Advanced Data Journalism: Doing More with R

Class 2: Joins and Strings

Andrew Ba Tran

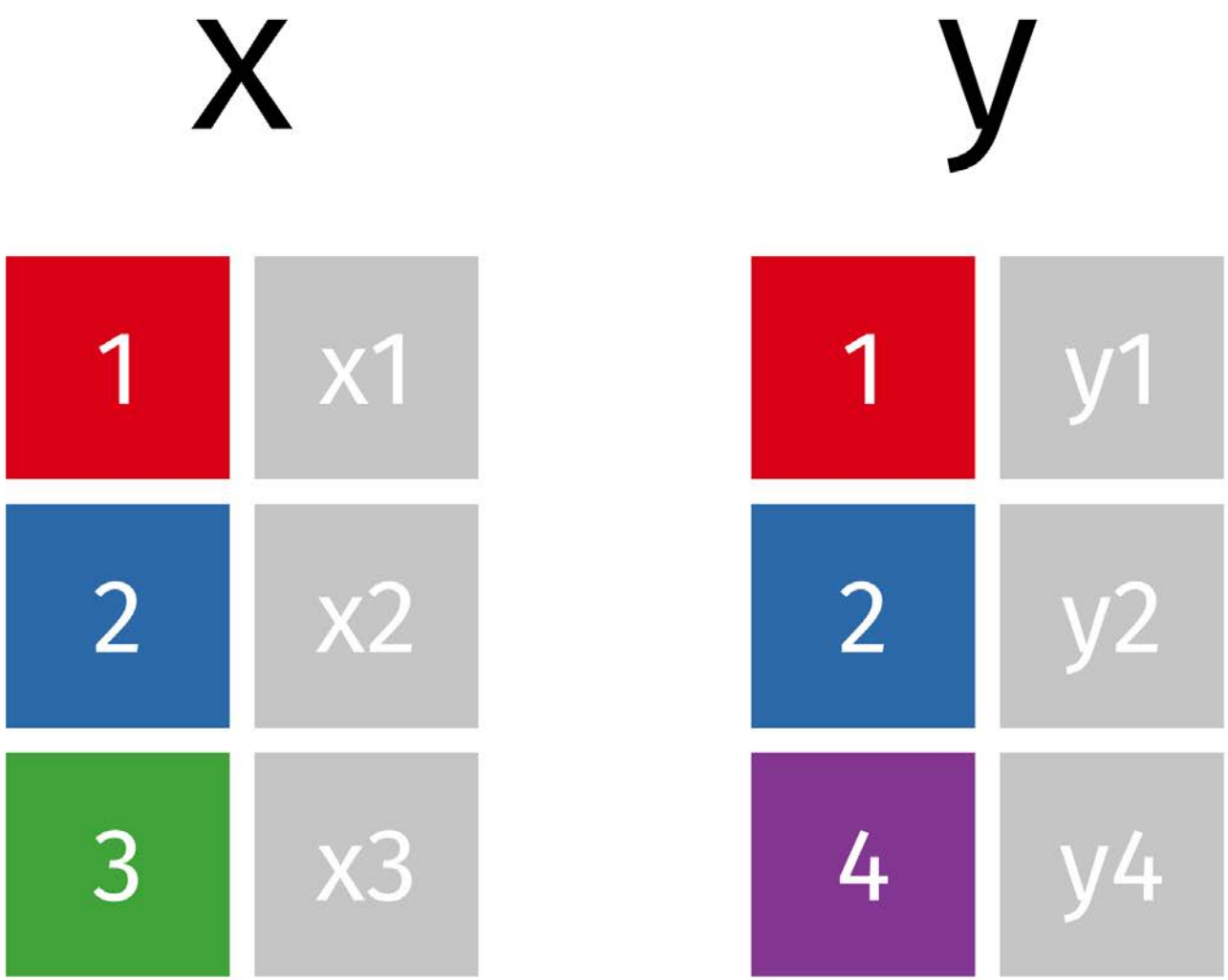
Let's start out with two data frames: x and y

```
x <- data.frame(id=c(1,2,3), x=c("x1", "x2", "x3"))

x
```

```
  id  x
1  1 x1
2  2 x2
3  3 x3
```

```
y <- data.frame(id=c(1,2,4), y=c("y1", "y2", "y4"))

y
```

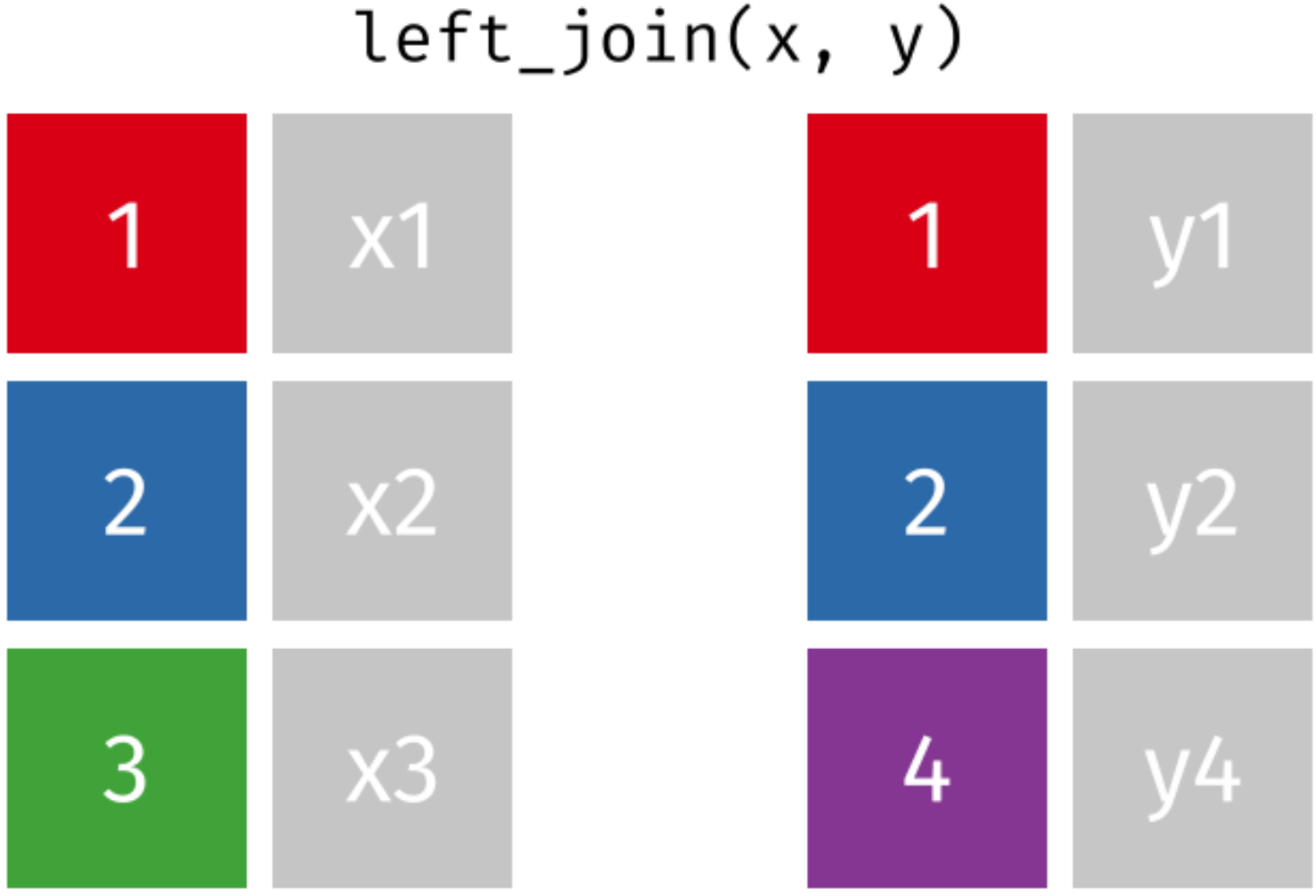
```
  id  y
1  1 y1
2  2 y2
3  4 y4
```


x
y
1
x1
2
x2
3
x3
1
y1
2
y2
4
y4

left_join()

```
left_join(x, y)
```

```
  id  x    y
1  1 x1   y1
2  2 x2   y2
3  3 x3 <NA>
```


left_join(x, y)
1
x1
2
x2
3
x3
1
y1
2
y2
4
y4

Two data frames: x and y but with different column names

```
x <- data.frame(id=c(1,2,3), x=c("x1", "x2", "x3"))

x
```

```
  id  x
1  1 x1
2  2 x2
3  3 x3
```

```
y <- data.frame(new_id=c(1,2,4), y=c("y1", "y2", "y4"))

y
```

```
  new_id  y
1      1 y1
2      2 y2
3      4 y4
```

```
left_join(x, y, by=c("id"="new_id"))
```

```
  id  x    y
1  1 x1   y1
2  2 x2   y2
3  3 x3 <NA>
```

Be careful of repeated data!

```
x <- data.frame(id=c(1,2,3),
                x=c("x1", "x2", "x3"))
```

Be careful of repeated data!

```
x <- data.frame(id=c(1,2,3),
                x=c("x1", "x2", "x3"))

x
```

```
  id  x
1  1 x1
2  2 x2
3  3 x3
```

Be careful of repeated data!

```
x <- data.frame(id=c(1,2,3),
                x=c("x1", "x2", "x3"))

x

y <- data.frame(id=c(1,2,4,2),
                y=c("y1", "y2", "y4", "y5"))
```

```
  id  x
1  1 x1
2  2 x2
3  3 x3
```

Be careful of repeated data!

