Parallel Iterations

Daniel Anderson

Week 5, Class 1

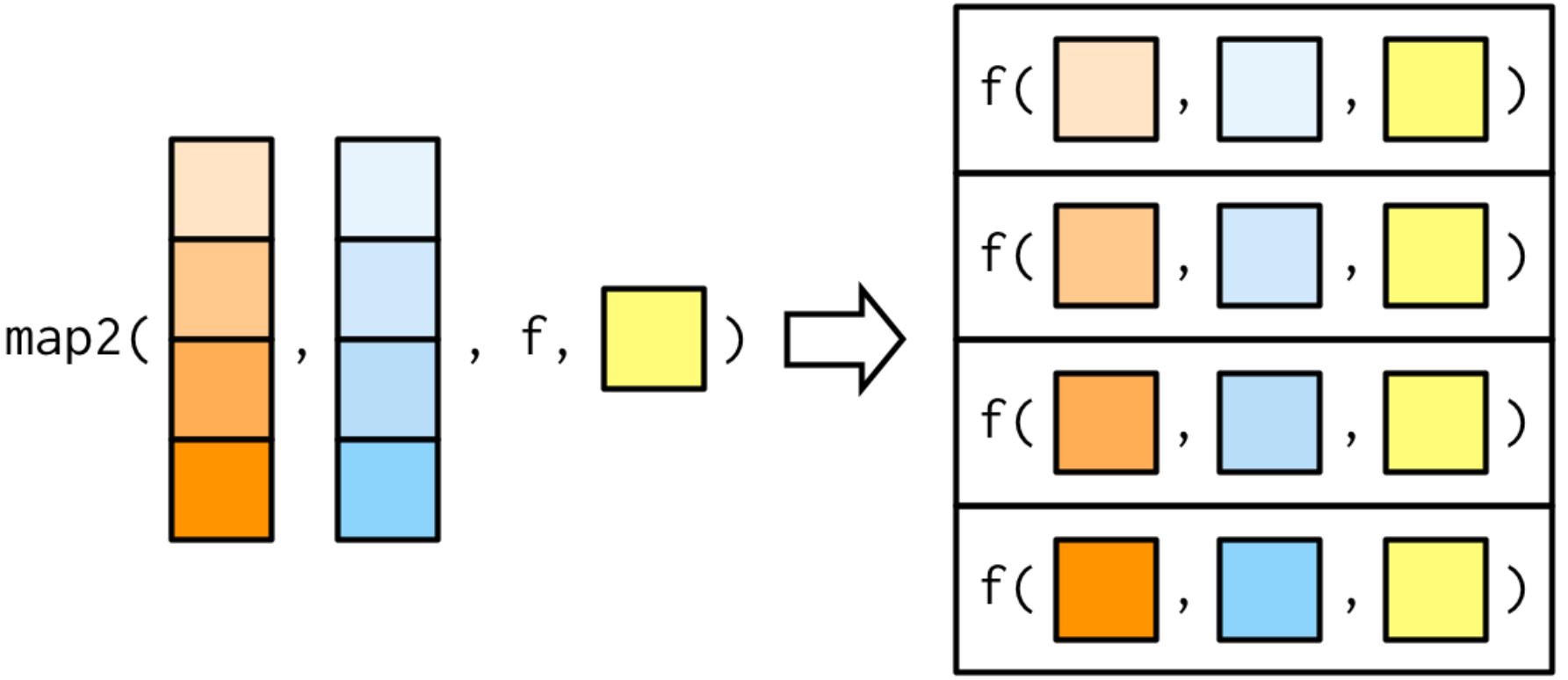
Agenda

- Finish up slides from last week
- Discuss `map2_*` and `pmap_*`

Learning objectives

- Understand the differences between `map`, `map2`, and `pmap`

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 mu 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

Please follow along

```
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(n = .x, mean = .y, sd = 10)
})
str(sim1)
```

```
## List of 510
##  $ : num [1:5] -8.89 11.59 -6.16 -7.1 3.81
##  $ : num [1:10] -5.967 -14.048 -0.242 -17.481 8.045 ...
##  $ : num [1:15] 5.103 -0.816 -5.772 -13.166 10.696 ...
##  $ : num [1:20] 3.3 -14.74 -4.39 7.39 -20.82 ...
##  $ : num [1:25] -5.64 17.8 -12.89 1.92 -1.03 ...
##  $ : num [1:30] -4.47 7.7 3.9 -3.88 10.41 ...
##  $ : num [1:35] 11.3211 5.2912 -0.0202 0.7237 2.5658 ...
##  $ : num [1:40] -5.36 14.02 1.73 -3.9 -6.73 ...
##  $ : num [1:45] 1.32 13.89 19.73 4.62 -16.47 ...
##  $ : num [1:50] -19.54 -1.02 -6.72 -8.03 12.07 ...
##  $ : num [1:55] -3.96 -2.31 -6.94 17.01 6.67 ...
##  $ : num [1:60] 0.169 6.871 -0.963 1.925 -6.045 ...
##  $ : num [1:65] 10.64 -16.511 -0.575 -6.33 -15.544 ...
##  $ : num [1:70] 4.84 -1.43 9.39 9.09 22.67 ...
##  $ : num [1:75] -7.264 0.494 -10.131 -15.417 -12.42 ...
##  $ : num [1:80] -4.49 5.16 -16.3 3.68 -5.99 ...
##  $ : num [1:85] -7 20.57 -3.59 -10.93 -6.08 ...
##  $ : num [1:90] 1.27 -4.96 7.39 10.66 -1.2 ...
##  $ : num [1:95] -0.562 -16.97 10.457 -0.644 -1.726 ...
##  $ : num [1:100] -7.589 0.757 -7.241 -19.506 -1.601 ...
```

