Data munging

Abhijit Dasgupta

March 25-27, 2019

Tidy data

Make the computer happy

Just as we want code to make humans happy, we want data to make computers happy

Tidy data is a principle promoted by Dr. Hadley Wickham to make computers happy

The properties of a tidy dataset are:

1. Each variable forms a column
2. Each observation forms a row
3. Each type of observational unit forms a table.

This forms a standardized way to structure a dataset, and so makes it easy for the analyst to develop standard pipelines.

A tidy dataset is tidy in one way, but a messy dataset can be messy in many ways

Hadley Wickham

Messy data

A dataset can be messy in many many ways. Many of the more common issues are listed below:

- Column names contain values, not just variable names
- Multiple variables are stored in one column
- Variables are stored in both rows and columns
- Multiple types of observational types are stored in the same table
- A single observational unit is stored in multiple tables

Sometimes the messier format is better for data entry, but bad for data analyses.

Variables in column names

```
library(tidyverse)
pew <- import('Data/FSI/pew.csv')
head(pew, 4)
```

```
  religion <$10k $10-20k $20-30k $30-40k $40-50k $50-75k $75-100k
1 Agnostic    27      34      60      81      76     137      122
2  Atheist    12      27      37      52      35      70       73
3 Buddhist    27      21      30      34      33      58       62
4 Catholic   418     617     732     670     638    1116      949
  $100-150k >150k Don't know/refused
1       109    84                 96
2        59    74                 76
3        39    53                 54
4       792   633               1489
```

- This dataset has actual data in the column headers, rather than variable names.
- We should ideally have 3 columns in this dataset: religion, income and frequency.
- We can achieve this using a function called `gather` which takes a wide dataset and makes it tall.

- Gather all the columns into two columns, `income` and `frequency`, by stacking the columns
- Don't include the variable `religion`

```
pew %>%
  gather(income, frequency, -religion) %>%
  as_tibble()
```

```
# A tibble: 180 x 3
   religion                income frequency
   <chr>                   <chr>      <int>
 1 Agnostic                <$10k         27
 2 Atheist                 <$10k         12
 3 Buddhist                <$10k         27
 4 Catholic                <$10k        418
 5 Don't know/refused      <$10k         15
 6 Evangelical Prot        <$10k        575
 7 Hindu                   <$10k          1
 8 Historically Black Prot <$10k        228
 9 Jehovah's Witness       <$10k         20
10 Jewish                  <$10k         19
# … with 170 more rows
```

```
head(pew)
```

```
            religion <$10k $10-20k $20-30k $30-40k $4
1           Agnostic    27      34      60      81
2            Atheist    12      27      37      52
3           Buddhist    27      21      30      34
4           Catholic   418     617     732     670
5 Don't know/refused    15      14      15      11
6   Evangelical Prot   575     869    1064     982
  $75-100k $100-150k >150k Don't know/refused
1      122       109    84                 96
2       73        59    74                 76
3       62        39    53                 54
4      949       792   633               1489
5       21        17    18                116
6      949       723   414               1529
```

```
pew %>%
  gather(income, frequency, -religion) %>% head(20)
```

```
                  religion  income frequency
1                 Agnostic   <$10k        27
2                  Atheist   <$10k        12
3                 Buddhist   <$10k        27
4                 Catholic   <$10k       418
5       Don't know/refused   <$10k        15
6         Evangelical Prot   <$10k       575
7                    Hindu   <$10k         1
8  Historically Black Prot   <$10k       228
9        Jehovah's Witness   <$10k        20
10                  Jewish   <$10k        19
11           Mainline Prot   <$10k       289
12                  Mormon   <$10k        29
13                  Muslim   <$10k         6
14                Orthodox   <$10k        13
15         Other Christian   <$10k         9
16            Other Faiths   <$10k        20
17   Other World Religions   <$10k         5
18             Unaffiliated   <$10k       217
19                Agnostic $10-20k        34
20                 Atheist $10-20k        27
```

Multiple variables in column names

```
tb <- import('Data/FSI/tb.csv') %>% as_tibble()
head(tb)
```

```
# A tibble: 6 x 22
  iso2   year   m04  m514  m014 m1524 m2534 m3544 m4554 m5564   m65    mu
  <chr> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
1 AD     1989    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
2 AD     1990    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
3 AD     1991    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
4 AD     1992    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
5 AD     1993    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
6 AD     1994    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
# … with 10 more variables: f04 <int>, f514 <int>, f014 <int>,
#   f1524 <int>, f2534 <int>, f3544 <int>, f4554 <int>, f5564 <int>,
#   f65 <int>, fu <int>
```

- column headers include both sex and age

```
tb %>%
  gather(sex_age, n, -iso2, -year, -fu)
```

```
# A tibble: 109,611 x 5
   iso2   year    fu sex_age     n
   <chr> <int> <int> <chr>   <int>
 1 AD     1989    NA m04        NA
 2 AD     1990    NA m04        NA
 3 AD     1991    NA m04        NA
 4 AD     1992    NA m04        NA
 5 AD     1993    NA m04        NA
 6 AD     1994    NA m04        NA
 7 AD     1996    NA m04        NA
 8 AD     1997    NA m04        NA
 9 AD     1998    NA m04        NA
10 AD     1999    NA m04        NA
# … with 109,601 more rows
```

```
tb %>%
  gather(sex_age, n, -iso2, -year, -fu, na.rm=T)
```

```
# A tibble: 35,478 x 5
   iso2   year    fu sex_age     n
   <chr> <int> <int> <chr>   <int>
 1 AD     2005     0 m04         0
 2 AD     2006     0 m04         0
 3 AD     2008     0 m04         0
 4 AE     2006    NA m04         0
 5 AE     2007    NA m04         0
 6 AE     2008     0 m04         0
 7 AG     2007    NA m04         0
 8 AL     2005     0 m04         0
 9 AL     2006     0 m04         1
10 AL     2007     0 m04         0
# … with 35,468 more rows
```

```
tb %>%
  gather(sex_age, n, -iso2, -year, -fu, na.rm=T) %>%
  separate(sex_age, c('sex','age'), sep=1) # by position
```

```
# A tibble: 35,478 x 6
   iso2   year    fu sex   age       n
   <chr> <int> <int> <chr> <chr> <int>
 1 AD     2005     0 m     04        0
 2 AD     2006     0 m     04        0
 3 AD     2008     0 m     04        0
 4 AE     2006    NA m     04        0
 5 AE     2007    NA m     04        0
 6 AE     2008     0 m     04        0
 7 AG     2007    NA m     04        0
 8 AL     2005     0 m     04        0
 9 AL     2006     0 m     04        1
10 AL     2007     0 m     04        0
# … with 35,468 more rows
```