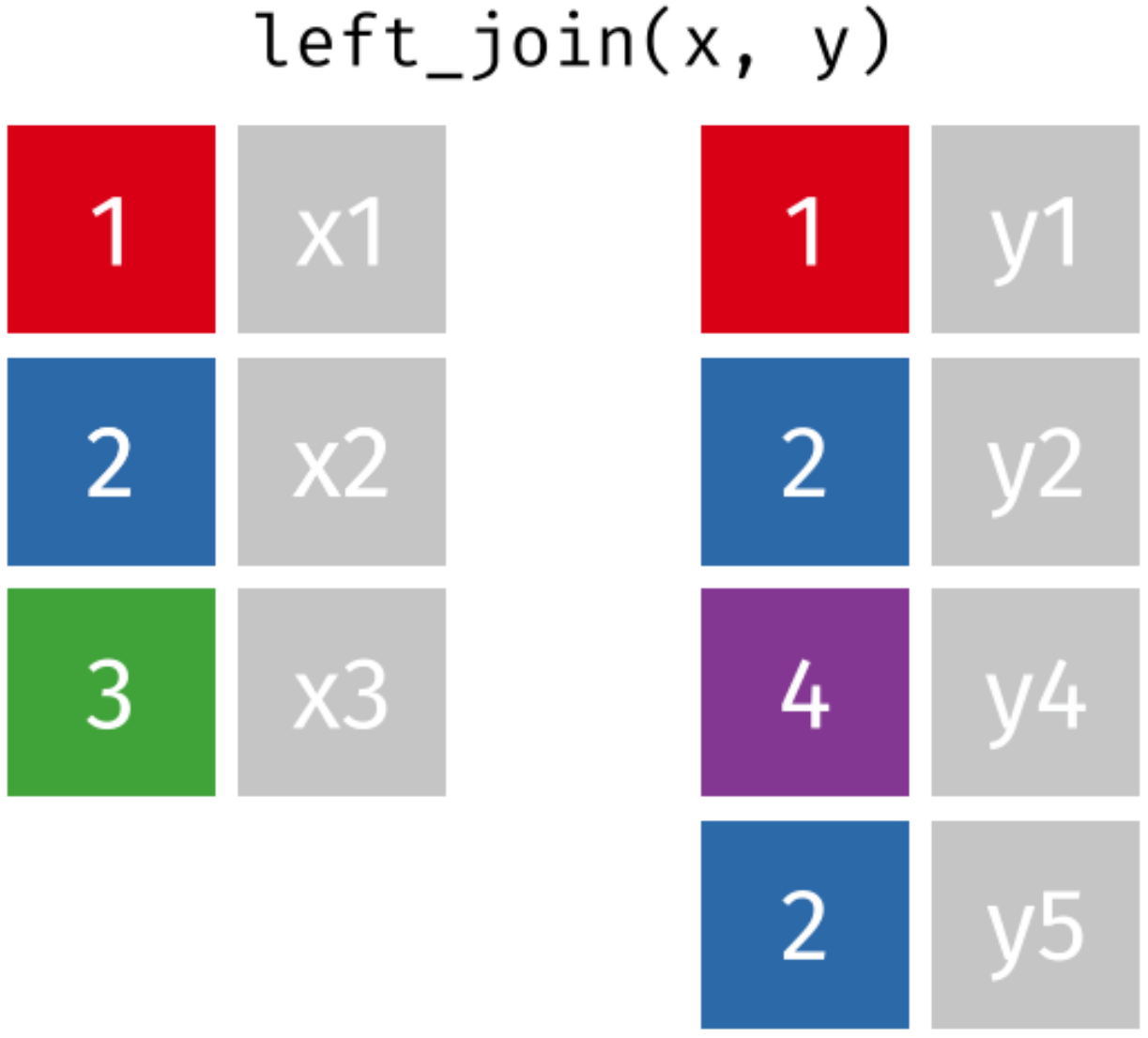
```
x <- data.frame(id=c(1,2,3),
                x=c("x1", "x2", "x3"))

x

y <- data.frame(id=c(1,2,4,2),
                y=c("y1", "y2", "y4", "y5"))

y
```

```
  id  x
1  1 x1
2  2 x2
3  3 x3

  id  y
1  1 y1
2  2 y2
3  4 y4
4  2 y5
```

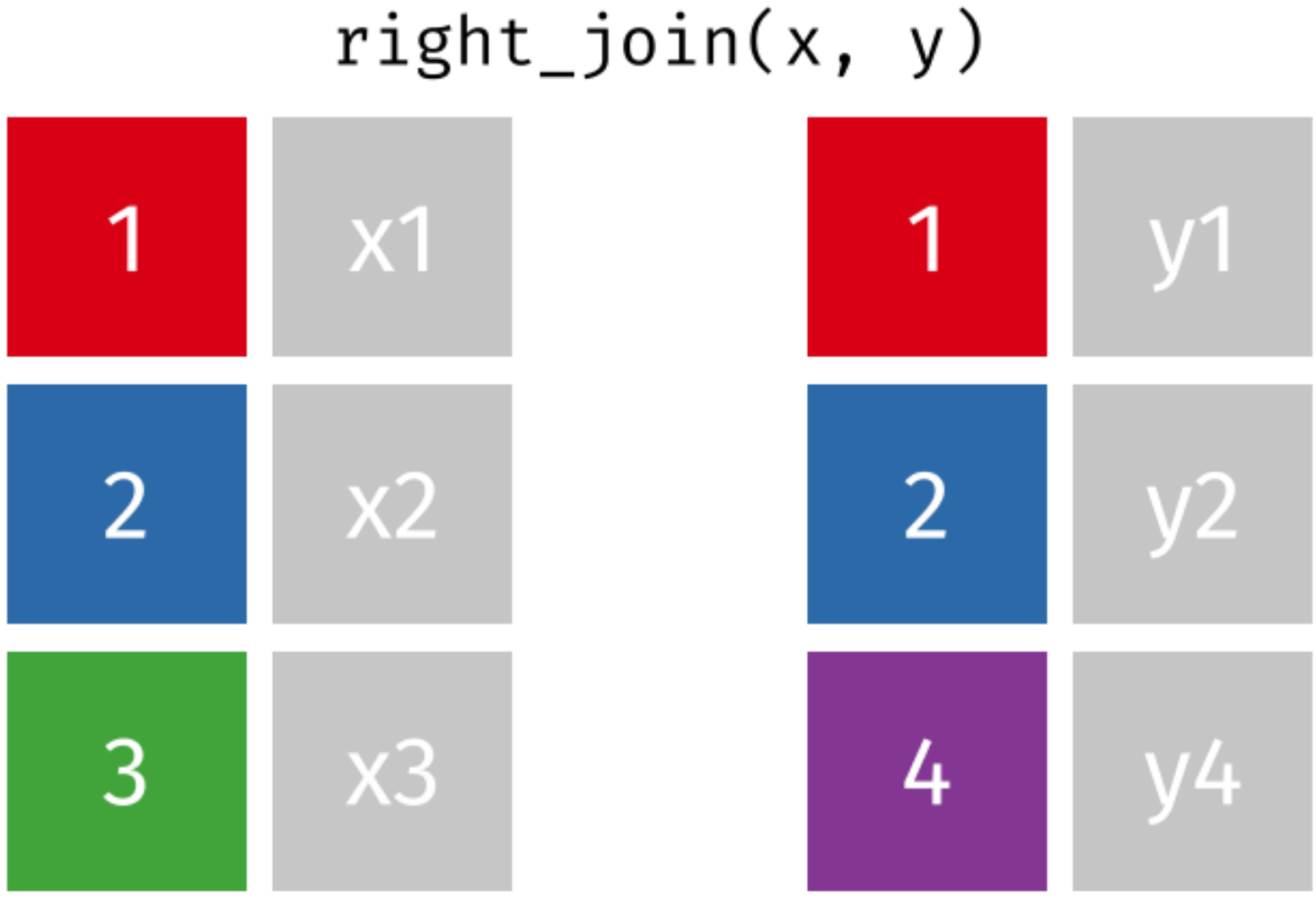
Be careful of repeated data!

```
x <- data.frame(id=c(1,2,3),
                x=c("x1", "x2", "x3"))

x

y <- data.frame(id=c(1,2,4,2),
                y=c("y1", "y2", "y4", "y5"))

y


left_join(x, y)
```

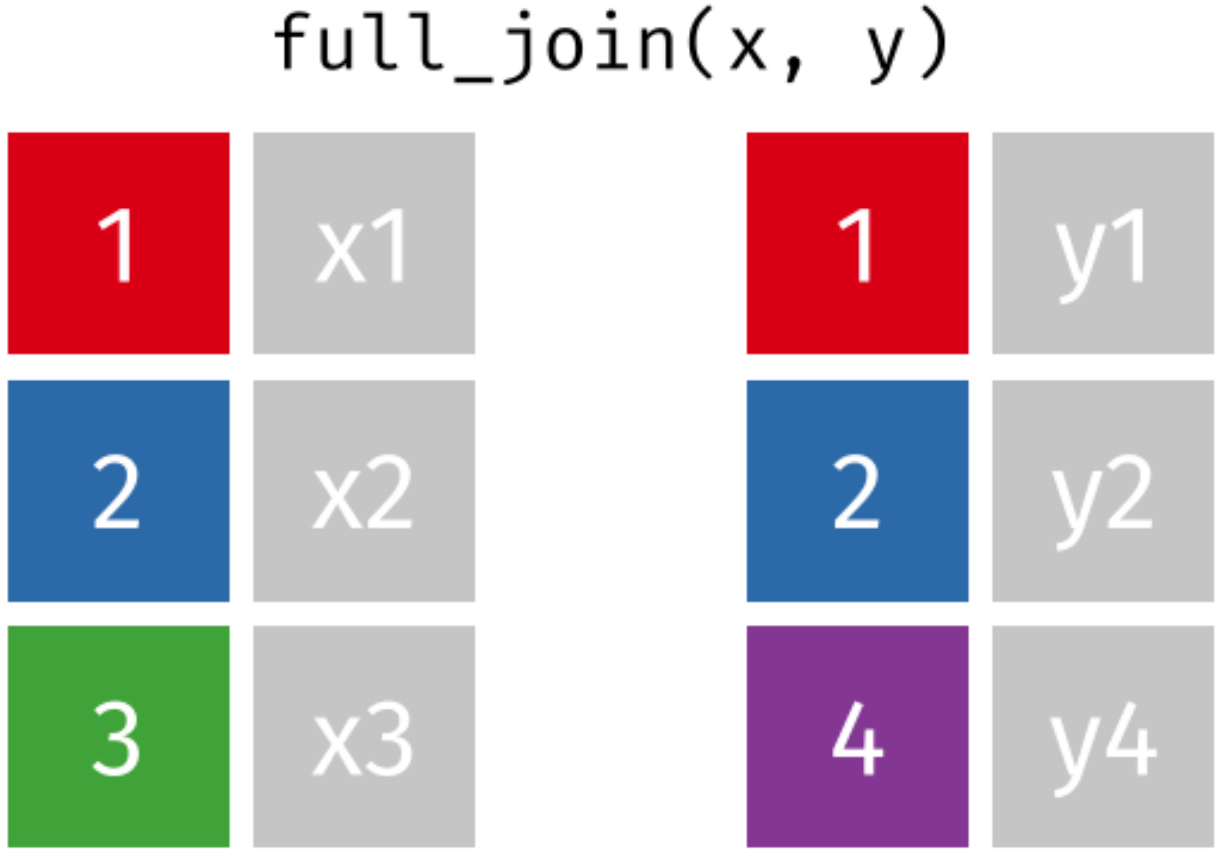
```
  id  x
1  1 x1
2  2 x2
3  3 x3

  id  y
1  1 y1
2  2 y2
3  4 y4
4  2 y5

  id  x    y
1  1 x1   y1
2  2 x2   y2
3  2 x2   y5
4  3 x3 <NA>
```


