Gov 50: 4. Data Wrangling

Matthew Blackwell

Harvard University

Roadmap

1. Data Wrangling
2. Operating on rows
3. Operating on columns
4. Operating on groups

1/ Data Wrangling

Data is messy

- Data almost **never** comes in the exact format you want to visualize

Data is messy

- Data almost **never** comes in the exact format you want to visualize
- The potential problems are numerous:

Data is messy

- Data almost **never** comes in the exact format you want to visualize
- The potential problems are numerous:
    - You need to change or transform a variable in some way.

Data is messy

- Data almost **never** comes in the exact format you want to visualize
- The potential problems are numerous:
    - You need to change or transform a variable in some way.
    - You need to focus on a particular part of the data.

Data is messy

- Data almost **never** comes in the exact format you want to visualize
- The potential problems are numerous:
    - You need to change or transform a variable in some way.
    - You need to focus on a particular part of the data.
    - You actually need a summary of a variable within groups.

Data is messy

- Data almost **never** comes in the exact format you want to visualize
- The potential problems are numerous:
    - You need to change or transform a variable in some way.
    - You need to focus on a particular part of the data.
    - You actually need a summary of a variable within groups.
    - You need to rename the variables.

Data is messy

- Data almost **never** comes in the exact format you want to visualize
- The potential problems are numerous:
    - You need to change or transform a variable in some way.
    - You need to focus on a particular part of the data.
    - You actually need a summary of a variable within groups.
    - You need to rename the variables.
    - You need to reorder the rows of the data.

Data is messy

- Data almost **never** comes in the exact format you want to visualize
- The potential problems are numerous:
    - You need to change or transform a variable in some way.
    - You need to focus on a particular part of the data.
    - You actually need a summary of a variable within groups.
    - You need to rename the variables.
    - You need to reorder the rows of the data.
- Today we'll talk about tools to do these tasks

data.frames vs tibbles

- The standard R object for datasets is the `data.frame`

data.frames vs tibbles

- The standard R object for datasets is the `data.frame`
    - Each column is a vector of the same length.

data.frames vs tibbles

- The standard R object for datasets is the `data.frame`
    - Each column is a vector of the same length.
    - Columns can be different types

data.frames vs tibbles

- The standard R object for datasets is the `data.frame`
    - Each column is a vector of the same length.
    - Columns can be different types
- Access columns with $: `mydata$myvariable`

data.frames vs tibbles

- The standard R object for datasets is the `data.frame`
  - Each column is a vector of the same length.
  - Columns can be different types
- Access columns with $: `mydata$myvariable`

data.frames vs tibbles

- The standard R object for datasets is the `data.frame`
    - Each column is a vector of the same length.
    - Columns can be different types
- Access columns with $: `mydata$myvariable`

```r
mtcars$mpg
```

```
##  [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8
## [12] 16.4 17.3 15.2 10.4 10.4 14.7 32.4 30.4 33.9 21.5 15.5
## [23] 15.2 13.3 19.2 27.3 26.0 30.4 15.8 19.7 15.0 21.4
```

Problems with data frames

```
mtcars
```

```
##                       mpg cyl  disp  hp drat   wt qsec vs am
## Mazda RX4            21.0   6 160.0 110 3.90 2.62 16.5  0  1
## Mazda RX4 Wag        21.0   6 160.0 110 3.90 2.88 17.0  0  1
## Datsun 710           22.8   4 108.0  93 3.85 2.32 18.6  1  1
## Hornet 4 Drive       21.4   6 258.0 110 3.08 3.21 19.4  1  0
## Hornet Sportabout    18.7   8 360.0 175 3.15 3.44 17.0  0  0
## Valiant              18.1   6 225.0 105 2.76 3.46 20.2  1  0
## Duster 360           14.3   8 360.0 245 3.21 3.57 15.8  0  0
## Merc 240D            24.4   4 146.7  62 3.69 3.19 20.0  1  0
## Merc 230             22.8   4 140.8  95 3.92 3.15 22.9  1  0
## Merc 280             19.2   6 167.6 123 3.92 3.44 18.3  1  0
## Merc 280C            17.8   6 167.6 123 3.92 3.44 18.9  1  0
## Merc 450SE           16.4   8 275.8 180 3.07 4.07 17.4  0  0
## Merc 450SL           17.3   8 275.8 180 3.07 3.73 17.6  0  0
## Merc 450SLC          15.2   8 275.8 180 3.07 3.78 18.0  0  0
## Cadillac Fleetwood   10.4   8 472.0 205 2.93 5.25 18.0  0  0
## Lincoln Continental  10.4   8 460.0 215 3.00 5.42 17.8  0  0
## Chrysler Imperial    14.7   8 440.0 230 3.23 5.34 17.4  0  0
## Fiat 128             32.4   4  78.7  66 4.08 2.20 19.5  1  1
## Honda Civic          30.4   4  75.7  52 4.93 1.61 18.5  1  1
## Toyota Corolla       33.9   4  71.1  65 4.22 1.83 19.9  1  1
```

