

Parallel Iterations

And a few additional {purrr} functions

Daniel Anderson
Week 5, Class 1



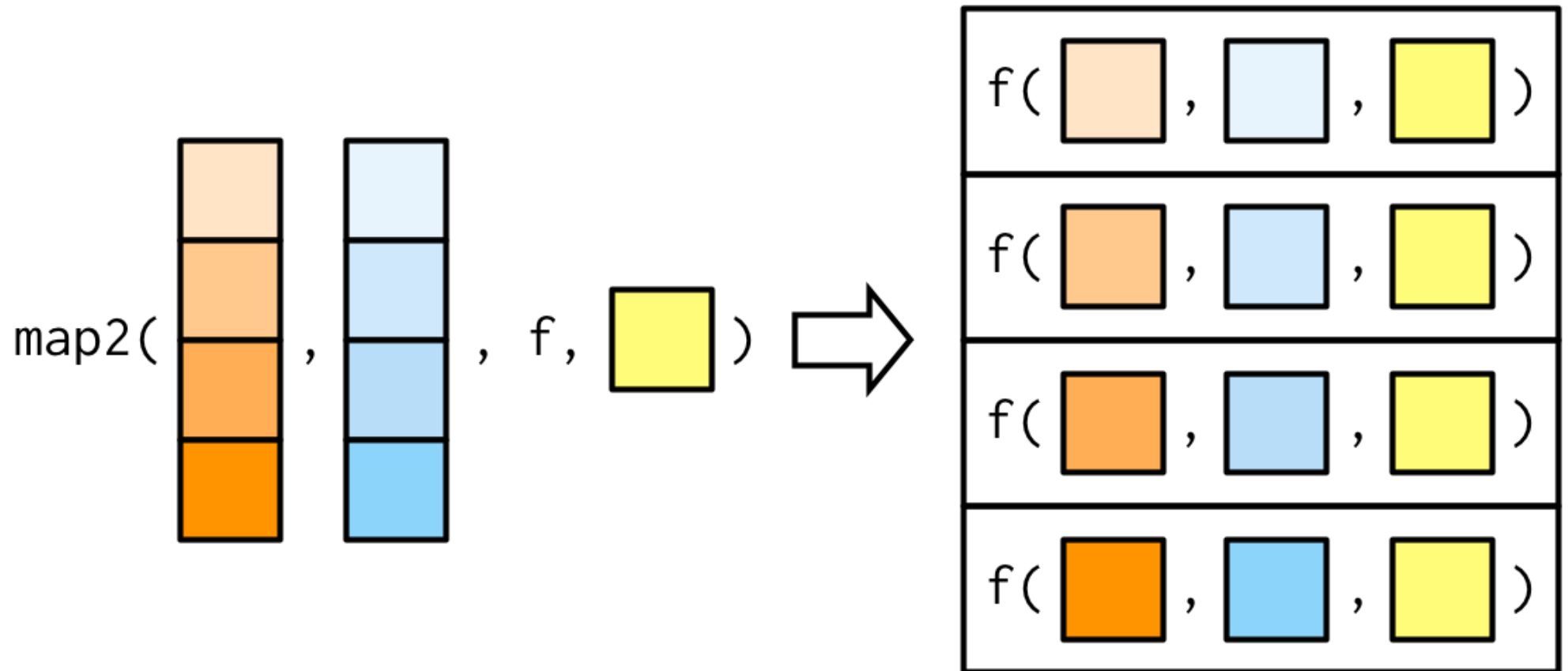
Agenda

- Finish up slides from Wednesday
- Discuss `map2_*` and `pmap_*`
- Introduce `walk`, `reduce` and `safely`

Learning objectives

- Understand the differences between `map`, `map2`, and `pmap`
- Know when to apply `walk` instead of `map`, and why it may be useful
- Be able to collapse lists with `reduce`
- Diagnose errors with `safely` (and understand other situations where it may be helpful)

map2



A few Examples

Basic simulations - iterating over two vectors

Plots by month, changing the title

Simulation

- Simulate data from a normal distribution
 - Vary n from 5 to 150 by increments of 5
 - For each n , vary μ from -2 to 2 by increments of 0.25

Simulation

- Simulate data from a normal distribution
 - Vary n from 5 to 150 by increments of 5
 - For each n , vary μ from -2 to 2 by increments of 0.25

How do we get all combinations

Simulation

- Simulate data from a normal distribution
 - Vary n from 5 to 150 by increments of 5
 - For each n , vary μ from -2 to 2 by increments of 0.25

How do we get all combinations

`expand.grid`

Example `expand.grid`

Bonus: It turns it into a data frame!

```
ints <- 1:3  
lets <- c("a", "b", "c")  
expand.grid(ints, lets)
```

```
##   Var1 Var2  
## 1     1    a  
## 2     2    a  
## 3     3    a  
## 4     1    b  
## 5     2    b  
## 6     3    b  
## 7     1    c  
## 8     2    c  
## 9     3    c
```

Set conditions

```
conditions <- expand.grid(n = seq(5, 150, 5),  
                          mu = seq(-2, 2, 0.25))  
  
head(conditions)
```

```
##      n mu  
## 1    5 -2  
## 2   10 -2  
## 3   15 -2  
## 4   20 -2  
## 5   25 -2  
## 6   30 -2
```

```
tail(conditions)
```

```
##      n mu  
## 505 125  2  
## 506 130  2  
## 507 135  2  
## 508 140  2  
## 509 145  2  
## 510 150  2
```

