

File I/O and basic data wrangling

John Tipton

Readings

- R for data science
 - Introduction
 - Chapters 3 (Data transformation with `dplyr`), 7 (Tibbles with `tibble`), 8 (Data import with `readr`), and 9 (Tidy data with `tidyr`)

The `dplyr` package

- The `dplyr` package can be loaded as part of the `tidyverse` library

```
library(tidyverse)
```

- Namespaces
 - Different modules might have the same name -- how does the computer know which function you meant?
 - In python, you can use the `mean()` function from the `statistics` module using `statistics.mean()`
 - In R, namespaces are defined using `::`
 - `dplyr::filter()` uses the `filter()` function in the `dplyr()` library

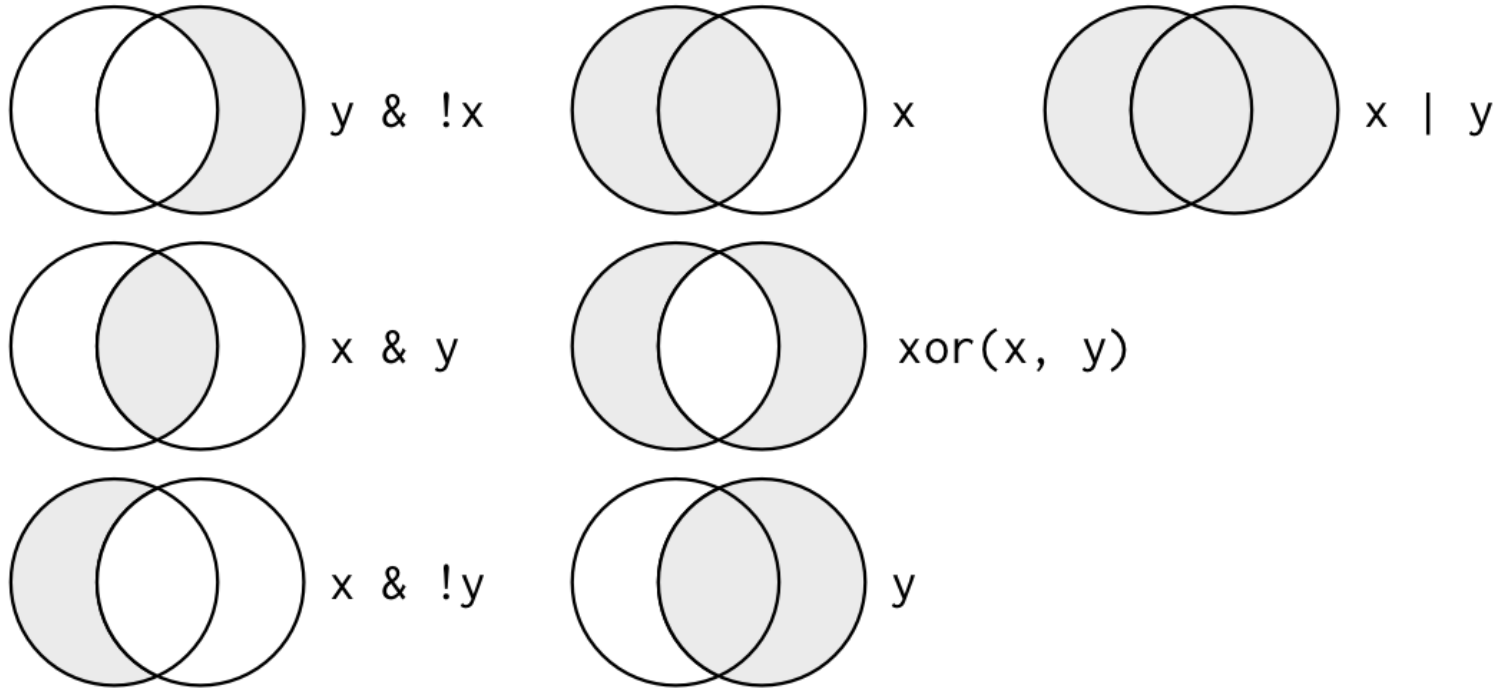
dplyr functions

- The big 5:
 - Choose observations (rows) based on conditional values with `filter()`
 - Reorder the variables with `arrange()`
 - Select variables by name with `select()`
 - Create new variables with `mutate()`
 - Create summary variables with `summarize()`
 - `summarise()` if you are British

Logical operators

- Test if A is greater than B `A > B`
- Test if A is less than B `A < B`
- Test if A is greater than or equal to B `A >= B`
- Test if A is less than or equal to B `A <= B`
- Test if A is equal to B `A == B`
- Test if A is not equal to B `A != B`
- Return TRUE if both A and B are TRUE `A & B`
- Return TRUE if A or B (or both) is TRUE `A | B`
- Return TRUE if A in B `A %in% B`

Logical operators



Penguins

```
library(palmerpenguins)
glimpse(penguins)
```

```
## Rows: 344
## Columns: 8
## $ species      <fct> Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, A
## $ island       <fct> Torgersen, Torgersen, Torgersen, Torgersen, Torgersen, Torgersen, Torgers
## $ bill_length_mm <dbl> 39.100000000000000142109, 39.50000000000000000000, 40.29999999999999715783
## $ bill_depth_mm <dbl> 18.69999999999999928946, 17.39999999999999857891, 18.00000000000000000000
## $ flipper_length_mm <int> 181, 186, 195, NA, 193, 190, 181, 195, 193, 190, 186, 180, 182, 191, 198,
## $ body_mass_g    <int> 3750, 3800, 3250, NA, 3450, 3650, 3625, 4675, 3475, 4250, 3300, 3700, 320
## $ sex            <fct> male, female, female, NA, female, male, female, male, NA, NA, NA, NA, fem
## $ year           <int> 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2
```

Filter

- Choose only the penguins from the island Torgersen

```
penguins %>%  
  filter(island == "Torgersen")
```

```
## # A tibble: 52 x 8  
##   species island   bill_length_mm bill_depth_mm flipper_length_mm body_mass_g sex    year  
##   <fct>   <fct>         <dbl>         <dbl>           <int>         <int> <fct> <int>  
## 1 Adelie Torgersen      39.1           18.7             181           3750 male   2007  
## 2 Adelie Torgersen      39.5           17.4             186           3800 female 2007  
## 3 Adelie Torgersen      40.3            18             195           3250 female 2007  
## 4 Adelie Torgersen      NA              NA              NA              NA <NA>   2007  
## 5 Adelie Torgersen      36.7           19.3             193           3450 female 2007  
## 6 Adelie Torgersen      39.3           20.6             190           3650 male   2007  
## 7 Adelie Torgersen      38.9           17.8             181           3625 female 2007  
## 8 Adelie Torgersen      39.2           19.6             195           4675 male   2007  
## 9 Adelie Torgersen      34.1           18.1             193           3475 <NA>   2007  
## 10 Adelie Torgersen      42             20.2             190           4250 <NA>   2007  
## # ... with 42 more rows
```


Filter

- Choose only the penguins that are species `Gentoo`

```
penguins %>%  
  filter(species == "Gentoo")
```

```
## # A tibble: 124 x 8  
##   species island bill_length_mm bill_depth_mm flipper_length_mm body_mass_g sex    year  
##   <fct>   <fct>         <dbl>         <dbl>             <int>      <int> <fct> <int>  
## 1 Gentoo  Biscoe         46.1           13.2              211        4500 female 2007  
## 2 Gentoo  Biscoe          50           16.3              230        5700 male   2007  
## 3 Gentoo  Biscoe         48.7           14.1              210        4450 female 2007  
## 4 Gentoo  Biscoe          50           15.2              218        5700 male   2007  
## 5 Gentoo  Biscoe         47.6           14.5              215        5400 male   2007  
## 6 Gentoo  Biscoe         46.5           13.5              210        4550 female 2007  
## 7 Gentoo  Biscoe         45.4           14.6              211        4800 female 2007  
## 8 Gentoo  Biscoe         46.7           15.3              219        5200 male   2007  
## 9 Gentoo  Biscoe         43.3           13.4              209        4400 female 2007  
## 10 Gentoo Biscoe         46.8           15.4              215        5150 male   2007  
## # ... with 114 more rows
```

Filter

- Choose only the penguins that are species `Chinstrap` and from the island `Biscoe`

```
penguins %>%  
  filter(species == "Chinstrap" & island == "Biscoe")
```

```
## # A tibble: 0 x 8
```

```
## # ... with 8 variables: species <fct>, island <fct>, bill_length_mm <dbl>, bill_depth_mm <dbl>, fl
```

- How many observations?