More powerful

Add it as a list column!

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

```
## # A tibble: 39,525 x 3
##        n    mu          sim
##    <dbl> <dbl>        <dbl>
##  1     5    -2   7.570682
##  2     5    -2  -2.268110
##  3     5    -2 -11.52502
##  4     5    -2  -5.372831
##  5     5    -2  -4.913946
##  6    10    -2   0.5334139
##  7    10    -2   6.504093
##  8    10    -2 -20.98906
##  9    10    -2  13.81959
## 10    10    -2 -16.19086
## # … with 39,515 more rows
```

Challenge

Can you replicate what we just did, but using a `rowwise()` approach?

```
conditions %>%
  rowwise() %>%
  mutate(sim = list(rnorm(n, mu, sd = 10))) %>%
  unnest(sim)
```

```
## # A tibble: 39,525 x 3
##         n    mu        sim
##     <dbl> <dbl>      <dbl>
##  1      5    -2  13.95355
##  2      5    -2  31.02931
##  3      5    -2  -2.671605
##  4      5    -2  -9.867195
##  5      5    -2  -2.897235
##  6     10    -2 -19.70467
##  7     10    -2  10.80861
##  8     10    -2   7.158649
##  9     10    -2   2.255881
## 10     10    -2   5.924527
## # … with 39,515 more rows
```

03:00

Vary the sd too?

---

pmap

Which we'll get to soon

Varying the title of a plot

The data

Please follow along

```
library(fivethirtyeight)
pulitzer
```

```
## # A tibble: 50 x 7
##    newspaper             circ2004 circ2013 pctchg_circ num_finals1990_20
##    <chr>                    <dbl>    <dbl>       <int>               <in
##  1 USA Today              2192098  1674306         -24
##  2 Wall Street Journal    2101017  2378827          13
##  3 New York Times         1119027  1865318          67
##  4 Los Angeles Times       983727   653868         -34
##  5 Washington Post         760034   474767         -38
##  6 New York Daily News     712671   516165         -28
##  7 New York Post           642844   500521         -22
##  8 Chicago Tribune         603315   414930         -31
##  9 San Jose Mercury News   558874   583998           4
## 10 Newsday                 553117   377744         -32
## # … with 40 more rows, and 1 more variable: num_finals1990_2014 <int>
```

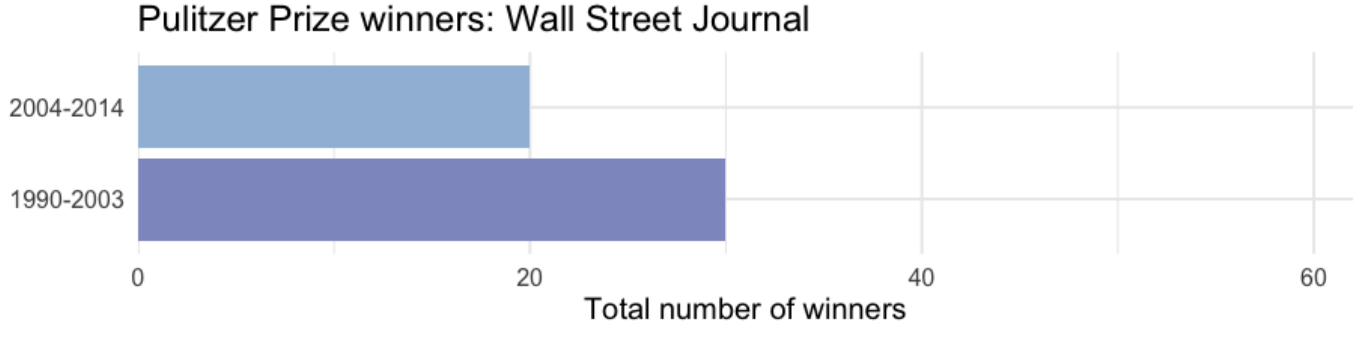
Prep data

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

head(pulitzer)
```

```
## # A tibble: 6 x 3
##   newspaper           year_range     n
##   <chr>               <chr>      <int>
## 1 USA Today           1990-2003      1
## 2 USA Today           2004-2014      1
## 3 Wall Street Journal 1990-2003     30
## 4 Wall Street Journal 2004-2014     20
## 5 New York Times      1990-2003     55
## 6 New York Times      2004-2014     62
```

One plot