Simulate!

```
sim1 <- map2(conditions$n, conditions$mu, ~rnorm(.x, .y, sd = 10))
str(sim1)
```

```
## List of 510
## $ : num [1:5] 1.47 4.21 -3.88 -1.42 -1.68
## $ : num [1:10] 7.75 -8.09 7.61 -1.84 -18.61 ...
## $ : num [1:15] -3.84 -2.19 9.58 -0.82 -4.13 ...
## $ : num [1:20] -7.421 3.697 -12.473 0.626 -5.097 ...
## $ : num [1:25] 12.75 -13.56 -3.93 -10.55 -10.48 ...
## $ : num [1:30] 12.63 -6.45 1.68 -2.72 -15.73 ...
## $ : num [1:35] 6.8 -35.46 4.9 5.21 -18.6 ...
## $ : num [1:40] -20.1641 -6.6494 3.8459 -0.0782 15.5604 ...
## $ : num [1:45] -0.139 -2.251 11.996 -15.19 -7.86 ...
## $ : num [1:50] 18.022 1.338 0.904 2.29 -5.007 ...
## $ : num [1:55] -8.61 -4.21 6.22 4.39 -9.99 ...
## $ : num [1:60] 9 -11.01 2.2 -5.56 -3.4 ...
## $ : num [1:65] -4.508 -2.29 -6.871 -0.758 -13.045 ...
## $ : num [1:70] 17.33 -9.2 -14.61 -12.83 6.87 ...
## $ : num [1:75] 8.72 -6.16 -4.67 2.92 -16 ...
## $ : num [1:80] -11.702 18.432 3.928 -13.13 0.929 ...
## $ : num [1:85] -13.439 -5.278 -0.867 -8.354 14.282 ...
## $ : num [1:90] 0.772 1.154 -19.369 8.707 1.572 ...
## $ : num [1:95] -3.095 0.157 -25.479 -3.716 13.151 ...
## $ : num [1:100] 2.3331 0.0165 -8.1208 -4.2457 5.284 ...
```

More powerful

Add it as a list column!

```
sim2 <- conditions %>%  
  as_tibble() %>% # Not required, but definitely helpful  
  mutate(sim = map2(n, mu, ~rnorm(.x, .y, sd = 10)))  
sim2
```

```
## # A tibble: 510 x 3  
##       n     mu sim  
##   <dbl> <dbl> <list>  
## 1      5     -2 <dbl [5]>  
## 2     10     -2 <dbl [10]>  
## 3     15     -2 <dbl [15]>  
## 4     20     -2 <dbl [20]>  
## 5     25     -2 <dbl [25]>  
## 6     30     -2 <dbl [30]>  
## 7     35     -2 <dbl [35]>  
## 8     40     -2 <dbl [40]>  
## 9     45     -2 <dbl [45]>  
## 10    50     -2 <dbl [50]>  
## # ... with 500 more rows
```

Unnest

```
conditions %>%  
  as_tibble() %>% # Not required, but definitely helpful  
  mutate(sim = map2(n, mu, ~rnorm(.x, .y, sd = 10))) %>%  
  unnest()
```

```
## # A tibble: 39,525 x 3  
##       n      mu      sim  
##   <dbl> <dbl>   <dbl>  
## 1     5    -2 -6.799818  
## 2     5    -2 -11.82891  
## 3     5    -2 -1.997035  
## 4     5    -2 -5.874381  
## 5     5    -2 -7.569429  
## 6    10    -2 -16.78940  
## 7    10    -2 -14.90955  
## 8    10    -2 -13.69314  
## 9    10    -2 -1.759368  
## 10   10    -2 -9.681130  
## # ... with 39,515 more rows
```

What if we wanted to vary the sd
too?

What if we wanted to vary the sd too?

pmap

Which we'll get to soon

Varying the title of a plot

The data

```
library(fivethirtyeight)
pulitzer
```

```
## # A tibble: 50 x 7
##   newspaper circ2004 circ2013 pctchg_circ num_finals1990_2003
##   <chr>         <dbl>    <dbl>         <int>         <int>
## 1 USA Today    2192098  1674306         -24             1
## 2 Wall Str...  2101017  2378827          13            30
## 3 New York...  1119027  1865318          67            55
## 4 Los Ange...   983727   653868         -34            44
## 5 Washingt...   760034   474767         -38            52
## 6 New York...   712671   516165         -28             4
## 7 New York...   642844   500521         -22             0
## 8 Chicago ...   603315   414930         -31            23
## 9 San Jose...   558874   583998           4             4
## 10 Newsday      553117   377744         -32            12
## # ... with 40 more rows, and 2 more variables: num_finals2004_2014 <int>,
## #   num_finals1990_2014 <int>
```