tibbles: a tidyverse alternative

```
midwest
```

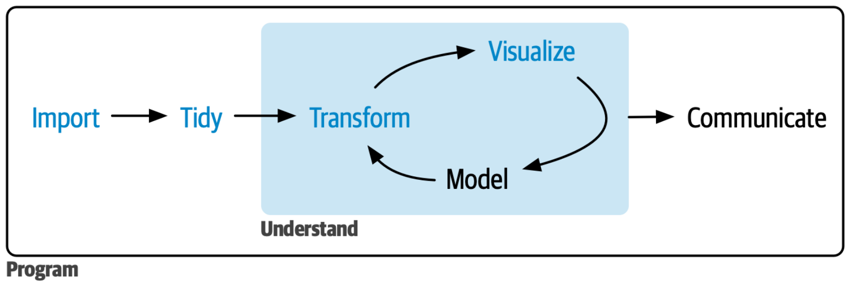
```
## # A tibble: 437 x 28
##       PID county    state  area poptotal popdensity
##     <int> <chr>     <chr> <dbl>    <int>      <dbl>
##  1    561 ADAMS     IL    0.052    66090      1271.
##  2    562 ALEXANDER IL    0.014    10626       759
##  3    563 BOND      IL    0.022    14991       681.
##  4    564 BOONE     IL    0.017    30806      1812.
##  5    565 BROWN     IL    0.018     5836       324.
##  6    566 BUREAU    IL    0.05     35688       714.
##  7    567 CALHOUN   IL    0.017     5322       313.
##  8    568 CARROLL   IL    0.027    16805       622.
##  9    569 CASS      IL    0.024    13437       560.
## 10    570 CHAMPAIGN IL    0.058   173025      2983.
## # ... with 427 more rows, and 22 more variables:
## #   popwhite <int>, popblack <int>,
## #   popamerindian <int>, popasian <int>,
## #   popother <int>, percwhite <dbl>, percblack <dbl>,
## #   percamerindan <dbl>, percasian <dbl>,
```

rows x columns

column types

abridged output

Transform-Visualize-Model cycle



Credit: Hadley Wickham

dplyr: a package for data transformation

- All `dplyr` functions:

dplyr: a package for data transformation

- All `dplyr` functions:
    - Take a dataset as their first argument

dplyr: a package for data transformation

- All `dplyr` functions:
  - Take a dataset as their first argument
  - Manipulate the dataset in some way

dplyr: a package for data transformation

- All `dplyr` functions:
  - Take a dataset as their first argument
  - Manipulate the dataset in some way
  - Returns the manipulated dataset

pipe

Nested calls can be hard to read (have to read inside out):

```
f(g(h(r(x))))
```

pipe

Nested calls can be hard to read (have to read inside out):

```
f(g(h(r(x))))
```

The pipe |> allows us to move output between functions (|> = “and then”):

```
x |>
  r() |>
  h() |>
  g() |>
  h()
```

The piped output goes to the first argument by default.

Local news data

- How does station ownership affect local news coverage?

Local news data

- How does station ownership affect local news coverage?
- Martin and McCrain (2019) use data on local news at TV stations before and after a large acquisition by a conglomorate.

Local news data

- How does station ownership affect local news coverage?
- Martin and McCrain (2019) use data on local news at TV stations before and after a large acquisition by a conglomorate.

Local news data

- How does station ownership affect local news coverage?
- Martin and McCrain (2019) use data on local news at TV stations before and after a large acquisition by a conglomorate.