This still needs to be cleaned

Variables stored in rows and columns

```
weather <- import('Data/FSI/weather.csv') %>% as_tibble()
weather
```

```
# A tibble: 22 x 35
   id     year month element    d1    d2    d3    d4    d5    d6    d7
   <chr> <int> <int> <chr>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
 1 MX17…  2010     1 tmax       NA    NA    NA    NA    NA    NA    NA
 2 MX17…  2010     1 tmin       NA    NA    NA    NA    NA    NA    NA
 3 MX17…  2010     2 tmax       NA  27.3  24.1    NA    NA    NA    NA
 4 MX17…  2010     2 tmin       NA  14.4  14.4    NA    NA    NA    NA
 5 MX17…  2010     3 tmax       NA    NA    NA    NA  32.1    NA    NA
 6 MX17…  2010     3 tmin       NA    NA    NA    NA  14.2    NA    NA
 7 MX17…  2010     4 tmax       NA    NA    NA    NA    NA    NA    NA
 8 MX17…  2010     4 tmin       NA    NA    NA    NA    NA    NA    NA
 9 MX17…  2010     5 tmax       NA    NA    NA    NA    NA    NA    NA
10 MX17…  2010     5 tmin       NA    NA    NA    NA    NA    NA    NA
# … with 12 more rows, and 24 more variables: d8 <dbl>, d9 <lgl>,
#   d10 <dbl>, d11 <dbl>, d12 <lgl>, d13 <dbl>, d14 <dbl>, d15 <dbl>,
#   d16 <dbl>, d17 <dbl>, d18 <lgl>, d19 <lgl>, d20 <lgl>, d21 <lgl>,
#   d22 <lgl>, d23 <dbl>, d24 <lgl>, d25 <dbl>, d26 <dbl>, d27 <dbl>,
#   d28 <dbl>, d29 <dbl>, d30 <dbl>, d31 <dbl>
```

```
weather %>%
  gather(day, temp, d1:d31)
```

```
# A tibble: 682 x 6
   id       year month element day    temp
   <chr>   <int> <int> <chr>   <chr> <dbl>
 1 MX17004  2010     1 tmax    d1       NA
 2 MX17004  2010     1 tmin    d1       NA
 3 MX17004  2010     2 tmax    d1       NA
 4 MX17004  2010     2 tmin    d1       NA
 5 MX17004  2010     3 tmax    d1       NA
 6 MX17004  2010     3 tmin    d1       NA
 7 MX17004  2010     4 tmax    d1       NA
 8 MX17004  2010     4 tmin    d1       NA
 9 MX17004  2010     5 tmax    d1       NA
10 MX17004  2010     5 tmin    d1       NA
# … with 672 more rows
```

`d1:d31` denotes all the variables physically between `d1` and `d31` in the dataset

> See what happens when you type `1:10` in the console

```
weather %>%
  gather(date, temp, d1:d31) %>%
  spread(element, temp)
```

```
# A tibble: 341 x 6
   id       year month date   tmax  tmin
   <chr>   <int> <int> <chr> <dbl> <dbl>
 1 MX17004  2010     1 d1       NA    NA
 2 MX17004  2010     1 d10      NA    NA
 3 MX17004  2010     1 d11      NA    NA
 4 MX17004  2010     1 d12      NA    NA
 5 MX17004  2010     1 d13      NA    NA
 6 MX17004  2010     1 d14      NA    NA
 7 MX17004  2010     1 d15      NA    NA
 8 MX17004  2010     1 d16      NA    NA
 9 MX17004  2010     1 d17      NA    NA
10 MX17004  2010     1 d18      NA    NA
# … with 331 more rows
```

Data cleaning

```
weather %>%
  gather(date, temp, d1:d31) %>%
  spread(element, temp)
```

```
# A tibble: 341 x 6
   id       year month date   tmax  tmin
   <chr>   <int> <int> <chr> <dbl> <dbl>
 1 MX17004  2010     1 d1       NA    NA
 2 MX17004  2010     1 d10      NA    NA
 3 MX17004  2010     1 d11      NA    NA
 4 MX17004  2010     1 d12      NA    NA
 5 MX17004  2010     1 d13      NA    NA
 6 MX17004  2010     1 d14      NA    NA
 7 MX17004  2010     1 d15      NA    NA
 8 MX17004  2010     1 d16      NA    NA
 9 MX17004  2010     1 d17      NA    NA
10 MX17004  2010     1 d18      NA    NA
# … with 331 more rows
```

- date column in alphabetical rather than numerical order

```
weather %>%
  gather(date, temp, d1:d31) %>%
  spread(element, temp) %>%
  mutate(date = parse_number(date))
```

```
# A tibble: 341 x 6
   id       year month  date  tmax  tmin
   <chr>   <int> <int> <dbl> <dbl> <dbl>
 1 MX17004  2010     1     1    NA    NA
 2 MX17004  2010     1    10    NA    NA
 3 MX17004  2010     1    11    NA    NA
 4 MX17004  2010     1    12    NA    NA
 5 MX17004  2010     1    13    NA    NA
 6 MX17004  2010     1    14    NA    NA
 7 MX17004  2010     1    15    NA    NA
 8 MX17004  2010     1    16    NA    NA
 9 MX17004  2010     1    17    NA    NA
10 MX17004  2010     1    18    NA    NA
# … with 331 more rows
```

```
weather %>%
  gather(date, temp, d1:d31) %>%
  spread(element, temp) %>%
  mutate(date = parse_number(date)) %>%
  arrange(date)
```

```
# A tibble: 341 x 6
   id       year month  date  tmax  tmin
   <chr>   <int> <int> <dbl> <dbl> <dbl>
 1 MX17004  2010     1     1    NA    NA
 2 MX17004  2010     2     1    NA    NA
 3 MX17004  2010     3     1    NA    NA
 4 MX17004  2010     4     1    NA    NA
 5 MX17004  2010     5     1    NA    NA
 6 MX17004  2010     6     1    NA    NA
 7 MX17004  2010     7     1    NA    NA
 8 MX17004  2010     8     1    NA    NA
 9 MX17004  2010    10     1    NA    NA
10 MX17004  2010    11     1    NA    NA
# … with 331 more rows
```

Not quite! We'd like dates ordered within months

```
weather %>%
  gather(date, temp, d1:d31) %>%
  spread(element, temp) %>%
  mutate(date = parse_number(date)) %>%
  arrange(month, date)
```

```
# A tibble: 341 x 6
   id       year month  date  tmax  tmin
   <chr>   <int> <int> <dbl> <dbl> <dbl>
 1 MX17004  2010     1     1    NA    NA
 2 MX17004  2010     1     2    NA    NA
 3 MX17004  2010     1     3    NA    NA
 4 MX17004  2010     1     4    NA    NA
 5 MX17004  2010     1     5    NA    NA
 6 MX17004  2010     1     6    NA    NA
 7 MX17004  2010     1     7    NA    NA
 8 MX17004  2010     1     8    NA    NA
 9 MX17004  2010     1     9    NA    NA
10 MX17004  2010     1    10    NA    NA
# … with 331 more rows
```

Good. Now to save it

```
weather2 <- weather %>%
  gather(date, temp, d1:d31) %>%
  spread(element, temp) %>%
  mutate(date = parse_number(date)) %>%
  arrange(month, date)
```

Exercise

The file `Data/FSI/mbta.xlsx` contains monthly data on number of commuter trips by different modalities on the MBTA system n Boston.

- It is in a messy format.
- It also has an additional quirk in that it has a title on the first line that isn't even data. You can avoid loading that in by using the option `skip=1` (i.e. skip the first line) when you import.