Filter

- Choose only the penguins that are species `Chinstrap` or from the island `Biscoe`

```
penguins %>%  
  filter(species == "Chinstrap" | island == "Biscoe")
```

```
## # A tibble: 236 x 8  
##   species island bill_length_mm bill_depth_mm flipper_length_mm body_mass_g sex   year  
##   <fct>   <fct>         <dbl>         <dbl>             <int>         <int> <fct> <int>  
## 1 Adelie Biscoe          37.8           18.3             174          3400 female 2007  
## 2 Adelie Biscoe          37.7           18.7             180          3600 male   2007  
## 3 Adelie Biscoe          35.9           19.2             189          3800 female 2007  
## 4 Adelie Biscoe          38.2           18.1             185          3950 male   2007  
## 5 Adelie Biscoe          38.8           17.2             180          3800 male   2007  
## 6 Adelie Biscoe          35.3           18.9             187          3800 female 2007  
## 7 Adelie Biscoe          40.6           18.6             183          3550 male   2007  
## 8 Adelie Biscoe          40.5           17.9             187          3200 female 2007  
## 9 Adelie Biscoe          37.9           18.6             172          3150 female 2007  
## 10 Adelie Biscoe          40.5           18.9             180          3950 male   2007  
## # ... with 226 more rows
```

- How many observations?

Filter

- Choose only the penguins that are species `Gentoo` or not from the island `Torgersen`

```
penguins %>%  
  filter(species == "Gentoo" | island != "Torgersen")
```

```
## # A tibble: 292 x 8  
##   species island bill_length_mm bill_depth_mm flipper_length_mm body_mass_g sex    year  
##   <fct>   <fct>         <dbl>         <dbl>             <int>         <int> <fct> <int>  
## 1 Adelie  Biscoe           37.8           18.3              174          3400 female 2007  
## 2 Adelie  Biscoe           37.7           18.7              180          3600 male   2007  
## 3 Adelie  Biscoe           35.9           19.2              189          3800 female 2007  
## 4 Adelie  Biscoe           38.2           18.1              185          3950 male   2007  
## 5 Adelie  Biscoe           38.8           17.2              180          3800 male   2007  
## 6 Adelie  Biscoe           35.3           18.9              187          3800 female 2007  
## 7 Adelie  Biscoe           40.6           18.6              183          3550 male   2007  
## 8 Adelie  Biscoe           40.5           17.9              187          3200 female 2007  
## 9 Adelie  Biscoe           37.9           18.6              172          3150 female 2007  
## 10 Adelie Biscoe           40.5           18.9              180          3950 male   2007  
## # ... with 282 more rows
```

Filter

- Choose penguins that are from the islands of Torgersen or Dream

```
penguins %>%  
  filter(island %in% c("Torgersen", "Dream"))
```

```
## # A tibble: 176 x 8  
##   species island    bill_length_mm bill_depth_mm flipper_length_mm body_mass_g sex    year  
##   <fct>   <fct>          <dbl>          <dbl>          <int>          <int> <fct> <int>  
## 1 Adelie Torgersen      39.1           18.7           181           3750 male   2007  
## 2 Adelie Torgersen      39.5           17.4           186           3800 female 2007  
## 3 Adelie Torgersen      40.3            18           195           3250 female 2007  
## 4 Adelie Torgersen      NA            NA            NA            NA <NA>   2007  
## 5 Adelie Torgersen      36.7           19.3           193           3450 female 2007  
## 6 Adelie Torgersen      39.3           20.6           190           3650 male   2007  
## 7 Adelie Torgersen      38.9           17.8           181           3625 female 2007  
## 8 Adelie Torgersen      39.2           19.6           195           4675 male   2007  
## 9 Adelie Torgersen      34.1           18.1           193           3475 <NA>   2007  
## 10 Adelie Torgersen      42            20.2           190           4250 <NA>   2007  
## # ... with 166 more rows
```

Filter

- Choose penguins that have `bill_length_mm` between 36 than 48 mm (inclusive)

```
penguins %>%  
  filter(bill_length_mm <= 48 & bill_length_mm >= 36)
```

```
## # A tibble: 226 x 8  
##   species island  bill_length_mm bill_depth_mm flipper_length_mm body_mass_g sex    year  
##   <fct>   <fct>      <dbl>         <dbl>         <int>         <int> <fct> <int>  
## 1 Adelie  Torgersen      39.1          18.7           181         3750 male   2007  
## 2 Adelie  Torgersen      39.5          17.4           186         3800 female 2007  
## 3 Adelie  Torgersen      40.3           18            195         3250 female 2007  
## 4 Adelie  Torgersen      36.7          19.3           193         3450 female 2007  
## 5 Adelie  Torgersen      39.3          20.6           190         3650 male   2007  
## 6 Adelie  Torgersen      38.9          17.8           181         3625 female 2007  
## 7 Adelie  Torgersen      39.2          19.6           195         4675 male   2007  
## 8 Adelie  Torgersen      42            20.2           190         4250 <NA>   2007  
## 9 Adelie  Torgersen      37.8          17.1           186         3300 <NA>   2007  
## 10 Adelie Torgersen      37.8          17.3           180         3700 <NA>   2007  
## # ... with 216 more rows
```

Arrange

- Arrange the penguins based on `bill_length_mm` in increasing order

```
penguins %>%  
  arrange(bill_length_mm)
```

```
## # A tibble: 344 x 8  
##   species island bill_length_mm bill_depth_mm flipper_length_mm body_mass_g sex   year  
##   <fct>   <fct>         <dbl>         <dbl>             <int>     <int> <fct> <int>  
## 1 Adelie Dream          32.1           15.5              188      3050 female 2009  
## 2 Adelie Dream          33.1           16.1              178      2900 female 2008  
## 3 Adelie Torgersen      33.5            19              190      3600 female 2008  
## 4 Adelie Dream          34            17.1              185      3400 female 2008  
## 5 Adelie Torgersen      34.1           18.1              193      3475 <NA> 2007  
## 6 Adelie Torgersen      34.4           18.4              184      3325 female 2007  
## 7 Adelie Biscoe         34.5           18.1              187      2900 female 2008  
## 8 Adelie Torgersen      34.6           21.1              198      4400 male   2007  
## 9 Adelie Torgersen      34.6           17.2              189      3200 female 2008  
## 10 Adelie Biscoe        35            17.9              190      3450 female 2008  
## # ... with 334 more rows
```

Arrange

- Arrange the penguins based on `bill_length_mm` in decreasing order

```
penguins %>%  
  arrange(desc(bill_length_mm))
```

```
## # A tibble: 344 x 8  
##   species    island bill_length_mm bill_depth_mm flipper_length_mm body_mass_g sex    year  
##   <fct>      <fct>         <dbl>         <dbl>           <int>         <int> <fct> <int>  
## 1 Gentoo    Biscoe         59.6          17             230          6050 male   2007  
## 2 Chinstrap Dream         58            17.8           181          3700 female 2007  
## 3 Gentoo    Biscoe         55.9          17             228          5600 male   2009  
## 4 Chinstrap Dream         55.8          19.8           207          4000 male   2009  
## 5 Gentoo    Biscoe         55.1          16             230          5850 male   2009  
## 6 Gentoo    Biscoe         54.3          15.7           231          5650 male   2008  
## 7 Chinstrap Dream         54.2          20.8           201          4300 male   2008  
## 8 Chinstrap Dream         53.5          19.9           205          4500 male   2008  
## 9 Gentoo    Biscoe         53.4          15.8           219          5500 male   2009  
## 10 Chinstrap Dream         52.8          20             205          4550 male   2008  
## # ... with 334 more rows
```


Select

- Select the 3rd through 5th variables

```
penguins %>%  
  select(3:5)
```

```
## # A tibble: 344 x 3  
##   bill_length_mm bill_depth_mm flipper_length_mm  
##         <dbl>         <dbl>         <int>  
## 1          39.1          18.7           181  
## 2          39.5          17.4           186  
## 3          40.3           18           195  
## 4           NA           NA             NA  
## 5          36.7          19.3           193  
## 6          39.3          20.6           190  
## 7          38.9          17.8           181  
## 8          39.2          19.6           195  
## 9          34.1          18.1           193  
## 10         42          20.2           190  
## # ... with 334 more rows
```

Select

- Select the variables `sex`, `island`, and `body_mass_g`

```
penguins %>%  
  select(sex, island, body_mass_g)
```

```
## # A tibble: 344 x 3  
##   sex      island  body_mass_g  
##   <fct>   <fct>         <int>  
## 1 male    Torgersen      3750  
## 2 female Torgersen      3800  
## 3 female Torgersen      3250  
## 4 <NA>    Torgersen         NA  
## 5 female Torgersen      3450  
## 6 male    Torgersen      3650  
## 7 female Torgersen      3625  
## 8 male    Torgersen      4675  
## 9 <NA>    Torgersen      3475  
## 10 <NA>   Torgersen      4250  
## # ... with 334 more rows
```

Select

- Select all the variables except for `flipper_length_mm`, and `year`

```
penguins %>%  
  select(-flipper_length_mm, -year)
```

```
## # A tibble: 344 x 6  
##   species island   bill_length_mm bill_depth_mm body_mass_g sex  
##   <fct>   <fct>         <dbl>         <dbl>         <int> <fct>  
## 1 Adelie  Torgersen         39.1          18.7          3750 male  
## 2 Adelie  Torgersen         39.5          17.4          3800 female  
## 3 Adelie  Torgersen         40.3           18          3250 female  
## 4 Adelie  Torgersen         NA           NA           NA <NA>  
## 5 Adelie  Torgersen         36.7          19.3          3450 female  
## 6 Adelie  Torgersen         39.3          20.6          3650 male  
## 7 Adelie  Torgersen         38.9          17.8          3625 female  
## 8 Adelie  Torgersen         39.2          19.6          4675 male  
## 9 Adelie  Torgersen         34.1          18.1          3475 <NA>  
## 10 Adelie Torgersen         42           20.2          4250 <NA>  
## # ... with 334 more rows
```

