Batch loading data

Daniel Anderson
Week 4, Class 1

UNIVERSITY OF
OREGON

Agenda

- Discuss the midterm
    - In-class portion (10 points; please don't stress)
    - Take home (40 points)
- Review Lab 1
- `map_df` and batch-loading data
- Introduce list columns

Learning objectives

- Understand when `map_df` can and should be applied
- Better understand file paths, and how `{fs}` can help
- Be able to batch load data of a specific type within a mixed-type directory
- Use filenames to pull data

Midterm

Questions?

Let's look at the take-home portion

Review Lab 1

`map_df`

- If each iteration returns a data frame, you can use `map_df` to automatically bind all the data frames together.

Example

- Create a function that simulates data

```
simulate <- function(n, mean = 0, sd = 1) {
    tibble(sample_id = seq_len(n),
           sample = rnorm(n, mean, sd))
}
simulate(10)
```

```
## # A tibble: 10 x 2
##    sample_id     sample
##        <int>      <dbl>
##  1         1  1.046262
##  2         2  0.8485443
##  3         3  0.9402922
##  4         4 -0.3246455
##  5         5 -0.1176745
##  6         6  0.4741537
##  7         7  1.209171
##  8         8  0.7537104
##  9         9  0.1545949
## 10        10  0.2740231
```

Simulation

- Assume we want to vary the sample size from 10 to 150 by increments of 5
- `mean` stays constant at 100, `sd` is constant at 10

Simulation

- Assume we want to vary the sample size from 10 to 150 by increments of 5
- `mean` stays constant at 100, `sd` is constant at 10

Try with `purrr::map`

```r
library(tidyverse)
sims <- map(seq(10, 150, 5), simulate, 100, 10)
```

```r
sims[1]
```

```
## [[1]]
## # A tibble: 10 x 2
##    sample_id    sample
##        <int>     <dbl>
##  1         1 100.2332
##  2         2 101.4359
##  3         3  86.00088
##  4         4  85.46788
##  5         5  98.35349
##  6         6 111.2937
##  7         7 105.5290
##  8         8 105.4931
##  9         9 111.1701
## 10        10 107.7481
```

```r
sims[2]
```

```
## [[1]]
## # A tibble: 15 x 2
##    sample_id    sample
##        <int>     <dbl>
##  1         1  97.82652
##  2         2 104.0230
##  3         3  93.88380
##  4         4 112.5834
##  5         5 126.6680
##  6         6  97.24947
##  7         7  95.56375
##  8         8  90.74221
##  9         9  90.25694
## 10        10  88.33989
## 11        11 108.3766
## 12        12  96.53626
## 13        13  91.62064
## 14        14  89.15980
## 15        15  77.26699
```

Swap for `map_df`

Try it - what happens?

Swap for `map_df`

Try it - what happens?

```
sims_df <- map_df(seq(10, 150, 5), simulate, 100, 10)
sims_df
```

```
## # A tibble: 2,320 x 2
##    sample_id    sample
##        <int>     <dbl>
##  1         1 105.2768
##  2         2 108.0377
##  3         3  92.80613
##  4         4 103.8668
##  5         5  94.49651
##  6         6 121.3657
##  7         7 118.5711
##  8         8 112.0742
##  9         9  88.31388
## 10        10 106.8642
## # … with 2,310 more rows
```

Notice a problem here

```
sims_df[1:15, ]
```

```
## # A tibble: 15 x 2
##    sample_id    sample
##        <int>     <dbl>
##  1         1 105.2768
##  2         2 108.0377
##  3         3  92.80613
##  4         4 103.8668
##  5         5  94.49651
##  6         6 121.3657
##  7         7 118.5711
##  8         8 112.0742
##  9         9  88.31388
## 10        10 106.8642
## 11         1 106.7368
## 12         2 105.6372
## 13         3 115.1260
## 14         4 118.1459
## 15         5 105.9971
```

.id argument

```
sims_df2 <- map_df(seq(10, 150, 5), simulate, 100, 10,
                   .id = "iteration")
sims_df2[1:14, ]
```

```
## # A tibble: 14 x 3
##    iteration sample_id    sample
##    <chr>          <int>     <dbl>
##  1 1                  1  92.88833
##  2 1                  2  87.79284
##  3 1                  3 109.8949
##  4 1                  4  99.09385
##  5 1                  5 117.5216
##  6 1                  6  90.13943
##  7 1                  7 100.4289
##  8 1                  8  95.15302
##  9 1                  9  98.92376
## 10 1                 10 106.7122
## 11 2                  1 102.8655
## 12 2                  2 124.4403
## 13 2                  3  95.16700
## 14 2                  4 113.5100
```

> `.id`: Either a string or NULL. If a string, the output will contain a variable with that name, storing either the name (if .x is named) or the index (if .x is unnamed) of the input. If NULL, the default, no variable will be created.

- {purrr} documentation

setNames

```
sample_size <- seq(10, 150, 5)
sample_size
```

```
##  [1]  10  15  20  25  30  35  40  45  50  55  60  65  70  75  80  85  90
## [18]  95 100 105 110 115 120 125 130 135 140 145 150
```

```
sample_size <- setNames(sample_size, seq(10, 150, 5))
sample_size
```

```
##  10  15  20  25  30  35  40  45  50  55  60  65  70  75  80  85  90  95
##  10  15  20  25  30  35  40  45  50  55  60  65  70  75  80  85  90  95
## 100 105 110 115 120 125 130 135 140 145 150
## 100 105 110 115 120 125 130 135 140 145 150
```

```
names(sample_size)
```

```
##  [1] "10"  "15"  "20"  "25"  "30"  "35"  "40"  "45"  "50"  "55"  "60"
## [12] "65"  "70"  "75"  "80"  "85"  "90"  "95"  "100" "105" "110" "115"
## [23] "120" "125" "130" "135" "140" "145" "150"
```

Try again

```
sims_df3 <- map_df(sample_size, simulate, 100, 10,
                   .id = "n")
sims_df3[1:14, ]
```

```
## # A tibble: 14 x 3
##    n     sample_id    sample
##    <chr>     <int>     <dbl>
##  1 10            1  88.00417
##  2 10            2 101.1348
##  3 10            3 106.8763
##  4 10            4  98.37171
##  5 10            5 102.7300
##  6 10            6 124.5342
##  7 10            7  97.52782
##  8 10            8  93.26488
##  9 10            9 109.4542
## 10 10           10  79.11504
## 11 15            1 116.7275
## 12 15            2 119.5852
## 13 15            3 100.0885
## 14 15            4 105.3158
```

Another quick example
`broom::tidy`

- The {broom} package helps us extract model output in a tidy format

```
lm(tvhours ~ age, gss_cat) %>%
    broom::tidy()
```

```
## # A tibble: 2 x 5
##   term          estimate   std.error statistic       p.value
##   <chr>            <dbl>       <dbl>     <dbl>         <dbl>
## 1 (Intercept) 1.992391   0.06980966   28.54033 4.672161e-173
## 2 age         0.02095679 0.001387361  15.10551 4.689310e- 51
```