Work through this process to clean this dataset into tidy form. I'll also note that you can "minus" columns by position as well as name, so `gather(date, avg_trips, -1, -mode)` is valid to not involve the first column and the `mode` column.

```
mbta <- import('Data/FSI/mbta.xlsx', skip = 1) %>% as_tibble()
mbta2 <- mbta %>%
  gather(date, avg_trips, -1, -mode) %>%
  separate(date, c("year", "month"), sep = '-')
mbta2
```

```
# A tibble: 638 x 5
     ..1 mode              year  month avg_trips
   <dbl> <chr>             <chr> <chr> <chr>
 1     1 All Modes by Qtr  2007  01    NA
 2     2 Boat              2007  01    4
 3     3 Bus               2007  01    335.819
 4     4 Commuter Rail     2007  01    142.2
 5     5 Heavy Rail        2007  01    435.294
 6     6 Light Rail        2007  01    227.231
 7     7 Pct Chg / Yr      2007  01    0.02
 8     8 Private Bus       2007  01    4.772
 9     9 RIDE              2007  01    4.9
10    10 Trackless Trolley 2007  01    12.757
# … with 628 more rows
```

- `year`, `month`, `avg_trips` are all character variables
- There is an odd column named `..1`
- The rows with "All Modes by Qtr" and "TOTAL" aren't necessary

```
mbta2 %>%
  mutate(
    year = parse_number(year),
    month = parse_number(month),
    avg_trips = parse_number(avg_trips)
  )
```

```
# A tibble: 638 x 5
     ..1 mode                year month avg_trips
   <dbl> <chr>              <dbl> <dbl>     <dbl>
 1     1 All Modes by Qtr    2007     1     NA
 2     2 Boat                2007     1      4
 3     3 Bus                 2007     1    336.
 4     4 Commuter Rail       2007     1    142.
 5     5 Heavy Rail          2007     1    435.
 6     6 Light Rail          2007     1    227.
 7     7 Pct Chg / Yr        2007     1      0.02
 8     8 Private Bus         2007     1      4.77
 9     9 RIDE                2007     1      4.9
10    10 Trackless Trolley   2007     1     12.8
# … with 628 more rows
```

```
mbta2 %>%
  mutate(
    year = parse_number(year),
    month = parse_number(month),
    avg_trips = parse_number(avg_trips)
  ) %>%
  select(-1)
```

```
# A tibble: 638 x 4
   mode               year month avg_trips
   <chr>             <dbl> <dbl>     <dbl>
 1 All Modes by Qtr   2007     1     NA
 2 Boat               2007     1      4
 3 Bus                2007     1    336.
 4 Commuter Rail      2007     1    142.
 5 Heavy Rail         2007     1    435.
 6 Light Rail         2007     1    227.
 7 Pct Chg / Yr       2007     1      0.02
 8 Private Bus        2007     1      4.77
 9 RIDE               2007     1      4.9
10 Trackless Trolley  2007     1     12.8
# … with 628 more rows
```

```
mbta2 %>%
  mutate(
    year = parse_number(year),
    month = parse_number(month),
    avg_trips = parse_number(avg_trips)
  ) %>%
  select(-1) %>%
  filter(mode != 'TOTAL', mode != "All Modes by Qtr")
```

```
# A tibble: 522 x 4
   mode               year month avg_trips
   <chr>             <dbl> <dbl>     <dbl>
 1 Boat               2007     1      4
 2 Bus                2007     1    336.
 3 Commuter Rail      2007     1    142.
 4 Heavy Rail         2007     1    435.
 5 Light Rail         2007     1    227.
 6 Pct Chg / Yr       2007     1      0.02
 7 Private Bus        2007     1      4.77
 8 RIDE               2007     1      4.9
 9 Trackless Trolley  2007     1     12.8
10 Boat               2007     2      3.6
# … with 512 more rows
```

Other cleaning tasks

1. `distinct()` keeps the unique (non-duplicate) rows of a dataset. Usage: `dataset %>% distinct()`

2. If you want to keep only rows with complete data, you can invoke `drop_na`. Usage: `dataset %>% drop_na()`. You can modify `drop_na` by specifying variables from which you want to drop the missing values.

3. If you want to convert a value to missing (commonly 99 is used for missing data), then you can use `replace_na` within `mutate` to change to missing values on a column-by-column basis. Usage: `dataset %>% mutate(var1 = na_if(var1, 99))`

Cleaning Excel data

Excel is used as a visual medium

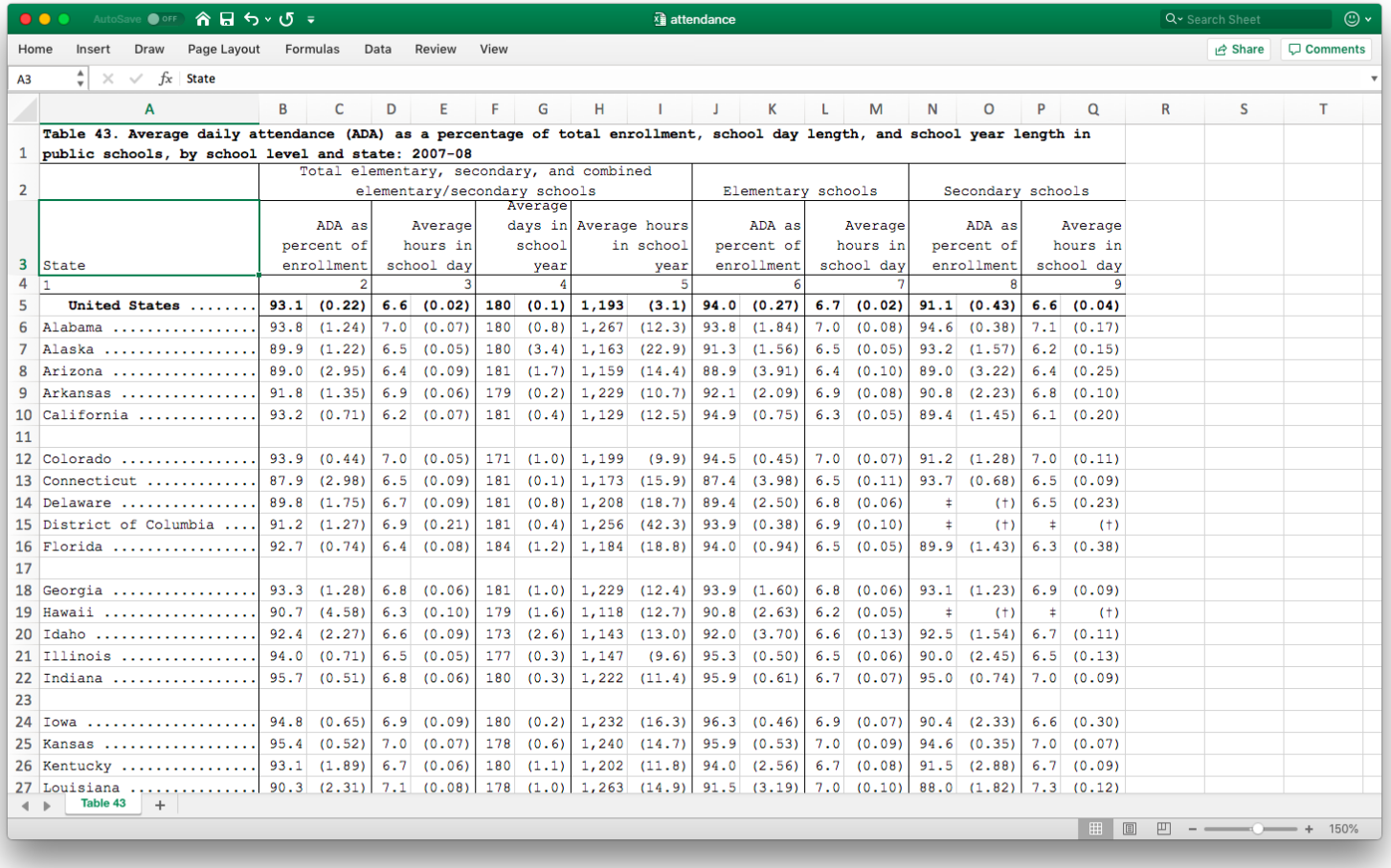
- Tables created to look good rather than being tidy or computer-friendly
  - Color being used to denote values of some variables
  - Multiple lines of headers
  - Multiple rows with variables
  - Typos leading to numeric variables become character

