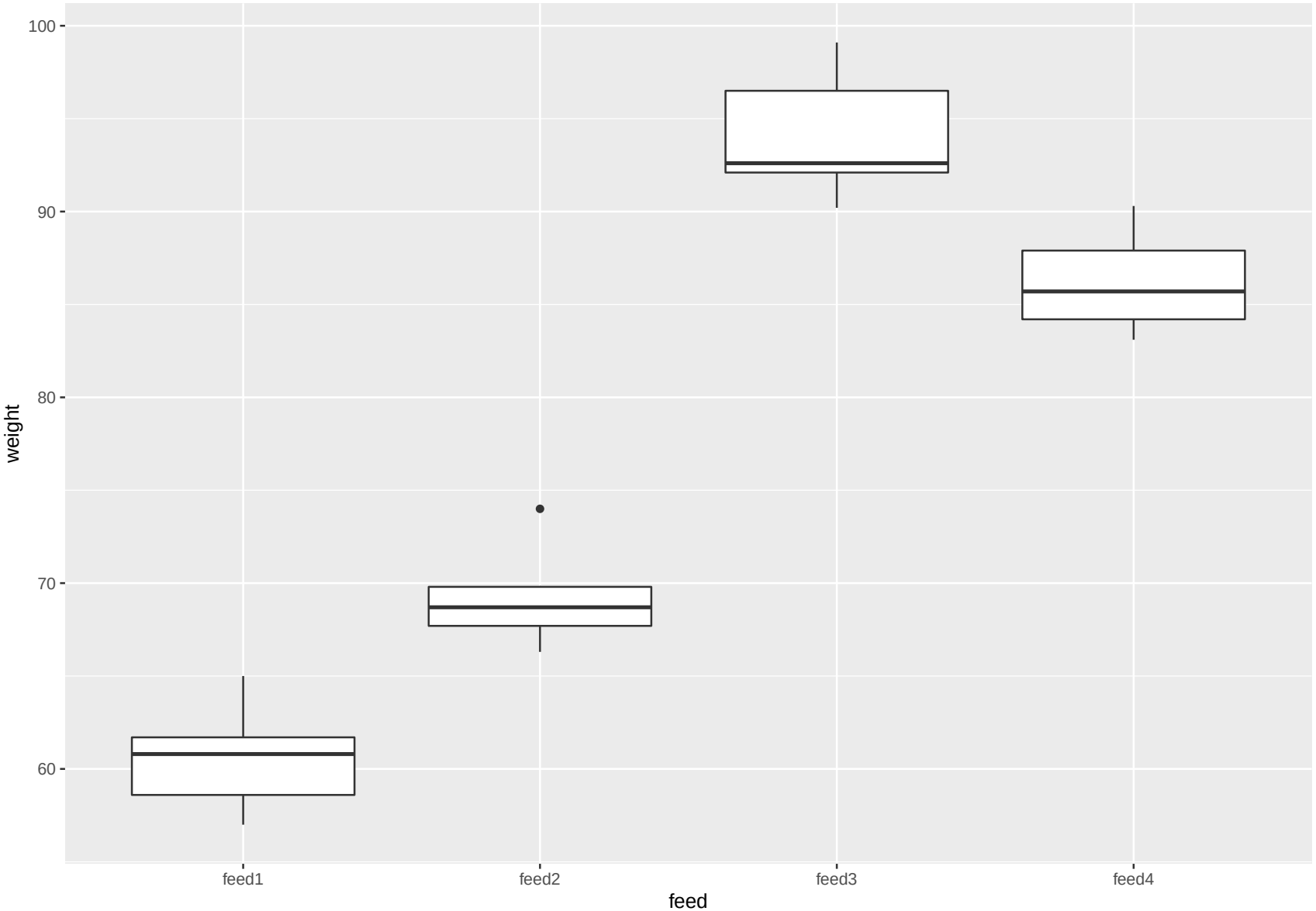
The results

```
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
```

Now we can make all 4 graphs at once

```
ggplot(pigs_longer, aes(x=feed, y=weight)) + geom_boxplot()
```



Making wider

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.

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`.

Gather

needs to take two steps:

```
prevalence %>%
  gather(disloc, freq, -Species) %>%
  separate(disloc, into=c("disease", "location"))
```

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

Making longer, the new way

Each column name encodes both disease and location, so put both of these in `names_to`:

```
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
```

How do I make this wider?