Fit separate models by year

Again - probs not best statistically

```
split(gss_cat, gss_cat$year) %>%
    map_df(~lm(tvhours ~ age, .x) %>%
             broom::tidy())
```

```
## # A tibble: 16 x 5
##    term          estimate   std.error statistic      p.value
##    <chr>            <dbl>       <dbl>     <dbl>        <dbl>
##  1 (Intercept) 2.080163   0.1709061   12.17138  7.995632e-33
##  2 age         0.01948584 0.003485199  5.591027 2.599011e- 8
##  3 (Intercept) 2.078999   0.2176829    9.550583 1.191266e-20
##  4 age         0.01963575 0.004400292  4.462375 9.137366e- 6
##  5 (Intercept) 1.767990   0.2464509    7.173804 1.531756e-12
##  6 age         0.02386070 0.005031548  4.742218 2.459650e- 6
##  7 (Intercept) 2.096054   0.1496431   14.00702  1.419772e-42
##  8 age         0.01781388 0.002977289  5.983256 2.589482e- 9
##  9 (Intercept) 1.855278   0.2156381    8.603668 2.167351e-17
## 10 age         0.02390720 0.004314567  5.541043 3.628675e- 8
## 11 (Intercept) 2.068914   0.2096397    9.868903 2.896085e-22
## 12 age         0.01989505 0.004086638  4.868317 1.251234e- 6
## 13 (Intercept) 1.878070   0.2258400    8.315932 2.280108e-16
## 14 age         0.02547794 0.004449295  5.726287 1.274840e- 8
```

.id

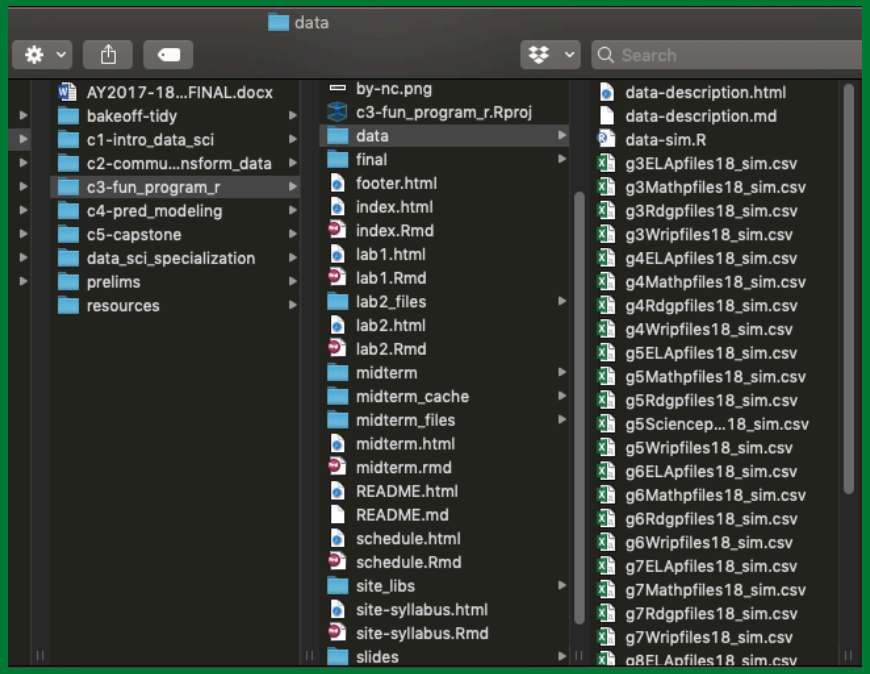
In cases like the preceding, `.id` becomes invaluable

```
split(gss_cat, gss_cat$year) %>%
     map_df(~lm(tvhours ~ age, .x) %>%
               broom::tidy(),
             .id = "year")
```

```
## # A tibble: 16 x 6
##    year  term          estimate   std.error statistic      p.value
##    <chr> <chr>            <dbl>       <dbl>     <dbl>        <dbl>
##  1 2000  (Intercept) 2.080163   0.1709061   12.17138  7.995632e-33
##  2 2000  age         0.01948584 0.003485199  5.591027 2.599011e- 8
##  3 2002  (Intercept) 2.078999   0.2176829    9.550583 1.191266e-20
##  4 2002  age         0.01963575 0.004400292  4.462375 9.137366e- 6
##  5 2004  (Intercept) 1.767990   0.2464509    7.173804 1.531756e-12
##  6 2004  age         0.02386070 0.005031548  4.742218 2.459650e- 6
##  7 2006  (Intercept) 2.096054   0.1496431   14.00702  1.419772e-42
##  8 2006  age         0.01781388 0.002977289  5.983256 2.589482e- 9
##  9 2008  (Intercept) 1.855278   0.2156381    8.603668 2.167351e-17
## 10 2008  age         0.02390720 0.004314567  5.541043 3.628675e- 8
## 11 2010  (Intercept) 2.068914   0.2096397    9.868903 2.896085e-22
## 12 2010  age         0.01989505 0.004086638  4.868317 1.251234e- 6
## 13 2012  (Intercept) 1.878070   0.2258400    8.315932 2.280108e-16
```

Batch-loading data

Please follow along



{fs}

- note - there are base equivalents. `{fs}` is just a a bit better across platforms and has better defaults.

```
# install.packages("fs")
library(fs)
dir_ls("../data")
```

```
## ../data/data-description.html      ../data/data-description.md
## ../data/data-sim.R                 ../data/g11ELApfiles18_sim.csv
## ../data/g11Mathpfiles18_sim.csv    ../data/g11Rdgpfiles18_sim.csv
## ../data/g11Sciencepfiles18_sim.csv ../data/g11Wripfiles18_sim.csv
## ../data/g3ELApfiles18_sim.csv      ../data/g3Mathpfiles18_sim.csv
## ../data/g3Rdgpfiles18_sim.csv      ../data/g3Wripfiles18_sim.csv
## ../data/g4ELApfiles18_sim.csv      ../data/g4Mathpfiles18_sim.csv
## ../data/g4Rdgpfiles18_sim.csv      ../data/g4Wripfiles18_sim.csv
## ../data/g5ELApfiles18_sim.csv      ../data/g5Mathpfiles18_sim.csv
## ../data/g5Rdgpfiles18_sim.csv      ../data/g5Sciencepfiles18_sim.csv
## ../data/g5Wripfiles18_sim.csv      ../data/g6ELApfiles18_sim.csv
## ../data/g6Mathpfiles18_sim.csv     ../data/g6Rdgpfiles18_sim.csv
## ../data/g6Wripfiles18_sim.csv      ../data/g7ELApfiles18_sim.csv
## ../data/g7Mathpfiles18_sim.csv     ../data/g7Rdgpfiles18_sim.csv
## ../data/g7Wripfiles18_sim.csv      ../data/g8ELApfiles18_sim.csv
## ../data/g8Mathpfiles18_sim.csv     ../data/g8Rdgpfiles18_sim.csv
```