left_join(x, y)
1
x1
2
x2
3
x3
1
y1
2
y2
4
y4
2
y5

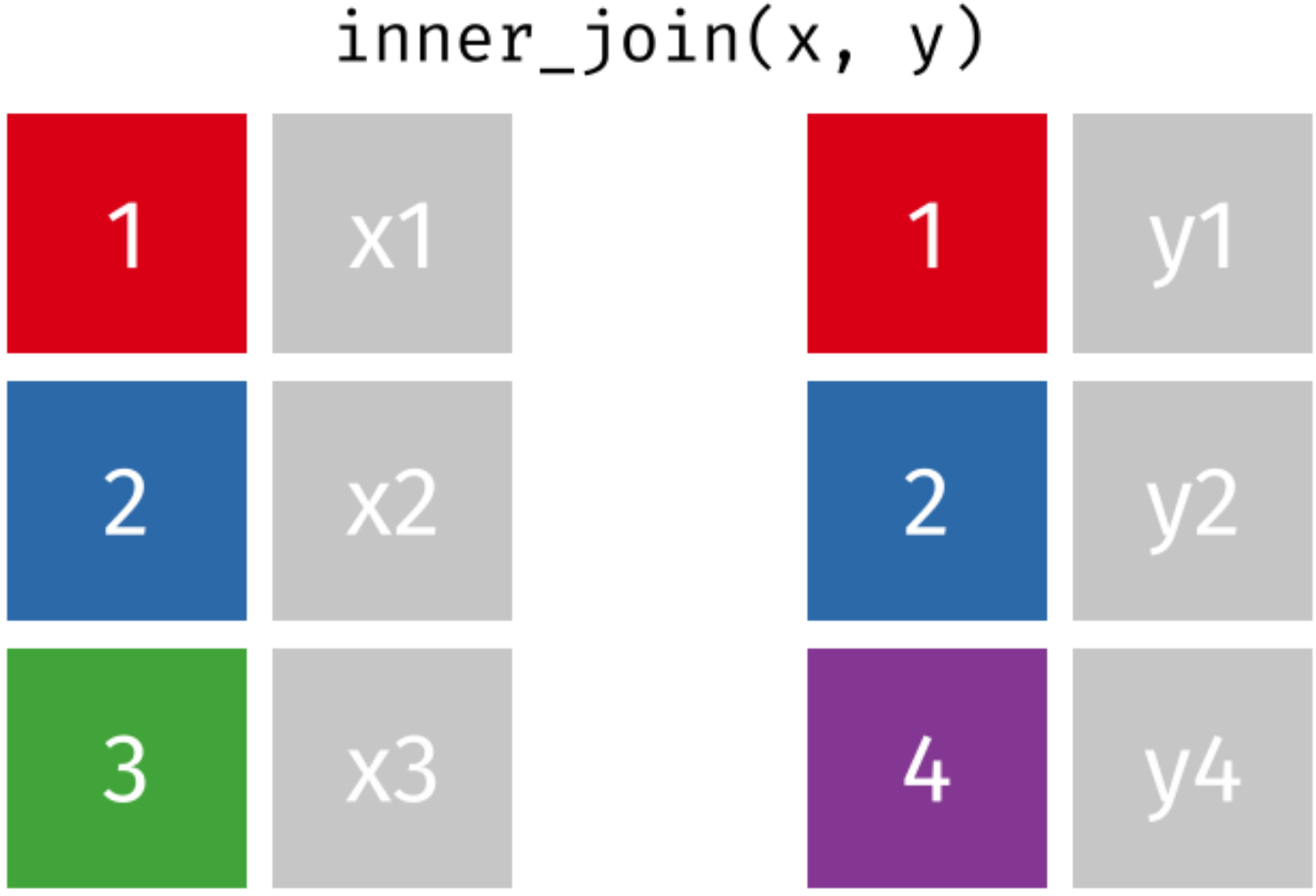
right_join()


right_join(x, y)
1
x1
2
x2
3
x3
1
y1
2
y2
4
y4

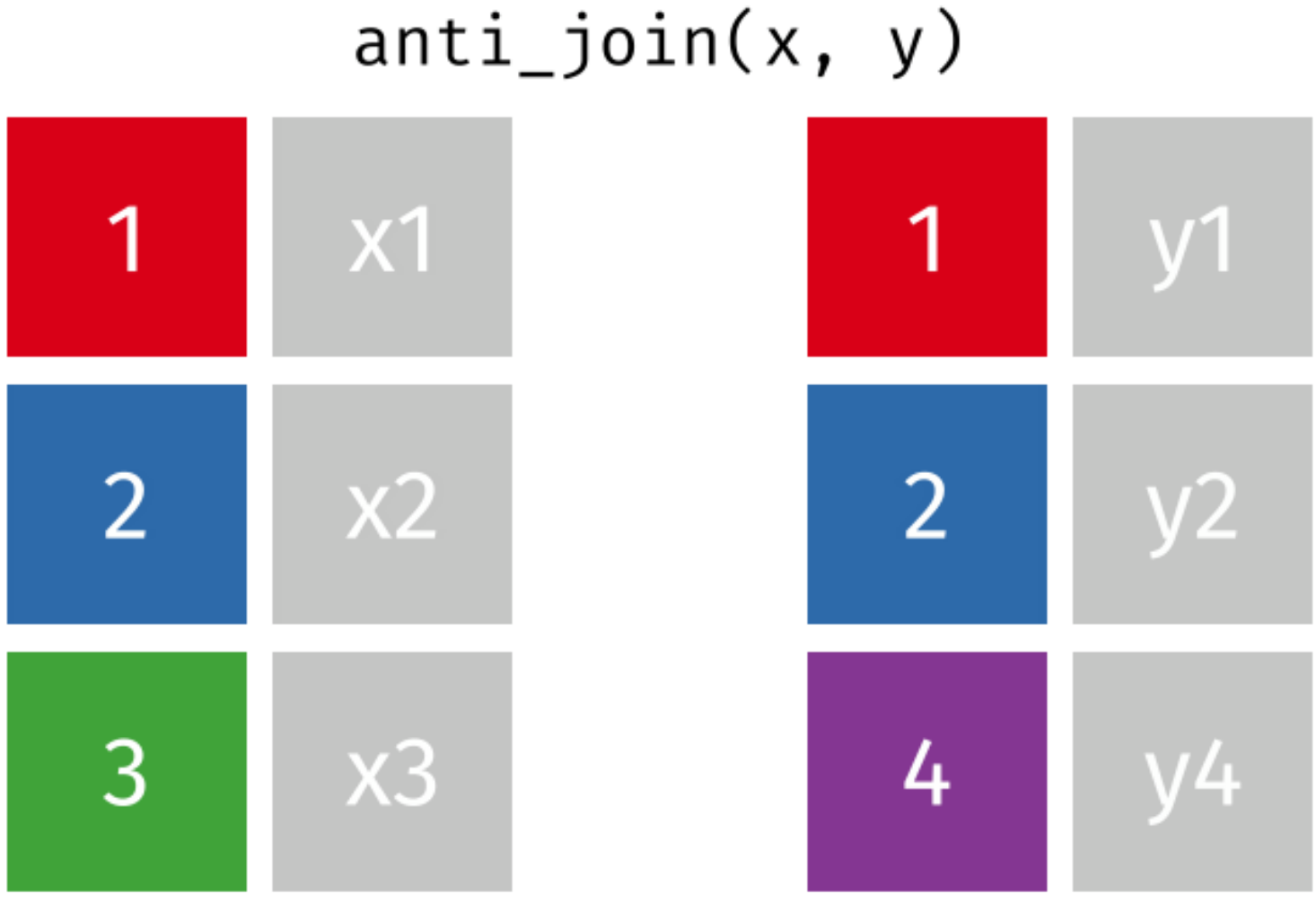
full_join()


full_join(x, y)
1
x1
2
x2
3
x3
1
y1
2
y2
4
y4

inner_join()


inner_join(x, y)
1
x1
2
x2
3
x3
1
y1
2
y2
4
y4

anti_join()


anti_join(x, y)
1
x1
2
x2
3
x3
1
y1
2
y2
4
y4

stringr package

Key `stringr` functions In this section, we will learn the following `stringr` functions:

- `str_to_upper()` `str_to_lower()` `str_to_title()`
- `str_trim()` `str_squish()`
- `str_c()`
- `str_detect()`
- `str_subset()`
- `str_sub()`

Change cases

```
test_text <- "tHiS iS A rANsOM noTE!"
```

Change cases

```
test_text <- "tHiS iS A rANsOM noTE!"

str_to_upper(test_text)
```

```
[1] "THIS IS A RANSOM NOTE!"
```

Change cases

```
test_text <- "tHiS iS A rANsOM noTE!"

str_to_upper(test_text)

str_to_lower(test_text)
```

```
[1] "THIS IS A RANSOM NOTE!"

[1] "this is a ransom note!"
```

Change cases

```
test_text <- "tHiS iS A rANsOM noTE!"

str_to_upper(test_text)

str_to_lower(test_text)

str_to_title(test_text)
```

```
[1] "THIS IS A RANSOM NOTE!"

[1] "this is a ransom note!"

[1] "This Is A Ransom Note!"
```

Trim text

```
test_text <- "  trim both   "
```

Trim text

```
test_text <- "  trim both   "

test_text
```

```
[1] "  trim both   "
```

Trim text

```
test_text <- "  trim both   "

test_text

str_trim(test_text, side="both")
```

```
[1] "  trim both   "

[1] "trim both"
```

Trim text

```
test_text <- "  trim both   "

test_text

str_trim(test_text, side="both")

str_trim(test_text, side="left")
```

```
[1] "  trim both   "

[1] "trim both"

[1] "trim both   "
```

Trim text

```
test_text <- "  trim both   "

test_text

str_trim(test_text, side="both")

str_trim(test_text, side="left")

str_trim(test_text, side="right")
```

```
[1] "  trim both   "

[1] "trim both"

[1] "trim both   "

[1] "  trim both"
```

Trim text

```
test_text <- "  trim both   "

test_text

str_trim(test_text, side="both")

str_trim(test_text, side="left")

str_trim(test_text, side="right")

messy_text <- "  sometimes  you get   this "
```

```
[1] "  trim both   "

[1] "trim both"

[1] "trim both   "

[1] "  trim both"
```

Trim text

```
test_text <- "  trim both   "

test_text

str_trim(test_text, side="both")

str_trim(test_text, side="left")

str_trim(test_text, side="right")

messy_text <- "  sometimes  you get   this "

str_squish(messy_text)
```

```
[1] "  trim both   "

[1] "trim both"

[1] "trim both   "

[1] "  trim both"

[1] "sometimes you get this"
```

Concatenate strings into one

```
text_a <- "one"
```

Concatenate strings into one

```
text_a <- "one"

text_b <- "two"
```

Concatenate strings into one

```
text_a <- "one"

text_b <- "two"

text_a
```

```
[1] "one"
```

Concatenate strings into one

```
text_a <- "one"

text_b <- "two"

text_a

text_b
```

```
[1] "one"

[1] "two"
```

Concatenate strings into one

```
text_a <- "one"

text_b <- "two"

text_a

text_b

str_c(text_a, text_b)
```

```
[1] "one"

[1] "two"

[1] "onetwo"
```