Variable	Description
`callsign`	Callsign of the station
`affiliation`	Network affiliation of the station
`date`	Airdate of news
`weekday`	Day of the week of airdate
`ideology`	Measure of news slant (bigger is more conservative)
`national_politics`	Avg proportion of segments on national politics
`local_politics`	Avg proportion of segments on national politics
`sinclair2017`	Station acquired by Sinclair group in Sept 2017
`post`	Date is before/after acquisition (0/1)

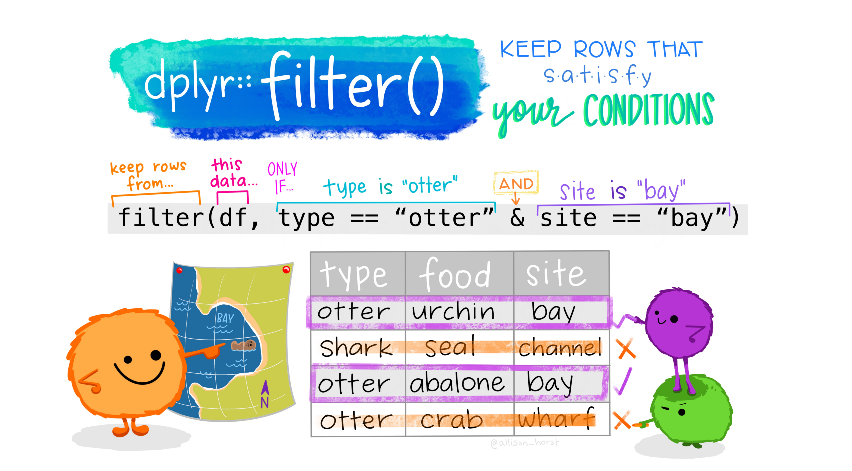
```r
library(gov50data)
news
```

```
## # A tibble: 3,137 x 10
##    callsign affiliation date       weekday ideology
##    <chr>    <chr>       <date>     <ord>      <dbl>
##  1 KRBC     NBC         2017-06-05 Mon      NA     
##  2 KTAB     CBS         2017-06-05 Mon      NA     
##  3 KXVA     FOX         2017-06-05 Mon      NA     
##  4 KPAX     CBS         2017-06-06 Tue      NA     
##  5 KTAB     CBS         2017-06-06 Tue      NA     
##  6 KECI     NBC         2017-06-07 Wed       0.0655
##  7 KPAX     CBS         2017-06-07 Wed       0.0853
##  8 KRBC     NBC         2017-06-07 Wed       0.0183
##  9 KTAB     CBS         2017-06-07 Wed       0.0850
## 10 KTMF     ABC         2017-06-07 Wed       0.0842
## # i 3,127 more rows
## # i 5 more variables: national_politics <dbl>,
## #   local_politics <dbl>, sinclair2017 <dbl>, post <dbl>,
## #   month <ord>
```

2/ Operating on rows

filter()

`filter()` selects rows that satisfy the argument you pass it:



```
filter(df, type == “otter” & site == “bay”)
```

type	food	site
otter	urchin	bay
shark	seal	channel
otter	abalone	bay
otter	crab	wharf

Credit: Allison Horst

```r
news |>
  filter(weekday == "Tue")
```

```
## # A tibble: 626 x 10
##    callsign affiliation date       weekday ideology
##    <chr>    <chr>       <date>     <ord>      <dbl>
##  1 KPAX     CBS         2017-06-06 Tue     NA
##  2 KTAB     CBS         2017-06-06 Tue     NA
##  3 KAEF     ABC         2017-06-13 Tue      0.0242
##  4 KBVU     FOX         2017-06-13 Tue      0.00894
##  5 KBZK     CBS         2017-06-13 Tue      0.129
##  6 KCVU     FOX         2017-06-13 Tue      0.114
##  7 KECI     NBC         2017-06-13 Tue      0.115
##  8 KHSL     CBS         2017-06-13 Tue      0.0821
##  9 KNVN     NBC         2017-06-13 Tue      0.120
## 10 KPAX     CBS         2017-06-13 Tue      0.0984
## # i 616 more rows
## # i 5 more variables: national_politics <dbl>,
## #   local_politics <dbl>, sinclair2017 <dbl>, post <dbl>,
## #   month <ord>
```