Limit files

- We really only want the `.csv`

```
dir_ls("../data", glob = "*.csv")
```

```
## ../data/g11ELApfiles18_sim.csv     ../data/g11Mathpfiles18_sim.csv
## ../data/g11Rdgpfiles18_sim.csv     ../data/g11Sciencepfiles18_sim.csv
## ../data/g11Wripfiles18_sim.csv     ../data/g3ELApfiles18_sim.csv
## ../data/g3Mathpfiles18_sim.csv     ../data/g3Rdgpfiles18_sim.csv
## ../data/g3Wripfiles18_sim.csv      ../data/g4ELApfiles18_sim.csv
## ../data/g4Mathpfiles18_sim.csv     ../data/g4Rdgpfiles18_sim.csv
## ../data/g4Wripfiles18_sim.csv      ../data/g5ELApfiles18_sim.csv
## ../data/g5Mathpfiles18_sim.csv     ../data/g5Rdgpfiles18_sim.csv
## ../data/g5Sciencepfiles18_sim.csv  ../data/g5Wripfiles18_sim.csv
## ../data/g6ELApfiles18_sim.csv      ../data/g6Mathpfiles18_sim.csv
## ../data/g6Rdgpfiles18_sim.csv      ../data/g6Wripfiles18_sim.csv
## ../data/g7ELApfiles18_sim.csv      ../data/g7Mathpfiles18_sim.csv
## ../data/g7Rdgpfiles18_sim.csv      ../data/g7Wripfiles18_sim.csv
## ../data/g8ELApfiles18_sim.csv      ../data/g8Mathpfiles18_sim.csv
## ../data/g8Rdgpfiles18_sim.csv      ../data/g8Sciencepfiles18_sim.csv
## ../data/g8Wripfiles18_sim.csv
```

If we want to be extra careful

- Combine with `here::here`. This will be pretty bullet-proof from a reproducibility perspective.

```
dir_ls(here::here("data"), glob = "*.csv")
```

```
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g11ELApfil
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g11Mathpfi
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g11Rdgpfil
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g11Science
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g11Wripfil
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g3ELApfile
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g3Mathpfil
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g3Rdgpfile
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g3Wripfile
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g4ELApfile
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g4Mathpfil
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g4Rdgpfile
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g4Wripfile
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g5ELApfile
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g5Mathpfil
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g5Rdgpfile
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g5Sciencep
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g5Wripfile
```

Batch load

- Loop through the directories and `import` or `read_csv`

```
files <- dir_ls(here::here("data"), glob = "*.csv")
batch <- map_df(files, read_csv)
batch
```

```
## # A tibble: 15,945 x 22
##    Entry  Theta Status Count RawScore       SE Infit Infit_Z Outfit
##    <dbl>  <dbl>  <dbl> <dbl>    <dbl>    <dbl> <dbl>   <dbl>  <dbl>
##  1   376 0.3483      1    36       17 0.3629    1.02    0.18   0.96
##  2   364 0.2156      1    36       14 0.3629    1.09    0.71   1.03
##  3   340 4.4752      1    36       35 1.0234    1.01    0.32   0.66
##  4   334 5.7137      0    36       36 1.8371    1       0      1
##  5   314 3.2773      1    36       33 0.6187    0.89   -0.12   0.72
##  6   146 1.4088      1    36       24 0.3774    1.04    0.48   1.03
##  7   106 3.2773      1    36       33 0.6187    1.09    0.33   1.02
##  8   290 2.9394      1    36       32 0.547800  0.82   -0.71   0.56
##  9   330 1.5541      1    36       25 0.3852    0.82   -1.02   0.74
## 10   251 2.6655      1    36       31 0.5008    1       0      0.67
## # … with 15,935 more rows, and 13 more variables: Outfit_Z <dbl>,
## #   Displacement <dbl>, PointMeasureCorr <dbl>, Weight <dbl>,
## #   ObservMatch <dbl>, ExpectMatch <dbl>, PointMeasureExpected <dbl>,
## #   RMSR <dbl>, WMLE <dbl>, testeventid <dbl>, ssid <dbl>,
```

Problem

- We've lost a lot of info - no way to identify which file is which

Try to fix it!

Add id

```
batch2 <- map_df(files, read_csv, .id = "file")
batch2
```

```
## # A tibble: 15,945 x 23
##    file  Entry  Theta Status Count RawScore       SE Infit Infit_Z Outfit
##    <chr> <dbl>  <dbl>  <dbl> <dbl>    <dbl>    <dbl> <dbl>   <dbl>  <dbl>
##  1 /Use…   376 0.3483      1    36       17 0.3629    1.02    0.18   0.96
##  2 /Use…   364 0.2156      1    36       14 0.3629    1.09    0.71   1.03
##  3 /Use…   340 4.4752      1    36       35 1.0234    1.01    0.32   0.66
##  4 /Use…   334 5.7137      0    36       36 1.8371    1       0      1
##  5 /Use…   314 3.2773      1    36       33 0.6187    0.89   -0.12   0.72
##  6 /Use…   146 1.4088      1    36       24 0.3774    1.04    0.48   1.03
##  7 /Use…   106 3.2773      1    36       33 0.6187    1.09    0.33   1.02
##  8 /Use…   290 2.9394      1    36       32 0.547800  0.82   -0.71   0.56
##  9 /Use…   330 1.5541      1    36       25 0.3852    0.82   -1.02   0.74
## 10 /Use…   251 2.6655      1    36       31 0.5008    1       0      0.67
## # … with 15,935 more rows, and 13 more variables: Outfit_Z <dbl>,
## #   Displacement <dbl>, PointMeasureCorr <dbl>, Weight <dbl>,
## #   ObservMatch <dbl>, ExpectMatch <dbl>, PointMeasureExpected <dbl>,
## #   RMSR <dbl>, WMLE <dbl>, testeventid <dbl>, ssid <dbl>,
## #   asmtprmrydsbltycd <dbl>, asmtscndrydsbltycd <dbl>
```

```
batch2 %>%
    count(file)
```

```
## # A tibble: 31 x 2
##    file                                                                n
##    <chr>                                                           <int>
##  1 /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/d…   453
##  2 /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/d…   460
##  3 /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/d…   453
##  4 /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/d…   438
##  5 /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/d…   453
##  6 /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/d…   540
##  7 /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/d…   536
##  8 /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/d…   540
##  9 /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/d…   540
## 10 /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/d…   585
## # … with 21 more rows
```

- Still not terrifically useful. What can we do?