Concatenate strings into one

```
text_a <- "one"

text_b <- "two"

text_a

text_b

str_c(text_a, text_b)

str_c(text_a, text_b, sep="-")
```

```
[1] "one"

[1] "two"

[1] "onetwo"

[1] "one-two"
```

Concatenate strings into one

```
text_a <- "one"

text_b <- "two"

text_a

text_b

str_c(text_a, text_b)

str_c(text_a, text_b, sep="-")

str_c(text_a, text_b, sep=" and a ")
```

```
[1] "one"

[1] "two"

[1] "onetwo"

[1] "one-two"

[1] "one and a two"
```

Concatenate strings into one

```
text_a <- "one"
```

```
[1] "one"
```

```
text_b <- "two"
```

```
[1] "two"
```

```
text_a
```

```
[1] "onetwo"
```

```
text_b
```

```
[1] "one-two"
```

```
str_c(text_a, text_b)
```

```
[1] "one and a two"
```

```
str_c(text_a, text_b, sep="-")
```

```
[1] "one and a two"
```

```
str_c(text_a, text_b, sep=" and a ")
```

```
str_c(text_a, " and a ", text_b)
```

Concatenate strings into one

```
text_a <- "one"
```

```
[1] "one"
```

```
text_b <- "two"
```

```
[1] "two"
```

```
text_a
```

```
[1] "onetwo"
```

```
text_b
```

```
[1] "one-two"
```

```
str_c(text_a, text_b)
```

```
[1] "one and a two"
```

```
str_c(text_a, text_b, sep="-")
```

```
[1] "one and a two"
```

```
str_c(text_a, text_b, sep=" and a ")
```

```
str_c(text_a, " and a ", text_b)
```

Extract and replace substrings

```
test_text <- "Hello world"
```

Extract and replace substrings

```
test_text <- "Hello world"

test_text
```

```
[1] "Hello world"
```

Extract and replace substrings

```
test_text <- "Hello world"

test_text

str_sub(test_text, start = 6)
```

```
[1] "Hello world"

[1] " world"
```

Extract and replace substrings

```
test_text <- "Hello world"

test_text

str_sub(test_text, start = 6)

str_sub(test_text, end = 5) <- "Howdy"
```

```
[1] "Hello world"

[1] " world"
```

Extract and replace substrings

```
test_text <- "Hello world"

test_text

str_sub(test_text, start = 6)

str_sub(test_text, end = 5) <- "Howdy"

test_text
```

```
[1] "Hello world"

[1] " world"

[1] "Howdy world"
```

Extract and replace substrings

```
test_text <- "Hello world"

test_text

str_sub(test_text, start = 6)

str_sub(test_text, end = 5) <- "Howdy"

test_text

cn <- "Kemp County, Georgia"
```

```
[1] "Hello world"

[1] " world"

[1] "Howdy world"
```

Extract and replace substrings

```
test_text <- "Hello world"

test_text

str_sub(test_text, start = 6)

str_sub(test_text, end = 5) <- "Howdy"

test_text

cn <- "Kemp County, Georgia"

cn
```

```
[1] "Hello world"

[1] " world"

[1] "Howdy world"

[1] "Kemp County, Georgia"
```

Extract and replace substrings

```
test_text <- "Hello world"

test_text

str_sub(test_text, start = 6)

str_sub(test_text, end = 5) <- "Howdy"

test_text

cn <- "Kemp County, Georgia"

cn

str_replace(cn, " County, .*", "")
```

```
[1] "Hello world"

[1] " world"

[1] "Howdy world"

[1] "Kemp County, Georgia"

[1] "Kemp"
```

Extract and replace substrings

```
test_text <- "Hello world"

test_text

str_sub(test_text, start = 6)

str_sub(test_text, end = 5) <- "Howdy"

test_text

cn <- "Kemp County, Georgia"

cn

str_replace(cn, " County, .*", "")
```

```
[1] "Hello world"

[1] " world"

[1] "Howdy world"

[1] "Kemp County, Georgia"

[1] "Kemp"
```

More functions in stringr and more info on regular expressions here.

parse_number()

(from the readr package)

```
messy_numbers <- c("$5.00", "9,343,200", "6.0%
```

```
messy_numbers <- c("$5.00", "9,343,200", "6.0%

messy_numbers
```

```
[1] "$5.00"     "9,343,200" "6.0%"
```

```
messy_numbers <- c("$5.00", "9,343,200", "6.0%
```

```
messy_numbers
```

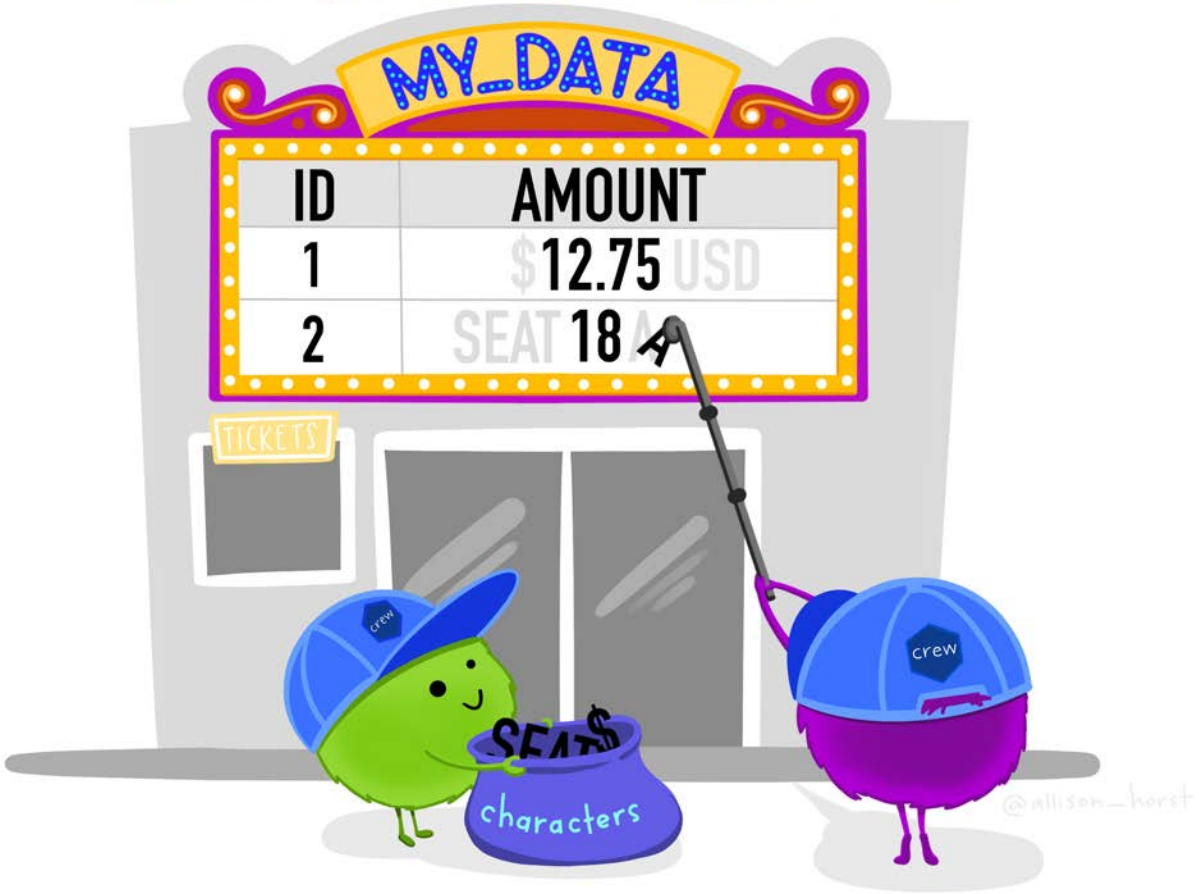
```
parse_number(messy_numbers)
```

```
[1] "$5.00"     "9,343,200" "6.0%"
```

```
[1]       5 9343200       6
```

readr::parse_number()

(just give me the numbers)



Advanced Data Journalism: Doing More with R

Class 2: Pivots and Dates

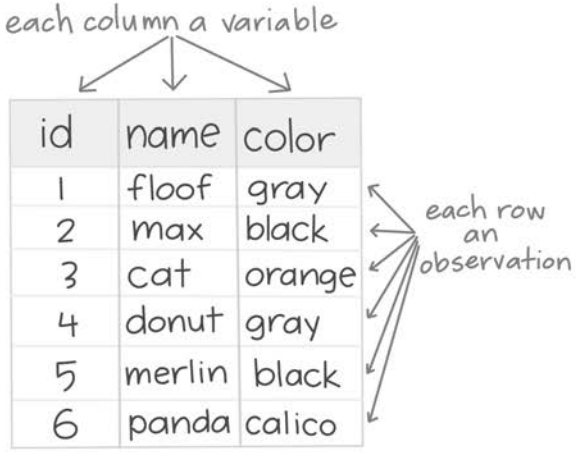
Andrew Ba Tran

"TIDY DATA is a standard way of mapping the meaning of a dataset to its structure."

–HADLEY WICKHAM

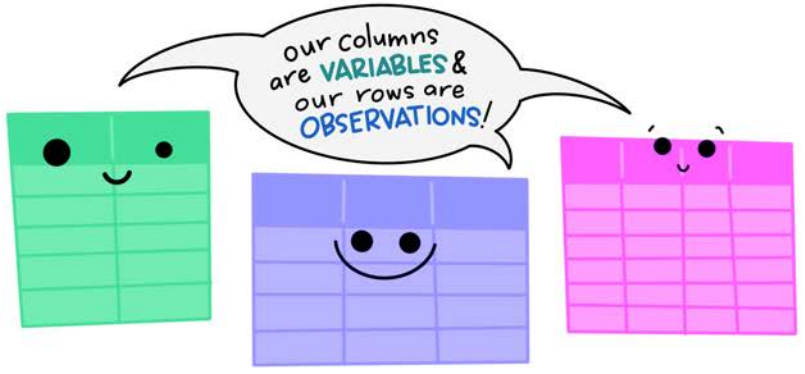
In tidy data:

- each variable forms a column
- each observation forms a row
- each cell is a single measurement



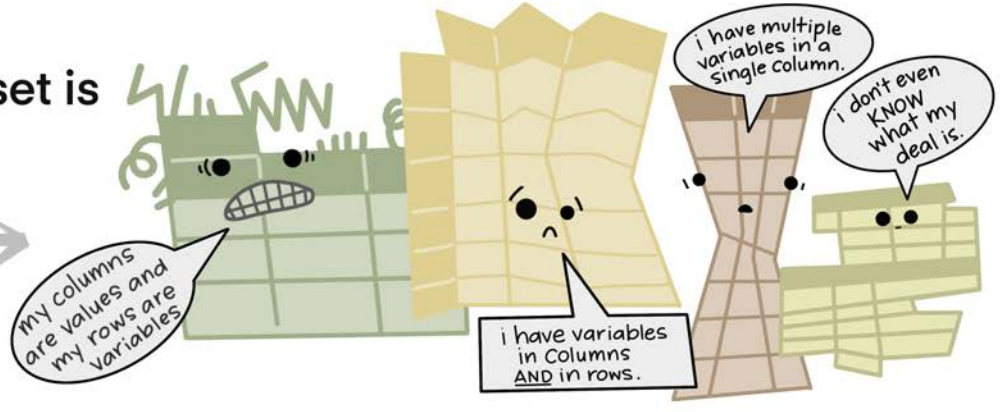
Wickham, H. (2014). Tidy Data. Journal of Statistical Software 59 (10). DOI: 10.18637/jss.v059.i10

The standard structure of tidy data means that "tidy datasets are all alike..."

