Examples - Using pivot_longer() and pivot_wider()

Example of *tibble()* and *tribble()* functions to create a data_frame.

```
library(tidyverse)

BP_narrow <- tibble(
  x = c("a", "b"),
  y = c(1,2),
  z = c(3.6, 8.5)
)

BP_narrow <- tribble(
  ~x, ~y,  ~z,
  "a", 2,  3.6,
  "b", 1,  8.5
)
```

```
BP_narrow <- tribble(
  ~subject, ~when, ~spb,
  "BHO", "before", 160,
  "GWB", "before", 120,
  "WJC", "before", 105,
  "BHO", "after", 115,
  "GWB", "after", 135,
  "WJC", "after", 145
)
```

Examples of *pivot_longer()* and *pivot_wider()*

```
BP_wide <- BP_narrow %>% pivot_wider(names_from = "when", values_from = "spb")
BP_wide
```

```
## # A tibble: 3 x 3
##   subject before after
##   <chr>    <dbl> <dbl>
## 1 BHO        160   115
## 2 GWB        120   135
## 3 WJC        105   145
```

```
BP_narrow_new <- BP_wide %>% pivot_longer(c("before", "after"), names_to = "when", values_to = "spb" )

BP_narrow_new
```

```
## # A tibble: 6 x 3
##   subject when     spb
```

```
##   <chr>   <chr>  <dbl>
## 1 BHO     before   160
## 2 BHO     after    115
## 3 GWB     before   120
## 4 GWB     after    135
## 5 WJC     before   105
## 6 WJC     after    145
```

Or we can remove the *subject* column and use the remaining two columns *before* and *after*.

```
BP_narrow_new <- BP_wide %>% pivot_longer(-subject, names_to = "when", values_to = "spb" )

BP_narrow_new
```

```
## # A tibble: 6 x 3
##   subject when     spb
##   <chr>   <chr>  <dbl>
## 1 BHO     before   160
## 2 BHO     after    115
## 3 GWB     before   120
## 4 GWB     after    135
## 5 WJC     before   105
## 6 WJC     after    145
```

Example *spread()*

Try the code in Section 5.2.4 on pages 101-103. Try to use the new *pivot_wider()* function.

```
library(babynames)

babynames
```

```
## # A tibble: 1,924,665 x 5
##     year sex   name          n   prop
##    <dbl> <chr> <chr>     <int>  <dbl>
##  1  1880 F     Mary       7065 0.0724
##  2  1880 F     Anna       2604 0.0267
##  3  1880 F     Emma       2003 0.0205
##  4  1880 F     Elizabeth  1939 0.0199
##  5  1880 F     Minnie     1746 0.0179
##  6  1880 F     Margaret   1578 0.0162
##  7  1880 F     Ida        1472 0.0151
##  8  1880 F     Alice      1414 0.0145
##  9  1880 F     Bertha     1320 0.0135
## 10  1880 F     Sarah      1288 0.0132
## # ... with 1,924,655 more rows
```

Example *for*

Try the code on pages 104-105

Example *apply()*

Try the code on pages 106-107

Next week we will take a look at the *map* functions from the *purrr* R package, which are modern alternatives to *for* loops and the *apply()* functions.