Step 1

- Remove the `here::here` path from string

```
batch2 <- batch2 %>%
    mutate(file = str_replace_all(file, here::here("data"), ""))

count(batch2, file)
```

```
## # A tibble: 31 x 2
##    file                          n
##    <chr>                     <int>
##  1 /g11ELApfiles18_sim.csv     453
##  2 /g11Mathpfiles18_sim.csv    460
##  3 /g11Rdgpfiles18_sim.csv     453
##  4 /g11Sciencepfiles18_sim.csv 438
##  5 /g11Wripfiles18_sim.csv     453
##  6 /g3ELApfiles18_sim.csv      540
##  7 /g3Mathpfiles18_sim.csv     536
##  8 /g3Rdgpfiles18_sim.csv      540
##  9 /g3Wripfiles18_sim.csv      540
## 10 /g4ELApfiles18_sim.csv      585
## # … with 21 more rows
```

Pull out pieces you need

- Regular expressions are most powerful here
    - We haven't talked about them much
- Try RegExplain

Pull grade

- Note - I'm not expecting you to just suddenly be able to do this. This is more for illustration. There's also other ways you could extract the same info

```
batch2 %>%
    mutate(grade = str_replace_all(file, "/g(\\d?\\d).+", "\\1")) %>%
    select(file, grade)
```

```
## # A tibble: 15,945 x 2
##    file                    grade
##    <chr>                   <chr>
##  1 /g11ELApfiles18_sim.csv 11
##  2 /g11ELApfiles18_sim.csv 11
##  3 /g11ELApfiles18_sim.csv 11
##  4 /g11ELApfiles18_sim.csv 11
##  5 /g11ELApfiles18_sim.csv 11
##  6 /g11ELApfiles18_sim.csv 11
##  7 /g11ELApfiles18_sim.csv 11
##  8 /g11ELApfiles18_sim.csv 11
##  9 /g11ELApfiles18_sim.csv 11
## 10 /g11ELApfiles18_sim.csv 11
## # … with 15,935 more rows
```

parse_number

- In this case `parse_number` also works - but note that it would not work to extract the year

```
batch2 %>%
mutate(grade = parse_number(file)) %>%
    select(file, grade)
```

```
## # A tibble: 15,945 x 2
##    file                    grade
##    <chr>                   <dbl>
##  1 /g11ELApfiles18_sim.csv    11
##  2 /g11ELApfiles18_sim.csv    11
##  3 /g11ELApfiles18_sim.csv    11
##  4 /g11ELApfiles18_sim.csv    11
##  5 /g11ELApfiles18_sim.csv    11
##  6 /g11ELApfiles18_sim.csv    11
##  7 /g11ELApfiles18_sim.csv    11
##  8 /g11ELApfiles18_sim.csv    11
##  9 /g11ELApfiles18_sim.csv    11
## 10 /g11ELApfiles18_sim.csv    11
## # … with 15,935 more rows
```

Extract year

- In this case `parse_number` also works - but note that it would not work to extract the year

```
batch2 %>%
    mutate(grade = str_replace_all(file, "/g(\\d?\\d).+", "\\1"),
        year = str_replace_all(file, ".+files(\\d\\d)_sim.+", "\\1")) %>%
    select(file, grade, year)
```

```
## # A tibble: 15,945 x 3
##    file                      grade year
##    <chr>                     <chr> <chr>
##  1 /g11ELApfiles18_sim.csv 11    18
##  2 /g11ELApfiles18_sim.csv 11    18
##  3 /g11ELApfiles18_sim.csv 11    18
##  4 /g11ELApfiles18_sim.csv 11    18
##  5 /g11ELApfiles18_sim.csv 11    18
##  6 /g11ELApfiles18_sim.csv 11    18
##  7 /g11ELApfiles18_sim.csv 11    18
##  8 /g11ELApfiles18_sim.csv 11    18
##  9 /g11ELApfiles18_sim.csv 11    18
## 10 /g11ELApfiles18_sim.csv 11    18
## # … with 15,935 more rows
```

Extract Content Area

```
batch2 %>%
    mutate(grade = str_replace_all(file, "/g(\\d?\\d).+", "\\1"),
           year = str_replace_all(file, ".+files(\\d\\d)_sim.+", "\\1"),
         content = str_replace_all(file, "/g\\d?\\d(.+)pfiles.+", "\\1")) %>%
    select(file, grade, year, content)
```

```
## # A tibble: 15,945 x 4
##    file                    grade year  content
##    <chr>                   <chr> <chr> <chr>
##  1 /g11ELApfiles18_sim.csv 11    18    ELA
##  2 /g11ELApfiles18_sim.csv 11    18    ELA
##  3 /g11ELApfiles18_sim.csv 11    18    ELA
##  4 /g11ELApfiles18_sim.csv 11    18    ELA
##  5 /g11ELApfiles18_sim.csv 11    18    ELA
##  6 /g11ELApfiles18_sim.csv 11    18    ELA
##  7 /g11ELApfiles18_sim.csv 11    18    ELA
##  8 /g11ELApfiles18_sim.csv 11    18    ELA
##  9 /g11ELApfiles18_sim.csv 11    18    ELA
## 10 /g11ELApfiles18_sim.csv 11    18    ELA
## # … with 15,935 more rows
```

Double checks: grade

```
batch2 %>%
    mutate(grade = str_replace_all(file,
                                   "/g(\\d?\\d).+",
                                   "\\1"),
           year = str_replace_all(file,
                                  ".+files(\\d\\d)_sim.+",
                                  "\\1"),
           content = str_replace_all(file,
                                     "/g\\d?\\d(.+)pfiles.+",
                                     "\\1")) %>%
    select(file, grade, year, content) %>%
    count(grade)
```

```
## # A tibble: 7 x 2
##   grade     n
##   <chr> <int>
## 1 11     2257
## 2 3      2156
## 3 4      2341
## 4 5      2632
## 5 6      2216
## 6 7      1962
## 7 8      2381
```

Double checks: year

```
batch2 %>%
    mutate(grade = str_replace_all(file,
                                   "/g(\\d?\\d).+",
                                   "\\1"),
           year = str_replace_all(file,
                                  ".+files(\\d\\d)_sim.+",
                                  "\\1"),
           content = str_replace_all(file,
                                     "/g\\d?\\d(.+)pfiles.+",
                                     "\\1")) %>%
    select(file, grade, year, content) %>%
    count(year)
```

```
## # A tibble: 1 x 2
##   year      n
##   <chr> <int>
## 1 18    15945
```

Double checks: content

```
batch2 %>%
    mutate(grade = str_replace_all(file,
                                   "/g(\\d?\\d).+",
                                   "\\1"),
           year = str_replace_all(file,
                                  ".+files(\\d\\d)_sim.+",
                                  "\\1"),
           content = str_replace_all(file,
                                     "/g\\d?\\d(.+)pfiles.+",
                                     "\\1")) %>%
    select(file, grade, year, content) %>%
    count(content)
```

```
## # A tibble: 5 x 2
##   content     n
##   <chr>   <int>
## 1 ELA      3627
## 2 Math     3629
## 3 Rdg      3627
## 4 Science  1435
## 5 Wri      3627
```

Finalize

```
d <- batch2 %>%
    mutate(grade = str_replace_all(file, "/g(\\d?\\d).+", "\\1"),
           grade = as.integer(grade),
           year = str_replace_all(file, ".+files(\\d\\d)_sim.+", "\\1"),
           year = as.integer(grade),
           content = str_replace_all(file, "/g\\d?\\d(.+)pfiles.+", "\\1"))
    select(-file) %>%
    select(ssid, grade, year, content, testeventid, asmtprmrydsbltycd,
           asmtscndrydsbltycd, Entry:WMLE)
```