Select

- Choose only variables starting with a string `starts_with()`
- Choose only variables ending with a string `ends_with()`
- Choose only variables containing a string `contains()`
- Choose only variables matching a regular expression `matches()`
- Choose only variables within a numeric range `num_range()`

Select

- Select all the variables ending with `mm` or beginning with `s`

```
penguins %>%  
  select(ends_with("mm") | starts_with("s"))
```

```
## # A tibble: 344 x 5  
##   bill_length_mm bill_depth_mm flipper_length_mm species sex  
##           <dbl>         <dbl>           <int> <fct>  <fct>  
## 1           39.1           18.7             181 Adelie  male  
## 2           39.5           17.4             186 Adelie  female  
## 3           40.3            18             195 Adelie  female  
## 4            NA            NA              NA Adelie  <NA>  
## 5           36.7           19.3             193 Adelie  female  
## 6           39.3           20.6             190 Adelie  male  
## 7           38.9           17.8             181 Adelie  female  
## 8           39.2           19.6             195 Adelie  male  
## 9           34.1           18.1             193 Adelie  <NA>  
## 10          42            20.2             190 Adelie  <NA>  
## # ... with 334 more rows
```

Mutate

- Create a variable called `bill_area` that approximates bill surface area in mm (assume a rectangular bill) and select the three bill variables.

```
penguins %>%  
  mutate(bill_area = bill_length_mm * bill_depth_mm) %>%  
  select(starts_with("bill"))
```

```
## # A tibble: 344 x 3  
##   bill_length_mm bill_depth_mm bill_area  
##           <dbl>         <dbl>    <dbl>  
## 1           39.1           18.7     731.  
## 2           39.5           17.4     687.  
## 3           40.3           18      725.  
## 4            NA            NA        NA  
## 5           36.7           19.3     708.  
## 6           39.3           20.6     810.  
## 7           38.9           17.8     692.  
## 8           39.2           19.6     768.  
## 9           34.1           18.1     617.  
## 10          42            20.2     848.  
## # ... with 334 more rows
```

Transmute

- Keep only the created variable
- Create a variable called `bill_area` that approximates bill surface area in mm (assume a rectangular bill) and select the three bill variables.

```
penguins %>%  
  transmute(bill_area = bill_length_mm * bill_depth_mm)
```

```
## # A tibble: 344 x 1  
##   bill_area  
##   <dbl>  
## 1     731.  
## 2     687.  
## 3     725.  
## 4      NA  
## 5     708.  
## 6     810.  
## 7     692.  
## 8     768.  
## 9     617.  
## 10    848.  
## # ... with 334 more rows
```

Transmute

- Keep only the created variable
- Create a variable called `bill_area` that approximates bill surface area on a log scale (assume a rectangular bill) and select the three bill variables (Why would you do this? No one knows...)

```
penguins %>%  
  transmute(bill_area = log(bill_length_mm * bill_depth_mm))
```

```
## # A tibble: 344 x 1  
##   bill_area  
##   <dbl>  
## 1      6.59  
## 2      6.53  
## 3      6.59  
## 4      NA  
## 5      6.56  
## 6      6.70  
## 7      6.54  
## 8      6.64  
## 9      6.43  
## 10     6.74  
## # ... with 334 more rows
```


Summarize

- Calculate the average `bill_length_mm` and save as `mean_bill_length`

```
penguins %>%  
  summarize(mean_bill_length_mm = mean(bill_length_mm))
```

```
## # A tibble: 1 x 1  
##   mean_bill_length_mm  
##                 <dbl>  
## 1                   NA
```

- What happened?

```
penguins %>%  
  summarize(mean_bill_length_mm = mean(bill_length_mm, na.rm = TRUE))
```

```
## # A tibble: 1 x 1  
##   mean_bill_length_mm  
##                 <dbl>  
## 1                 43.9
```

- What does `na.rm = TRUE` do?

Summarize

- What is the average `body_mass_g` by species?

```
penguins %>%  
  group_by(species) %>%  
  summarize(mean_body_mass_by_species = mean(body_mass_g, na.rm = TRUE))
```

```
## # A tibble: 3 x 2  
##   species    mean_body_mass_by_species  
##   <fct>          <dbl>  
## 1 Adelie          3701.  
## 2 Chinstrap       3733.  
## 3 Gentoo          5076.
```

```
penguins %>%  
  filter(!is.na(body_mass_g)) %>%  
  group_by(species) %>%  
  summarize(mean_body_mass_by_species = mean(body_mass_g))
```

```
## # A tibble: 3 x 2  
##   species    mean_body_mass_by_species  
##   <fct>          <dbl>  
## 1 Adelie          3701.  
## 2 Chinstrap       3733.  
## 3 Gentoo          5076.
```

Summarize

```
data("starwars")
glimpse(starwars)
```

```
## Rows: 87
## Columns: 14
## $ name      <chr> "Luke Skywalker", "C-3PO", "R2-D2", "Darth Vader", "Leia Organa", "Owen Lars"
## $ height    <int> 172, 167, 96, 202, 150, 178, 165, 97, 183, 182, 188, 180, 228, 180, 173, 175,
## $ mass      <dbl> 77.0000000000000000000000000000000000, 75.0000000000000000000000000000000000, 32.0000000000000000000000000000000000, 13.
## $ hair_color <chr> "blond", NA, NA, "none", "brown", "brown, grey", "brown", NA, "black", "auburn"
## $ skin_color <chr> "fair", "gold", "white, blue", "white", "light", "light", "light", "white, red", "white, blue"
## $ eye_color  <chr> "blue", "yellow", "red", "yellow", "brown", "blue", "blue", "red", "brown", "blue"
## $ birth_year <dbl> 19.0000000000000000000000000000000000, 112.0000000000000000000000000000000000, 33.0000000000000000000000000000000000, 4
## $ sex        <chr> "male", "none", "none", "male", "female", "male", "female", "none", "male", "female"
## $ gender     <chr> "masculine", "masculine", "masculine", "masculine", "feminine", "masculine", "feminine", "masculine", "feminine"
## $ homeworld  <chr> "Tatooine", "Tatooine", "Naboo", "Tatooine", "Alderaan", "Tatooine", "Tatooine", "Tatooine", "Tatooine"
## $ species    <chr> "Human", "Droid", "Droid", "Human", "Human", "Human", "Human", "Human", "Droid", "Human"
## $ films      <list> <"The Empire Strikes Back", "Revenge of the Sith", "Return of the Jedi", "A New Hope"
## $ vehicles   <list> <"Snowspeeder", "Imperial Speeder Bike">, <>, <>, <>, "Imperial Speeder Bike"
## $ starships  <list> <"X-wing", "Imperial shuttle">, <>, <>, "TIE Advanced x1", <>, <>, <>, <>, <>
```

Summarize

- Count using `n()`
- Important to make sure you check the group sample size

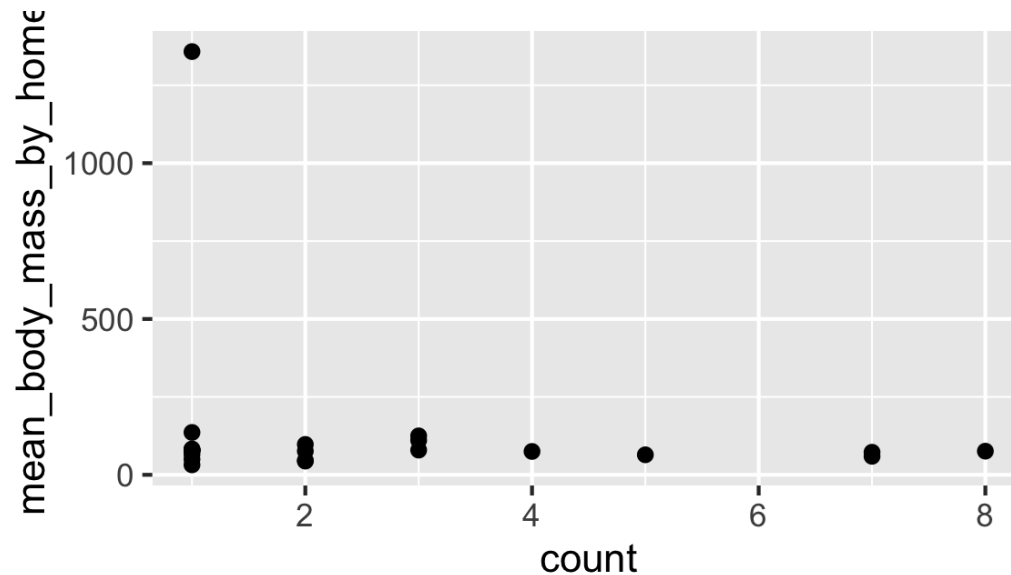
```
starwars %>%  
  filter(!is.na(mass) & !is.na(films)) %>%  
  group_by(films) %>%  
  summarize(mean_body_mass_by_home = mean(mass),  
            count = n())
```

```
## # A tibble: 23 x 3  
##   films      mean_body_mass_by_home count  
##   <list>          <dbl> <int>  
## 1 <chr [5]>          79.3     3  
## 2 <chr [6]>          76       2  
## 3 <chr [7]>          32       1  
## 4 <chr [4]>         136       1  
## 5 <chr [3]>         97.5     2  
## 6 <chr [1]>          75       4  
## 7 <chr [3]>         75.9     8  
## 8 <chr [4]>          80       1  
## 9 <chr [3]>        1358       1  
## 10 <chr [3]>         77       1  
## # ... with 13 more rows
```