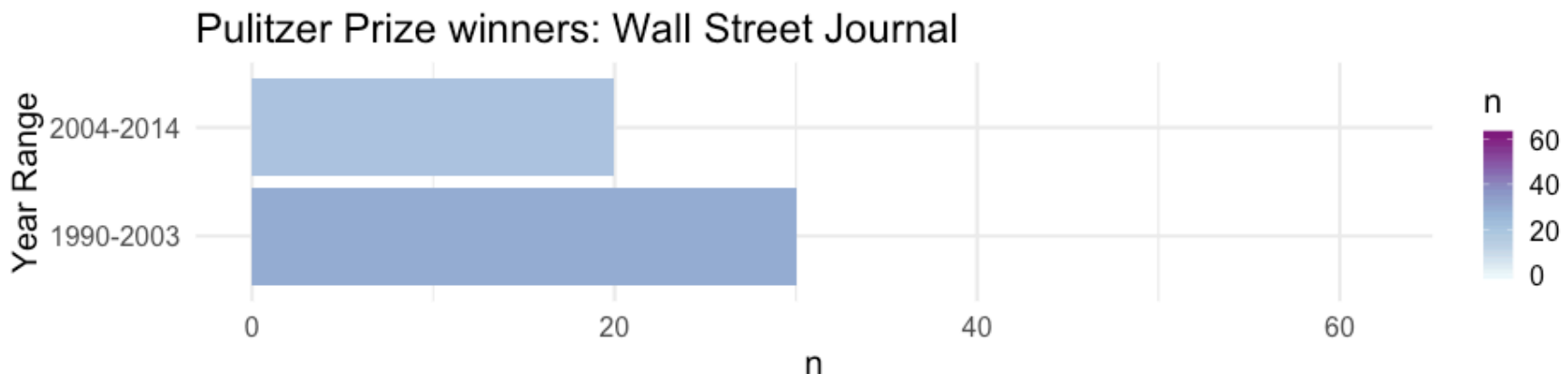

Prep data

```
pulitzer <- pulitzer %>%  
  select(newspaper, starts_with("num")) %>%  
  gather(year_range, n, -newspaper) %>%  
  mutate(year_range = str_replace_all(year_range, "num_finals", ""),  
         year_range = str_replace_all(year_range, "_", "-")) %>%  
  filter(year_range != "1990-2014")  
pulitzer
```

```
## # A tibble: 100 x 3  
##   newspaper      year_range      n  
##   <chr>         <chr>    <int>  
## 1 USA Today      1990-2003      1  
## 2 Wall Street Journal 1990-2003     30  
## 3 New York Times    1990-2003     55  
## 4 Los Angeles Times 1990-2003     44  
## 5 Washington Post   1990-2003     52  
## 6 New York Daily News 1990-2003      4  
## 7 New York Post      1990-2003      0  
## 8 Chicago Tribune    1990-2003     23  
## 9 San Jose Mercury News 1990-2003      4  
## 10 Newsday           1990-2003     12  
## # ... with 90 more rows
```

One plot

```
pulitzer %>%  
  filter(newspaper == "Wall Street Journal") %>%  
  ggplot(aes(year_range, n)) +  
    geom_col(aes(fill = n)) +  
    scale_fill_distiller(type = "seq",  
                        limits = c(0, max(pulitzer$n)),  
                        palette = "BuPu",  
                        direction = 1) +  
    ylim(0, max(pulitzer$n)) +  
    coord_flip() +  
    labs(title = "Pulitzer Prize winners: Wall Street Journal",  
         x = "Year Range")
```



Nest data

```
pulitzer %>%  
  group_by(newspaper) %>%  
  nest()
```

```
## # A tibble: 50 x 2  
##   newspaper      data  
##   <chr>         <list>  
## 1 USA Today     <tibble [2 x 2]>  
## 2 Wall Street Journal <tibble [2 x 2]>  
## 3 New York Times <tibble [2 x 2]>  
## 4 Los Angeles Times <tibble [2 x 2]>  
## 5 Washington Post <tibble [2 x 2]>  
## 6 New York Daily News <tibble [2 x 2]>  
## 7 New York Post   <tibble [2 x 2]>  
## 8 Chicago Tribune <tibble [2 x 2]>  
## 9 San Jose Mercury News <tibble [2 x 2]>  
## 10 Newsday        <tibble [2 x 2]>  
## # ... with 40 more rows
```

Produce all plots

You try first!

Don't worry about the correct title yet, if you don't want

Produce all plots

You try first!

Don't worry about the correct title yet, if you don't want

```
pulitzer %>%
  group_by(newspaper) %>%
  nest() %>%
  mutate(plot = map(data, ~ggplot(.x, aes(year_range, n)) +
    geom_col(aes(fill = n)) +
    scale_fill_distiller(type = "seq",
      limits = c(0, max(pulitzer$n)),
      palette = "BuPu",
      direction = 1) +
    ylim(0, max(pulitzer$n)) +
    coord_flip() +
    labs(title = "Pulitzer Prize winners")))
```

```
## # A tibble: 50 x 3
##   newspaper      data      plot
##   <chr>         <list>    <list>
## 1 USA Today    <tibble [2 x 2]> <S3: gg>
## 2 Wall Street Journal <tibble [2 x 2]> <S3: gg>
## 3 New York Times <tibble [2 x 2]> <S3: gg>
```

Add title

```
library(glue)

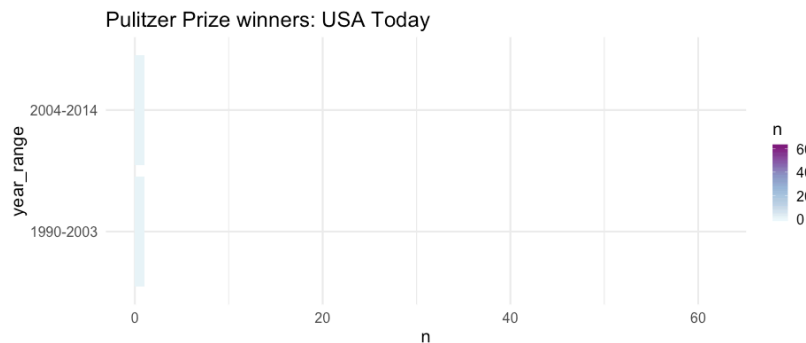
p <- pulitzer %>%
  group_by(newspaper) %>%
  nest() %>%
  mutate(plot = map2(data, newspaper, ~
    ggplot(.x, aes(year_range, n)) +
    geom_col(aes(fill = n)) +
    scale_fill_distiller(type = "seq",
                        limits = c(0, max(pulitzer$n)),
                        palette = "BuPu",
                        direction = 1) +
    ylim(0, max(pulitzer$n)) +
    coord_flip() +
    labs(title = glue("Pulitzer Prize winners: {.y}")))))
```