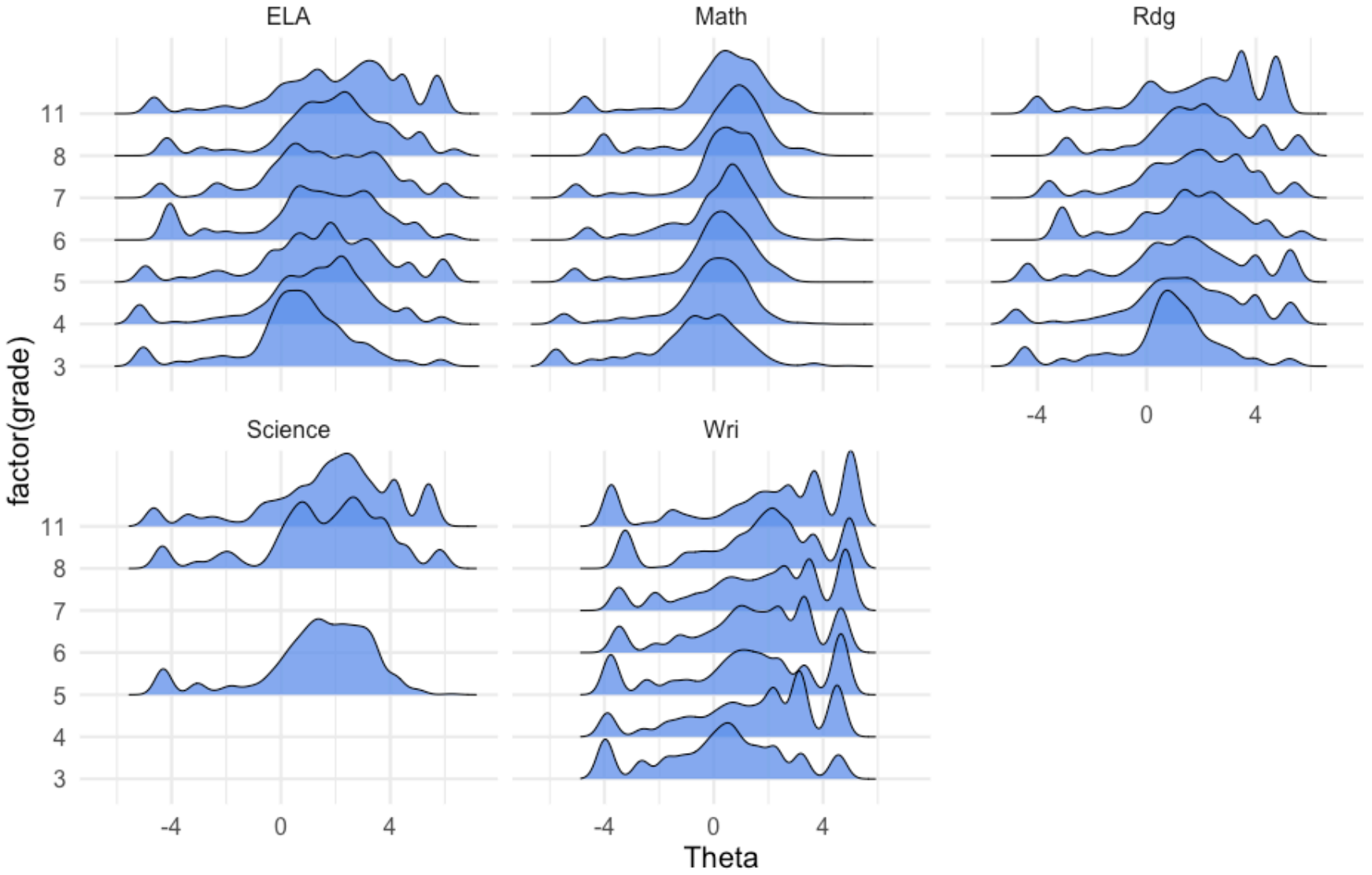
Final product

- In this case, we basically have a tidy data frame already!
- We've reduced our problem from 31 files to a single file

```
d
```

```
## # A tibble: 15,945 x 25
##        ssid grade  year content testeventid asmtprmrydsbltycd
##       <dbl> <int> <int> <chr>          <dbl>               <dbl>
##  1 11424133    11    11 ELA           151867                  10
##  2 11032405    11    11 ELA           144283                  82
##  3 10809295    11    11 ELA           146312                  10
##  4 10815457    11    11 ELA           146977                  10
##  5 10638008    11    11 ELA           144402                  10
##  6 10006109    11    11 ELA           149010                  10
##  7  9305807    11    11 ELA           146880                  10
##  8 10396942    11    11 ELA           143994                  10
##  9 10801367    11    11 ELA           143860                  10
## 10 10246495    11    11 ELA           144434                  80
## # … with 15,935 more rows, and 19 more variables:
## #   asmtscndrydsbltycd <dbl>, Entry <dbl>, Theta <dbl>, Status <dbl>,
## #   Count <dbl>, RawScore <dbl>, SE <dbl>, Infit <dbl>, Infit_Z <dbl>,
## #   Outfit <dbl>, Outfit_Z <dbl>, Displacement <dbl>,
```

Quick look at distributions



Calculate some summary stats

```
d %>%
    group_by(grade, content, asmtprmrydsbltycd) %>%
    summarize(mean = mean(Theta)) %>%
    spread(content, mean)
```

```
## # A tibble: 77 x 7
## # Groups:   grade [7]
##    grade asmtprmrydsbltycd        ELA       Math         Rdg Science
##    <int>              <dbl>      <dbl>      <dbl>       <dbl>   <dbl>
##  1     3                  0  0.7972462 -3.38815    0.8141667        NA
##  2     3                 10  0.4113426 -0.7972930  0.7428369        NA
##  3     3                 20  0.4086200 -1.46495    0.7528571        NA
##  4     3                 40 -2.766267  -2.925338   -2               NA
##  5     3                 50  1.407710  -0.1980967  0.7541463        NA
##  6     3                 60  1.33732   -0.9878643  0.7753333        NA
##  7     3                 70  0.2105250 -1.658856  -0.9320000        NA
##  8     3                 74  1.504025   0.2944778  0.02600000       NA
##  9     3                 80  1.025756  -0.6844064  0.7501333        NA
## 10     3                 82  0.3758255 -0.8861006  0.5980100        NA
## # … with 67 more rows, and 1 more variable: Wri <dbl>
```

Backing up a bit

- What if we wanted only math files?

```
dir_ls(here::here("data"), regexp = "Math")
```

```
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g11Mathpfi
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g3Mathpfil
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g4Mathpfil
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g5Mathpfil
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g6Mathpfil
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g7Mathpfil
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g8Mathpfil
```