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

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

Interlude

```
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?

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

```
## Error: Column `r` must be length 1 (a summary value), no
```

the right way to do it

```
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 tidy from broom:

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

```
## Warning: 'tidy.numeric' is deprecated.
## See help("Deprecated")
```

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

and so

```
pigs_longer %>%
  group_by(feed) %>%
  summarize(r=list(tidy(quantile(weight, c(0.25, 0.75))))) %>%
  unnest(r) %>%
  pivot_wider(names_from=names, values_from=x)
```

```
## Warning: 'tidy.numeric' is deprecated.
```

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
##    <chr>     <chr>     <dbl>  <dbl> <dbl> <dbl>  <dbl> <
##  1 A.1       A             3     13     9     0      0
##  2 A.2       A             0     14    10     6      6
##  3 A.3       A             4      6    17     8      2
##  4 A.4       A             7      7    13     7      6
##  5 A.5       A             3     12    11     6     12
##  6 A.6       A            10     14     8    13      3
##  7 B.1       B             9     11    17     8     11
##  8 B.2       B             4     16    13     9      3
##  9 B.3       B             8     10     9    12      0
## 10 B.4       B             5      9    13     3      0
## 11 B.5       B             0     15    11     3      0
## 12 B.6       B             4     11    14     4      2
## 13 Control_1 Control      10     12    15     4      3
```

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
## #     with 98 more rows
```

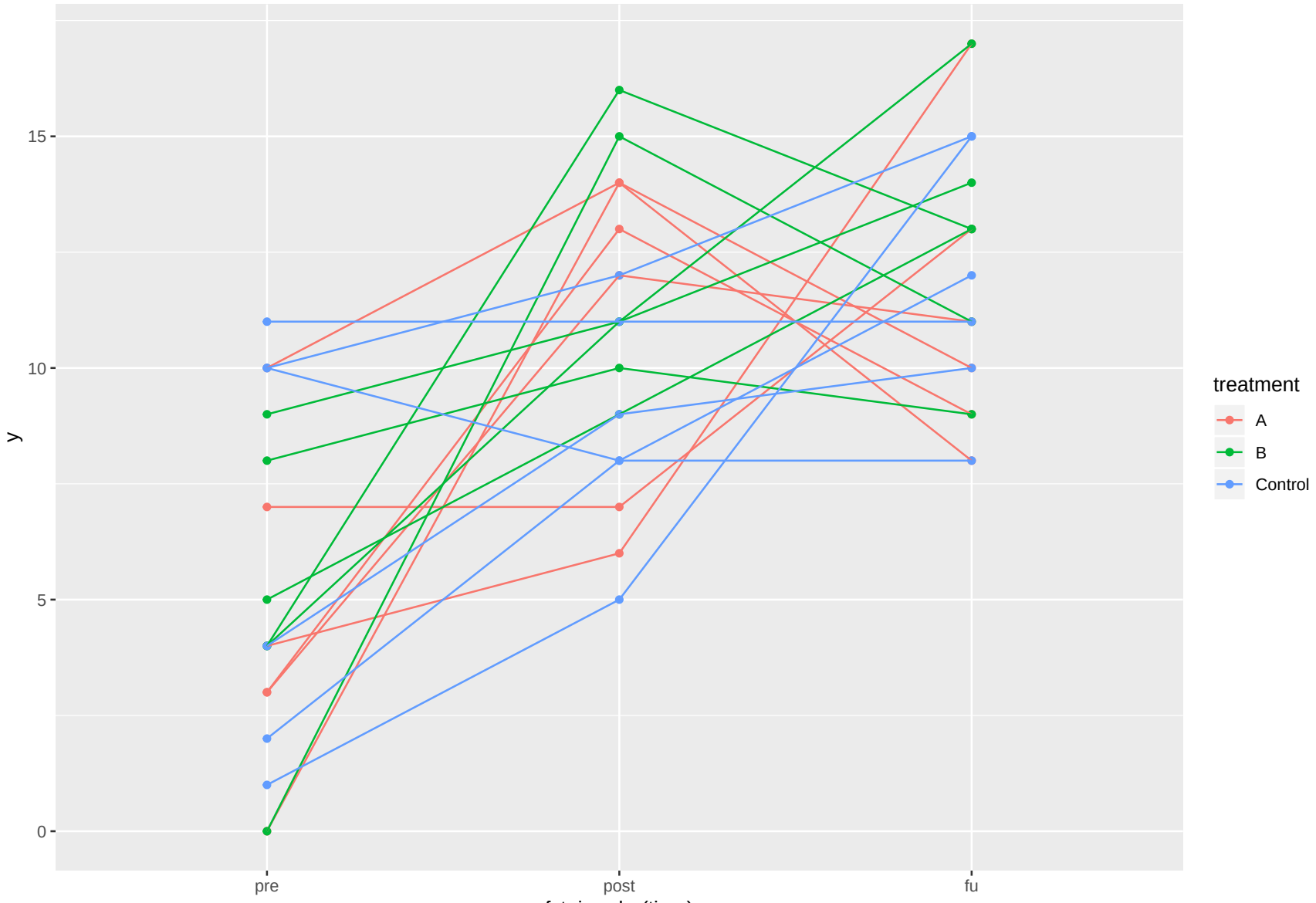
Attempt 2

```
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
```

make a graph

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



thank you!