Summarize

- pipe output into `ggplot`

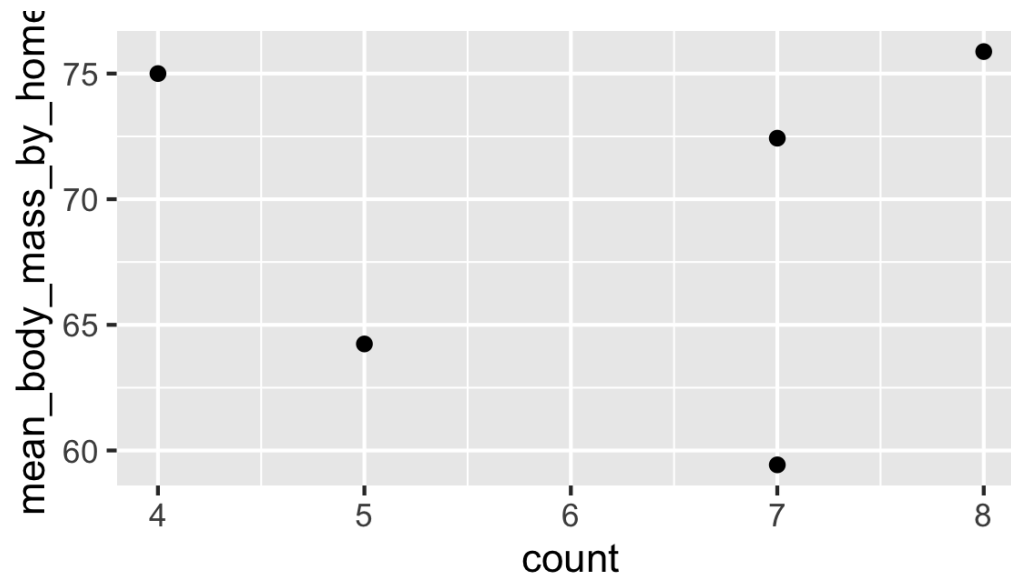
```
starwars %>%  
  filter(!is.na(mass) & !is.na(films)) %>%  
  group_by(films) %>%  
  summarize(mean_body_mass_by_home = mean(mass),  
            count = n()) %>%  
  ggplot(aes(x = count, y = mean_body_mass_by_home)) +  
  geom_point()
```



Summarize

- Only plot the average body masses for characters that appear in 4 or more movies

```
starwars %>%  
  filter(!is.na(mass) & !is.na(films)) %>%  
  group_by(films) %>%  
  summarize(mean_body_mass_by_home = mean(mass),  
            count = n()) %>%  
  filter(count >= 4) %>%  
  ggplot(aes(x = count, y = mean_body_mass_by_home)) +  
  geom_point()
```



Question

- Which character has the highest body mass?

```
starwars %>%  
  select(name, mass) %>%  
  arrange(desc(mass))
```

```
## # A tibble: 87 x 2  
##   name                mass  
##   <chr>              <dbl>  
## 1 Jabba Desilijic Tiure 1358  
## 2 Grievous             159  
## 3 IG-88                140  
## 4 Darth Vader          136  
## 5 Tarfful              136  
## 6 Owen Lars            120  
## 7 Bossk                113  
## 8 Chewbacca            112  
## 9 Jek Tono Porkins     110  
## 10 Dexter Jettster     102  
## # ... with 77 more rows
```

Question

- Plot the average height by species vs. the average mass by species

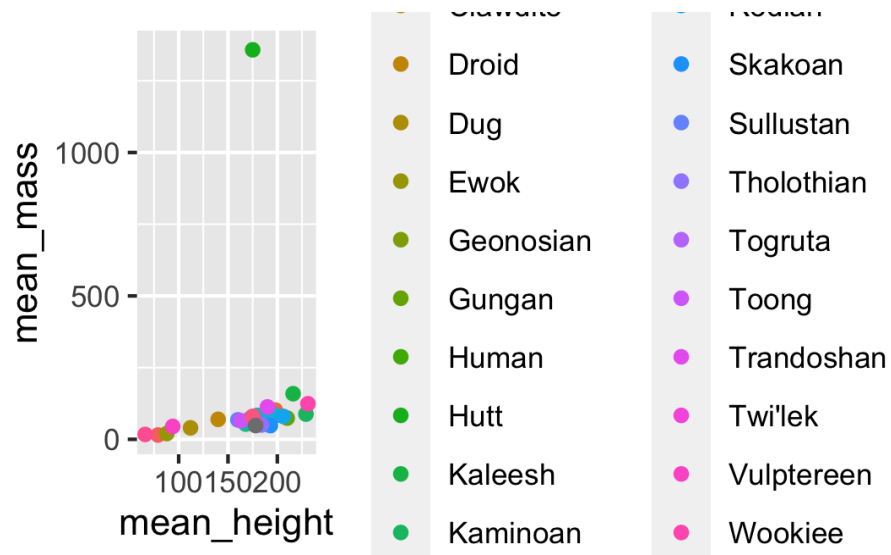
```
starwars %>%
  group_by(species) %>%
  filter(!is.na(height) & !is.na(mass)) %>%
  summarize(
    mean_height = mean(height),
    mean_mass = mean(mass),
    count = n()
  )
```

```
## # A tibble: 32 x 4
##   species    mean_height mean_mass count
##   <chr>         <dbl>     <dbl> <int>
## 1 Aleena           79         15      1
## 2 Besalisk        198        102      1
## 3 Cerean          198         82      1
## 4 Clawdite        168         55      1
## 5 Droid           140        69.8     4
## 6 Dug             112         40      1
## 7 Ewok             88         20      1
## 8 Geonosian       183         80      1
## 9 Gungan          210         74      2
## 10 Human          180.        82.8    22
## # ... with 22 more rows
```


Question

- Plot the average height by species vs. the average mass by species

```
starwars %>%  
  group_by(species) %>%  
  filter(!is.na(height) & !is.na(mass)) %>%  
  summarize(  
    mean_height = mean(height),  
    mean_mass = mean(mass)  
  ) %>%  
  ggplot(aes(x = mean_height, y = mean_mass, color = species)) +  
  geom_point()
```



Grouping

- Can `group_by()` multiple variables
- Count the number of penguins observed from each `species` and `island`

```
penguins %>%  
  group_by(species, island) %>%  
  summarize(count = n())
```

`summarise()` has grouped output by 'species'. You can override using the `.groups` argument.

```
## # A tibble: 5 x 3  
## # Groups:   species [3]  
##   species    island    count  
##   <fct>     <fct>     <int>  
## 1 Adelie    Biscoe        44  
## 2 Adelie    Dream         56  
## 3 Adelie    Torgersen     52  
## 4 Chinstrap Dream         68  
## 5 Gentoo    Biscoe       124
```

Grouping

- Then `ungroup()` to resume the calculations

```
penguins %>%  
  group_by(species, island) %>%  
  summarize(count = n()) %>%  
  ungroup() %>%  
  summarize(total = sum(count))
```

`summarise()` has grouped output by 'species'. You can override using the `.groups` argument.

```
## # A tibble: 1 x 1  
##   total  
##   <int>  
## 1    344
```

Grouping

- Filter the data to only contain penguin species have 100 observations or more?

```
penguins %>%  
  group_by(species) %>%  
  filter(n() >= 100)
```

```
## # A tibble: 276 x 8  
## # Groups:   species [2]  
##   species island    bill_length_mm bill_depth_mm flipper_length_mm body_mass_g sex    year  
##   <fct>   <fct>          <dbl>          <dbl>          <int>          <int> <fct> <int>  
## 1 Adelie Torgersen      39.1           18.7           181           3750 male   2007  
## 2 Adelie Torgersen      39.5           17.4           186           3800 female 2007  
## 3 Adelie Torgersen      40.3            18           195           3250 female 2007  
## 4 Adelie Torgersen      NA             NA             NA             NA <NA>   2007  
## 5 Adelie Torgersen      36.7           19.3           193           3450 female 2007  
## 6 Adelie Torgersen      39.3           20.6           190           3650 male   2007  
## 7 Adelie Torgersen      38.9           17.8           181           3625 female 2007  
## 8 Adelie Torgersen      39.2           19.6           195           4675 male   2007  
## 9 Adelie Torgersen      34.1           18.1           193           3475 <NA>   2007  
## 10 Adelie Torgersen      42             20.2           190           4250 <NA>   2007  
## # ... with 266 more rows
```

Grouping

- Is there a difference in mean `body_mass_g` between penguin `species` on the different `islands`?

```
penguins %>%  
  group_by(species, island) %>%  
  filter(!is.na(body_mass_g)) %>%  
  summarize(mean_mass = mean(body_mass_g),  
            count = n())
```