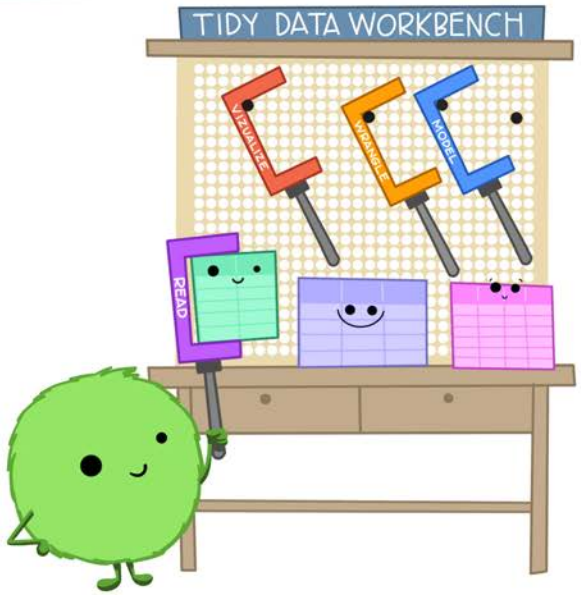


"...but every messy dataset is messy in its own way."

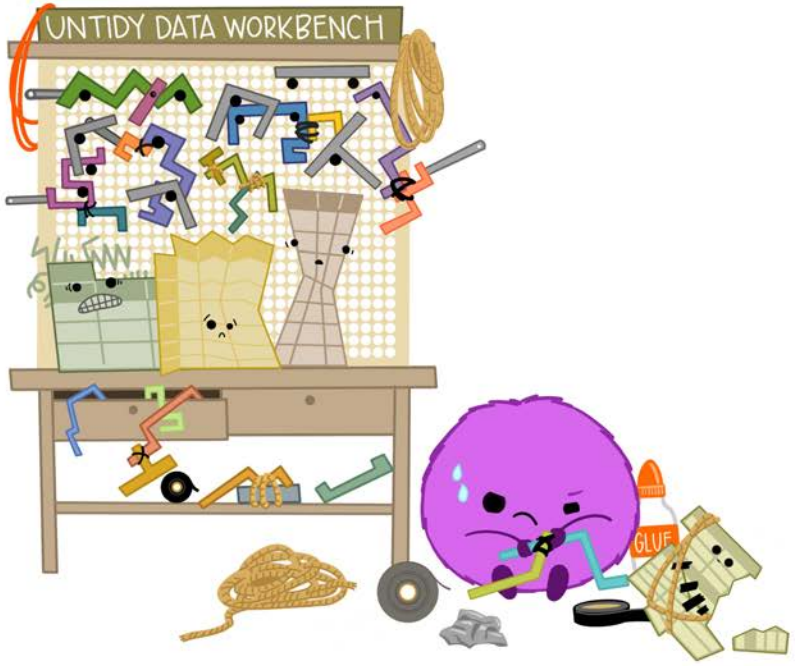
–HADLEY WICKHAM

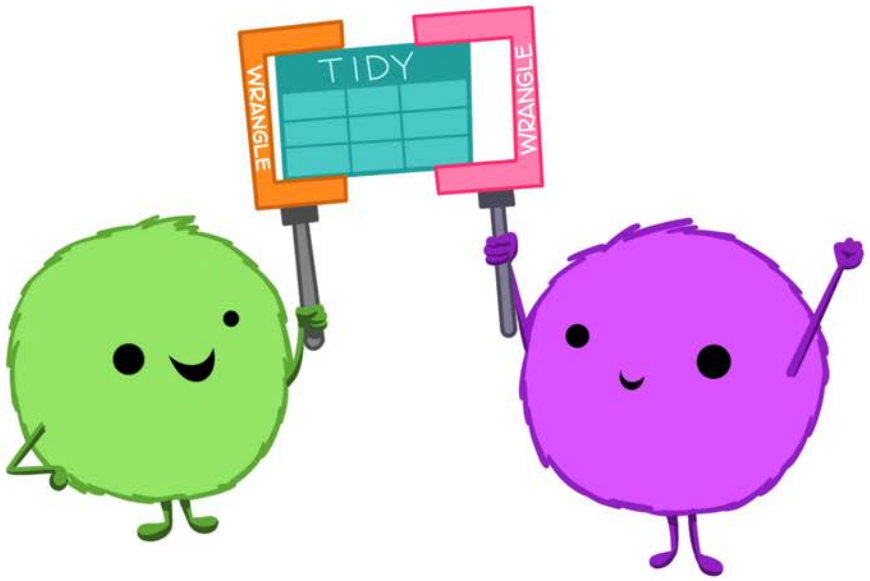


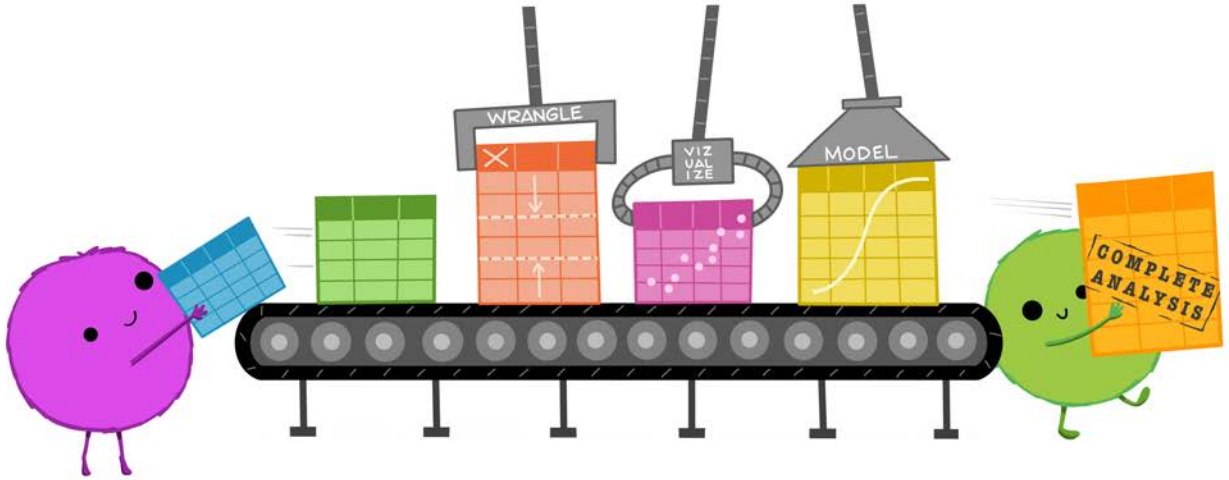
When working with tidy data, we can use the same tools in similar ways for different datasets...

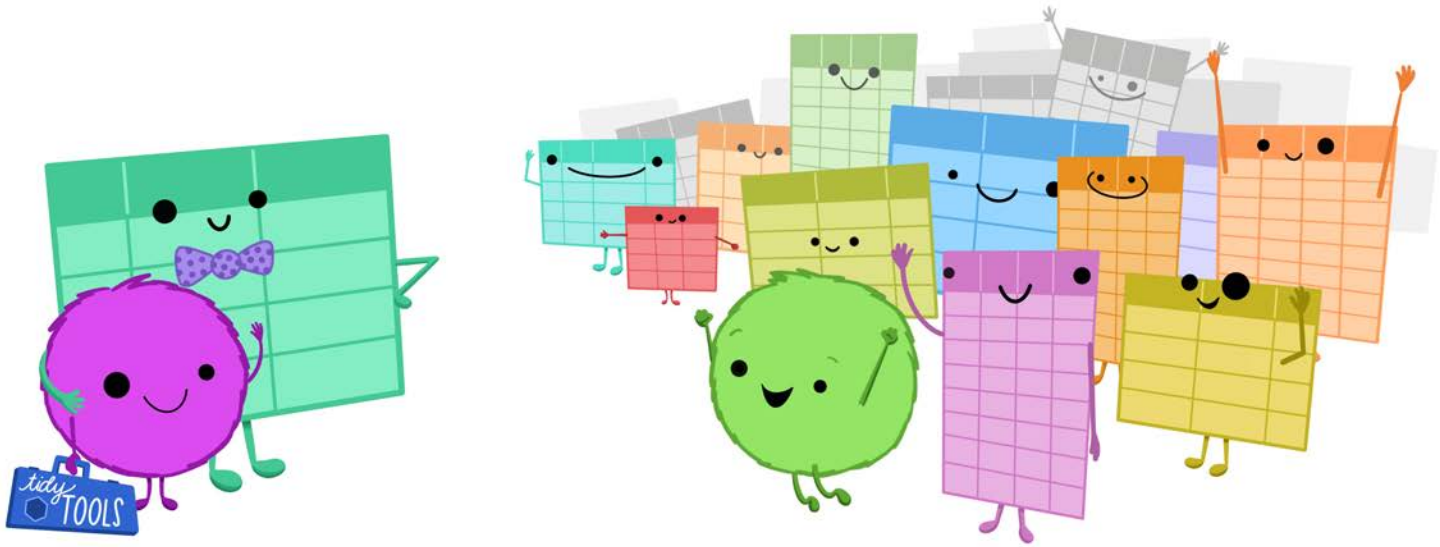


...but working with untidy data often means reinventing the wheel with one-time approaches that are hard to iterate or reuse.




TIDY
WRANGLE
WRANGLE


WRANGLE
VIZ
UAL
IZE
MODEL
COMPLETE
ANALYSIS


tidy
TOOLS


make friends with tidy data.