Only Grade 5

You try

Only Grade 5

You try

```
dir_ls(here::here("data"), regexp = "g5")
```

```
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g5ELApfile
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g5Mathpfil
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g5Rdgpfile
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g5Sciencep
## /Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g5Wripfile
```

Base equivalents

```
list.files(here::here("data"))
```

```
##  [1] "data-description.html"      "data-description.md"
##  [3] "data-sim.R"                 "g11ELApfiles18_sim.csv"
##  [5] "g11Mathpfiles18_sim.csv"    "g11Rdgpfiles18_sim.csv"
##  [7] "g11Sciencepfiles18_sim.csv" "g11Wripfiles18_sim.csv"
##  [9] "g3ELApfiles18_sim.csv"      "g3Mathpfiles18_sim.csv"
## [11] "g3Rdgpfiles18_sim.csv"      "g3Wripfiles18_sim.csv"
## [13] "g4ELApfiles18_sim.csv"      "g4Mathpfiles18_sim.csv"
## [15] "g4Rdgpfiles18_sim.csv"      "g4Wripfiles18_sim.csv"
## [17] "g5ELApfiles18_sim.csv"      "g5Mathpfiles18_sim.csv"
## [19] "g5Rdgpfiles18_sim.csv"      "g5Sciencepfiles18_sim.csv"
## [21] "g5Wripfiles18_sim.csv"      "g6ELApfiles18_sim.csv"
## [23] "g6Mathpfiles18_sim.csv"     "g6Rdgpfiles18_sim.csv"
## [25] "g6Wripfiles18_sim.csv"      "g7ELApfiles18_sim.csv"
## [27] "g7Mathpfiles18_sim.csv"     "g7Rdgpfiles18_sim.csv"
## [29] "g7Wripfiles18_sim.csv"      "g8ELApfiles18_sim.csv"
## [31] "g8Mathpfiles18_sim.csv"     "g8Rdgpfiles18_sim.csv"
## [33] "g8Sciencepfiles18_sim.csv"  "g8Wripfiles18_sim.csv"
## [35] "pfiles18"
```

Full path

```
list.files(here::here("data"), full.names = TRUE)
```

```
##  [1] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/data
##  [2] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/data
##  [3] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/data
##  [4] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g11E
##  [5] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g11M
##  [6] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g11R
##  [7] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g11S
##  [8] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g11W
##  [9] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g3EL
## [10] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g3Ma
## [11] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g3Rd
## [12] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g3Wr
## [13] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g4EL
## [14] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g4Ma
## [15] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g4Rd
## [16] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g4Wr
## [17] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g5EL
## [18] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g5Ma
## [19] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g5Rd
## [20] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g5Sc
## [21] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g5Wr
## [22] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g6EL
```

Only csvs

```
list.files(here::here("data"),
           full.names = TRUE,
           pattern = "*.csv")
```

```
##  [1] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g11E
##  [2] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g11M
##  [3] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g11R
##  [4] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g11S
##  [5] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g11W
##  [6] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g3EL
##  [7] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g3Ma
##  [8] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g3Rd
##  [9] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g3Wr
## [10] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g4EL
## [11] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g4Ma
## [12] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g4Rd
## [13] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g4Wr
## [14] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g5EL
## [15] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g5Ma
## [16] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g5Rd
## [17] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g5Sc
## [18] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g5Wr
## [19] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g6EL
## [20] "/Users/Daniel/Teaching/data_sci_specialization/c3-fun_program_r/data/g6Ma
```

Why not use base?

- We could, but `{fs}` plays a little nicer with `{purrr}`

Why not use base?

- We could, but `{fs}` plays a little nicer with `{purrr}`

```
files <- list.files(here::here("data"), pattern = "*.csv")
batch3 <- map_df(files, read_csv, .id = "file")
```

```
## Error: 'g11ELApfiles18_sim.csv' does not exist in current working directory ('/
```

Why not use base?

- We could, but `{fs}` plays a little nicer with `{purrr}`

```
files <- list.files(here::here("data"), pattern = "*.csv")
batch3 <- map_df(files, read_csv, .id = "file")
```

```
## Error: 'g11ELApfiles18_sim.csv' does not exist in current working directory ('/
```

- Need to return full names

```
files
```

```
##  [1] "g11ELApfiles18_sim.csv"     "g11Mathpfiles18_sim.csv"
##  [3] "g11Rdgpfiles18_sim.csv"     "g11Sciencepfiles18_sim.csv"
##  [5] "g11Wripfiles18_sim.csv"     "g3ELApfiles18_sim.csv"
##  [7] "g3Mathpfiles18_sim.csv"     "g3Rdgpfiles18_sim.csv"
##  [9] "g3Wripfiles18_sim.csv"      "g4ELApfiles18_sim.csv"
## [11] "g4Mathpfiles18_sim.csv"     "g4Rdgpfiles18_sim.csv"
## [13] "g4Wripfiles18_sim.csv"      "g5ELApfiles18_sim.csv"
## [15] "g5Mathpfiles18_sim.csv"     "g5Rdgpfiles18_sim.csv"
## [17] "g5Sciencepfiles18_sim.csv"  "g5Wripfiles18_sim.csv"
## [19] "g6ELApfiles18_sim.csv"      "g6Mathpfiles18_sim.csv"
## [21] "g6Rdgpfiles18_sim.csv"      "g6Wripfiles18_sim.csv"
```

Try again

```
files <- list.files(here::here("data"),
                    pattern = "*.csv",
                    full.names = TRUE)
batch3 <- map_df(files, read_csv, .id = "file")
batch3
```

```
## # A tibble: 15,945 x 23
##    file  Entry  Theta Status Count RawScore       SE Infit Infit_Z Outfit
##    <chr> <dbl>  <dbl>  <dbl> <dbl>    <dbl>    <dbl> <dbl>   <dbl>  <dbl>
##  1 1       376 0.3483      1    36       17 0.3629    1.02    0.18   0.96
##  2 1       364 0.2156      1    36       14 0.3629    1.09    0.71   1.03
##  3 1       340 4.4752      1    36       35 1.0234    1.01    0.32   0.66
##  4 1       334 5.7137      0    36       36 1.8371    1       0      1
##  5 1       314 3.2773      1    36       33 0.6187    0.89   -0.12   0.72
##  6 1       146 1.4088      1    36       24 0.3774    1.04    0.48   1.03
##  7 1       106 3.2773      1    36       33 0.6187    1.09    0.33   1.02
##  8 1       290 2.9394      1    36       32 0.547800  0.82   -0.71   0.56
##  9 1       330 1.5541      1    36       25 0.3852    0.82   -1.02   0.74
## 10 1       251 2.6655      1    36       31 0.5008    1       0      0.67
## # … with 15,935 more rows, and 13 more variables: Outfit_Z <dbl>,
## #   Displacement <dbl>, PointMeasureCorr <dbl>, Weight <dbl>,
## #   ObservMatch <dbl>, ExpectMatch <dbl>, PointMeasureExpected <dbl>,
## #   RMSR <dbl>, WMLE <dbl>, testeventid <dbl>, ssid <dbl>,
## #   asmtprmrydsbltycd <dbl>, asmtscndrydsbltycd <dbl>
```

indexes

- The prior example gave us indexes, rather than the file path. Why?

indexes

- The prior example gave us indexes, rather than the file path. Why?