Multiple conditions means “and”

```r
news |>
  filter(weekday == "Tue",
         affiliation == "FOX")
```

```
## # A tibble: 130 x 10
##    callsign affiliation date       weekday ideology
##    <chr>    <chr>       <date>     <ord>      <dbl>
##  1 KBVU     FOX         2017-06-13 Tue      0.00894
##  2 KCVU     FOX         2017-06-13 Tue      0.114
##  3 WEMT     FOX         2017-06-13 Tue      0.235
##  4 WYDO     FOX         2017-06-13 Tue      0.0949
##  5 KBVU     FOX         2017-06-20 Tue     NA
##  6 KCVU     FOX         2017-06-20 Tue     NA
##  7 KXVA     FOX         2017-06-20 Tue     NA
##  8 WEMT     FOX         2017-06-20 Tue      0.268
##  9 WYDO     FOX         2017-06-20 Tue      0.0590
## 10 KBVU     FOX         2017-06-27 Tue     NA
## # i 120 more rows
## # i 5 more variables: national_politics <dbl>,
## #   local_politics <dbl>, sinclair2017 <dbl>, post <dbl>,
## #   month <ord>
```

logicals

- Comparing two values/vectors:

logicals

- Comparing two values/vectors:
    - `>`/`>=`: greater than/greater than or equal to

logicals

- Comparing two values/vectors:
  - >/>=: greater than/greater than or equal to
  - </<=: less than/less than or equal to

logicals

- Comparing two values/vectors:
  - >/>=: greater than/greater than or equal to
  - </<=: less than/less than or equal to
  - ==/!=: equal to/not equal to

logicals

- Comparing two values/vectors:
    - >/>=: greater than/greater than or equal to
    - </<=: less than/less than or equal to
    - ==/!=: equal to/not equal to
- Combining multiple logical statements:

logicals

- Comparing two values/vectors:
  - >/>=: greater than/greater than or equal to
  - </<=: less than/less than or equal to
  - ==/!=: equal to/not equal to
- Combining multiple logical statements:
  - &: and

logicals

- Comparing two values/vectors:
  - >/>=: greater than/greater than or equal to
  - </<=: less than/less than or equal to
  - ==/!=: equal to/not equal to
- Combining multiple logical statements:
  - &: and
  - |: or

Common gotcha!

```r
news |>
  filter(weekday = "Tue")
```

```
## Error in `filter()`:
## ! We detected a named input.
## i This usually means that you've used `=` instead of `==`.
## i Did you mean `weekday == "Tue"`?
```

```r
news |>
  filter(affiliation == "FOX" | affiliation == "ABC")
```

```
## # A tibble: 1,525 x 10
##    callsign affiliation date       weekday  ideology
##    <chr>    <chr>       <date>     <ord>       <dbl>
##  1 KXVA     FOX         2017-06-05 Mon     NA
##  2 KTMF     ABC         2017-06-07 Wed      0.0842
##  3 KTXS     ABC         2017-06-07 Wed     -0.000488
##  4 KXVA     FOX         2017-06-07 Wed     NA
##  5 KAEF     ABC         2017-06-08 Thu      0.0426
##  6 KBVU     FOX         2017-06-08 Thu     -0.0860
##  7 KTMF     ABC         2017-06-08 Thu      0.0433
##  8 KTXS     ABC         2017-06-08 Thu      0.0627
##  9 KXVA     FOX         2017-06-08 Thu     NA
## 10 WCTI     ABC         2017-06-08 Thu      0.139
## # i 1,515 more rows
## # i 5 more variables: national_politics <dbl>,
## #   local_politics <dbl>, sinclair2017 <dbl>, post <dbl>,
## #   month <ord>
```

```r
news |>
  filter(ideology < 0 & weekday == "Tue")
```

```
## # A tibble: 66 x 10
##    callsign affiliation date       weekday ideology
##    <chr>    <chr>       <date>     <ord>      <dbl>
##  1 KAEF     ABC         2017-06-27 Tue     -0.0117
##  2 KECI     NBC         2017-06-27 Tue     -0.00362
##  3 KHSL     CBS         2017-06-27 Tue     -0.0735
##  4 KNVN     NBC         2017-06-27 Tue     -0.0175
##  5 KPAX     CBS         2017-06-27 Tue     -0.134
##  6 KTXS     ABC         2017-06-27 Tue     -0.0307
##  7 WCTI     ABC         2017-06-27 Tue     -0.0308
##  8 WITN     NBC         2017-06-27 Tue     -0.0233
##  9 WJHL     CBS         2017-06-27 Tue     -0.00388
## 10 WNCT     CBS         2017-06-27 Tue     -0.130
## # i 56 more rows
## # i 5 more variables: national_politics <dbl>,
## #   local_politics <dbl>, sinclair2017 <dbl>, post <dbl>,
## #   month <ord>
```