Excel is not reproducible, prone to mistakes by click

- Two special cases of Excel errors in the press
    - Duke cancer scandal with Dr. Anil Potti's group
    - Reinhart & Rogoff models for economic growth
- 35% of datasets in Nature (the journal) have Excel errors (The Economist, 2016)

> A gene named `1MAR` is entered in Excel. What does it become?

- Real data lies in the paired statistics and standard error columns
- The headers are basically different groupings and categories and should be variables



attendance

Home Insert Draw Page Layout Formulas Data Review View

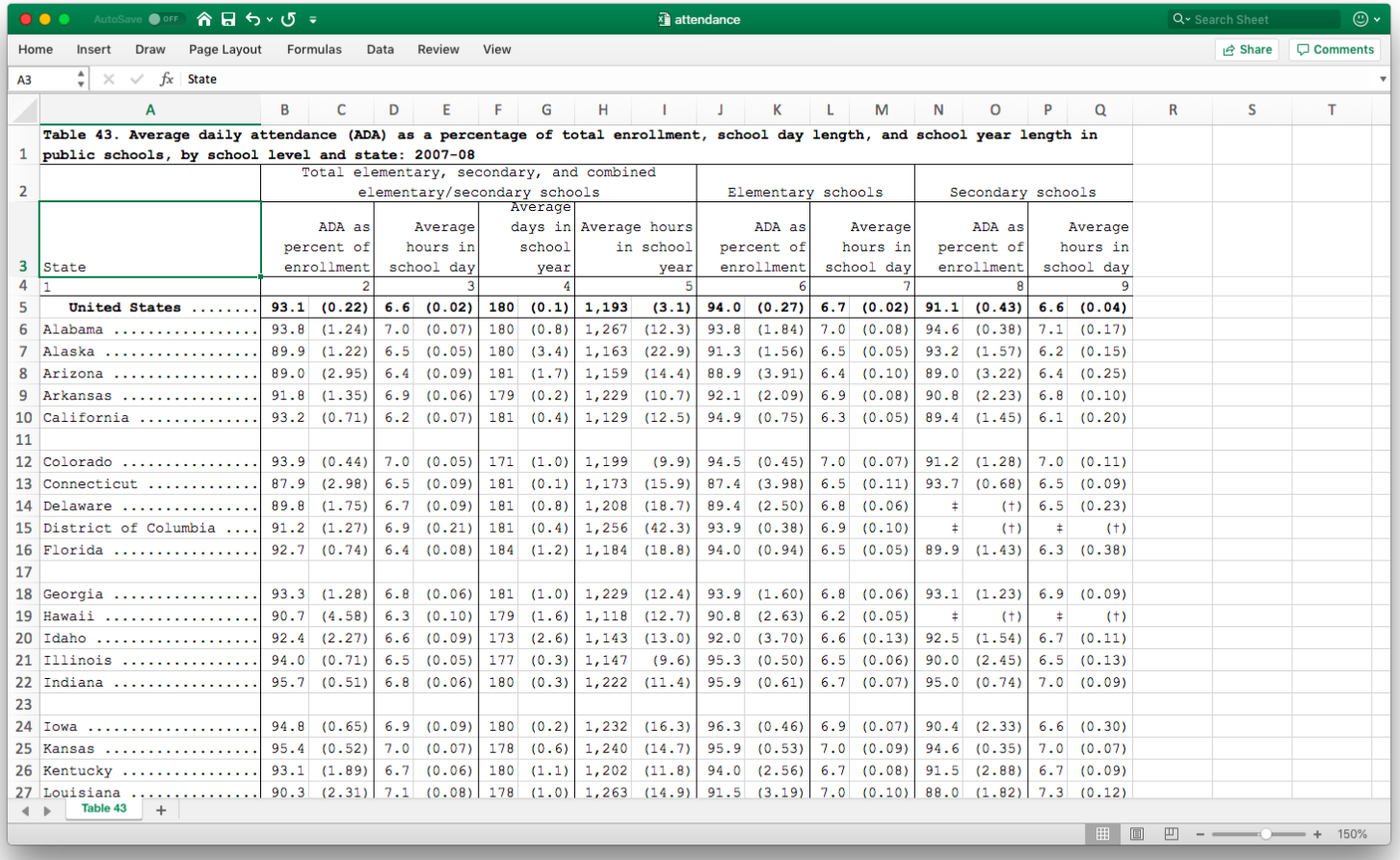
A3 State

Table 43. Average daily attendance (ADA) as a percentage of total enrollment, school day length, and school year length in public schools, by school level and state: 2007-08