```
pulitzer%>%
    filter(newspaper == "Wall Street Journal") %>%
  ggplot(aes(n, year_range)) +
  geom_col(aes(fill = n)) +
  scale_fill_distiller(type = "seq",
                       limits = c(0, max(pulitzer$n)),
                       palette = "BuPu",
                       direction = 1) +
  scale_x_continuous(limits = c(0, max(pulitzer$n)),
                     expand = c(0, 0)) +
  guides(fill = "none") +
  labs(title = "Pulitzer Prize winners: Wall Street Journal",
       x = "Total number of winners",
       y = "")
```



Nest data

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

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

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(aes(n, year_range)) +
      geom_col(aes(fill = n)) +
      scale_fill_distiller(type = "seq",
                           limits = c(0, max(pulitzer$n)),
                           palette = "BuPu",
                           direction = 1) +
      scale_x_continuous(limits = c(0, max(pulitzer$n)),
                         expand = c(0, 0)) +
      guides(fill = "none") +
      labs(title = "Pulitzer Prize winners",
           x = "Total number of winners",
           y = "")
    })
```

Add title

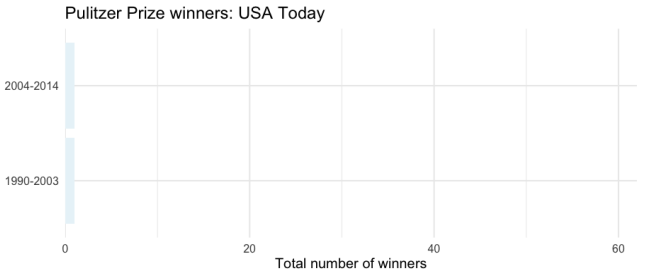
```
library(glue)

p <- pulitzer%>%
    group_by(newspaper) %>%
    nest() %>%
mutate(plot = map2(data, newspaper, ~{
      ggplot(.x, aes(n, year_range)) +
      geom_col(aes(fill = n)) +
      scale_fill_distiller(type = "seq",
                           limits = c(0, max(pulitzer$n)),
                           palette = "BuPu",
                           direction = 1) +
      scale_x_continuous(limits = c(0, max(pulitzer$n)),
                         expand = c(0, 0)) +
      guides(fill = "none") +
      labs(title = glue("Pulitzer Prize winners: {.y}"),
           x = "Total number of winners",
           y = "")
    })
)
```

```
p
```

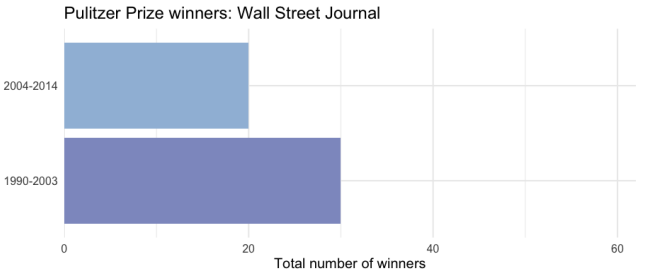
```
## # A tibble: 50 x 3
## # Groups:   newspaper [50]
##    newspaper             data                 plot
##    <chr>                 <list>               <list>
##  1 USA Today             <tibble[,2] [2 × 2]> <gg>
##  2 Wall Street Journal   <tibble[,2] [2 × 2]> <gg>
##  3 New York Times      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

Look at a couple plots

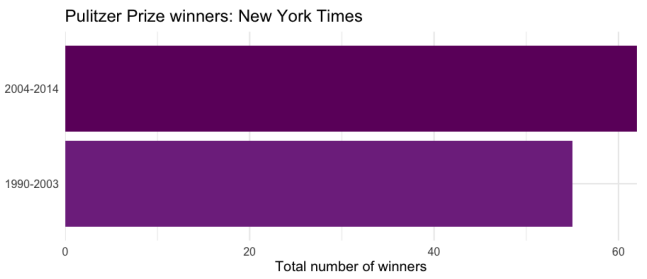
```
p$plot[[1]]
```



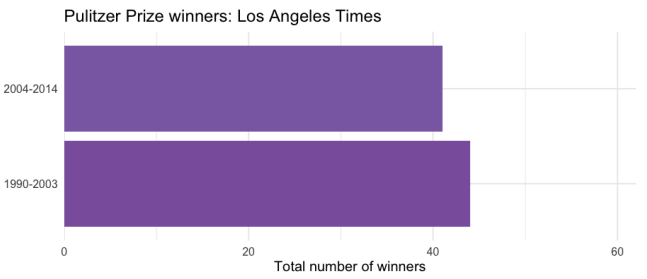

```
p$plot[[2]]
```




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




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



Challenge

(You can probably guess where this is going)

Can you reproduce the prior plots using a `rowwise()` approach?