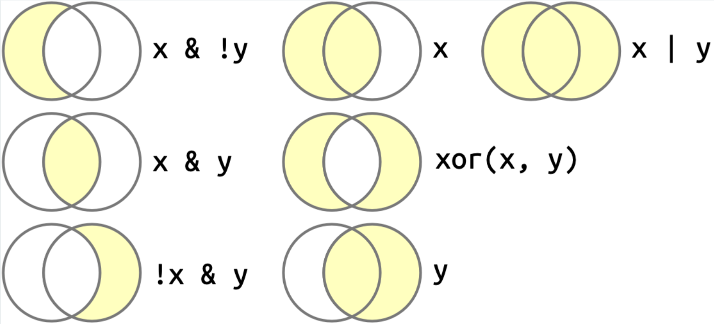
Combining %in%

When combining | and ==, useful to use %in%:

```
news |>
  filter(weekday %in% c("Mon", "Fri"))
```

```
## # A tibble: 1,253 x 10
##    callsign affiliation date       weekday ideology
##    <chr>    <chr>       <date>     <ord>      <dbl>
##  1 KRBC     NBC         2017-06-05 Mon       NA
##  2 KTAB     CBS         2017-06-05 Mon       NA
##  3 KXVA     FOX         2017-06-05 Mon       NA
##  4 KAEF     ABC         2017-06-09 Fri        0.0870
##  5 KBVU     FOX         2017-06-09 Fri       NA
##  6 KECI     NBC         2017-06-09 Fri        0.115
##  7 KPAX     CBS         2017-06-09 Fri        0.0882
##  8 KRBC     NBC         2017-06-09 Fri        0.0929
##  9 KTAB     CBS         2017-06-09 Fri        0.0588
## 10 KTMF     ABC         2017-06-09 Fri       NA
## # i 1,243 more rows
## # i 5 more variables: national_politics <dbl>,
## #   local_politics <dbl>, sinclair2017 <dbl>, post <dbl>,
## #   month <ord>
```

Complicated logicals



`arrange()`

`arrange()` will reorder the rows based on the values of the columns.

arrange()

arrange() will reorder the rows based on the values of the columns.

With multiple arguments, sort by first argument, then second, then third...

Arrange by callsign then date

```r
news |>
  arrange(callsign, date)
```

```
## # A tibble: 3,137 x 10
##    callsign affiliation date       weekday ideology
##    <chr>    <chr>       <date>     <ord>      <dbl>
##   1 KAEF     ABC         2017-06-08 Thu         0.0426
##   2 KAEF     ABC         2017-06-09 Fri         0.0870
##   3 KAEF     ABC         2017-06-12 Mon         0.0135
##   4 KAEF     ABC         2017-06-13 Tue         0.0242
##   5 KAEF     ABC         2017-06-14 Wed         0.123
##   6 KAEF     ABC         2017-06-15 Thu         0.0778
##   7 KAEF     ABC         2017-06-16 Fri        NA
##   8 KAEF     ABC         2017-06-19 Mon         0.778
##   9 KAEF     ABC         2017-06-20 Tue         0.115
## 10 KAEF     ABC         2017-06-21 Wed        -0.315
## # i 3,127 more rows
## # i 5 more variables: national_politics <dbl>,
## #   local_politics <dbl>, sinclair2017 <dbl>, post <dbl>,
## #   month <ord>
```

Which station-dates were the most liberal?

```r
news |>
  arrange(ideology)
```

```
## # A tibble: 3,137 x 10
##    callsign affiliation date       weekday ideology
##    <chr>    <chr>       <date>     <ord>      <dbl>
##  1 KRBC     NBC         2017-10-19 Thu       -0.674
##  2 WJHL     CBS         2017-12-08 Fri       -0.673
##  3 KRBC     NBC         2017-10-18 Wed       -0.586
##  4 KCVU     FOX         2017-06-22 Thu       -0.414
##  5 KRBC     NBC         2017-12-11 Mon       -0.365
##  6 KAEF     ABC         2017-06-21 Wed       -0.315
##  7 KTMF     ABC         2017-12-01 Fri       -0.303
##  8 KWYB     ABC         2017-12-01 Fri       -0.303
##  9 KTVM     NBC         2017-09-01 Fri       -0.302
## 10 KNVN     NBC         2017-12-08 Fri       -0.299
## # i 3,127 more rows
## # i 5 more variables: national_politics <dbl>,
## #   local_politics <dbl>, sinclair2017 <dbl>, post <dbl>,
## #   month <ord>
```

Which station-dates were the most conservative?