State	Total elementary, secondary, and combined elementary/secondary schools				Elementary schools		Secondary schools	
	ADA as percent of enrollment	Average hours in school day	Average days in school year	Average hours in school year	ADA as percent of enrollment	Average hours in school day	ADA as percent of enrollment	Average hours in school day
1	2	3	4	5	6	7	8	9
United States	**93.1 (0.22)**	**6.6 (0.02)**	**180 (0.1)**	**1,193 (3.1)**	**94.0 (0.27)**	**6.7 (0.02)**	**91.1 (0.43)**	**6.6 (0.04)**
Alabama	93.8 (1.24)	7.0 (0.07)	180 (0.8)	1,267 (12.3)	93.8 (1.84)	7.0 (0.08)	94.6 (0.38)	7.1 (0.17)
Alaska	89.9 (1.22)	6.5 (0.05)	180 (3.4)	1,163 (22.9)	91.3 (1.56)	6.5 (0.05)	93.2 (1.57)	6.2 (0.15)
Arizona	89.0 (2.95)	6.4 (0.09)	181 (1.7)	1,159 (14.4)	88.9 (3.91)	6.4 (0.10)	89.0 (3.22)	6.4 (0.25)
Arkansas	91.8 (1.35)	6.9 (0.06)	179 (0.2)	1,229 (10.7)	92.1 (2.09)	6.9 (0.08)	90.8 (2.23)	6.8 (0.10)
California	93.2 (0.71)	6.2 (0.07)	181 (0.4)	1,129 (12.5)	94.9 (0.75)	6.3 (0.05)	89.4 (1.45)	6.1 (0.20)
Colorado	93.9 (0.44)	7.0 (0.05)	171 (1.0)	1,199 (9.9)	94.5 (0.45)	7.0 (0.07)	91.2 (1.28)	7.0 (0.11)
Connecticut	87.9 (2.98)	6.5 (0.09)	181 (0.1)	1,173 (15.9)	87.4 (3.98)	6.5 (0.11)	93.7 (0.68)	6.5 (0.09)
Delaware	89.8 (1.75)	6.7 (0.09)	181 (0.8)	1,208 (18.7)	89.4 (2.50)	6.8 (0.06)	‡ (†)	6.5 (0.23)
District of Columbia	91.2 (1.27)	6.9 (0.21)	181 (0.4)	1,256 (42.3)	93.9 (0.38)	6.9 (0.10)	‡ (†)	‡ (†)
Florida	92.7 (0.74)	6.4 (0.08)	184 (1.2)	1,184 (18.8)	94.0 (0.94)	6.5 (0.05)	89.9 (1.43)	6.3 (0.38)
Georgia	93.3 (1.28)	6.8 (0.06)	181 (1.0)	1,229 (12.4)	93.9 (1.60)	6.8 (0.06)	93.1 (1.23)	6.9 (0.09)
Hawaii	90.7 (4.58)	6.3 (0.10)	179 (1.6)	1,118 (12.7)	90.8 (2.63)	6.2 (0.05)	‡ (†)	‡ (†)
Idaho	92.4 (2.27)	6.6 (0.09)	173 (2.6)	1,143 (13.0)	92.0 (3.70)	6.6 (0.13)	92.5 (1.54)	6.7 (0.11)
Illinois	94.0 (0.71)	6.5 (0.05)	177 (0.3)	1,147 (9.6)	95.3 (0.50)	6.5 (0.06)	90.0 (2.45)	6.5 (0.13)
Indiana	95.7 (0.51)	6.8 (0.06)	180 (0.3)	1,222 (11.4)	95.9 (0.61)	6.7 (0.07)	95.0 (0.74)	7.0 (0.09)
Iowa	94.8 (0.65)	6.9 (0.09)	180 (0.2)	1,232 (16.3)	96.3 (0.46)	6.9 (0.07)	90.4 (2.33)	6.6 (0.30)
Kansas	95.4 (0.52)	7.0 (0.07)	178 (0.6)	1,240 (14.7)	95.9 (0.53)	7.0 (0.09)	94.6 (0.35)	7.0 (0.07)
Kentucky	93.1 (1.89)	6.7 (0.06)	180 (1.1)	1,202 (11.8)	94.0 (2.56)	6.7 (0.08)	91.5 (2.88)	6.7 (0.09)
Louisiana	90.3 (2.31)	7.1 (0.08)	178 (1.0)	1,263 (14.9)	91.5 (3.19)	7.0 (0.10)	88.0 (1.82)	7.3 (0.12)

Table 43

- `import` fails horribly
- Two packages, `tidyxl` and `unpivotr`, by Duncan Garmonsway, save the day



attendance

Table 43. Average daily attendance (ADA) as a percentage of total enrollment, school day length, and school year length in public schools, by school level and state: 2007-08

State	Total elementary, secondary, and combined elementary/secondary schools						Elementary schools				Secondary schools			
	ADA as percent of enrollment		Average hours in school day		Average days in school year	Average hours in school year	ADA as percent of enrollment		Average hours in school day		ADA as percent of enrollment		Average hours in school day	
1	2		3		4	5	6		7		8		9	
United States	**93.1**	**(0.22)**	**6.6**	**(0.02)**	**180**	**(0.1)**	**1,193**	**(3.1)**	**94.0**	**(0.27)**	**6.7**	**(0.02)**	**91.1**	**(0.43)**	**6.6**	**(0.04)**
Alabama	93.8	(1.24)	7.0	(0.07)	180	(0.8)	1,267	(12.3)	93.8	(1.84)	7.0	(0.08)	94.6	(0.38)	7.1	(0.17)
Alaska	89.9	(1.22)	6.5	(0.05)	180	(3.4)	1,163	(22.9)	91.3	(1.56)	6.5	(0.05)	93.2	(1.57)	6.2	(0.15)
Arizona	89.0	(2.95)	6.4	(0.09)	181	(1.7)	1,159	(14.4)	88.9	(3.91)	6.4	(0.10)	89.0	(3.22)	6.4	(0.25)
Arkansas	91.8	(1.35)	6.9	(0.06)	179	(0.2)	1,229	(10.7)	92.1	(2.09)	6.9	(0.08)	90.8	(2.23)	6.8	(0.10)
California	93.2	(0.71)	6.2	(0.07)	181	(0.4)	1,129	(12.5)	94.9	(0.75)	6.3	(0.05)	89.4	(1.45)	6.1	(0.20)
Colorado	93.9	(0.44)	7.0	(0.05)	171	(1.0)	1,199	(9.9)	94.5	(0.45)	7.0	(0.07)	91.2	(1.28)	7.0	(0.11)
Connecticut	87.9	(2.98)	6.5	(0.09)	181	(0.1)	1,173	(15.9)	87.4	(3.98)	6.5	(0.11)	93.7	(0.68)	6.5	(0.09)
Delaware	89.8	(1.75)	6.7	(0.09)	181	(0.8)	1,208	(18.7)	89.4	(2.50)	6.8	(0.06)	‡	(†)	6.5	(0.23)
District of Columbia	91.2	(1.27)	6.9	(0.21)	181	(0.4)	1,256	(42.3)	93.9	(0.38)	6.9	(0.10)	‡	(†)	‡	(†)
Florida	92.7	(0.74)	6.4	(0.08)	184	(1.2)	1,184	(18.8)	94.0	(0.94)	6.5	(0.05)	89.9	(1.43)	6.3	(0.38)
Georgia	93.3	(1.28)	6.8	(0.06)	181	(1.0)	1,229	(12.4)	93.9	(1.60)	6.8	(0.06)	93.1	(1.23)	6.9	(0.09)
Hawaii	90.7	(4.58)	6.3	(0.10)	179	(1.6)	1,118	(12.7)	90.8	(2.63)	6.2	(0.05)	‡	(†)	‡	(†)
Idaho	92.4	(2.27)	6.6	(0.09)	173	(2.6)	1,143	(13.0)	92.0	(3.70)	6.6	(0.13)	92.5	(1.54)	6.7	(0.11)
Illinois	94.0	(0.71)	6.5	(0.05)	177	(0.3)	1,147	(9.6)	95.3	(0.50)	6.5	(0.06)	90.0	(2.45)	6.5	(0.13)
Indiana	95.7	(0.51)	6.8	(0.06)	180	(0.3)	1,222	(11.4)	95.9	(0.61)	6.7	(0.07)	95.0	(0.74)	7.0	(0.09)
Iowa	94.8	(0.65)	6.9	(0.09)	180	(0.2)	1,232	(16.3)	96.3	(0.46)	6.9	(0.07)	90.4	(2.33)	6.6	(0.30)
Kansas	95.4	(0.52)	7.0	(0.07)	178	(0.6)	1,240	(14.7)	95.9	(0.53)	7.0	(0.09)	94.6	(0.35)	7.0	(0.07)
Kentucky	93.1	(1.89)	6.7	(0.06)	180	(1.1)	1,202	(11.8)	94.0	(2.56)	6.7	(0.08)	91.5	(2.88)	6.7	(0.09)
Louisiana	90.3	(2.31)	7.1	(0.08)	178	(1.0)	1,263	(14.9)	91.5	(3.19)	7.0	(0.10)	88.0	(1.82)	7.3	(0.12)

```
library(tidyxl)
dataset1 <- xlsx_cells('Data/FSI/attendance.xlsx')
dataset1
```

```
# A tibble: 1,173 x 21
   sheet address   row   col is_blank data_type error logical numeric
   <chr> <chr>   <int> <int> <lgl>    <chr>     <chr> <lgl>     <dbl>
 1 Tabl… A1          1     1 FALSE    character <NA>  NA           NA
 2 Tabl… B1          1     2 TRUE     blank     <NA>  NA           NA
 3 Tabl… C1          1     3 TRUE     blank     <NA>  NA           NA
 4 Tabl… D1          1     4 TRUE     blank     <NA>  NA           NA
 5 Tabl… E1          1     5 TRUE     blank     <NA>  NA           NA
 6 Tabl… F1          1     6 TRUE     blank     <NA>  NA           NA
 7 Tabl… G1          1     7 TRUE     blank     <NA>  NA           NA
 8 Tabl… H1          1     8 TRUE     blank     <NA>  NA           NA
 9 Tabl… I1          1     9 TRUE     blank     <NA>  NA           NA
10 Tabl… J1          1    10 TRUE     blank     <NA>  NA           NA
# … with 1,163 more rows, and 12 more variables: date <dttm>,
#   character <chr>, character_formatted <list>, formula <chr>,
#   is_array <lgl>, formula_ref <chr>, formula_group <int>, comment <chr>,
#   height <dbl>, width <dbl>, style_format <chr>, local_format_id <int>
```