No names

```
names(files)
```

```
## NULL
```

- We **need** the file path! An index isn't nearly as useful.

Base method that works

```
files <- list.files(here::here("data"),
                    pattern = "*.csv",
                    full.names = TRUE)
files <- setNames(files, files)

batch4 <- map_df(files, read_csv, .id = "file")
batch4
```

```
## # A tibble: 15,945 x 23
##    file  Entry  Theta Status Count RawScore       SE Infit Infit_Z Outfit
##    <chr> <dbl>  <dbl>  <dbl> <dbl>    <dbl>    <dbl> <dbl>   <dbl>  <dbl>
##  1 /Use…   376 0.3483      1    36       17 0.3629    1.02    0.18   0.96
##  2 /Use…   364 0.2156      1    36       14 0.3629    1.09    0.71   1.03
##  3 /Use…   340 4.4752      1    36       35 1.0234    1.01    0.32   0.66
##  4 /Use…   334 5.7137      0    36       36 1.8371    1       0      1
##  5 /Use…   314 3.2773      1    36       33 0.6187    0.89   -0.12   0.72
##  6 /Use…   146 1.4088      1    36       24 0.3774    1.04    0.48   1.03
##  7 /Use…   106 3.2773      1    36       33 0.6187    1.09    0.33   1.02
##  8 /Use…   290 2.9394      1    36       32 0.547800  0.82   -0.71   0.56
##  9 /Use…   330 1.5541      1    36       25 0.3852    0.82   -1.02   0.74
## 10 /Use…   251 2.6655      1    36       31 0.5008    1       0      0.67
## # … with 15,935 more rows, and 13 more variables: Outfit_Z <dbl>,
## #   Displacement <dbl>, PointMeasureCorr <dbl>, Weight <dbl>,
## #   ObservMatch <dbl>, ExpectMatch <dbl>, PointMeasureExpected <dbl>,
```

My recommendation

- If you're working interactively, no reason not to use `{fs}`
- If you are building functions that take generic paths, might be worth considering skipping the dependency

My recommendation

- If you're working interactively, no reason not to use `{fs}`
- If you are building functions that take generic paths, might be worth considering skipping the dependency

Note

I am **not** saying skip it, but rather that you should **consider** whether it is really needed or not.

List columns

Comparing models

Let's say we wanted to fit/compare a set of models for each content area

1. `lm(Theta ~ asmtprmrydsbltycd)`
2. `lm(Theta ~ asmtprmrydsbltycd + asmtscndrydsbltycd)`
3. `lm(Theta ~ asmtprmrydsbltycd + asmtscndrydsbltycd + asmtprmrydsbltycd:asmtscndrydsbltycd)`

Split the data

The base method we've been using...

```
splt_content <- split(d, d$content)
str(splt_content)
```

```
## List of 5
##  $ ELA    :Classes 'tbl_df', 'tbl' and 'data.frame':    3627 obs. of  25 variab
##   ..$ ssid                : num [1:3627] 11424133 11032405 10809295 10815457 10
##   ..$ grade               : int [1:3627] 11 11 11 11 11 11 11 11 11 11 ...
##   ..$ year                : int [1:3627] 11 11 11 11 11 11 11 11 11 11 ...
##   ..$ content             : chr [1:3627] "ELA" "ELA" "ELA" "ELA" ...
##   ..$ testeventid         : num [1:3627] 151867 144283 146312 146977 144402 ...
##   ..$ asmtprmrydsbltycd   : num [1:3627] 10 82 10 10 10 10 10 10 10 80 ...
##   ..$ asmtscndrydsbltycd  : num [1:3627] 80 50 20 0 10 0 0 0 0 0 ...
##   ..$ Entry               : num [1:3627] 376 364 340 334 314 146 106 290 330 25
##   ..$ Theta               : num [1:3627] 0.348 0.216 4.475 5.714 3.277 ...
##   ..$ Status              : num [1:3627] 1 1 1 0 1 1 1 1 1 1 ...
##   ..$ Count               : num [1:3627] 36 36 36 36 36 36 36 36 36 36 ...
##   ..$ RawScore            : num [1:3627] 17 14 35 36 33 24 33 32 25 31 ...
##   ..$ SE                  : num [1:3627] 0.363 0.363 1.023 1.837 0.619 ...
##   ..$ Infit               : num [1:3627] 1.02 1.09 1.01 1 0.89 1.04 1.09 0.82 0
##   ..$ Infit_Z             : num [1:3627] 0.18 0.71 0.32 0 -0.12 0.48 0.33 -0.71
##   ..$ Outfit              : num [1:3627] 0.96 1.03 0.66 1 0.72 1.03 1.02 0.56 0
```

We could use this method

```
m1 <- map(splt_content, ~lm(Theta ~ asmtprmrydsbltycd,
                                          data = .x))
m2 <- map(splt_content, ~lm(Theta ~ asmtprmrydsbltycd +
                                    asmtscndrydsbltycd,
                                                data = .x))
m3 <- map(splt_content, ~lm(Theta ~ asmtprmrydsbltycd*asmtscndrydsbltycd,
                                          data = .x))
```

- We could then go through and conduct tests to see which model was better, etc.

Alternative

- Create a data frame with a list column

```
d %>%
    nest(-content)
```

```
## # A tibble: 5 x 2
##   content data
##   <chr>   <list>
## 1 ELA     <tibble [3,627 × 24]>
## 2 Math    <tibble [3,629 × 24]>
## 3 Rdg     <tibble [3,627 × 24]>
## 4 Science <tibble [1,435 × 24]>
## 5 Wri     <tibble [3,627 × 24]>
```

Add model list column

```
mods <- d %>%
    nest(-content) %>%
    mutate(m1 = map(data, ~lm(Theta ~ asmtprmrydsbltycd,
                                                data = .x)),
           m2 = map(data, ~lm(Theta ~ asmtprmrydsbltycd + asmtscndrydsbltycd,
                                                data = .x)),
           m3 = map(data, ~lm(Theta ~ asmtprmrydsbltycd*asmtscndrydsbltycd,
                                                data = .x)))
mods
```

```
## # A tibble: 5 x 5
##   content data                    m1       m2       m3
##   <chr>   <list>                  <list>   <list>   <list>
## 1 ELA     <tibble [3,627 × 24]> <S3: lm> <S3: lm> <S3: lm>
## 2 Math    <tibble [3,629 × 24]> <S3: lm> <S3: lm> <S3: lm>
## 3 Rdg     <tibble [3,627 × 24]> <S3: lm> <S3: lm> <S3: lm>
## 4 Science <tibble [1,435 × 24]> <S3: lm> <S3: lm> <S3: lm>
## 5 Wri     <tibble [3,627 × 24]> <S3: lm> <S3: lm> <S3: lm>
```