Use `desc()` to reverse the order:

```
news |>
  arrange(desc(ideology))
```

```
## # A tibble: 3,137 x 10
##    callsign affiliation date       weekday ideology
##    <chr>    <chr>       <date>     <ord>      <dbl>
##  1 KAEF     ABC         2017-06-19 Mon        0.778
##  2 WYDO     FOX         2017-07-19 Wed        0.580
##  3 KRCR     ABC         2017-10-03 Tue        0.566
##  4 KAEF     ABC         2017-10-18 Wed        0.496
##  5 KBVU     FOX         2017-11-16 Thu        0.491
##  6 KTMF     ABC         2017-11-06 Mon        0.455
##  7 KAEF     ABC         2017-06-29 Thu        0.447
##  8 KPAX     CBS         2017-11-23 Thu        0.437
##  9 KTAB     CBS         2017-11-16 Thu        0.427
## 10 KCVU     FOX         2017-07-06 Thu        0.406
## # i 3,127 more rows
## # i 5 more variables: national_politics <dbl>,
## #   local_politics <dbl>, sinclair2017 <dbl>, post <dbl>,
## #   month <ord>
```

3/ Operating on columns

select():

`select()` selects columns via their names.

Selecting based on names

```r
news |>
  select(callsign, date, ideology)
```

```
## # A tibble: 3,137 x 3
##    callsign date       ideology
##    <chr>    <date>        <dbl>
##  1 KRBC     2017-06-05   NA
##  2 KTAB     2017-06-05   NA
##  3 KXVA     2017-06-05   NA
##  4 KPAX     2017-06-06   NA
##  5 KTAB     2017-06-06   NA
##  6 KECI     2017-06-07    0.0655
##  7 KPAX     2017-06-07    0.0853
##  8 KRBC     2017-06-07    0.0183
##  9 KTAB     2017-06-07    0.0850
## 10 KTMF     2017-06-07    0.0842
## # i 3,127 more rows
```

Selecting based on a range of variables

```r
news |>
  select(callsign:ideology)
```

```
## # A tibble: 3,137 x 5
##    callsign affiliation date       weekday ideology
##    <chr>    <chr>       <date>     <ord>      <dbl>
##  1 KRBC     NBC         2017-06-05 Mon       NA
##  2 KTAB     CBS         2017-06-05 Mon       NA
##  3 KXVA     FOX         2017-06-05 Mon       NA
##  4 KPAX     CBS         2017-06-06 Tue       NA
##  5 KTAB     CBS         2017-06-06 Tue       NA
##  6 KECI     NBC         2017-06-07 Wed        0.0655
##  7 KPAX     CBS         2017-06-07 Wed        0.0853
##  8 KRBC     NBC         2017-06-07 Wed        0.0183
##  9 KTAB     CBS         2017-06-07 Wed        0.0850
## 10 KTMF     ABC         2017-06-07 Wed        0.0842
## # i 3,127 more rows
```

Selecting all not in a range

```r
news |>
  select(!callsign:ideology)
```

```
## # A tibble: 3,137 x 5
##    national_politics local_politics sinclair2017  post month
##                <dbl>          <dbl>        <dbl> <dbl> <ord>
##  1           0.0286          0.0190            0     0 Jun
##  2           0.0286          0.0190            0     0 Jun
##  3           0.0393          0.0262            0     0 Jun
##  4           0.00357         0.194             0     0 Jun
##  5           0.0945          0.109             0     0 Jun
##  6           0.225           0.148             1     0 Jun
##  7           0.283           0.123             0     0 Jun
##  8           0.130           0.189             0     0 Jun
##  9           0.0901          0.138             0     0 Jun
## 10           0.152           0.129             0     0 Jun
## # i 3,127 more rows
```

Selecting all numeric columns

```r
news |>
  select(where(is.numeric))
```

```
## # A tibble: 3,137 x 5
##    ideology national_politics local_politics sinclair2017
##       <dbl>             <dbl>          <dbl>        <dbl>
##  1  NA                0.0286          0.0190            0
##  2  NA                0.0286          0.0190            0
##  3  NA                0.0393          0.0262            0
##  4  NA                0.00357         0.194             0
##  5  NA                0.0945          0.109             0
##  6   0.0655           0.225           0.148             1
##  7   0.0853           0.283           0.123             0
##  8   0.0183           0.130           0.189             0
##  9   0.0850           0.0901          0.138             0
## 10   0.0842           0.152           0.129             0
## # i 3,127 more rows
## # i 1 more variable: post <dbl>
```