``summarise()`` has grouped output by 'species'. You can override using the ``.groups`` argument.

```
## # A tibble: 5 x 4  
## # Groups:   species [3]  
##   species island    mean_mass count  
##   <fct>    <fct>      <dbl> <int>  
## 1 Adelie  Biscoe         3710.    44  
## 2 Adelie  Dream          3688.    56  
## 3 Adelie  Torgersen      3706.    51  
## 4 Chinstrap Dream        3733.    68  
## 5 Gentoo  Biscoe         5076.   123
```

data.frames and tibbles

- The default data object in R is the `data.frame`
- A `tibble` is a `data.frame` with extra bells and whistles

```
data("iris")  
class(iris)
```

```
## [1] "data.frame"
```

```
iris_tibble <- as_tibble(iris)  
class(iris_tibble)
```

```
## [1] "tbl_df"      "tbl"        "data.frame"
```

Creating tibbles

```
dat <- data.frame(x = 1:5, y = rnorm(5), z = letters[1:5])  
glimpse(dat)
```

```
## Rows: 5  
## Columns: 3  
## $ x <int> 1, 2, 3, 4, 5  
## $ y <dbl> -1.07908605196321638608481, -0.01375981518563124117449, 0.73639613403315984374586, -0.01375981518563124117449, 0.73639613403315984374586  
## $ z <chr> "a", "b", "c", "d", "e"
```

```
dat_tibble <- tibble(x = 1:5, y = rnorm(5), z = letters[1:5])  
glimpse(dat_tibble)
```

```
## Rows: 5  
## Columns: 3  
## $ x <int> 1, 2, 3, 4, 5  
## $ y <dbl> -0.5636274879810749816045, 0.9625499602714389091318, -1.1225359576985030063412, 0.32310749810749816045, -0.5636274879810749816045  
## $ z <chr> "a", "b", "c", "d", "e"
```

Working with `tibbles`

- Better printing of data
- Easier to perform grouping and nesting operations
- Subsetting tibbles
 - `$` and `[[`
 - `[[` can subset by variable name or index (counting base starts at 1)
 - `$` subsets by variable name only

starwars\$name

## [1]	"Luke Skywalker"	"C-3P0"	"R2-D2"	"Darth Vader"
## [8]	"R5-D4"	"Biggs Darklighter"	"Obi-Wan Kenobi"	"Anakin Skywalker"
## [15]	"Greedo"	"Jabba Desilijic Tiure"	"Wedge Antilles"	"Jek Tono Porkins"
## [22]	"IG-88"	"Bossk"	"Lando Calrissian"	"Lobot"
## [29]	"Wicket Systri Warrick"	"Nien Nunb"	"Qui-Gon Jinn"	"Nute Gunray"
## [36]	"Rugor Nass"	"Ric Olié"	"Watto"	"Sebulba"
## [43]	"Bib Fortuna"	"Ayla Secura"	"Dud Bolt"	"Gasgano"
## [50]	"Kit Fisto"	"Eeth Koth"	"Adi Gallia"	"Saesee Tiin"
## [57]	"Gregar Typho"	"Cordé"	"Cliegg Lars"	"Poggle the Lesser"
## [64]	"Dooku"	"Bail Prestor Organa"	"Jango Fett"	"Zam Wesell"
## [71]	"Jocasta Nu"	"Ratts Tyerell"	"R4-P17"	"Wat Tambor"
## [78]	"Tarfful"	"Raymus Antilles"	"Sly Moore"	"Tion Medon"
## [85]	"BB8"	"Captain Phasma"	"Padmé Amidala"	

```
starwars[["name"]]
```

## [1] "Luke Skywalker"	"C-3P0"	"R2-D2"	"Darth Vader"
## [8] "R5-D4"	"Biggs Darklighter"	"Obi-Wan Kenobi"	"Anakin Skywalker"
## [15] "Greedo"	"Jabba Desilijic Tiure"	"Wedge Antilles"	"Jek Tono Porkins"
## [22] "IG-88"	"Bossk"	"Lando Calrissian"	"Lobot"
## [29] "Wicket Systri Warrick"	"Nien Nunb"	"Qui-Gon Jinn"	"Nute Gunray"
## [36] "Rugor Nass"	"Ric Olié"	"Watto"	"Sebulba"
## [43] "Bib Fortuna"	"Ayla Secura"	"Dud Bolt"	"Gasgano"
## [50] "Kit Fisto"	"Eeth Koth"	"Adi Gallia"	"Saesee Tiin"
## [57] "Gregar Typho"	"Cordé"	"Cliegg Lars"	"Poggle the Lesser"
## [64] "Dooku"	"Bail Prestor Organa"	"Jango Fett"	"Zam Wesell"
## [71] "Jocasta Nu"	"Ratts Tyerell"	"R4-P17"	"Wat Tambor"
## [78] "Tarfful"	"Raymus Antilles"	"Sly Moore"	"Tion Medon"
## [85] "BB8"	"Captain Phasma"	"Padmé Amidala"	

```
starwars[[1]]
```

## [1] "Luke Skywalker"	"C-3P0"	"R2-D2"	"Darth Vader"
## [8] "R5-D4"	"Biggs Darklighter"	"Obi-Wan Kenobi"	"Anakin Skywalker"
## [15] "Greedo"	"Jabba Desilijic Tiure"	"Wedge Antilles"	"Jek Tono Porkins"
## [22] "IG-88"	"Bossk"	"Lando Calrissian"	"Lobot"
## [29] "Wicket Systri Warrick"	"Nien Nunb"	"Qui-Gon Jinn"	"Nute Gunray"
## [36] "Rugor Nass"	"Ric Olié"	"Watto"	"Sebulba"
## [43] "Bib Fortuna"	"Ayla Secura"	"Dud Bolt"	"Gasgano"
## [50] "Kit Fisto"	"Eeth Koth"	"Adi Gallia"	"Saesee Tiin"
## [57] "Gregar Typho"	"Cordé"	"Cliegg Lars"	"Poggle the Lesser"
## [64] "Dooku"	"Bail Prestor Organa"	"Jango Fett"	"Zam Wesell"
## [71] "Jocasta Nu"	"Ratts Tyerell"	"R4-P17"	"Wat Tambor"
## [78] "Tarfful"	"Raymus Antilles"	"Sly Moore"	"Tion Medon"
## [85] "BB8"	"Captain Phasma"	"Padmé Amidala"	

```
starwars %>%  
  .$name
```

## [1]	"Luke Skywalker"	"C-3P0"	"R2-D2"	"Darth Vader"
## [8]	"R5-D4"	"Biggs Darklighter"	"Obi-Wan Kenobi"	"Anakin Skywalker"
## [15]	"Greedo"	"Jabba Desilijic Tiure"	"Wedge Antilles"	"Jek Tono Porkins"
## [22]	"IG-88"	"Bossk"	"Lando Calrissian"	"Lobot"
## [29]	"Wicket Systri Warrick"	"Nien Nunb"	"Qui-Gon Jinn"	"Nute Gunray"
## [36]	"Rugor Nass"	"Ric Olié"	"Watto"	"Sebulba"
## [43]	"Bib Fortuna"	"Ayla Secura"	"Dud Bolt"	"Gasgano"
## [50]	"Kit Fisto"	"Eeth Koth"	"Adi Gallia"	"Saesee Tiin"
## [57]	"Gregar Typho"	"Cordé"	"Cliegg Lars"	"Poggle the Lesser"
## [64]	"Dooku"	"Bail Prestor Organa"	"Jango Fett"	"Zam Wesell"
## [71]	"Jocasta Nu"	"Ratts Tyerell"	"R4-P17"	"Wat Tambor"
## [78]	"Tarfful"	"Raymus Antilles"	"Sly Moore"	"Tion Medon"
## [85]	"BB8"	"Captain Phasma"	"Padmé Amidala"	

Reading files

- Many options to read files
 - Different file types can have different methods to read files
- Base R options
 - `read.table()`, `read.csv()`, `read.csv2()`, `read.delim()`, `read.delim2()`
- Excel file type methods using the `readxl` package
 - `read_excel()`, `read_xls()`, `read_xlsx()`
- Reading files using the `readr` package
 - `read_csv()` reads comma separated value (csv) files
 - `read_csv2()` reads semicolon separated files (Europeans and others use , instead of . in decimal numbers)
 - `read_tsv()` reads tab delimited files
 - `read_delim()` reads generic delimiter files