p

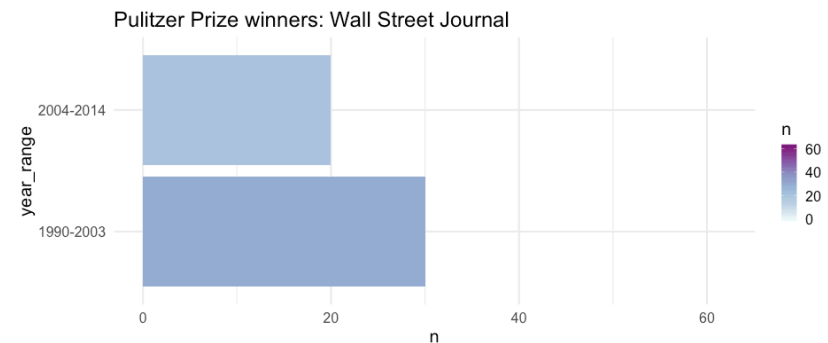
```
## # A tibble: 50 x 3
##   newspaper      data      plot
##   <chr>         <list>    <list>
## 1 USA Today    <tibble [2 x 2]> <S3: gg>
```

Look at a couple plots

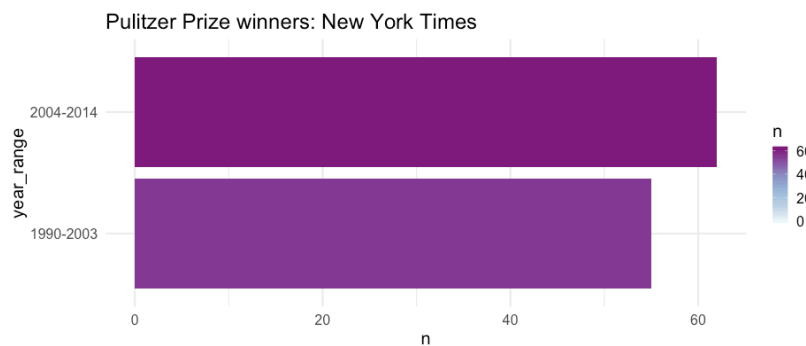
```
p$plot[[1]]
```



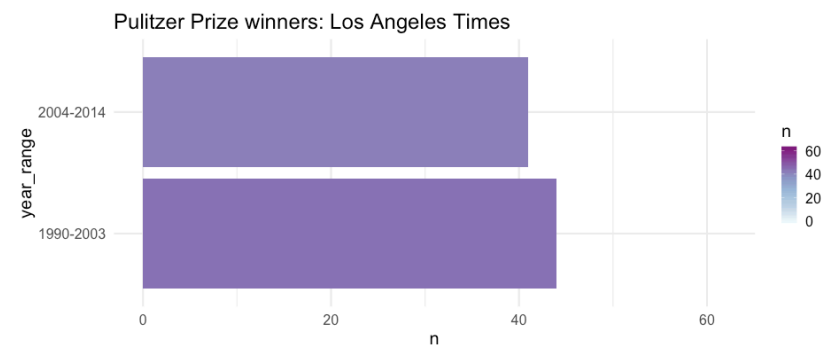
```
p$plot[[2]]
```



```
p$plot[[3]]
```