Combining multiple selections

```r
news |>
  select(callsign:weekday, ends_with("politics"))
```

```
## # A tibble: 3,137 x 6
##    callsign affiliation date       weekday national_politics
##    <chr>    <chr>       <date>     <ord>               <dbl>
##  1 KRBC     NBC         2017-06-05 Mon               0.0286
##  2 KTAB     CBS         2017-06-05 Mon               0.0286
##  3 KXVA     FOX         2017-06-05 Mon               0.0393
##  4 KPAX     CBS         2017-06-06 Tue               0.00357
##  5 KTAB     CBS         2017-06-06 Tue               0.0945
##  6 KECI     NBC         2017-06-07 Wed               0.225
##  7 KPAX     CBS         2017-06-07 Wed               0.283
##  8 KRBC     NBC         2017-06-07 Wed               0.130
##  9 KTAB     CBS         2017-06-07 Wed               0.0901
## 10 KTMF     ABC         2017-06-07 Wed               0.152
## # i 3,127 more rows
## # i 1 more variable: local_politics <dbl>
```

rename()

`rename(new_name = old_name)` renames the `old_name` variable to `new_name`

```r
news |>
  rename(call_sign = callsign)
```

```
## # A tibble: 3,137 x 10
##    call_sign affiliation date       weekday ideology
##    <chr>     <chr>       <date>     <ord>      <dbl>
##  1 KRBC      NBC         2017-06-05 Mon      NA
##  2 KTAB      CBS         2017-06-05 Mon      NA
##  3 KXVA      FOX         2017-06-05 Mon      NA
##  4 KPAX      CBS         2017-06-06 Tue      NA
##  5 KTAB      CBS         2017-06-06 Tue      NA
##  6 KECI      NBC         2017-06-07 Wed       0.0655
##  7 KPAX      CBS         2017-06-07 Wed       0.0853
##  8 KRBC      NBC         2017-06-07 Wed       0.0183
##  9 KTAB      CBS         2017-06-07 Wed       0.0850
## 10 KTMF      ABC         2017-06-07 Wed       0.0842
## # i 3,127 more rows
## # i 5 more variables: national_politics <dbl>,
## #   local_politics <dbl>, sinclair2017 <dbl>, post <dbl>,
## #   month <ord>
```

mutate()

`mutate(new_var = fun(old_vars))` adds new columns that are functions of existing columns.

```r
news |>
  mutate(
    national_local_diff = national_politics - local_politics,
    national_politics_perc = national_politics * 100
  ) |>
  select(callsign, date, national_politics, local_politics,
         national_local_diff, national_politics_perc)
```

```
## # A tibble: 3,137 x 6
##    callsign date       national_politics local_politics national_local_diff national_politics_perc
##    <chr>    <date>                 <dbl>          <dbl>               <dbl>                  <dbl>
##  1 KRBC     2017-06-05           0.0286          0.0190             0.00952                  2.86
##  2 KTAB     2017-06-05           0.0286          0.0190             0.00952                  2.86
##  3 KXVA     2017-06-05           0.0393          0.0262             0.0131                   3.93
##  4 KPAX     2017-06-06           0.00357         0.194             -0.191                    0.357
##  5 KTAB     2017-06-06           0.0945          0.109             -0.0145                   9.45
##  6 KECI     2017-06-07           0.225           0.148              0.0761                  22.5
##  7 KPAX     2017-06-07           0.283           0.123              0.160                   28.3
##  8 KRBC     2017-06-07           0.130           0.189             -0.0589                  13.0
##  9 KTAB     2017-06-07           0.0901          0.138             -0.0476                   9.01
## 10 KTMF     2017-06-07           0.152           0.129              0.0229                  15.2
## # i 3,127 more rows
```

`if_else()`

`if_else(test_condition, yes, no)` allows us to create a vector that depends on a logical

`if_else()`

`if_else(test_condition, yes, no)` allows us to create a vector that depends on a logical