Reading files

- Player performance for NHL players

```
file_path <- here::here("data", "game_goals.csv")
file_path
```

```
## [1] "/Users/tips/dasc1104-teaching/data/game_goals.csv"
```

```
dat <- read_csv(file_path)
```

```
##
## — Column specification —————
## cols(
##   .default = col_double(),
##   player = col_character(),
##   date = col_date(format = ""),
##   age = col_character(),
##   team = col_character(),
##   at = col_character(),
##   opp = col_character(),
##   location = col_character(),
##   outcome = col_character()
## )
## i Use `spec()` for the full column specifications.
```

```
glimpse(dat)
```

```
## Rows: 49,384
## Columns: 25
## $ player      <chr> "Alex Ovechkin", "Alex Ovechkin", "Alex Ovechkin", "Alex Ovechkin", "A
## $ season      <dbl> 2006, 2006, 2006, 2006, 2006, 2006, 2006, 2006, 2006, 2006, 2006, 2006
## $ rank        <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
## $ date        <date> 2005-10-05, 2005-10-07, 2005-10-08, 2005-10-10, 2005-10-12, 2005-10-1
## $ game_num    <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
## $ age         <chr> "20-018", "20-020", "20-021", "20-023", "20-025", "20-026", "20-029",
## $ team        <chr> "WSH", "WSH", "WSH", "WSH", "WSH", "WSH", "WSH", "WSH", "WSH", "WSH", "WSH",
## $ at          <chr> NA, NA, "@", NA, "@", NA, NA, "@", NA, "@", "@", "@", NA, NA, "@", NA,
## $ opp         <chr> "CBJ", "ATL", "ATL", "NYR", "CAR", "NYI", "TBL", "FLA", "CAR", "BUF",
## $ location    <chr> "Home", "Home", "Away", "Home", "Away", "Home", "Home", "Away", "Home",
## $ outcome     <chr> "W", "L", "L", "W", "L", "L", "W", "L", "L", "W", "L", "L", "W", "W",
## $ goals       <dbl> 2, 0, 0, 1, 1, 0, 0, 2, 0, 0, 2, 0, 0, 2, 2, 1, 0, 1, 1, 0, 0, 0, 0, 0
## $ assists     <dbl> 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1
## $ points      <dbl> 2, 1, 1, 1, 1, 1, 1, 2, 0, 1, 2, 0, 0, 2, 2, 1, 0, 1, 2, 0, 1, 1, 1
## $ plus_minus  <dbl> 1, -2, 0, 1, 0, -1, 1, 1, 0, 0, 0, -2, 1, -1, -2, 0, 0, 0, -1, 1, 1, -
## $ penalty_min <dbl> 2, 0, 4, 2, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 4, 0, 0, 0, 2, 0, 0, 0, 0
## $ goals_even  <dbl> 1, 0, 0, 0, 1, 0, 0, 2, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0
## $ goals_powerplay <dbl> 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 1, 0, 0, 0, 1, 0, 0, 0, 0
## $ goals_short <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
## $ goals_gamewinner <dbl> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0
## $ assists_even <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA
## $ assists_powerplay <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA
## $ assists_short <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA
## $ shots       <dbl> 5, 1, 3, 6, 6, 5, 2, 10, 2, 5, 4, 7, 7, 8, 7, 5, 3, 9, 7, 5, 4, 0, 9,
## $ shot_percent <dbl> 40.0000000000000000000000000000000000, 0.0000000000000000000000, 0.0000000000000000000000
```

- For players who took at least 100 shots in the 2014 season, which player had the highest mean shot percentage?

```
dat %>%  
  filter(season == 2014) %>%  
  group_by(player) %>%  
  filter(!is.na(shots) & !is.na(shot_percent)) %>%  
  summarize(total_shots = sum(shots),  
            mean_shot_percent = mean(shot_percent)) %>%  
  filter(total_shots >= 100) %>%  
  arrange(desc(mean_shot_percent))
```

```
## # A tibble: 29 x 3  
##   player      total_shots mean_shot_percent  
##   <chr>          <dbl>          <dbl>  
## 1 Steven Stamkos      124            20.5  
## 2 Anze Kopitar        200            19.1  
## 3 Joe Pavelski        225            17.0  
## 4 Corey Perry        280            16.9  
## 5 Jarome Iginla       209            16.0  
## 6 Brad Marchand       149            15.9  
## 7 Ryan Getzlaf        204            15.5  
## 8 Patrick Kane        227            14.9  
## 9 Jamie Benn          279            14.8  
## 10 Jaromir Jagr        231            14.3  
## # ... with 19 more rows
```


- Data about **measles vaccine**

```
file_path <- here::here("data", "measles.csv")
file_path
```

```
## [1] "/Users/tips/dasc1104-teaching/data/measles.csv"
```

```
dat <- read_csv(file_path)
```

```
##
## — Column specification
## cols(
##   index = col_double(),
##   state = col_character(),
##   year = col_character(),
##   name = col_character(),
##   type = col_character(),
##   city = col_character(),
##   county = col_character(),
##   district = col_logical(),
##   enroll = col_double(),
##   mmr = col_double(),
##   overall = col_double(),
##   xrel = col_logical(),
##   xmed = col_double(),
##   xper = col_double(),
##   lat = col_double(),
##   lng = col_double()
## )
```

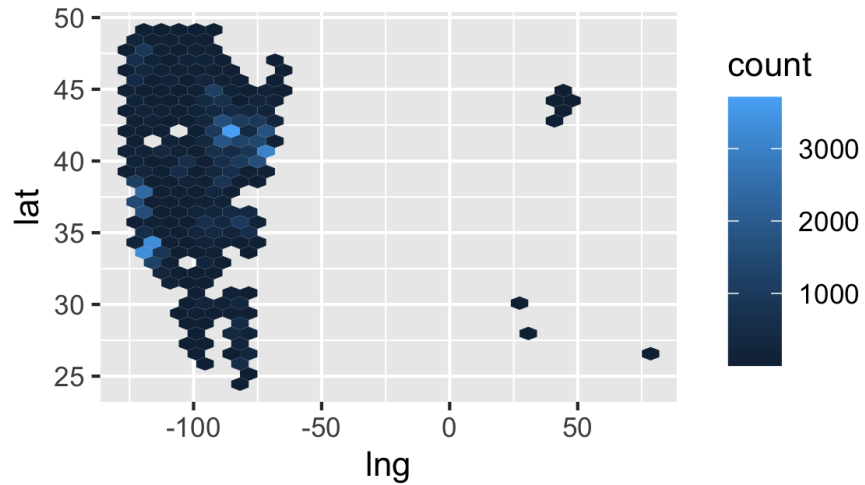
```
glimpse(dat)
```

```
## Rows: 66,113
## Columns: 16
## $ index      <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 11, 12, 13, 14, 15, 15, 16, 17, 18, 19, 20,
## $ state      <chr> "Arizona", "Arizona", "Arizona", "Arizona", "Arizona", "Arizona", "Arizona", "A
## $ year       <chr> "2018-19", "2018-19", "2018-19", "2018-19", "2018-19", "2018-19", "2018-19", "2
## $ name       <chr> "A J Mitchell Elementary", "Academy Del Sol", "Academy Del Sol - Hope", "Academ
## $ type       <chr> "Public", "Charter", "Charter", "Charter", "Charter", "Public", "Charter", "Cha
## $ city       <chr> "Nogales", "Tucson", "Tucson", "Phoenix", "Phoenix", "Phoenix", "Phoenix", "Yum
## $ county     <chr> "Santa Cruz", "Pima", "Pima", "Maricopa", "Maricopa", "Maricopa", "Maricopa", "
## $ district   <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
## $ enroll     <dbl> 51, 22, 85, 60, 43, 36, 24, 22, 26, 78, 78, 35, 54, 54, 34, 57, 57, 47, 54, 98,
## $ mmr        <dbl> 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,
## $ overall    <dbl> -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
## $ xrel       <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
## $ xmed       <dbl> NA, NA, NA, NA, 2.330000000000000071054, NA, NA, NA, NA, NA, NA, NA, 2.859999999999
## $ xper       <dbl> NA, NA, NA, NA, 2.330000000000000071054, NA, 4.16999999999999928946, NA, NA, M
## $ lat        <dbl> 31.347818900000000004193, 32.221921500000000054888, 32.130493199999999653419, 33.4
## $ lng        <dbl> -110.93803139999999998487, -110.89610310000000048152, -111.11700480000000036836, -
```

- Plot the locations of measles outbreaks

```
dat %>%  
  ggplot(aes(x = lng, y = lat)) +  
  geom_hex()
```

Warning: Removed 1549 rows containing non-finite values (stat_binhex).