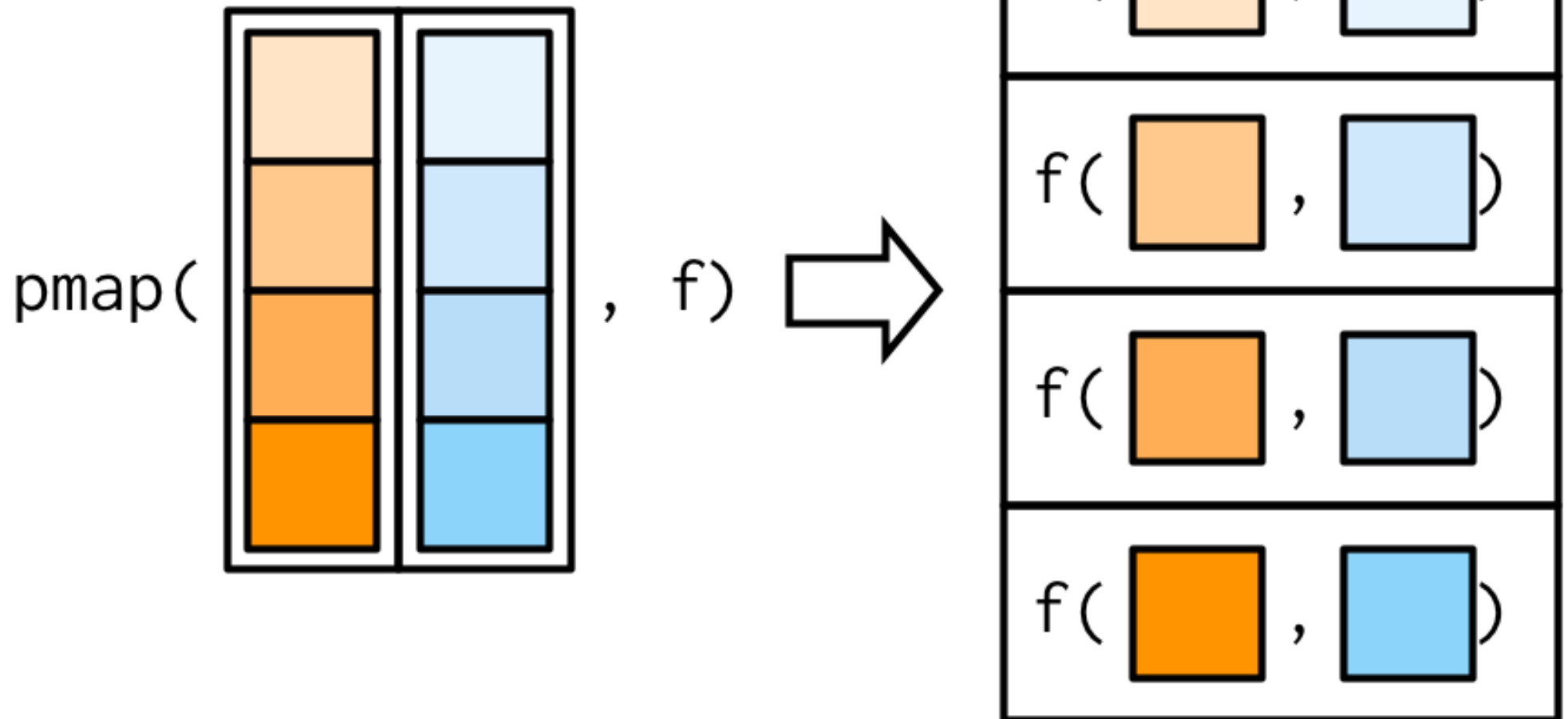


```
p$plot[[4]]
```



iterating over n vectors
`pmap`

pmap



Simulation

- Simulate data from a normal distribution
 - Vary n from 5 to 150 by increments of 5
 - For each n , vary μ from -2 to 2 by increments of 0.25
 - For each σ from 1 to 3 by increments of 0.1

```
full_conditions <- expand.grid(n = seq(5, 150, 5),  
                               mu = seq(-2, 2, 0.25),  
                               sd = seq(1, 3, .1))  
  
head(full_conditions)
```

```
##      n mu sd  
## 1   5 -2  1  
## 2  10 -2  1  
## 3  15 -2  1  
## 4  20 -2  1  
## 5  25 -2  1  
## 6  30 -2  1
```

```
tail(full_conditions)
```

```
##           n mu sd  
## 10705 125  2  3  
## 10706 130  2  3  
## 10707 135  2  3  
## 10708 140  2  3  
## 10709 145  2  3  
## 10710 150  2  3
```

Full Simulation

```
fsim <- pmap(list(number = full_conditions$n,  
                  average = full_conditions$mu,  
                  stdev = full_conditions$sd),  
             function(number, average, stdev) {  
               rnorm(n = number, mean = average, sd = stdev)  
             })  
str(fsim)
```

```
## List of 10710  
## $ : num [1:5] -0.68509 -0.00408 -2.12953 -3.06301 -1.7573  
## $ : num [1:10] -1.043 -2.461 -2.128 -0.523 -2 ...  
## $ : num [1:15] -2.82 -1.555 -1.828 -0.666 -2.829 ...  
## $ : num [1:20] -4.17 -1.3 -2.86 -1.99 -4.05 ...  
## $ : num [1:25] -1.8358 0.0138 -2.1699 -3.817 -2.1927 ...  
## $ : num [1:30] -1.739 -0.773 -2.072 -0.425 -1.092 ...  
## $ : num [1:35] -2.42 -3.08 -2.65 -2.21 -1.99 ...  
## $ : num [1:40] -1.551 -1.527 0.317 -0.25 -1.227 ...  
## $ : num [1:45] -2.44 -1.29 -1.56 -2.49 -3.1 ...  
## $ : num [1:50] -1.78 -2.26 -2.56 -3.54 -1.67 ...  
## $ : num [1:55] -1.937 -1.24 -0.9 -0.597 -2.127 ...  
## $ : num [1:60] -2.79 -2.3 -1.01 -2.44 -2.44 ...  
## $ : num [1:65] -2.8 -1.74 -2.33 -1.52 -1.08 ...  
## $ : num [1:70] -2.1 -3.04 -2.81 -2.45 -1.15 ...  
## $ : num [1:75] -1.717 -2.612 0.117 -1.497 -2.025 ...
```