Let's start out with a simple data frame

2 rows x 3 columns

```
df <- data.frame(id=c(1,2), x=c("a", "b"),
                 y=c("c", "d"), z=c("e", "f"))

df
```

```
  id x y z
1  1 a c e
2  2 b d f
```

wide

id	x	y	z
1	a	c	e
2	b	d	f

long

id	key	val
1	x	a
2	x	b
1	y	c
2	y	d
1	z	e
2	z	f

pivot_longer()

```
df
```

```
  id x y z
1  1 a c e
2  2 b d f
```

```
df %>%
  pivot_longer(cols=x:z,
               names_to="key",
               values_to="val")
```

```
# A tibble: 6 × 3
     id key   val
  <dbl> <chr> <chr>
1     1 x     a
2     1 y     c
3     1 z     e
4     2 x     b
5     2 y     d
6     2 z     f
```

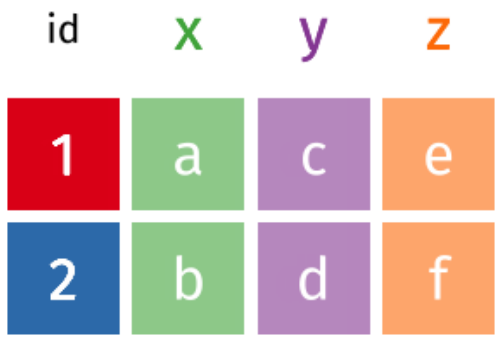
```
df %>%
  pivot_longer(cols=x:z,
               names_to="key",
               values_to="val")

df
```

```
# A tibble: 6 × 3
     id key   val
  <dbl> <chr> <chr>
1     1 x     a
2     1 y     c
3     1 z     e
4     2 x     b
5     2 y     d
6     2 z     f

  id x y z
1  1 a c e
2  2 b d f
```

```
df %>%
  pivot_longer(cols=x:z,
               names_to="key",
               values_to="val")

df %>%
  pivot_longer(cols=2:4,
               names_to="key",
               values_to="val")
```

```
# A tibble: 6 × 3
     id key   val
  <dbl> <chr> <chr>
1     1 x     a
2     1 y     c
3     1 z     e
4     2 x     b
5     2 y     d
6     2 z     f

# A tibble: 6 × 3
     id key   val
  <dbl> <chr> <chr>
1     1 x     a
2     1 y     c
3     1 z     e
4     2 x     b
5     2 y     d
6     2 z     f
```

wide



```r
df <- data.frame(state=c("TX", "NY", "FL"),
                 ducks=c(23, 39, 47),
                 fish=c(6,30,20),
                 birds=c(99,3,64))
```

```r
df <- data.frame(state=c("TX", "NY", "FL"),
                 ducks=c(23, 39, 47),
                 fish=c(6,30,20),
                 birds=c(99,3,64))

df
```

```
  state ducks fish birds
1    TX    23    6    99
2    NY    39   30     3
3    FL    47   20    64
```

```r
df <- data.frame(state=c("TX", "NY", "FL"),
                 ducks=c(23, 39, 47),
                 fish=c(6,30,20),
                 birds=c(99,3,64))

df %>%
  pivot_longer(cols=ducks:birds,
               names_to="animals",
               values_to="total")
```

```
# A tibble: 9 × 3
  state animals total
  <chr> <chr>   <dbl>
1 TX    ducks      23
2 TX    fish        6
3 TX    birds      99
4 NY    ducks      39
5 NY    fish       30
6 NY    birds       3
7 FL    ducks      47
8 FL    fish       20
9 FL    birds      64
```

```r
df <- data.frame(state=c("TX", "NY", "FL"),
                 ducks=c(23, 39, 47),
                 fish=c(6,30,20),
                 birds=c(99,3,64))

df %>%
  pivot_longer(cols=ducks:birds,
               names_to="animals",
               values_to="total")

df
```

```
# A tibble: 9 × 3
  state animals total
  <chr> <chr>   <dbl>
1 TX    ducks      23
2 TX    fish        6
3 TX    birds      99
4 NY    ducks      39
5 NY    fish       30
6 NY    birds       3
7 FL    ducks      47
8 FL    fish       20
9 FL    birds      64

  state ducks fish birds
1    TX    23    6    99
2    NY    39   30     3
3    FL    47   20    64
```

```r
df <- data.frame(state=c("TX", "NY", "FL"),
                 ducks=c(23, 39, 47),
                 fish=c(6,30,20),
                 birds=c(99,3,64))

df %>%
  pivot_longer(cols=ducks:birds,
               names_to="animals",
               values_to="total")

df %>%
  pivot_longer(cols=2:4,
               names_to="animals",
               values_to="totals")
```

```
# A tibble: 9 × 3
  state animals total
  <chr> <chr>   <dbl>
1 TX    ducks      23
2 TX    fish        6
3 TX    birds      99
4 NY    ducks      39
5 NY    fish       30
6 NY    birds       3
7 FL    ducks      47
8 FL    fish       20
9 FL    birds      64

# A tibble: 9 × 3
  state animals totals
  <chr> <chr>    <dbl>
1 TX    ducks       23
2 TX    fish         6
3 TX    birds       99
4 NY    ducks       39
5 NY    fish        30
6 NY    birds        3
7 FL    ducks       47
8 FL    fish        20
9 FL    birds       64
```

```r
df <- data.frame(state=c("TX", "NY", "FL"),
                 ducks=c(23, 39, 47),
                 fish=c(6,30,20),
                 birds=c(99,3,64))
```

```r
df <- data.frame(state=c("TX", "NY", "FL"),
                 ducks=c(23, 39, 47),
                 fish=c(6,30,20),
                 birds=c(99,3,64))

df
```

```
  state ducks fish birds
1    TX    23    6    99
2    NY    39   30     3
3    FL    47   20    64
```

```r
df <- data.frame(state=c("TX", "NY", "FL"),
                 ducks=c(23, 39, 47),
                 fish=c(6,30,20),
                 birds=c(99,3,64))

df %>%
  pivot_longer(cols=ducks:birds,
               names_to="animals",
               values_to="total")
```

```
# A tibble: 9 × 3
  state animals total
  <chr> <chr>   <dbl>
1 TX    ducks      23
2 TX    fish        6
3 TX    birds      99
4 NY    ducks      39
5 NY    fish       30
6 NY    birds       3
7 FL    ducks      47
8 FL    fish       20
9 FL    birds      64
```

```r
df <- data.frame(state=c("TX", "NY", "FL"),
                 ducks=c(23, 39, 47),
                 fish=c(6,30,20),
                 birds=c(99,3,64))

df %>%
  pivot_longer(cols=ducks:birds,
               names_to="animals",
               values_to="total") %>%
  group_by(state)
```

```
# A tibble: 9 × 3
# Groups:   state [3]
  state animals total
  <chr> <chr>   <dbl>
1 TX    ducks      23
2 TX    fish        6
3 TX    birds      99
4 NY    ducks      39
5 NY    fish       30
6 NY    birds       3
7 FL    ducks      47
8 FL    fish       20
9 FL    birds      64
```

```r
df <- data.frame(state=c("TX", "NY", "FL"),
                 ducks=c(23, 39, 47),
                 fish=c(6,30,20),
                 birds=c(99,3,64))

df %>%
  pivot_longer(cols=ducks:birds,
               names_to="animals",
               values_to="total") %>%
  group_by(state) %>%
  mutate(percent=
           round(total/sum(total)*100,1))
```

```
# A tibble: 9 × 4
# Groups:   state [3]
  state animals total percent
  <chr> <chr>   <dbl>   <dbl>
1 TX    ducks      23    18
2 TX    fish        6     4.7
3 TX    birds      99    77.3
4 NY    ducks      39    54.2
5 NY    fish       30    41.7
6 NY    birds       3     4.2
7 FL    ducks      47    35.9
8 FL    fish       20    15.3
9 FL    birds      64    48.9
```

```
df <- data.frame(state=c("TX", "NY", "FL"),
                 ducks=c(23, 39, 47),
                 fish=c(6,30,20),
                 birds=c(99,3,64))

df %>%
  pivot_longer(cols=ducks:birds,
               names_to="animals",
               values_to="total") %>%
  group_by(state) %>%
  mutate(percent=
           round(total/sum(total)*100,1))
```

```
# A tibble: 9 × 4
# Groups:   state [3]
  state animals total percent
  <chr> <chr>   <dbl>   <dbl>
1 TX    ducks      23    18
2 TX    fish        6     4.7
3 TX    birds      99    77.3
4 NY    ducks      39    54.2
5 NY    fish       30    41.7
6 NY    birds       3     4.2
7 FL    ducks      47    35.9
8 FL    fish       20    15.3
9 FL    birds      64    48.9
```