- This grabs a bunch of meta-data about the Excel entries, including color and formatting features
- The data has been blown up on a cell-by-cell basis
- Use `tidyverse` tools to fix this? Nope. `unpivotr` is more powerful in this case.

```
library(unpivotr)
```

```
Warning: package 'unpivotr' was built under R version 3.5.2
```

```
dataset1 %>%
  filter(row != 1, row != 4, row < 65) %>%
  behead('N', tophead) %>%
  behead('N', head2) %>%
  behead('W', State) %>%
  select(row, col, data_type, numeric, tophead, head2, State)
```

```
# A tibble: 960 x 7
     row   col data_type  numeric tophead                head2     State
   <int> <int> <chr>        <dbl> <chr>                  <chr>     <chr>
 1     5     2 numeric    9.31e+1 Total elementary, se… ADA as p… "   Unit…
 2     5     3 numeric    2.19e-1 <NA>                   <NA>      "   Unit…
 3     5     4 numeric    6.64e+0 <NA>                   Average … "   Unit…
 4     5     5 numeric    1.76e-2 <NA>                   <NA>      "   Unit…
 5     5     6 numeric    1.80e+2 <NA>                   Average … "   Unit…
 6     5     7 numeric    1.43e-1 <NA>                   <NA>      "   Unit…
 7     5     8 numeric    1.19e+3 <NA>                   Average … "   Unit…
 8     5     9 numeric    3.09e+0 <NA>                   <NA>      "   Unit…
 9     5    10 numeric    9.40e+1 Elementary schools     ADA as p… "   Unit…
10     5    11 numeric    2.69e-1 <NA>                   <NA>      "   Unit…
# … with 950 more rows
```

- Pull off the two headers first with behead. Tell the function what direction (N, W, S, E or angles) the header is sitting in relation to the data

```
library(unpivotr)
dataset1 %>%
  filter(row != 1, row != 4, row < 65) %>%
  behead('N', tophead) %>%
  behead('N', head2) %>%
  behead('W', State) %>%
  select(row, col, data_type, numeric, tophead, head2, State) %>%
  mutate(header = ifelse(col %% 2 == 0, 'stats','se'))
```

```
# A tibble: 960 x 8
     row   col data_type  numeric tophead          head2    State    header
   <int> <int> <chr>        <dbl> <chr>            <chr>    <chr>    <chr>
 1     5     2 numeric    9.31e+1 Total elementar… ADA as … "   Uni… stats
 2     5     3 numeric    2.19e-1 <NA>             <NA>     "   Uni… se
 3     5     4 numeric    6.64e+0 <NA>             Average… "   Uni… stats
 4     5     5 numeric    1.76e-2 <NA>             <NA>     "   Uni… se
 5     5     6 numeric    1.80e+2 <NA>             Average… "   Uni… stats
 6     5     7 numeric    1.43e-1 <NA>             <NA>     "   Uni… se
 7     5     8 numeric    1.19e+3 <NA>             Average… "   Uni… stats
 8     5     9 numeric    3.09e+0 <NA>             <NA>     "   Uni… se
 9     5    10 numeric    9.40e+1 Elementary scho… ADA as … "   Uni… stats
10     5    11 numeric    2.69e-1 <NA>             <NA>     "   Uni… se
# … with 950 more rows
```

- even columns are stats, odd columns are standard errors
- %% gives the remainder when left side is divided by right side

```
library(unpivotr)
dataset1 %>%
  filter(row != 1, row != 4, row < 65) %>%
  behead('N', tophead) %>%
  behead('N', head2) %>%
  behead('W', State) %>%
  select(row, col, data_type, numeric, tophead, head2, State) %>%
  mutate(header = ifelse(col %% 2 == 0, 'stats','se')) %>%
  fill(tophead) %>% fill(head2)
```

```
# A tibble: 960 x 8
     row   col data_type  numeric tophead          head2    State    header
   <int> <int> <chr>        <dbl> <chr>            <chr>    <chr>    <chr>
 1     5     2 numeric    9.31e+1 Total elementar… ADA as … "   Uni… stats
 2     5     3 numeric    2.19e-1 Total elementar… ADA as … "   Uni… se
 3     5     4 numeric    6.64e+0 Total elementar… Average… "   Uni… stats
 4     5     5 numeric    1.76e-2 Total elementar… Average… "   Uni… se
 5     5     6 numeric    1.80e+2 Total elementar… Average… "   Uni… stats
 6     5     7 numeric    1.43e-1 Total elementar… Average… "   Uni… se
 7     5     8 numeric    1.19e+3 Total elementar… Average… "   Uni… stats
 8     5     9 numeric    3.09e+0 Total elementar… Average… "   Uni… se
 9     5    10 numeric    9.40e+1 Elementary scho… ADA as … "   Uni… stats
10     5    11 numeric    2.69e-1 Elementary scho… ADA as … "   Uni… se
# … with 950 more rows
```