Alternative spec

```
fsim <- pmap(list(full_conditions$n,  
                  full_conditions$mu,  
                  full_conditions$sd),  
             ~rnorm(n = ..1, mean = ..2, sd = ..3))  
str(fsim)
```

```
## List of 10710  
## $ : num [1:5] -1.732 -4.046 -0.882 -1.318 0.102  
## $ : num [1:10] -2.59 -2.39 -1.57 -1.56 -3.3 ...  
## $ : num [1:15] -0.849 -1.381 -1.831 -1.879 -1.121 ...  
## $ : num [1:20] -2.345 -0.497 -0.884 -1.316 -3.192 ...  
## $ : num [1:25] -0.3437 -1.6887 -2.7514 0.0419 -1.5099 ...  
## $ : num [1:30] -2.043 -0.678 -0.745 -1.69 -1.088 ...  
## $ : num [1:35] -2.65 -1.48 -1.65 -2.84 -2.23 ...  
## $ : num [1:40] -2.85 -2.425 -1.35 -0.432 -3.335 ...  
## $ : num [1:45] -2.91 -3.59 -2.7 -1.84 -1.56 ...  
## $ : num [1:50] -2.557 -2.953 -0.053 -1.976 -1.742 ...  
## $ : num [1:55] -2.66 -1.94 -0.917 -2.746 -1.57 ...  
## $ : num [1:60] -1.05 -2.824 -2.057 -0.907 -2.253 ...  
## $ : num [1:65] -2.7 -2.18 -1.11 -1.32 -1.28 ...  
## $ : num [1:70] -3.48 -3.93 -1.51 -2.56 -1.46 ...  
## $ : num [1:75] -1.277 0.255 -1.15 -2.819 -1.231 ...  
## $ : num [1:80] -2.243 -2.165 -0.542 -2.482 -1.958 ...  
## $ : num [1:85] -1.601 -0.107 -0.9 -0.619 -2.8 ...
```

Simpler

Maybe a little too clever

- A data frame is a list so...

```
fsim <- pmap(full_conditions, ~rnorm(n = ..1,  
                                     mean = ..2,  
                                     sd = ..3))  
  
str(fsim)
```

```
## List of 10710  
## $ : num [1:5] -2.67 -1.71 -3.72 -1.05 -2.57  
## $ : num [1:10] -1.88 -1.52 -1.98 -1.13 -1.28 ...  
## $ : num [1:15] -2.022 -2.413 -0.846 -2.005 -0.864 ...  
## $ : num [1:20] -2.945 -3.135 -2.305 -2.507 -0.111 ...  
## $ : num [1:25] -2.114 -1.132 -3.318 -0.686 -2.381 ...  
## $ : num [1:30] -2.08 -3.18 -1.35 -1.13 -4.8 ...  
## $ : num [1:35] -2.13 -1.2 -2.34 -1.93 -2.08 ...  
## $ : num [1:40] -1.3611 -3.8349 -2.3402 -1.12 -0.0739 ...  
## $ : num [1:45] -1.547 -0.761 -2.548 -2.908 -0.806 ...  
## $ : num [1:50] -2.01 -1.32 -1.51 -1.96 -1.19 ...  
## $ : num [1:55] -3.077 -2.245 -2.603 -0.235 -1.557 ...  
## $ : num [1:60] -2.68 -1.55 -3.45 -2.22 -2.24 ...  
## $ : num [1:65] -3.09 -3.21 -3.11 -2.58 -1.64 ...
```

List column version

```
full_conditions %>%  
  as_tibble() %>%  
  mutate(sim = pmap(list(n, mu, sd), ~rnorm(..1, ..2, ..3)))
```

```
## # A tibble: 10,710 x 4  
##       n      mu      sd sim  
##   <dbl> <dbl> <dbl> <list>  
## 1      5     -2      1 <dbl [5]>  
## 2     10     -2      1 <dbl [10]>  
## 3     15     -2      1 <dbl [15]>  
## 4     20     -2      1 <dbl [20]>  
## 5     25     -2      1 <dbl [25]>  
## 6     30     -2      1 <dbl [30]>  
## 7     35     -2      1 <dbl [35]>  
## 8     40     -2      1 <dbl [40]>  
## 9     45     -2      1 <dbl [45]>  
## 10    50     -2      1 <dbl [50]>  
## # ... with 10,700 more rows
```

Unnest

```
full_conditions %>%  
  as_tibble() %>%  
  mutate(sim = pmap(list(n, mu, sd), ~rnorm(..1, ..2, ..3))) %>%  
  unnest()
```

```
## # A tibble: 830,025 x 4  
##       n      mu    sd      sim  
##   <dbl> <dbl> <dbl>   <dbl>  
## 1     5     -2     1 -1.152654  
## 2     5     -2     1 -2.468304  
## 3     5     -2     1 -0.5156141  
## 4     5     -2     1 -1.163307  
## 5     5     -2     1 -2.303434  
## 6    10     -2     1 -0.3618123  
## 7    10     -2     1 -3.325788  
## 8    10     -2     1 -0.7039727  
## 9    10     -2     1 -1.897031  
## 10   10     -2     1  1.001286  
## # ... with 830,015 more rows
```


Plot

Add a caption stating the total number of pulizter prize winners across years

Add column for total

```
pulitzer <- pulitzer %>%  
  group_by(newspaper) %>%  
  mutate(tot = sum(n))  
pulitzer
```

```
## # A tibble: 100 x 4  
## # Groups:   newspaper [50]  
##   newspaper      year_range      n    tot  
##   <chr>          <chr>    <int> <int>  
## 1 USA Today      1990-2003      1      2  
## 2 Wall Street Journal 1990-2003     30     50  
## 3 New York Times    1990-2003     55    117  
## 4 Los Angeles Times 1990-2003     44     85  
## 5 Washington Post   1990-2003     52    100  
## 6 New York Daily News 1990-2003      4      6  
## 7 New York Post      1990-2003      0      0  
## 8 Chicago Tribune    1990-2003     23     38  
## 9 San Jose Mercury News 1990-2003      4      6  
## 10 Newsday           1990-2003     12     18  
## # ... with 90 more rows
```