```
pulitzer%>%
nest_by(newspaper) %>%
    mutate(
      plot = list(
     ggplot(data, aes(n, year_range)) +
        geom_col(aes(fill = n)) +
        scale_fill_distiller(type = "seq",
                             limits = c(0, max(pulitzer$n)),
                             palette = "BuPu",
                             direction = 1) +
        scale_x_continuous(limits = c(0, max(pulitzer$n)),
                           expand = c(0, 0)) +
        guides(fill = "none") +
        labs(title = glue("Pulitzer Prize winners: {newspaper}")
           x = "Total number of winners",
           y = "")
    )
)
```

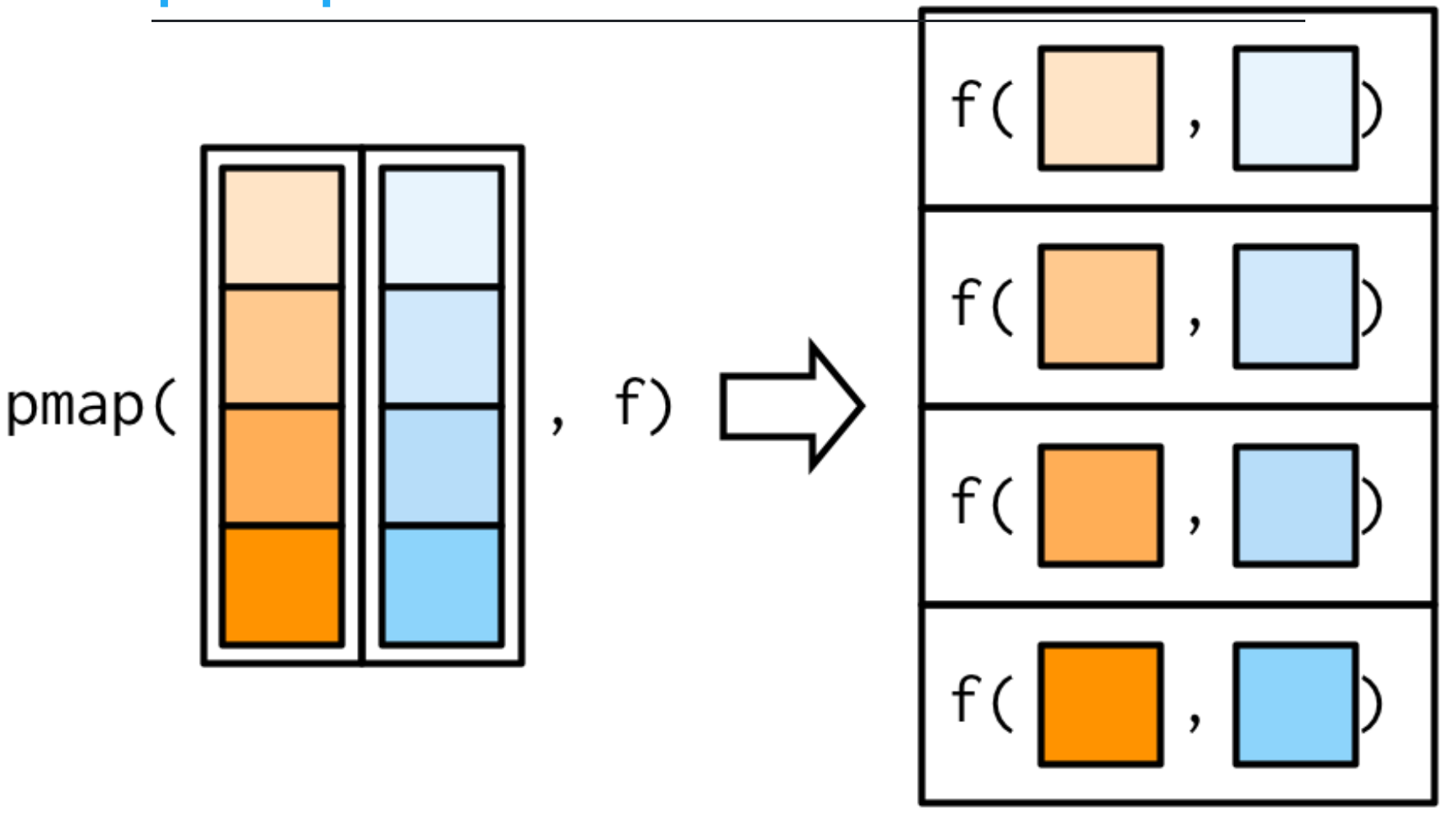
```
## # A tibble: 50 x 3
## # Rowwise:  newspaper
##    newspaper                                  data plot
##    <chr>                          <list<tibble[,2]>> <list>
##  1 Arizona Republic                       [2 × 2] <gg>
##  2 Atlanta Journal Constitution           [2 × 2] <gg>
##  3 Baltimore Sun                          [2 × 2] <gg>
##  4 Boston Globe                           [2 × 2] <gg>
##  5 Boston Herald                          [2 × 2] <gg>
```

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 mu 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] -1.536 -2.266 -1.205 -2.234 -0.944
##  $ : num [1:10] -0.841 -2.637 -2.427 -1.969 -2.696 ...
##  $ : num [1:15] -2.397 -1.581 -1.349 -1.856 0.587 ...
##  $ : num [1:20] -1.767 -3.026 -1.364 -2.747 -0.197 ...
##  $ : num [1:25] -2.56 -2.13 -1.33 -2.9 -3.86 ...
##  $ : num [1:30] -3.95 -1.88 -1.21 -1.5 -1.5 ...
##  $ : num [1:35] -1.56 -1.69 -4.14 -3.05 -1.93 ...
##  $ : num [1:40] -2.979 -1.432 -1.607 -1.905 -0.628 ...
##  $ : num [1:45] -1.829 -1.772 -1.432 -0.315 -2.345 ...
##  $ : num [1:50] -2.25 -1.93 -3.97 -1.59 -2.74 ...
##  $ : num [1:55] -3.562 -0.966 -1.718 -1.661 -1.809 ...
##  $ : num [1:60] -1.35 -1.15 -1.74 -2.3 -3.64 ...
##  $ : num [1:65] -2.47 -1.24 -2.43 -2.27 -2.34 ...
##  $ : num [1:70] -1.349 -2.941 -0.651 -2.054 -1.731 ...
```