New vector gets `yes` expression when `test_condition` is TRUE, no otherwise

```
news |>
  mutate(Ownership = if_else(sinclair2017 == 1,
                             "Acquired by Sinclair",
                             "Not Acquired")) |>
  select(callsign, affiliation, date, Ownership)
```

```
## # A tibble: 3,137 x 4
##    callsign affiliation date       Ownership
##    <chr>    <chr>       <date>     <chr>
##  1 KRBC     NBC         2017-06-05 Not Acquired
##  2 KTAB     CBS         2017-06-05 Not Acquired
##  3 KXVA     FOX         2017-06-05 Not Acquired
##  4 KPAX     CBS         2017-06-06 Not Acquired
##  5 KTAB     CBS         2017-06-06 Not Acquired
##  6 KECI     NBC         2017-06-07 Acquired by Sinclair
##  7 KPAX     CBS         2017-06-07 Not Acquired
##  8 KRBC     NBC         2017-06-07 Not Acquired
##  9 KTAB     CBS         2017-06-07 Not Acquired
## 10 KTMF     ABC         2017-06-07 Not Acquired
## # i 3,127 more rows
```

4/ Operating on groups

group_by()

group_by(var) divides the data into groups based on the var variable.

`group_by()`

`group_by(var)` divides the data into groups based on the `var` variable.

Doesn't change data yet, but subsequent operations will by `var`.

```r
news |>
  group_by(month)
```

```
## # A tibble: 3,137 x 10
## # Groups:   month [7]
##    callsign affiliation date       weekday ideology national_politics
##    <chr>    <chr>       <date>     <ord>      <dbl>             <dbl>
##  1 KRBC     NBC         2017-06-05 Mon      NA                 0.0286
##  2 KTAB     CBS         2017-06-05 Mon      NA                 0.0286
##  3 KXVA     FOX         2017-06-05 Mon      NA                 0.0393
##  4 KPAX     CBS         2017-06-06 Tue      NA                 0.00357
##  5 KTAB     CBS         2017-06-06 Tue      NA                 0.0945
##  6 KECI     NBC         2017-06-07 Wed       0.0655            0.225
##  7 KPAX     CBS         2017-06-07 Wed       0.0853            0.283
##  8 KRBC     NBC         2017-06-07 Wed       0.0183            0.130
##  9 KTAB     CBS         2017-06-07 Wed       0.0850            0.0901
## 10 KTMF     ABC         2017-06-07 Wed       0.0842            0.152
## # i 3,127 more rows
## # i 4 more variables: local_politics <dbl>, sinclair2017 <dbl>,
## #   post <dbl>, month <ord>
```

summarize()

`summarize(sum_var = fun(curr_var))` calculates summaries of variables by groups.

Ideological slant by weekday

```
news |>
  group_by(month) |>
  summarize(
    slant_mean = mean(ideology, na.rm = TRUE)
  )
```

```
## # A tibble: 7 x 2
##   month slant_mean
##   <ord>      <dbl>
## 1 Jun       0.0786
## 2 Jul       0.103
## 3 Aug       0.105
## 4 Sep       0.0751
## 5 Oct       0.0862
## 6 Nov       0.0972
## 7 Dec       0.0774
```

Summaries by ownership and pre/post

```r
news |>
  group_by(sinclair2017, post) |>
  summarize(
    slant_mean = mean(ideology, na.rm = TRUE),
    national_mean = mean(national_politics, na.rm = TRUE)
  )
```

```
## # A tibble: 4 x 4
## # Groups:   sinclair2017 [2]
##   sinclair2017  post slant_mean national_mean
##          <dbl> <dbl>      <dbl>         <dbl>
## 1            0     0     0.100          0.118
## 2            0     1     0.0768         0.107
## 3            1     0     0.0936         0.124
## 4            1     1     0.0938         0.144
```

Summarize across types of variables

`across()` will apply a summary function across many variables

```r
news |>
  group_by(sinclair2017, post) |>
  summarize(
    across(where(is.numeric), mean, na.rm = TRUE),
  )
```

```
## # A tibble: 4 x 5
## # Groups:   sinclair2017 [2]
##   sinclair2017  post ideology national_politics local_politics
##          <dbl> <dbl>    <dbl>             <dbl>          <dbl>
## 1            0     0   0.100              0.118          0.158
## 2            0     1   0.0768             0.107          0.150
## 3            1     0   0.0936             0.124          0.170
## 4            1     1   0.0938             0.144          0.147
```