Easiest way (imo)

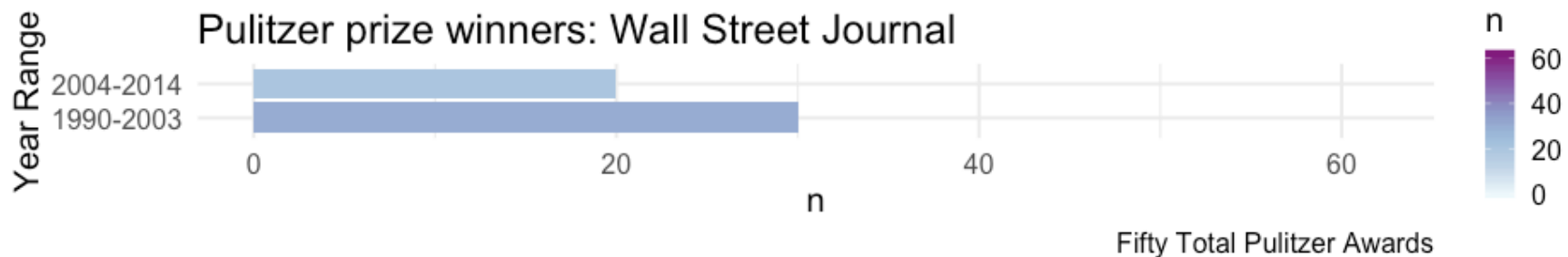
Create a column to represent exactly the label you want.

```
#install.packages("english")
library(english)
pulitzer <- pulitzer %>%
  mutate(label =
    glue("{str_to_title(as.english(tot))} Total Pulitzer Awards"))
pulitzer
```

```
## # A tibble: 100 x 5
## # Groups:   newspaper [50]
##   newspaper      year_range      n    tot label
##   <chr>          <chr>    <int> <int> <chr>
## 1 USA Today      1990-2003      1      2 Two Total Pulitzer Awards
## 2 Wall Street Journ... 1990-2003     30     50 Fifty Total Pulitzer Awards
## 3 New York Times    1990-2003     55    117 One Hundred Seventeen Total P...
## 4 Los Angeles Times  1990-2003     44     85 Eighty Five Total Pulitzer Aw...
## 5 Washington Post   1990-2003     52    100 One Hundred Total Pulitzer Aw...
## 6 New York Daily Ne... 1990-2003      4      6 Six Total Pulitzer Awards
## 7 New York Post      1990-2003      0      0 Zero Total Pulitzer Awards
## 8 Chicago Tribune    1990-2003     23     38 Thirty Eight Total Pulitzer A...
## 9 San Jose Mercury ... 1990-2003      4      6 Six Total Pulitzer Awards
## 10 Newsday           1990-2003     12     18 Eighteen Total Pulitzer Awards
```

Produce one plot

```
tmp <- pulitzer %>%  
  filter(newspaper == "Wall Street Journal")  
  
ggplot(tmp, aes(year_range, n)) +  
  geom_col(aes(fill = n)) +  
  scale_fill_distiller(type = "seq",  
                        limits = c(0, max(pulitzer$n)),  
                        palette = "BuPu",  
                        direction = 1) +  
  ylim(0, max(pulitzer$n)) +  
  coord_flip() +  
  labs(title = "Pulitzer prize winners: Wall Street Journal",  
        x = "Year Range",  
        caption = unique(tmp$label))
```



Produce all plots

Nest first

```
pulitzer %>%  
  group_by(newspaper, label) %>%  
  nest()
```

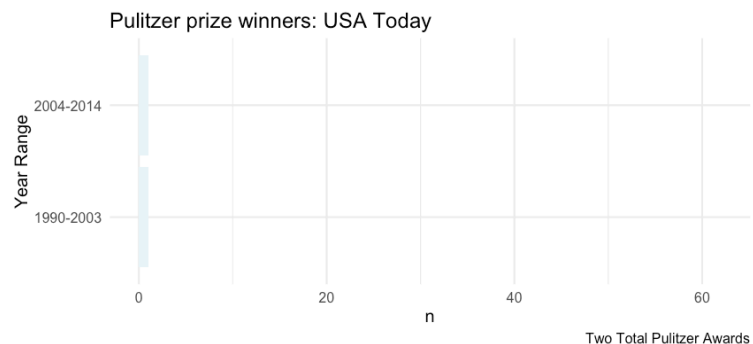
```
## # A tibble: 50 x 3  
##   newspaper      label      data  
##   <chr>         <chr>    <list>  
## 1 USA Today      Two Total Pulitzer Awards    <tibble [2 x ...  
## 2 Wall Street Journal Fifty Total Pulitzer Awards  <tibble [2 x ...  
## 3 New York Times  One Hundred Seventeen Total Pulitze... <tibble [2 x ...  
## 4 Los Angeles Times Eighty Five Total Pulitzer Awards  <tibble [2 x ...  
## 5 Washington Post  One Hundred Total Pulitzer Awards  <tibble [2 x ...  
## 6 New York Daily News Six Total Pulitzer Awards    <tibble [2 x ...  
## 7 New York Post    Zero Total Pulitzer Awards    <tibble [2 x ...  
## 8 Chicago Tribune  Thirty Eight Total Pulitzer Awards <tibble [2 x ...  
## 9 San Jose Mercury Ne... Six Total Pulitzer Awards    <tibble [2 x ...  
## 10 Newsday         Eighteen Total Pulitzer Awards  <tibble [2 x ...  
## # ... with 40 more rows
```