- A subset of speeches at Trump rallies

```
file_path <- here::here("data", "Trump_rallies")
all_files <- list.files(file_path, pattern = ".txt")

dat <- list()

for (i in 1:length(all_files)) {
  dat[[i]] <- read_file(paste(file_path, all_files[i], sep = "/"))
}
```

- Trump rally speeches

```
glimpse(dat)
```

```
## List of 35
```

```
## $ : chr "Thank you. Thank you. Thank you to Vice President Pence. He's a good guy. We've done a  
## $ : chr "There's a lot of people. That's great. Thank you very much. Thank you very much. That  
## $ : chr "Thank you. Thank you. Thank you. All I can say is that the fake news just doesn't get  
## $ : chr "I want to thank you very much. North Carolina, thank you very much. I'm thrilled to ba  
## $ : chr "Thank you all. Thank you very much. Thank you to Vice President Mike Pence, and hello  
## $ : chr "Hello Colorado. We love Colorado, most beautiful place And I'm thrilled to be back in  
## $ : chr "Thank you. Thank you very much. Hello Dallas. It's great to be with you tonight. Thank  
## $ : chr "I worked so hard for this state. I worked so hard. You just got two of the greatest tr  
## $ : chr "What a crowd, what a crowd. Get those people over here. See me. Let them come over. WH  
## $ : chr " Thank you everybody. Thank you and Vice President Mike Pence, thank you very much. In  
## $ : chr "We brought you a lot of car plants, Michigan. We brought you a lot of car plants. You  
## $ : chr "Thank you very much. Thank you. Thank you. Thank you to Greenville, North Carolina, I  
## $ : chr "Thank you, thank you. Wow. Wow, and I'm thrilled to be here with you in Henderson. Tho  
## $ : chr " Well, thank you to Vice President Pence. Thank you, Mike. And hello Pennsylvania. Hel  
## $ : chr "Well, thank you very much. And hello Las Vegas. Great to be with you. They have a big  
## $ : chr "So thank you Pennsylvania, very much. I'm thrilled to be in Latrobe, the home of the 1  
## $ : chr "Thank you very much and thank you to the original Lee Greenwood. Thank you. Thank you,  
## $ : chr "Well thank you very much. And I'm thrilled to be back in Wisconsin where we had a very  
## $ : chr "Well, I thank you very much. So I want to start by saying, \"Hello, Nevada. How are yo  
## $ : chr "Thank you very much. Thank you, Minnesota. This is a great state. We are going to win  
## $ : chr "Thank you, thank you very much. Thank you very much. That is a beautiful site right be  
## $ : chr " Thank you very much everybody. Thank you. Wow. I will never, ever let you down, that  
## $ : chr "Hello, everybody. Hello, everybody. Wow. Hello, everybody. Thank you. Thank you. And I  
## $ : chr "Hello, Manchester, and I am thrilled to be in the great state of New Hampshire with th  
## $ : chr " Wow, thank you. Thank you, New Mexico. Thank you. We love being with you. We love be  
## $ : chr "Wow, that's a big crowd. This is a big crowd. Thank you very much, everybody. Hello to  
## $ : chr " Thank you very much, Phoenix. We love to be back. We'll be back a lot. We're going to  
## $ : chr "Doesn't have the power. Doesn't have the staying power. You see what's happening in Ca  
## $ : chr "Hello, Houston. I am so thrilled to be here in the great state of Texas with one of the Am
```

- examine the first speech

```
library(tidytext)
# examine the first speech
dat[[1]] %>%
  tibble(text = .) %>%
  drop_na() %>%
  unnest_tokens(word, text) %>%
  group_by(word) %>%
  summarize(count = n()) %>%
  arrange(desc(count)) ## from tidytext package
```

```
## # A tibble: 2,113 x 2
##   word    count
##   <chr> <int>
## 1 the      702
## 2 i        508
## 3 and      497
## 4 you      480
## 5 to       427
## 6 a        366
## 7 they     318
## 8 of       311
## 9 it       279
## 10 that    260
## # ... with 2,103 more rows
```

- examine the fist speech (remove the "stop" words)

```
library(tidytext)
dat[[1]] %>%
  tibble(text = .) %>%
  drop_na() %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  group_by(word) %>%
  summarize(count = n()) %>%
  arrange(desc(count)) ## from tidytext package
```

```
## Joining, by = "word"
```

```
## # A tibble: 1,693 x 2
##   word      count
##   <chr>    <int>
## 1 people     112
## 2 michigan    50
## 3 country     47
## 4 time        42
## 5 lot         39
## 6 president   38
## 7 love        33
## 8 deal        30
## 9 remember    27
## 10 democrats  24
## # ... with 1,683 more rows
```

Examine the fist speech with a word cloud

```
library(tidytext)
library(wordcloud2)
dat[[1]] %>%
  tibble(text = .) %>%
  drop_na() %>%
  unnest_tokens(word, text) %>% ## from tidytext package
  anti_join(stop_words) %>% ## from tidytext package
  group_by(word) %>%
  summarize(count = n()) %>%
  wordcloud2()
```


Examine the fist speech with a word cloud

```
<div id="htmlwidget-766398535c149d60ef72" style="width:60%;height:648px;" class="wordclo  
<script type="application/json" data-for="htmlwidget-766398535c149d60ef72">{"x":{"word":
```

File I/O

- Data about Jelle's Marble Run

```
filename <- here::here("data", "marbles.csv")
dat <- read_csv(filename)
```

```
##
```

```
## — Column specification —————
```

```
## cols(
```

```
##   `# Data from Jelle's Marble Run https://www.youtube.com/channel/UCYJdpnjuSWV0LgGT9fIzL0g` = co
```

```
## )
```

```
## Warning: 257 parsing failures.
```

```
## row col  expected      actual                                file
```

```
##   1  -- 1 columns 14 columns '/Users/tips/dasc1104-teaching/data/marbles.csv'
```

```
##   2  -- 1 columns 14 columns '/Users/tips/dasc1104-teaching/data/marbles.csv'
```

```
##   3  -- 1 columns 14 columns '/Users/tips/dasc1104-teaching/data/marbles.csv'
```

```
##   4  -- 1 columns 14 columns '/Users/tips/dasc1104-teaching/data/marbles.csv'
```

```
##   5  -- 1 columns 14 columns '/Users/tips/dasc1104-teaching/data/marbles.csv'
```

```
## ... ..
```

```
## See problems(...) for more details.
```

- What went wrong?

```
glimpse(dat)
```

```
## Rows: 257  
## Columns: 1  
## $ `# Data from Jelle's Marble Run https://www.youtube.com/channel/UCYJdpnjuSWVOLgGT9fIzL0g` <chr>
```

- Let's look at the file in the terminal using the head command

```
head ./data/marbles.csv
```

```
# Data from Jelle's Marble Run https://www.youtube.com/channel/UCYJdpnjuSWVOLgGT9fIzL0g  
date,race,site,source,marble_name,team_name,time_s,pole,points,track_length_m,number_laps,avg_time_s  
15-Feb-20,S1Q1,Savage Speedway,https://youtu.be/JtsQ_UydjEI?t=356,Clementin,O'rangers,28.11,P1,NA,12.81,1  
15-Feb-20,S1Q1,Savage Speedway,https://youtu.be/JtsQ_UydjEI?t=356,Starry,Team Galactic,28.37,P2,NA,12.81,1  
15-Feb-20,S1Q1,Savage Speedway,https://youtu.be/JtsQ_UydjEI?t=356,Momo,Team Momo,28.4,P3,NA,12.81,1  
15-Feb-20,S1Q1,Savage Speedway,https://youtu.be/JtsQ_UydjEI?t=356,Yellow,Mellow Yellow,28.7,P4,NA,12.81,1  
15-Feb-20,S1Q1,Savage Speedway,https://youtu.be/JtsQ_UydjEI?t=356,Snowy,Snowballs,28.71,P5,NA,12.81,1  
15-Feb-20,S1Q1,Savage Speedway,https://youtu.be/JtsQ_UydjEI?t=356,Razzy,Raspberry Racers,28.72,P6,NA,12.81,1  
15-Feb-20,S1Q1,Savage Speedway,https://youtu.be/JtsQ_UydjEI?t=356,Prim,Team Primary,28.96,P7,NA,12.81,1  
15-Feb-20,S1Q1,Savage Speedway,https://youtu.be/JtsQ_UydjEI?t=356,Vespa,Hornets,29.11,P8,NA,12.81,1
```

- There is a line before the variable names!

Skipping lines when reading files

```
filename <- here::here("data", "marbles.csv")  
dat <- read_csv(filename, skip = 1)
```

```
##  
## — Column specification —————  
## cols(  
##   date = col_character(),  
##   race = col_character(),  
##   site = col_character(),  
##   source = col_character(),  
##   marble_name = col_character(),  
##   team_name = col_character(),  
##   time_s = col_double(),  
##   pole = col_character(),  
##   points = col_double(),  
##   track_length_m = col_double(),  
##   number_laps = col_double(),  
##   avg_time_lap = col_double(),  
##   host = col_character(),  
##   notes = col_character()  
## )
```

Parsing vectors

- The `parse_*()` functions

```
parse_logical(c(TRUE, FALSE, "TRUE", "FALSE", 1, 0, NA, "ABC"))
```

```
## Warning: 1 parsing failure.
## row col      expected actual
##   8  -- 1/0/T/F/TRUE/FALSE   ABC

## [1] TRUE FALSE TRUE FALSE TRUE FALSE   NA   NA
## attr(,"problems")
## # A tibble: 1 x 4
##   row col expected      actual
##   <int> <int> <chr>      <chr>
## 1     8   NA 1/0/T/F/TRUE/FALSE ABC
```

```
parse_double(c(2.4, "5.7", 22/7, NA, "ABC"))
```

```
## Warning: 1 parsing failure.
## row col expected actual
##   5  -- a double   ABC

## [1] 2.39999999999999911182 5.700000000000000177636 3.142857142857140129166
## attr(,"problems")
## # A tibble: 1 x 4
##   row col expected actual
##   <int> <int> <chr>      <chr>
## 1     5   NA a double ABC
```

NA

Parsing vectors

```
parse_integer(c(1, "3", 4.5, NA, "ABC"))
```

```
## Warning: 2 parsing failures.
## row col          expected actual
##   3  -- no trailing characters    4.5
##   5  -- an integer                ABC

## [1] 1 3 NA NA NA
## attr(,"problems")
## # A tibble: 2 x 4
##   row col expected          actual
##   <int> <int> <chr>          <chr>
## 1     3     NA no trailing characters 4.5
## 2     5     NA an integer                ABC
```

```
parse_factor(c("A", "B", "C", "A"))
```

```
## [1] A B C A
## Levels: A B C
```