pivot_wider()

```
df_long <- df
```

```
df_long <- df %>%
  pivot_longer(cols=ducks:birds,
               names_to="animals",
               values_to="total")
```

```
df_long <- df %>%
  pivot_longer(cols=ducks:birds,
               names_to="animals",
               values_to="total") %>%
  group_by(state)
```

```
df_long <- df %>%
  pivot_longer(cols=ducks:birds,
               names_to="animals",
               values_to="total") %>%
  group_by(state) %>%
  mutate(percent=
           round(total/sum(total)*100,1))
```

```
df_long <- df %>%
  pivot_longer(cols=ducks:birds,
               names_to="animals",
               values_to="total") %>%
  group_by(state) %>%
  mutate(percent=
           round(total/sum(total)*100,1))

df_long
```

```
# A tibble: 9 × 4
# Groups:   state [3]
  state animals total percent
  <chr> <chr>   <dbl>   <dbl>
1 TX    ducks      23    18
2 TX    fish        6     4.7
3 TX    birds      99    77.3
4 NY    ducks      39    54.2
5 NY    fish       30    41.7
6 NY    birds       3     4.2
7 FL    ducks      47    35.9
8 FL    fish       20    15.3
9 FL    birds      64    48.9
```

```r
df_long <- df %>%
  pivot_longer(cols=ducks:birds,
               names_to="animals",
               values_to="total") %>%
  group_by(state) %>%
  mutate(percent=
           round(total/sum(total)*100,1))

df_long %>%
  pivot_wider(names_from="animals",
              values_from="percent")
```

```
# A tibble: 9 × 5
# Groups:   state [3]
  state total ducks  fish birds
  <chr> <dbl> <dbl> <dbl> <dbl>
1 TX       23  18    NA    NA
2 TX        6  NA     4.7  NA
3 TX       99  NA    NA    77.3
4 NY       39  54.2  NA    NA
5 NY       30  NA    41.7  NA
6 NY        3  NA    NA     4.2
7 FL       47  35.9  NA    NA
8 FL       20  NA    15.3  NA
9 FL       64  NA    NA    48.9
```

```r
df_long <- df %>%
  pivot_longer(cols=ducks:birds,
               names_to="animals",
               values_to="total") %>%
  group_by(state) %>%
  mutate(percent=
           round(total/sum(total)*100,1))

df_long %>%
  pivot_wider(names_from="animals",
              values_from="percent")

df_long
```

```
# A tibble: 9 × 5
# Groups:   state [3]
  state total ducks  fish birds
  <chr> <dbl> <dbl> <dbl> <dbl>
1 TX       23  18    NA    NA
2 TX        6  NA     4.7  NA
3 TX       99  NA    NA    77.3
4 NY       39  54.2  NA    NA
5 NY       30  NA    41.7  NA
6 NY        3  NA    NA     4.2
7 FL       47  35.9  NA    NA
8 FL       20  NA    15.3  NA
9 FL       64  NA    NA    48.9
```

```
# A tibble: 9 × 4
# Groups:   state [3]
  state animals total percent
  <chr> <chr>   <dbl>   <dbl>
1 TX    ducks      23    18
2 TX    fish        6     4.7
3 TX    birds      99    77.3
4 NY    ducks      39    54.2
5 NY    fish       30    41.7
6 NY    birds       3     4.2
7 FL    ducks      47    35.9
8 FL    fish       20    15.3
9 FL    birds      64    48.9
```

```r
df_long <- df %>%
  pivot_longer(cols=ducks:birds,
               names_to="animals",
               values_to="total") %>%
  group_by(state) %>%
  mutate(percent=
           round(total/sum(total)*100,1))

df_long %>%
  pivot_wider(names_from="animals",
              values_from="percent")

df_long %>%
  select(-total)
```

```
# A tibble: 9 × 5
# Groups:   state [3]
  state total ducks  fish birds
  <chr> <dbl> <dbl> <dbl> <dbl>
1 TX       23  18    NA    NA
2 TX        6  NA     4.7  NA
3 TX       99  NA    NA    77.3
4 NY       39  54.2  NA    NA
5 NY       30  NA    41.7  NA
6 NY        3  NA    NA     4.2
7 FL       47  35.9  NA    NA
8 FL       20  NA    15.3  NA
9 FL       64  NA    NA    48.9

# A tibble: 9 × 3
# Groups:   state [3]
  state animals percent
  <chr> <chr>     <dbl>
1 TX    ducks      18
2 TX    fish        4.7
3 TX    birds      77.3
4 NY    ducks      54.2
5 NY    fish       41.7
6 NY    birds       4.2
7 FL    ducks      35.9
8 FL    fish       15.3
9 FL    birds      48.9
```

```
df_long <- df %>%
  pivot_longer(cols=ducks:birds,
               names_to="animals",
               values_to="total") %>%
  group_by(state) %>%
  mutate(percent=
           round(total/sum(total)*100,1))

df_long %>%
  pivot_wider(names_from="animals",
              values_from="percent")

df_long %>%
  select(-total) %>%
  pivot_wider(names_from="animals",
              values_from="percent")
```

```
# A tibble: 9 × 5
# Groups:   state [3]
  state total ducks  fish birds
  <chr> <dbl> <dbl> <dbl> <dbl>
1 TX       23  18    NA    NA
2 TX        6  NA     4.7  NA
3 TX       99  NA    NA    77.3
4 NY       39  54.2  NA    NA
5 NY       30  NA    41.7  NA
6 NY        3  NA    NA     4.2
7 FL       47  35.9  NA    NA
8 FL       20  NA    15.3  NA
9 FL       64  NA    NA    48.9

# A tibble: 3 × 4
# Groups:   state [3]
  state ducks  fish birds
  <chr> <dbl> <dbl> <dbl>
1 TX     18     4.7  77.3
2 NY     54.2  41.7   4.2
3 FL     35.9  15.3  48.9
```

```r
df_long <- df %>%
  pivot_longer(cols=ducks:birds,
               names_to="animals",
               values_to="total") %>%
  group_by(state) %>%
  mutate(percent=
           round(total/sum(total)*100,1))

df_long %>%
  pivot_wider(names_from="animals",
              values_from="percent")

df_long %>%
  select(-total) %>%
  pivot_wider(names_from="animals",
              values_from="percent") %>%
  mutate(birds_fish_diff=
           birds-fish)
```

```
# A tibble: 9 × 5
# Groups:   state [3]
  state total ducks  fish birds
  <chr> <dbl> <dbl> <dbl> <dbl>
1 TX       23  18    NA    NA
2 TX        6  NA     4.7  NA
3 TX       99  NA    NA    77.3
4 NY       39  54.2  NA    NA
5 NY       30  NA    41.7  NA
6 NY        3  NA    NA     4.2
7 FL       47  35.9  NA    NA
8 FL       20  NA    15.3  NA
9 FL       64  NA    NA    48.9

# A tibble: 3 × 5
# Groups:   state [3]
  state ducks  fish birds birds_fish_diff
  <chr> <dbl> <dbl> <dbl>           <dbl>
1 TX     18     4.7  77.3            72.6
2 NY     54.2  41.7   4.2           -37.5
3 FL     35.9  15.3  48.9            33.6
```

```
df_long <- df %>%
  pivot_longer(cols=ducks:birds,
               names_to="animals",
               values_to="total") %>%
  group_by(state) %>%
  mutate(percent=
           round(total/sum(total)*100,1))

df_long %>%
  pivot_wider(names_from="animals",
              values_from="percent")

df_long %>%
  select(-total) %>%
  pivot_wider(names_from="animals",
              values_from="percent") %>%
  mutate(birds_fish_diff=
           birds-fish)
```

```
# A tibble: 9 × 5
# Groups:   state [3]
  state total ducks  fish birds
  <chr> <dbl> <dbl> <dbl> <dbl>
1 TX       23  18    NA    NA
2 TX        6  NA     4.7  NA
3 TX       99  NA    NA    77.3
4 NY       39  54.2  NA    NA
5 NY       30  NA    41.7  NA
6 NY        3  NA    NA     4.2
7 FL       47  35.9  NA    NA
8 FL       20  NA    15.3  NA
9 FL       64  NA    NA    48.9

# A tibble: 3 × 5
# Groups:   state [3]
  state ducks  fish birds birds_fish_diff
  <chr> <dbl> <dbl> <dbl>           <dbl>
1 TX     18     4.7  77.3            72.6
2 NY     54.2  41.7   4.2           -37.5
3 FL     35.9  15.3  48.9            33.6
```

```
df_long <- df
```

```
df_long <- df %>%
  pivot_longer(cols=ducks:birds,
               names_to="animals",
               values_to="total")
```

```
df_long <- df %>%
  pivot_longer(cols=ducks:birds,
               names_to="animals",
               values_to="total") %>%
  group_by(state)
```

```
df_long <- df %>%
  pivot_longer(cols=ducks:birds,
               names_to="animals",
               values_to="total") %>%
  group_by(state) %>%
  mutate(percent=
           round(total/sum(total)*100,1))
```

```
df_long <- df %>%
  pivot_longer(cols=ducks:birds,
               names_to="animals",
               values_to="total") %>%
  group_by(state) %>%
  mutate(percent=
           round(total/sum(total)*100,1))

df_long
```

```
# A tibble: 9 × 4
# Groups:   state [3]
  state animals total percent
  <chr> <chr>   <dbl>   <dbl>
1 TX    ducks      23    18
2 TX    fish        6     4.7
3 TX    birds      99    77.3
4 NY    ducks      39    54.2
5 NY    fish       30    41.7
6 NY    birds       3     4.2
7 FL    ducks      47    35.9
8 FL    fish       20    15.3
9 FL    birds      64    48.9
```

```r
df_long <- df %>%
  pivot_longer(cols=ducks:birds,
               names_to="animals",
               values_to="total") %>%
  group_by(state) %>%
  mutate(percent=
           round(total/sum(total)*100,1))

df_long %>%
  pivot_wider(names_from="animals",
              values_from=c("total", "percent"
```

```
# A tibble: 3 × 7
# Groups:   state [3]
  state total_ducks total_fish total_birds percent_ducks percent_fish percent_…¹
  <chr>       <dbl>      <dbl>       <dbl>          <dbl>        <dbl>       <dbl>
1 TX             23          6          99           18            4.7        77.3
2 NY             39         30           3           54.2         41.7         4.2
3 FL             47         20          64           35.9         15.3        48.9
# … with abbreviated variable name ¹percent_birds
```

```
df_long <- df %>%
  pivot_longer(cols=ducks:birds,
               names_to="animals",
               values_to="total") %>%
  group_by(state) %>%
  mutate(percent=
           round(total/sum(total)*100,1))

df_long %>%
  pivot_wider(names_from="animals",
              values_from=c("total", "percent"
```

```
# A tibble: 3 × 7
# Groups:   state [3]
  state total_ducks total_fish total_birds percent_ducks percent_fish percent_…¹
  <chr>       <dbl>      <dbl>       <dbl>          <dbl>        <dbl>      <dbl>
1 TX             23          6          99           18            4.7       77.3
2 NY             39         30           3           54.2         41.7        4.2
3 FL             47         20          64           35.9         15.3       48.9
# … with abbreviated variable name ¹percent_birds
```

lubridate package


lubridate:
TIME CONTROL!
FEED:
FLUX
CAPACITOR
Studio
cheat sheet
LUBRDAT
Horst '18

```
library(lubridate)
```

```
library(lubridate)

data <- data.frame(First=c("Charlie", "Lucy",
                   Last=c("Brown", "van Pelt",
                   birthday=c("10-31-06", "2/4
```

```r
library(lubridate)

data <- data.frame(First=c("Charlie", "Lucy",
                   Last=c("Brown", "van Pelt",
                   birthday=c("10-31-06", "2/4

data
```

```
        First     Last     birthday
1    Charlie    Brown     10-31-06
2       Lucy van Pelt     2/4/2007
3 Peppermint    Patty June 1, 2005
```

```r
library(lubridate)

data <- data.frame(First=c("Charlie", "Lucy",
                   Last=c("Brown", "van Pelt",
                   birthday=c("10-31-06", "2/4

data %>%
  mutate(birthday_clean=mdy(birthday))
```

```
        First     Last     birthday birthday_clean
1    Charlie    Brown     10-31-06     2006-10-31
2       Lucy van Pelt     2/4/2007     2007-02-04
3 Peppermint    Patty June 1, 2005     2005-06-01
```

Reading dates

Order of elements in date-time	Parse function
year, month, day	`ymd()`
year, day, month	`ydm()`
month, day, year	`mdy()`
day, month, year	`dmy()`
hour, minute	`hm()`
hour, minute, second	`hms()`
year, month, day, hour, minute, second	`ymd_hms()`

Accessing date parts

Date component	Function
Year	`year()`
Month	`month()`
Week	`week()`
Day of year	`yday()`
Day of month	`mday()`
Day of week	`wday()`
Hour	`hour()`
Minute	`minute()`
Second	`ymd_hms()`
Time zone	`ymd_hms()`

```
data <- data.frame(First=c("Charlie", "Lucy",
                   Last=c("Brown", "van Pelt",
                   birthday=c("10-31-06", "2/4
```

```r
data <- data.frame(First=c("Charlie", "Lucy",
                   Last=c("Brown", "van Pelt",
                   birthday=c("10-31-06", "2/4

data
```

```
       First     Last     birthday
1    Charlie    Brown     10-31-06
2       Lucy van Pelt     2/4/2007
3 Peppermint    Patty June 1, 2005
```

```
data <- data.frame(First=c("Charlie", "Lucy",
                   Last=c("Brown", "van Pelt",
                   birthday=c("10-31-06", "2/4

data %>%
  mutate(birthday_clean=mdy(birthday))
```

```
        First     Last     birthday birthday_clean
1    Charlie    Brown     10-31-06     2006-10-31
2       Lucy van Pelt     2/4/2007     2007-02-04
3 Peppermint    Patty June 1, 2005     2005-06-01
```

```
data <- data.frame(First=c("Charlie", "Lucy",
                   Last=c("Brown", "van Pelt",
                   birthday=c("10-31-06", "2/4

data %>%
  mutate(birthday_clean=mdy(birthday)) %>%
  mutate(month=month(birthday_clean))
```

```
       First     Last     birthday birthday_clean month
1    Charlie    Brown     10-31-06     2006-10-31    10
2       Lucy van Pelt     2/4/2007     2007-02-04     2
3 Peppermint    Patty June 1, 2005     2005-06-01     6
```

```r
data <- data.frame(First=c("Charlie", "Lucy",
                   Last=c("Brown", "van Pelt",
                   birthday=c("10-31-06", "2/4

data %>%
  mutate(birthday_clean=mdy(birthday)) %>%
  mutate(month=month(birthday_clean)) %>%
  mutate(year=year(birthday_clean))
```

```
       First     Last     birthday birthday_clean month year
1    Charlie    Brown     10-31-06     2006-10-31    10 2006
2       Lucy van Pelt     2/4/2007     2007-02-04     2 2007
3 Peppermint    Patty June 1, 2005     2005-06-01     6 2005
```

```
data <- data.frame(First=c("Charlie", "Lucy",
                   Last=c("Brown", "van Pelt",
                   birthday=c("10-31-06", "2/4

data %>%
  mutate(birthday_clean=mdy(birthday)) %>%
  mutate(month=month(birthday_clean)) %>%
  mutate(year=year(birthday_clean)) %>%
  mutate(week=week(birthday_clean))
```