Produce plots

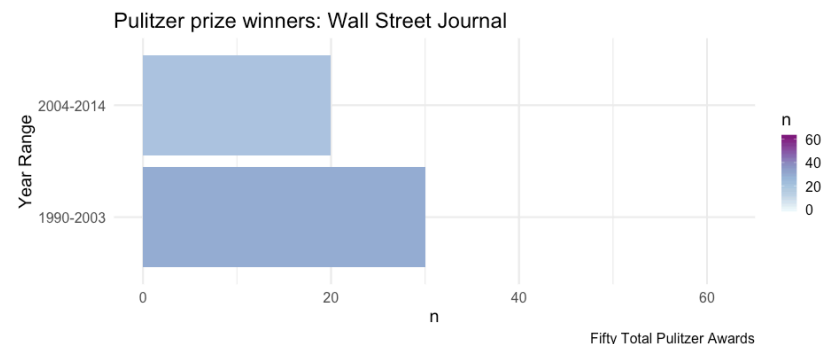
```
final_plots <- pulitzer %>%  
  group_by(newspaper, label) %>%  
  nest() %>%  
  mutate(plots = pmap(list(newspaper, label, data),  
    ~ggplot(..3, aes(year_range, n)) +  
      geom_col(aes(fill = n)) +  
      scale_fill_distiller(type = "seq",  
        limits = c(0, max(pulitzer$n)),  
        palette = "BuPu",  
        direction = 1) +  
      ylim(0, max(pulitzer$n)) +  
      coord_flip() +  
      labs(title = glue("Pulitzer prize winners: {..1}"),  
        x = "Year Range",  
        caption = ..2)))
```

Look at a couple plots

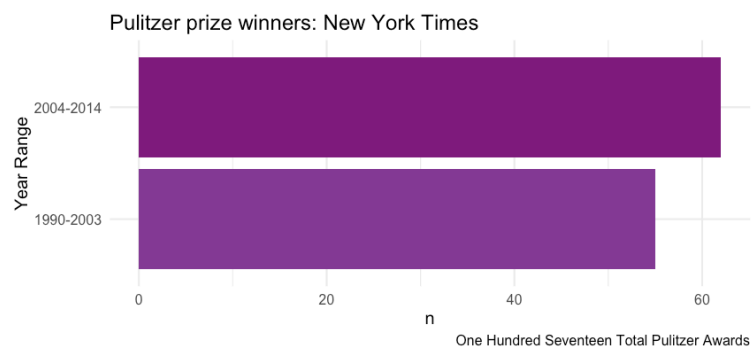
```
final_plots$plots[[1]]
```



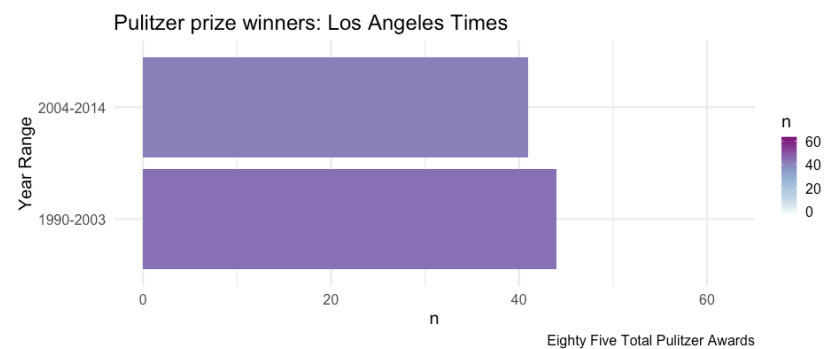
```
final_plots$plots[[2]]
```



```
final_plots$plots[[3]]
```



```
final_plots$plots[[4]]
```



Save all plots

We'll have to iterate across at least two things: (a) file path/names, and (b) the plots themselves

walk

- Why `walk` for saving instead of `map`?

Walk is an alternative to map that you use when you want to call a function for its side effects, rather than for its return value. You typically do this because you want to render output to the screen or save files to disk - the important thing is the action, not the return value.

-r4ds

Example

Create a directory

```
fs::dir_create(here::here("plots", "pulitzers"))
```

Example

Create a directory

```
fs::dir_create(here::here("plots", "pulitzers"))
```

Create file paths

```
files <- str_replace_all(tolower(final_plots$newspaper), " ", "-")  
paths <- here::here("plots", "pulitzers", glue("{files}.png"))  
paths
```

```
## [1] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/plots/pul  
## [2] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/plots/pul  
## [3] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/plots/pul  
## [4] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/plots/pul  
## [5] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/plots/pul  
## [6] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/plots/pul  
## [7] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/plots/pul  
## [8] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/plots/pul  
## [9] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/plots/pul  
## [10] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/plots/pul  
## [11] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/plots/pul  
## [12] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/plots/pul
```

Save plots

```
walk2(paths, final_plots$plots, ggsave,  
      width = 9.5,  
      height = 6.5,  
      dpi = 500)
```

Wrap-up

- Lots more to `{purrr}` we weren't able to get to
 - May be interested in `safely`, `reduce`, `modify`, and `pull` and `pluck`
- Function programming can *really* help your efficiency, and even if it slows you down initially, I'd recommend striving toward it, because it will ultimately be a huge help.

Questions?