- column headers spanned several columns visually, but rested in left-most column internally
- used *last value carried forward* to fill in the other columns

```
library(unpivotr)
tidy_dataset <- dataset1 %>%
  filter(row != 1, row != 4, row < 65) %>%
  behead('N', tophead) %>%
  behead('N', head2) %>%
  behead('W', State) %>%
  select(row, col, data_type, numeric, tophead, head2, State) %>%
  mutate(header = ifelse(col %% 2 == 0, 'stats','se')) %>%
  fill(tophead) %>% fill(head2) %>%
  select(-col) %>%
  spatter(header, numeric) %>%
  select(-row)
tidy_dataset
```

```
# A tibble: 480 x 5
   tophead                          head2          State            se  stats
   <chr>                            <chr>          <chr>         <dbl>  <dbl>
 1 Elementary schools               ADA as percen… "   United … 0.269  9.40e1
 2 Elementary schools               Average hours… "   United … 0.0160 6.66e0
 3 Secondary schools                ADA as percen… "   United … 0.432  9.11e1
 4 Secondary schools                Average hours… "   United … 0.0403 6.59e0
 5 Total elementary, secondary, … ADA as percen… "   United … 0.219  9.31e1
 6 Total elementary, secondary, … Average days … "   United … 0.143  1.80e2
 7 Total elementary, secondary, … Average hours… "   United … 0.0176 6.64e0
 8 Total elementary, secondary, … Average hours… "   United … 3.09   1.19e3
 9 Elementary schools               ADA as percen… Alabama ...… 1.84   9.38e1
10 Elementary schools               Average hours… Alabama ...… 0.0759 7.04e0
# … with 470 more rows
```

- `spatter` works like `spread`, but is more robust for this kind of weird data

```
tidy_dataset <- tidy_dataset %>%
  mutate(State = str_remove(State, '\\.+')) %>%
  mutate(State = str_trim(State))
tidy_dataset
```

```
# A tibble: 480 x 5
   tophead                          head2           State        se  stats
   <chr>                            <chr>           <chr>     <dbl>  <dbl>
 1 Elementary schools               ADA as percent… United … 0.269  9.40e1
 2 Elementary schools               Average hours … United … 0.0160 6.66e0
 3 Secondary schools                ADA as percent… United … 0.432  9.11e1
 4 Secondary schools                Average hours … United … 0.0403 6.59e0
 5 Total elementary, secondary, and… ADA as percent… United … 0.219  9.31e1
 6 Total elementary, secondary, and… Average days i… United … 0.143  1.80e2
 7 Total elementary, secondary, and… Average hours … United … 0.0176 6.64e0
 8 Total elementary, secondary, and… Average hours … United … 3.09   1.19e3
 9 Elementary schools               ADA as percent… Alabama  1.84   9.38e1
10 Elementary schools               Average hours … Alabama  0.0759 7.04e0
# … with 470 more rows
```

- We're using a **regular expression** to identify and remove all the dots
    - Rich tool for text searching
- Next we trim away any white space that is around the string

Save the data

```
saveRDS(tidy_dataset, file = 'Data/FSI/schools.rds')
```

The RDS format is an open standard and a fast way to store and retrieve datasets in R

What about being colorful?

```
library(tidyxl)
library(unpivotr)

dataset2 <- xlsx_cells('Data/FSI/classlist.xlsx')
formats <- xlsx_formats('Data/FSI/classlist.xlsx')
```

We need to grab the formats too, now

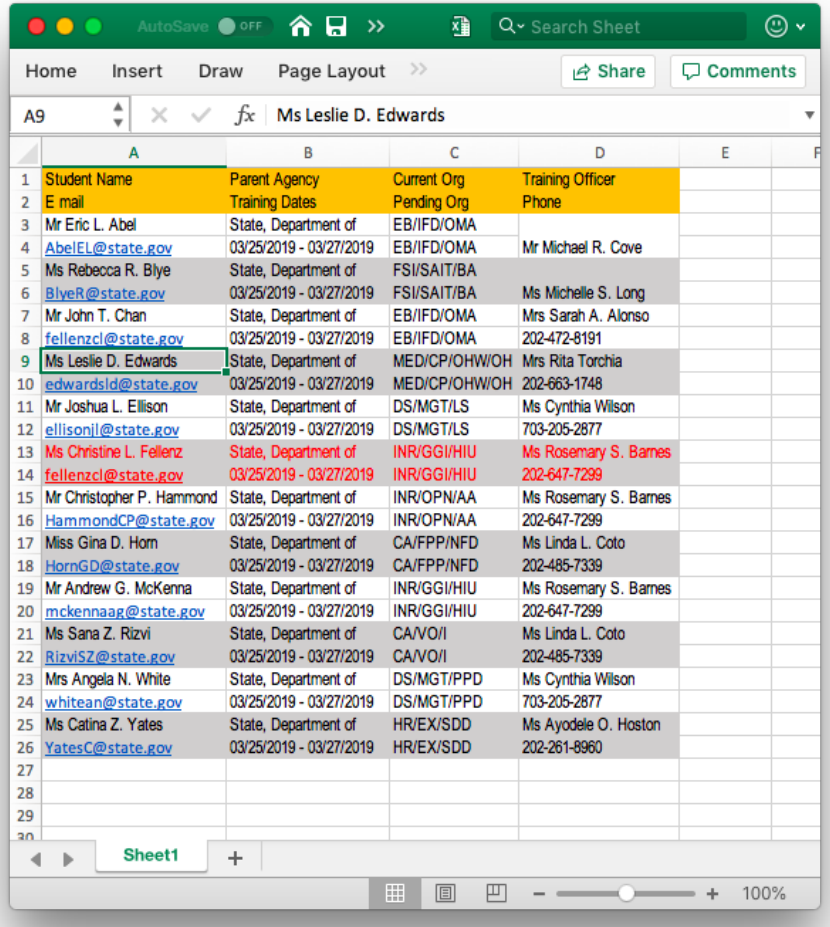
```
format_id <- dataset2$local_format_id
dataset2$font_color <- formats$local$font$color$rgb[f
dataset2$bg_color <- formats$local$fill$patternFill$f

unique(dataset2$font_color)
```

```
[1] "FF000000" "FF0563C1" "FFFF0000"
```

```
unique(dataset2$bg_color)
```

```
[1] "FFFFC000" NA         "FFE7E6E6"
```