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] -2.15 -2.45 -2.93 -3.36 -1.7
##  $ : num [1:10] -1.777 -0.882 -2.075 -2.826 -2.757 ...
##  $ : num [1:15] -1.52044 -1.45154 -0.30679 -0.16667 -0.00603 ...
##  $ : num [1:20] -0.627 -1.191 -0.71 -2.197 -1.363 ...
##  $ : num [1:25] -1.7 -1.83 -1.86 -2.42 -2.94 ...
##  $ : num [1:30] -1.32 -2.53 -1.6 -1.02 -3.23 ...
##  $ : num [1:35] 0.0321 -0.798 -1.0541 -1.5625 -0.2858 ...
##  $ : num [1:40] -2.44 -1.53 -2.03 -2.34 -2.6 ...
##  $ : num [1:45] -2.22 -2.16 -3.81 -2.88 -0.73 ...
##  $ : num [1:50] -1.81 -3.08 -1.15 -2.61 -2.3 ...
##  $ : num [1:55] -1.87 -2.224 -3.966 -0.403 -2.465 ...
##  $ : num [1:60] -2.86 -2.27 -2.95 -2.23 -3.99 ...
##  $ : num [1:65] 0.54 -1.2 -0.74 -2.49 -2.55 ...
##  $ : num [1:70] -0.342 -3.39 -2.711 0.766 -0.39 ...
##  $ : num [1:75] -1.81 -1.652 -2.138 -0.816 -2.304 ...
##  $ : num [1:80] -1.747 -2.63 -1.113 -2.78 -0.929 ...
##  $ : num [1:85] -2.21 -1.85 -2.64 -0.85 -1.15 ...
##  $ : num [1:90] -2.881 -0.366 -0.616 -2.621 -2.1 ...
##  $ : num [1:95] -0.416 -2.188 -2.981 -1.865 -2.759 ...
```

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] -2.28 -1.9 -2.7 -3.14 -1.78
##  $ : num [1:10] -1.21 -3.23 -0.97 -1.28 -2.06 ...
##  $ : num [1:15] -2.07 -1.24 -1.65 -0.27 -1.99 ...
##  $ : num [1:20] -1.772 -2.843 -0.871 -1.46 -0.763 ...
##  $ : num [1:25] 0.685 -0.119 -2.856 -0.688 -1.409 ...
##  $ : num [1:30] -1.743 -2.922 -1.887 -2.03 -0.667 ...
##  $ : num [1:35] -0.723 -1.828 -2.902 -0.284 -0.931 ...
##  $ : num [1:40] -2.258 -3.35 -0.359 -0.794 -1.552 ...
##  $ : num [1:45] -1.48 -1.92 -2.06 -1.99 -2.11 ...
##  $ : num [1:50] -1.77 -1.84 -1.22 -1.44 -1.53 ...
##  $ : num [1:55] -2.09 -2.6 -2.34 -3.12 -2.65 ...
##  $ : num [1:60] -0.874 -0.363 -1.584 -1.647 -0.697 ...
```

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(sim)
```

```
## # A tibble: 830,025 x 4
##        n    mu    sd        sim
##    <dbl> <dbl> <dbl>      <dbl>
##  1     5    -2     1 -0.9769357
##  2     5    -2     1 -1.026245
##  3     5    -2     1 -1.983785
##  4     5    -2     1 -1.262065
##  5     5    -2     1 -0.3616705
##  6    10    -2     1 -1.136676
##  7    10    -2     1 -1.665104
##  8    10    -2     1 -3.062858
##  9    10    -2     1 -4.412271
## 10    10    -2     1 -0.9440350
## # … with 830,015 more rows
```

Replicate with `nest_by()`

You try first