Part of the benefit

It's a normal data frame!

```
mods %>%
    gather(model, output, m1:m3)
```

```
## # A tibble: 15 x 4
##    content data                    model output
##    <chr>   <list>                  <chr> <list>
##  1 ELA     <tibble [3,627 × 24]> m1    <S3: lm>
##  2 Math    <tibble [3,629 × 24]> m1    <S3: lm>
##  3 Rdg     <tibble [3,627 × 24]> m1    <S3: lm>
##  4 Science <tibble [1,435 × 24]> m1    <S3: lm>
##  5 Wri     <tibble [3,627 × 24]> m1    <S3: lm>
##  6 ELA     <tibble [3,627 × 24]> m2    <S3: lm>
##  7 Math    <tibble [3,629 × 24]> m2    <S3: lm>
##  8 Rdg     <tibble [3,627 × 24]> m2    <S3: lm>
##  9 Science <tibble [1,435 × 24]> m2    <S3: lm>
## 10 Wri     <tibble [3,627 × 24]> m2    <S3: lm>
## 11 ELA     <tibble [3,627 × 24]> m3    <S3: lm>
## 12 Math    <tibble [3,629 × 24]> m3    <S3: lm>
## 13 Rdg     <tibble [3,627 × 24]> m3    <S3: lm>
## 14 Science <tibble [1,435 × 24]> m3    <S3: lm>
## 15 Wri     <tibble [3,627 × 24]> m3    <S3: lm>
```

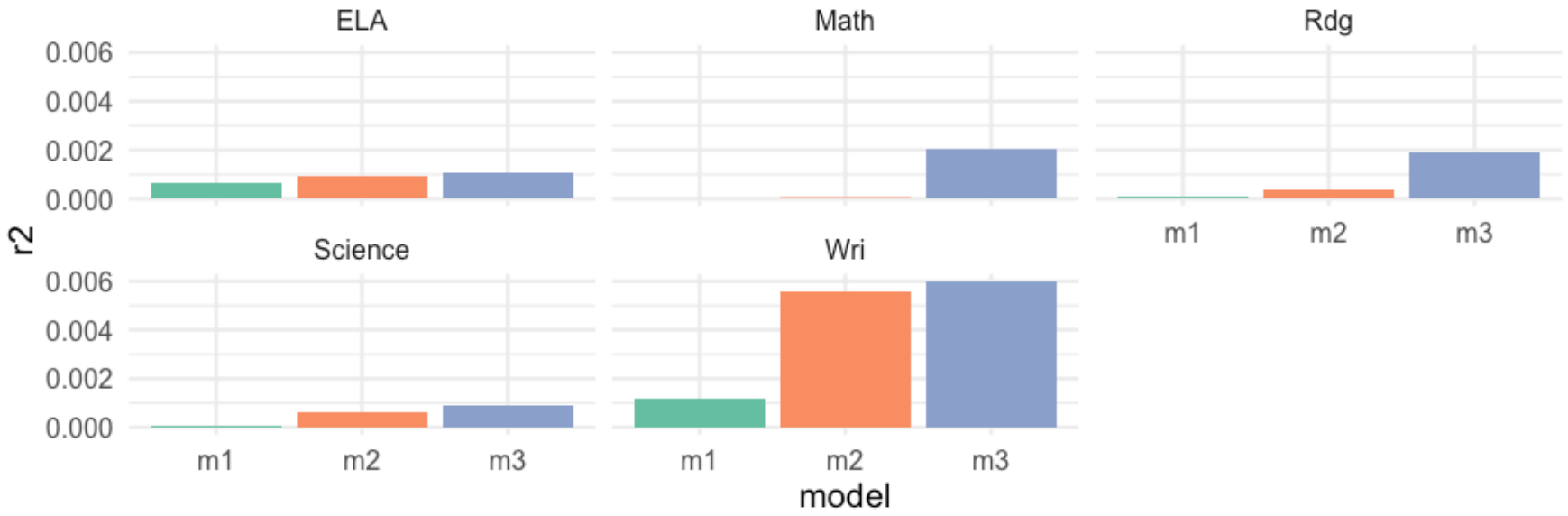
Extract all R^2

```
mods %>%
    gather(model, output, m1:m3) %>%
    mutate(r2 = map_dbl(output, ~summary(.x)$r.squared))
```

```
## # A tibble: 15 x 5
##    content data                    model output            r2
##    <chr>   <list>                  <chr> <list>         <dbl>
##  1 ELA     <tibble [3,627 × 24]> m1    <S3: lm> 6.394420e-4
##  2 Math    <tibble [3,629 × 24]> m1    <S3: lm> 3.151964e-7
##  3 Rdg     <tibble [3,627 × 24]> m1    <S3: lm> 7.138071e-5
##  4 Science <tibble [1,435 × 24]> m1    <S3: lm> 4.447519e-5
##  5 Wri     <tibble [3,627 × 24]> m1    <S3: lm> 1.162675e-3
##  6 ELA     <tibble [3,627 × 24]> m2    <S3: lm> 9.458086e-4
##  7 Math    <tibble [3,629 × 24]> m2    <S3: lm> 3.677477e-5
##  8 Rdg     <tibble [3,627 × 24]> m2    <S3: lm> 3.865820e-4
##  9 Science <tibble [1,435 × 24]> m2    <S3: lm> 6.533086e-4
## 10 Wri     <tibble [3,627 × 24]> m2    <S3: lm> 5.590844e-3
## 11 ELA     <tibble [3,627 × 24]> m3    <S3: lm> 1.035428e-3
## 12 Math    <tibble [3,629 × 24]> m3    <S3: lm> 2.056597e-3
## 13 Rdg     <tibble [3,627 × 24]> m3    <S3: lm> 1.889568e-3
## 14 Science <tibble [1,435 × 24]> m3    <S3: lm> 8.766337e-4
## 15 Wri     <tibble [3,627 × 24]> m3    <S3: lm> 6.002932e-3
```

Plot

```
mods %>%
    gather(model, output, m1:m3) %>%
    mutate(r2 = map_dbl(output, ~summary(.x)$r.squared)) %>%
ggplot(aes(model, r2)) +
    geom_col(aes(fill = model)) +
    facet_wrap(~content) +
    guides(fill = "none") +
    scale_fill_brewer(palette = "Set2")
```



Summary

- Batch processing is **really** powerful
- Much of the tools we've learned in the past can be applied once we get the data in a more workable format
- List columns are also **really** nice for organization and using our data frame toolkit

Summary

- Batch processing is **really** powerful
- Much of the tools we've learned in the past can be applied once we get the data in a more workable format
- List columns are also **really** nice for organization and using our data frame toolkit

Wednesday

- We'll talk more about list columns

Summary

- Batch processing is **really** powerful
- Much of the tools we've learned in the past can be applied once we get the data in a more workable format
- List columns are also **really** nice for organization and using our data frame toolkit

Wednesday

- We'll talk more about list columns

Any time left?

- Let's get started on the midterm