Parsing vectors

```
parse_date(c("2020-10-31", "2020/10/28", NA))
```

```
## [1] "2020-10-31" "2020-10-28" NA
```

```
parse_date("02/18/2020", "%m/%d/%Y")
```

```
## [1] "2020-02-18"
```

```
parse_time(c("6:22:16", "22:16"))
```

```
## 06:22:16
```

```
## 22:16:00
```

Parsing a file

- When you load a file using `read_*()` functions, R guesses which `parse_*()` functions should be applied to each column

```
guess_parser(c(2.5, 7))
```

```
## [1] "double"
```

```
guess_parser(c(2.5, "3"))
```

```
## [1] "double"
```

```
guess_parser(c("ABC", "FALSE"))
```

```
## [1] "character"
```

```
guess_parser(c(TRUE, FALSE, NA))
```

```
## [1] "logical"
```


Writing to files

- Use `write_csv()` to write data to a csv file
 - Can load the data with `read_csv()`
- Use `write_rds()` and `read_rds()` to save and load compressed R data files
 - very useful when running long code to save the output to a file
- Use the `feather` package for file types that are compatible with both R and python
 - Cross-language and very fast
- Can check if a directory exists and create it if it doesn't

```
if(!dir.exists(here::here("results"))) {  
  dir.create(here::here("results"))  
}
```

Writing to files

```
write_csv(dat, path = here::here("results", "marbles-clean.csv"))
```

```
write_rds(dat, path = here::here("results", "marbles-clean.rds"))
```

```
# remove the data.frame
rm(dat)
glimpse(dat)
```

```
## Error in glimpse(dat): object 'dat' not found
```

```
# useful for saving long-running chunks of code
dat <- read_rds(here::here("results", "marbles-clean.rds"))
```

```
glimpse(dat)
```

```
## Rows: 256
## Columns: 14
## $ date      <chr> "15-Feb-20", "15-Feb-20", "15-Feb-20", "15-Feb-20", "15-Feb-20", "15-Feb-20"
## $ race      <chr> "S1Q1", "S1Q1", "S1Q1", "S1Q1", "S1Q1", "S1Q1", "S1Q1", "S1Q1", "S1Q1", "S1Q1"
## $ site      <chr> "Savage Speedway", "Savage Speedway", "Savage Speedway", "Savage Speedway",
## $ source    <chr> "https://youtu.be/JtsQ_UydjEI?t=356", "https://youtu.be/JtsQ_UydjEI?t=356",
## $ marble_name <chr> "Clementin", "Starry", "Momo", "Yellow", "Snowy", "Razzy", "Prim", "Vespa",
## $ team_name  <chr> "O'rangers", "Team Galactic", "Team Momo", "Mellow Yellow", "Snowballs", "Ra
## $ time_s     <dbl> 28.10999999999999943157, 28.370000000000000099476, 28.39999999999999857891, 2
## $ pole      <chr> "P1", "P2", "P3", "P4", "P5", "P6", "P7", "P8", "P9", "P10", "P11", "P12", "
## $ points     <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 25, 18, 15,
## $ track_length_m <dbl> 12.810000000000000049738, 12.810000000000000049738, 12.810000000000000049738, 1
```

Tidy data

- Consistent data formats make analysis much easier
- Tidy data has each observation as a row and each variable as a column

Tall vs. wide data

- "Tall" data

```
tall_data <- read.table(header=TRUE, text =
  'subject sex condition measurement
    1      M    control          7.9
    1      M    cond1           12.3
    1      M    cond2           10.7
    2      F    control          6.3
    2      F    cond1           10.6
    2      F    cond2           11.1
    3      F    control          9.5
    3      F    cond1           13.1
    3      F    cond2           13.8
    4      M    control          11.5
    4      M    cond1           13.4
    4      M    cond2           12.9
  ')
# Make sure the subject column is a factor
tall_data$subject <- factor(tall_data$subject)
```

- "Wide" data

```
wide_data <- read.table(header=TRUE, text =
  'subject sex control cond1 cond2
    1      M      7.9  12.3  10.7
    2      F      6.3  10.6  11.1
    3      F      9.5  13.1  13.8
    4      M     11.5  13.4  12.9
  ')
# Make sure the subject column is a factor
wide_data$subject <- factor(wide_data$subject)
```

Tall vs. wide data

tall_data

wide_data

##	subject	sex	condition	measurements	##	subject	sex	control
## 1	1	M	control	7.900000000000000355271	## 1	1	M	7.900000000000000355271 12.300000000000000710543
## 2	1	M	cond1	12.300000000000000710543	## 2	2	F	6.299999999999999822364 10.599999999999999644729
## 3	1	M	cond2	10.69999999999999929457	## 3	3	F	9.500000000000000000000 13.099999999999999644729
## 4	2	F	control	6.299999999999999822364	## 4	4	M	11.500000000000000000000 13.400000000000000710543
## 5	2	F	cond1	10.599999999999999644729				
## 6	2	F	cond2	11.099999999999999644729				
## 7	3	F	control	9.500000000000000000000				
## 8	3	F	cond1	13.099999999999999644729				
## 9	3	F	cond2	13.800000000000000710543				
## 10	4	M	control	11.500000000000000000000				
## 11	4	M	cond1	13.400000000000000355271				
## 12	4	M	cond2	12.900000000000000355271				

Tall vs. wide data

- In general, tall data is preferred
 - Easier to generate summaries of the data
 - Can generate key-value pairs (dictionaries)
 - Most statistical models require long data for inputs

Convert from tall to wide

```
tall_data
```

```
##      subject sex condition      measurement
## 1         1   M   control 7.900000000000000355271
## 2         1   M    cond1 12.300000000000000710543
## 3         1   M    cond2 10.699999999999999289457
## 4         2   F   control 6.299999999999999822364
## 5         2   F    cond1 10.599999999999999644729
## 6         2   F    cond2 11.099999999999999644729
## 7         3   F   control 9.500000000000000000000
## 8         3   F    cond1 13.099999999999999644729
## 9         3   F    cond2 13.800000000000000710543
## 10        4   M   control 11.500000000000000000000
## 11        4   M    cond1 13.400000000000000355271
## 12        4   M    cond2 12.900000000000000355271
```

```
tall_data %>%
  pivot_wider(names_from = condition, values_from = measurement)
```

```
## # A tibble: 4 x 5
##   subject sex   control cond1 cond2
##   <fct>   <chr>   <dbl> <dbl> <dbl>
## 1 1      M       7.9  12.3  10.7
## 2 2      F       6.3  10.6  11.1
## 3 3      F       9.5  13.1  13.8
## 4 4      M      11.5  13.4  12.9
```

Convert from wide to tall

```
wide_data
```

```
##   subject sex      control      cond1      cond2
## 1      1    M 7.900000000000000355271 12.30000000000000071054 10.699999999999928946
## 2      2    F 6.299999999999999822364 10.59999999999999964473 11.09999999999999964473
## 3      3    F 9.500000000000000000000 13.09999999999999964473 13.80000000000000071054
## 4      4    M 11.500000000000000000000 13.40000000000000035527 12.90000000000000035527
```

```
wide_data %>%
  pivot_longer(cols = c(control, cond1, cond2), names_to = "measurement")
```

```
## # A tibble: 12 x 4
##   subject sex measurement value
##   <fct>   <chr> <chr>      <dbl>
## 1 1      M    control    7.9
## 2 1      M    cond1     12.3
## 3 1      M    cond2     10.7
## 4 2      F    control    6.3
## 5 2      F    cond1     10.6
## 6 2      F    cond2     11.1
## 7 3      F    control    9.5
## 8 3      F    cond1     13.1
## 9 3      F    cond2     13.8
## 10 4     M    control   11.5
## 11 4     M    cond1     13.4
## 12 4     M    cond2     12.9
```


Completing data

- The `tibble` below does not have data for every case

```
df <- tibble(  
  group = c(1:2, 1),  
  item_id = c(1:2, 2),  
  item_name = c("a", "b", "b"),  
  value1 = 1:3,  
  value2 = 4:6  
)
```

- Complete the group variable so that `item_id` and `item_name` have all their possible combinations filled in

```
df %>% complete(group, nesting(item_id, item_name))
```

```
## # A tibble: 4 x 5  
##   group item_id item_name value1 value2  
##   <dbl>   <dbl> <chr>      <int> <int>  
## 1     1     1     a          1     4  
## 2     1     2     b          3     6  
## 3     2     1     a         NA    NA  
## 4     2     2     b          2     5
```

Completing data

- Fill in the completed values

```
df %>%  
  complete(group, nesting(item_id, item_name),  
            fill = list(value1 = "foo", value2 = "bar"))
```

```
## # A tibble: 4 x 5  
##   group item_id item_name value1 value2  
##   <dbl>   <dbl> <chr>      <chr> <chr>  
## 1     1       1 a         1     4  
## 2     1       2 b         3     6  
## 3     2       1 a         foo    bar  
## 4     2       2 b         2     5
```