```
full_conditions %>%
  rowwise() %>%
  mutate(sim = list(rnorm(n, mu, sd))) %>%
  unnest(sim)
```

```
## # A tibble: 830,025 x 4
##         n    mu    sd        sim
##    <dbl> <dbl> <dbl>      <dbl>
##  1     5    -2     1 -4.246447
##  2     5    -2     1 -3.394033
##  3     5    -2     1 -2.688046
##  4     5    -2     1 -4.004150
##  5     5    -2     1 -2.636363
##  6    10    -2     1 -0.5402081
##  7    10    -2     1 -2.075118
##  8    10    -2     1 -1.927610
##  9    10    -2     1 -2.871188
## 10    10    -2     1 -0.9080542
## # … with 830,015 more rows
```

02:00

Plot

Add a caption stating the total number of Pulitzer 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 USA Today           2004-2014      1     2
##  3 Wall Street Journal 1990-2003     30    50
##  4 Wall Street Journal 2004-2014     20    50
##  5 New York Times      1990-2003     55   117
##  6 New York Times      2004-2014     62   117
##  7 Los Angeles Times   1990-2003     44    85
##  8 Los Angeles Times   2004-2014     41    85
##  9 Washington Post     1990-2003     52   100
## 10 Washington Post     2004-2014     48   100
## # … with 90 more rows
```

Easiest way (imo)

Create a column to represent exactly the label you want.

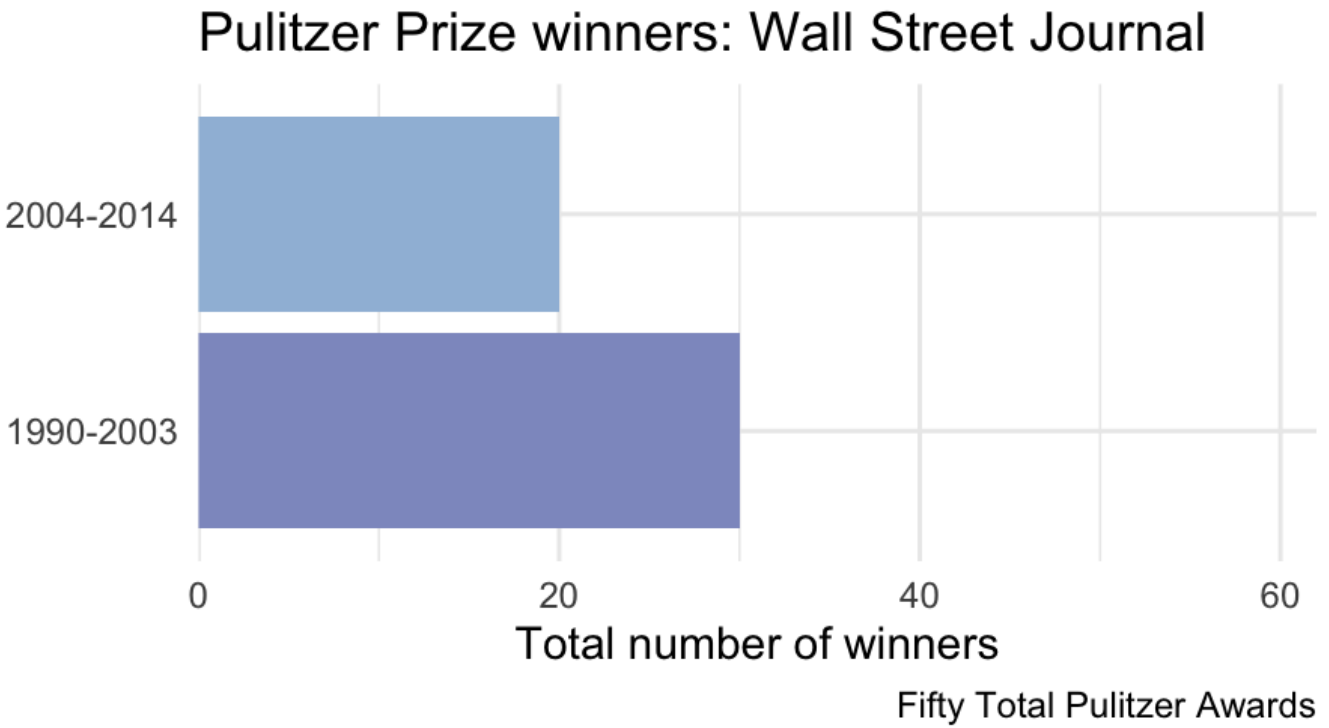
```
#install.packages("english")
library(english)
pulitzer<- pulitzer%>%
    mutate(
      label = glue(
        "{str_to_title(as.english(tot))} Total Pulitzer Awards"
    )
)
```

```
select(pulitzer, newspaper, label)
```

```
## # A tibble: 100 x 2
## # Groups:   newspaper [50]
##    newspaper           label
##    <chr>               <glue>
##  1 USA Today           Two Total Pulitzer Awards
##  2 USA Today           Two Total Pulitzer Awards
##  3 Wall Street Journal Fifty Total Pulitzer Awards
##  4 Wall Street Journal Fifty Total Pulitzer Awards
##  5 New York Times      One Hundred Seventeen Total Pulitzer Awards
##  6 New York Times      One Hundred Seventeen Total Pulitzer Awards
##  7 Los Angeles Times   Eighty-Five Total Pulitzer Awards
##  8 Los Angeles Times   Eighty-Five Total Pulitzer Awards
##  9 Washington Post     One Hundred Total Pulitzer Awards
## 10 Washington Post     One Hundred Total Pulitzer Awards
## # … with 90 more rows
```