	A	B	C	D
1	Student Name	Parent Agency	Current Org	Training Officer
2	E mail	Training Dates	Pending Org	Phone
3	Mr Eric L. Abel	State, Department of	EB/IFD/OMA	
4	AbelEL@state.gov	03/25/2019 - 03/27/2019	EB/IFD/OMA	Mr Michael R. Cove
5	Ms Rebecca R. Blye	State, Department of	FSI/SAIT/BA	
6	BlyeR@state.gov	03/25/2019 - 03/27/2019	FSI/SAIT/BA	Ms Michelle S. Long
7	Mr John T. Chan	State, Department of	EB/IFD/OMA	Mrs Sarah A. Alonso
8	fellenzcl@state.gov	03/25/2019 - 03/27/2019	EB/IFD/OMA	202-472-8191
9	Ms Leslie D. Edwards	State, Department of	MED/CP/OHW/OH	Mrs Rita Torchia
10	edwardsld@state.gov	03/25/2019 - 03/27/2019	MED/CP/OHW/OH	202-663-1748
11	Mr Joshua L. Ellison	State, Department of	DS/MGT/LS	Ms Cynthia Wilson
12	ellisonjl@state.gov	03/25/2019 - 03/27/2019	DS/MGT/LS	703-205-2877
13	Ms Christine L. Fellenz	State, Department of	INR/GGI/HIU	Ms Rosemary S. Barnes
14	fellenzcl@state.gov	03/25/2019 - 03/27/2019	INR/GGI/HIU	202-647-7299
15	Mr Christopher P. Hammond	State, Department of	INR/OPN/AA	Ms Rosemary S. Barnes
16	HammondCP@state.gov	03/25/2019 - 03/27/2019	INR/OPN/AA	202-647-7299
17	Miss Gina D. Horn	State, Department of	CA/FPP/NFD	Ms Linda L. Coto
18	HornGD@state.gov	03/25/2019 - 03/27/2019	CA/FPP/NFD	202-485-7339
19	Mr Andrew G. McKenna	State, Department of	INR/GGI/HIU	Ms Rosemary S. Barnes
20	mckennaag@state.gov	03/25/2019 - 03/27/2019	INR/GGI/HIU	202-647-7299
21	Ms Sana Z. Rizvi	State, Department of	CA/VO/I	Ms Linda L. Coto
22	RizviSZ@state.gov	03/25/2019 - 03/27/2019	CA/VO/I	202-485-7339
23	Mrs Angela N. White	State, Department of	DS/MGT/PPD	Ms Cynthia Wilson
24	whitean@state.gov	03/25/2019 - 03/27/2019	DS/MGT/PPD	703-205-2877
25	Ms Catina Z. Yates	State, Department of	HR/EX/SDD	Ms Ayodele O. Hoston
26	YatesC@state.gov	03/25/2019 - 03/27/2019	HR/EX/SDD	202-261-8960
27				
28				
29				

Grab the red rows

```
red_rows <- dataset2 %>% filter(font_color=='FFFF0000') %>%
  select(row, col, data_type, character) %>%
  mutate(row=2, col = 1:8)
headers <- dataset2 %>% filter(bg_color == 'FFFFC000') %>%
  select(row, col, data_type, character) %>%
  mutate(row = 1, col = 1:8)

bind_rows(headers, red_rows) %>%
  behead('N', header) %>%
  select(-col) %>%
  spatter(header) %>%
  select(-row)
```

```
# A tibble: 1 x 8
  `Current Org` `E mail` `Parent Agency` `Pending Org` Phone `Student Name`
  <chr>         <chr>    <chr>           <chr>         <chr> <chr>
1 INR/GGI/HIU   fellenz… State, Departm… INR/GGI/HIU   202-… Ms Christine …
# … with 2 more variables: `Training Dates` <chr>, `Training
#   Officer` <chr>
```

Tidying this data

There are really two datasets interwoven here

- The odd rows form one dataset
- The even rows form another dataset

We need to put these two datasets side-by-side

Tidying this data

```
dat1 <- dataset2 %>%
  filter( row %% 2 == 1) %>% # odd rows
  behead('N', header) %>%
  mutate(row = (row+1)/2) # make the row numbers sequ

dat2 <- dataset2 %>%
  filter(row %% 2 == 0) %>%  # even rows
  behead('N', header) %>%
  mutate(row = row/2) %>% # make row numbers sequenti
  mutate(col = col+4) # These will be the last 4 cols

tidy_dataset2 <-
  rbind(dat1, dat2) %>% # Put datsets on top of each
  select(row, data_type, numeric, character, header)
  spatter(header) %>%
  select(-row, -numeric)
```

Tidying this data

```
dat1 <- dataset2 %>%
  filter( row %% 2 == 1) %>% # odd rows
  behead('N', header) %>%
  mutate(row = (row+1)/2) # make the row numbers sequ

dat2 <- dataset2 %>%
  filter(row %% 2 == 0) %>%  # even rows
  behead('N', header) %>%
  mutate(row = row/2) %>% # make row numbers sequenti
  mutate(col = col+4) # These will be the last 4 cols

tidy_dataset2 <-
  rbind(dat1, dat2) %>% # Put datsets on top of each
  select(row, data_type, numeric, character, header)
  spatter(header) %>%
  select(-row, -numeric)
```

```
# A tibble: 12 x 8
   `Current Org` `E mail` `Parent Agency` `Pending Or
   <chr>         <chr>    <chr>           <chr>
 1 EB/IFD/OMA    AbelEL@… State, Departm… EB/IFD/OMA
 2 FSI/SAIT/BA   BlyeR@s… State, Departm… FSI/SAIT/BA
 3 EB/IFD/OMA    fellenz… State, Departm… EB/IFD/OMA
 4 MED/CP/OHW/OH edwards… State, Departm… MED/CP/OHW/
 5 DS/MGT/LS     ellison… State, Departm… DS/MGT/LS
 6 INR/GGI/HIU   fellenz… State, Departm… INR/GGI/HIU
 7 INR/OPN/AA    Hammond… State, Departm… INR/OPN/AA
 8 CA/FPP/NFD    HornGD@… State, Departm… CA/FPP/NFD
 9 INR/GGI/HIU   mckenna… State, Departm… INR/GGI/HIU
10 CA/VO/I       RizviSZ… State, Departm… CA/VO/I
11 DS/MGT/PPD    whitean… State, Departm… DS/MGT/PPD
12 HR/EX/SDD     YatesC@… State, Departm… HR/EX/SDD
# … with 3 more variables: `Student Name` <chr>, `Tra
#   `Training Officer` <chr>
```

Tidying this data

```
tidy_dataset2 <- tidy_dataset2 %>%
  set_names(make.names(names(.))) %>%
  select(Student.Name, everything())
```

```
# A tibble: 12 x 8
   Student.Name Current.Org E.mail Parent.Agency Pending.Org Phone
   <chr>        <chr>       <chr>  <chr>         <chr>       <chr>
 1 Mr Eric L. … EB/IFD/OMA  AbelE… State, Depar… EB/IFD/OMA  <NA>
 2 Ms Rebecca … FSI/SAIT/BA BlyeR… State, Depar… FSI/SAIT/BA <NA>
 3 Mr John T. … EB/IFD/OMA  felle… State, Depar… EB/IFD/OMA  202-…
 4 Ms Leslie D… MED/CP/OHW… edwar… State, Depar… MED/CP/OHW… 202-…
 5 Mr Joshua L… DS/MGT/LS   ellis… State, Depar… DS/MGT/LS   703-…
 6 Ms Christin… INR/GGI/HIU felle… State, Depar… INR/GGI/HIU 202-…
 7 Mr Christop… INR/OPN/AA  Hammo… State, Depar… INR/OPN/AA  202-…
 8 Miss Gina D… CA/FPP/NFD  HornG… State, Depar… CA/FPP/NFD  202-…
 9 Mr Andrew G… INR/GGI/HIU mcken… State, Depar… INR/GGI/HIU 202-…
10 Ms Sana Z. … CA/VO/I     Rizvi… State, Depar… CA/VO/I     202-…
11 Mrs Angela … DS/MGT/PPD  white… State, Depar… DS/MGT/PPD  703-…
12 Ms Catina Z… HR/EX/SDD   Yates… State, Depar… HR/EX/SDD   202-…
# … with 2 more variables: Training.Dates <chr>, Training.Officer <chr>
```