Produce one plot

```
tmp <- pulitzer%>%
    filter(newspaper == "Wall Street Journal")

ggplot(tmp, aes(n, year_range)) +
  geom_col(aes(fill = n)) +
  scale_fill_distiller(type = "seq",
                       limits = c(0, max(pulitzer$n)),
                       palette = "BuPu",
                       direction = 1) +
  scale_x_continuous(limits = c(0, max(pulitzer$n)),
                     expand = c(0, 0)) +
  guides(fill = "none") +
  labs(
    title = glue("Pulitzer Prize winners: Wall Street Journal"),
    x = "Total number of winners",
    y = "",
    caption = unique(tmp$label)
  )
```


Pulitzer Prize winners: Wall Street Journal
2004-2014
1990-2003
0
20
40
60
Total number of winners
Fifty Total Pulitzer Awards

Produce all plots

Nest first

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

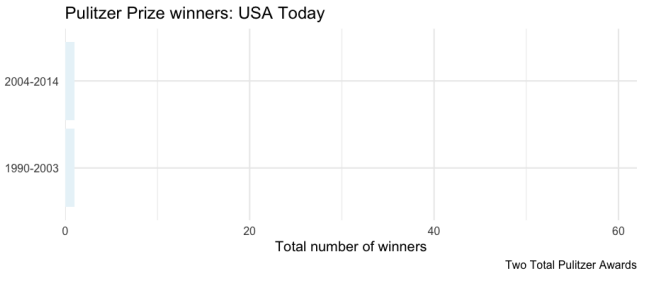
```
## # A tibble: 50 x 3
## # Groups:   newspaper, label [50]
##    newspaper             label                                    dat
##    <chr>                 <glue>                                   <li
##  1 USA Today             Two Total Pulitzer Awards                <ti
##  2 Wall Street Journal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

Produce plots

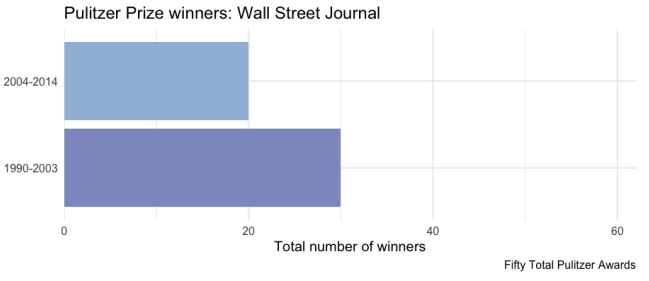
```
final_plots <- pulitzer%>%
    group_by(newspaper, label) %>%
    nest() %>%
    mutate(plots = pmap(list(newspaper, label, data), ~{
    ggplot(..3, aes(n, year_range)) +
      geom_col(aes(fill = n)) +
      scale_fill_distiller(type = "seq",
                           limits = c(0, max(pulitzer$n)),
                           palette = "BuPu",
                           direction = 1) +
        scale_x_continuous(limits = c(0, max(pulitzer$n)),
                           expand = c(0, 0)) +
        guides(fill = "none") +
     labs(title = glue("Pulitzer Prize winners: {..1}"),
            x = "Total number of winners",
            y = "",
          caption = ..2)
      })
)
```

Look at a couple plots

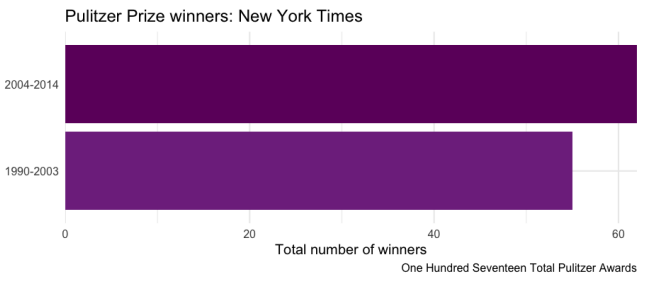
```
final_plots$plots[[1]]
```



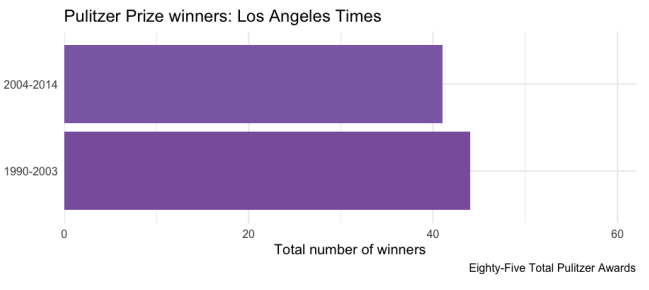

```
final_plots$plots[[2]]
```




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




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



Replicate with `nest_by()`

You try first

03:00

```r
final_plots2 <- pulitzer%>%
  ungroup() %>%
    nest_by(newspaper, label) %>%
    mutate(
      plots = list(
        ggplot(data, aes(n, year_range)) +
          geom_col(aes(fill = n)) +
        scale_fill_distiller(type = "seq",
                             limits = c(0, max(pulitzer$n)),
                             palette = "BuPu",
                             direction = 1) +
          scale_x_continuous(limits = c(0, max(pulitzer$n)),
                             expand = c(0, 0)) +
          guides(fill = "none") +
          labs(title = glue("Pulitzer Prize winners: {newspaper}"
               x = "Total number of winners",
               y = "",
               caption = label)
      )
)
```

```
final_plots2
```

```
## # A tibble: 50 x 4
## # Rowwise:  newspaper, label
##    newspaper                    label                                    <list
##    <chr>                        <glue>                                   <list
##  1 Arizona Republic             Seven Total Pulitzer Awards
##  2 Atlanta Journal Constitution Six Total Pulitzer Awards
##  3 Baltimore Sun                Thirteen Total Pulitzer Awards
##  4 Boston Globe                 Forty-One Total Pulitzer Awards
##  5 Boston Herald                Zero Total Pulitzer Awards
##  6 Charlotte Observer           Four Total Pulitzer Awards
##  7 Chicago Sun-Times            Two Total Pulitzer Awards
##  8 Chicago Tribune              Thirty-Eight Total Pulitzer Awards
##  9 Cleveland Plain Dealer       Eleven Total Pulitzer Awards
## 10 Columbus Dispatch            One Total Pulitzer Awards
## # … with 40 more rows
```

Save all plots

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

We can do this with the `map()` family, but instead we'll use a different function, which we'll talk about more next week.

As an aside, what are the **steps** we would need to take to do this?

Could we use a `nest_by()` solution?

Try with `nest_by()`

You try first:

- Create a vector of file paths
- "loop" through the file paths and the plots to save them

04:00

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/2020-21/c3-fp-2021/
##  [2] "/Users/daniel/Teaching/data_sci_specialization/2020-21/c3-fp-2021/
##  [3] "/Users/daniel/Teaching/data_sci_specialization/2020-21/c3-fp-2021/
##  [4] "/Users/daniel/Teaching/data_sci_specialization/2020-21/c3-fp-2021/
##  [5] "/Users/daniel/Teaching/data_sci_specialization/2020-21/c3-fp-2021/
##  [6] "/Users/daniel/Teaching/data_sci_specialization/2020-21/c3-fp-2021/
##  [7] "/Users/daniel/Teaching/data_sci_specialization/2020-21/c3-fp-2021/
##  [8] "/Users/daniel/Teaching/data_sci_specialization/2020-21/c3-fp-2021/
##  [9] "/Users/daniel/Teaching/data_sci_specialization/2020-21/c3-fp-2021/
## [10] "/Users/daniel/Teaching/data_sci_specialization/2020-21/c3-fp-2021/
## [11] "/Users/daniel/Teaching/data_sci_specialization/2020-21/c3-fp-2021/
## [12] "/Users/daniel/Teaching/data_sci_specialization/2020-21/c3-fp-2021/
```

Add paths to data frame

```
final_plots %>%
  ungroup() %>%
  mutate(path = paths) %>%
  select(plots, path)
```

```
## # A tibble: 50 x 2
##    plots
##    <list>
##  1 <gg>
##  2 <gg>
##  3 <gg>
##  4 <gg>
##  5 <gg>
##  6 <gg>
##  7 <gg>
##  8 <gg>
##  9 <gg>
## 10 <gg>
## # … with 40 more rows, and 1 more variable: path <chr>
```

Save

```
final_plots %>%
  ungroup() %>%
  mutate(path = paths) %>%
  rowwise() %>%
  summarize(
    ggsave(
      path,
      plots,
      width = 9.5,
      height = 6.5,
      dpi = 500
    )
  )
```

```
## # A tibble: 50 x 0
```

Wrap–up

- Parallel iterations greatly increase the things you can do – iterating through at least two things simultaneously is pretty common

- The `nest_by()` approach can regularly get you the same result as `group_by() %>% nest() %>% mutate() %>% map()`

    - Caveat – must be in a data frame, which means working with list columns
    - My view – it's still worth learning both. Looping with **{purrr}** is super flexible and often safer than base versions (type safe). Doesn't have to be used within a data frame.

Next time

Looping variants