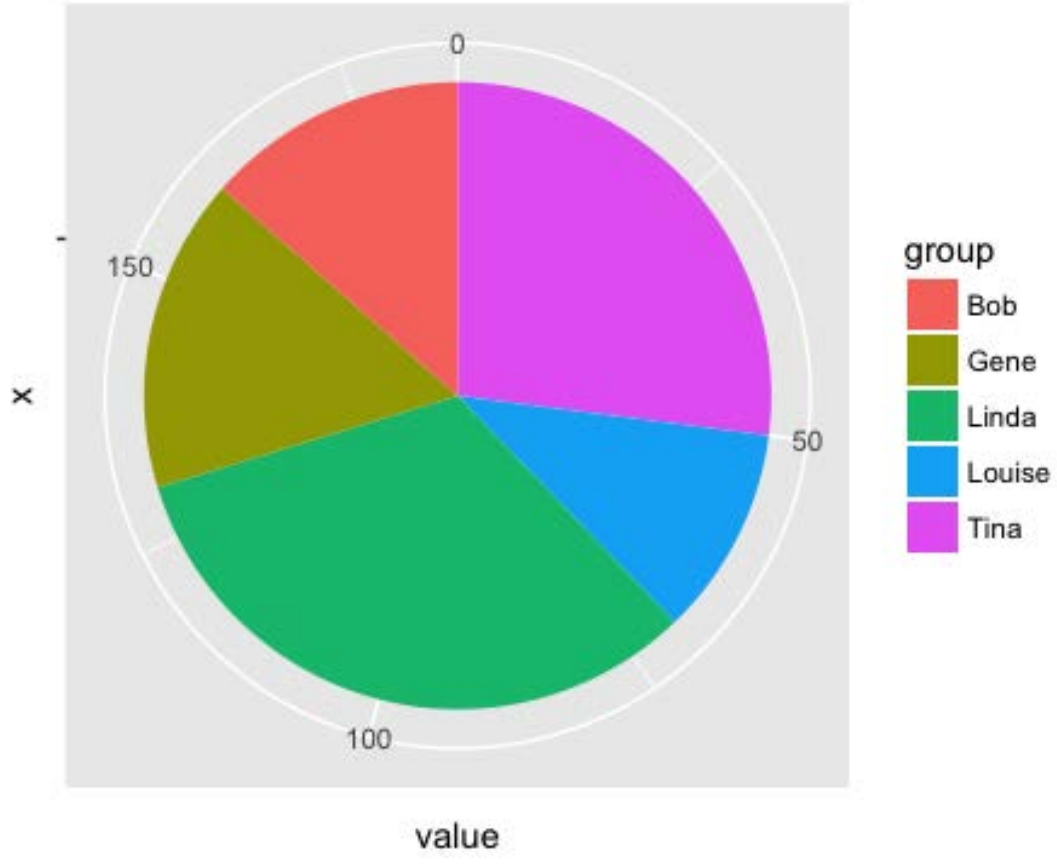
```
       First     Last     birthday birthday_clean month year week
1    Charlie    Brown     10-31-06     2006-10-31    10 2006   44
2       Lucy van Pelt     2/4/2007     2007-02-04     2 2007    5
3 Peppermint    Patty June 1, 2005     2005-06-01     6 2005   22
```


LUBRIDATE: wrangle times + dates!
YEARS
MONTHS
DAYS
Horst '18

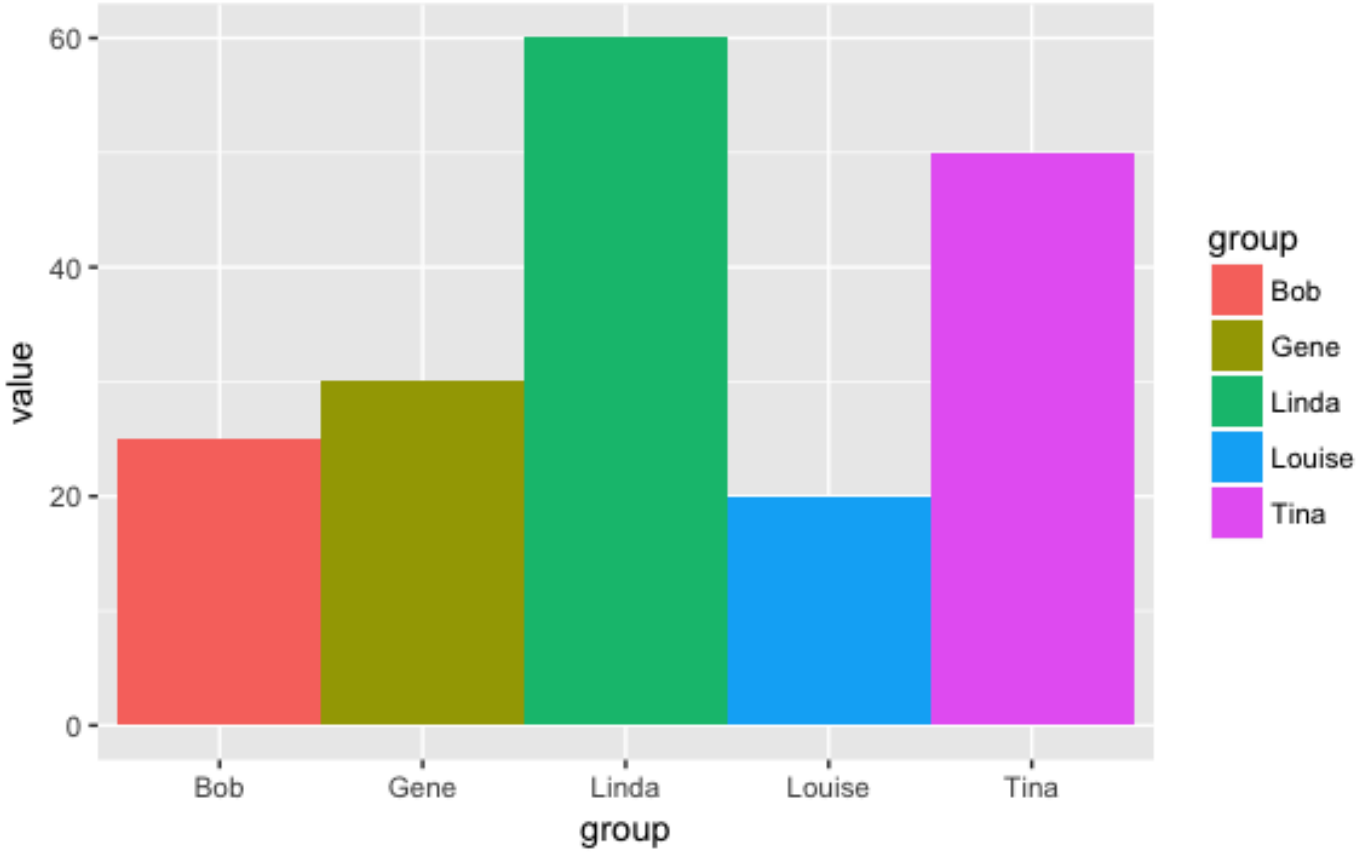
Advanced Data Journalism: Doing More with R

Class 2: Visualizations

Andrew Ba Tran

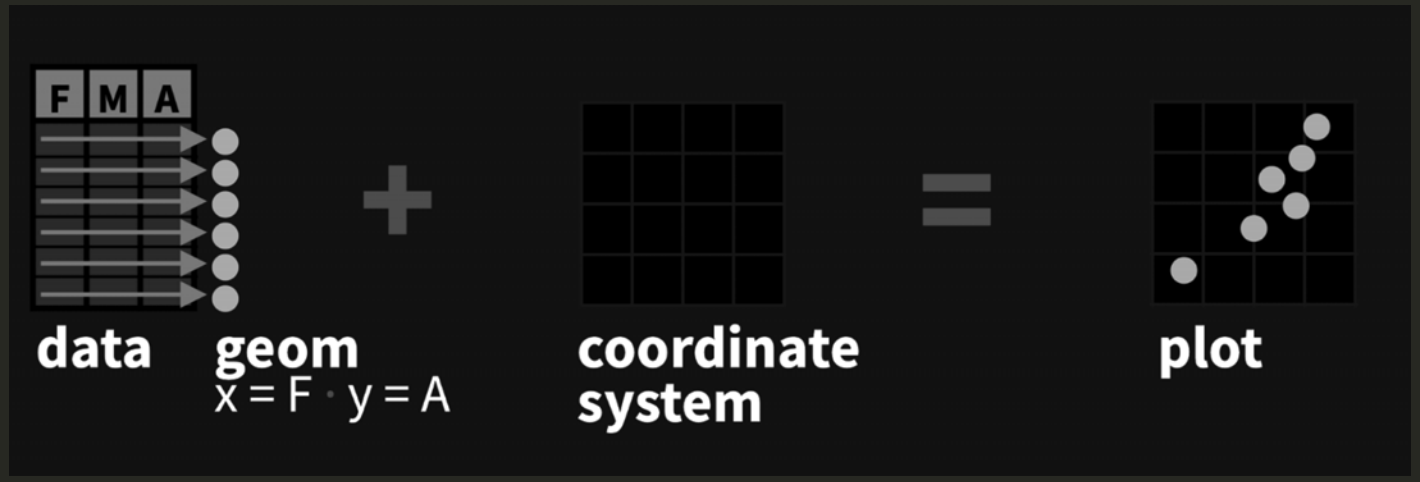

0
50
100
150
x
value
group
Bob
Gene
Linda
Louise
Tina


150
100
50
0
value
x
group
Bob
Gene
Linda
Louise
Tina

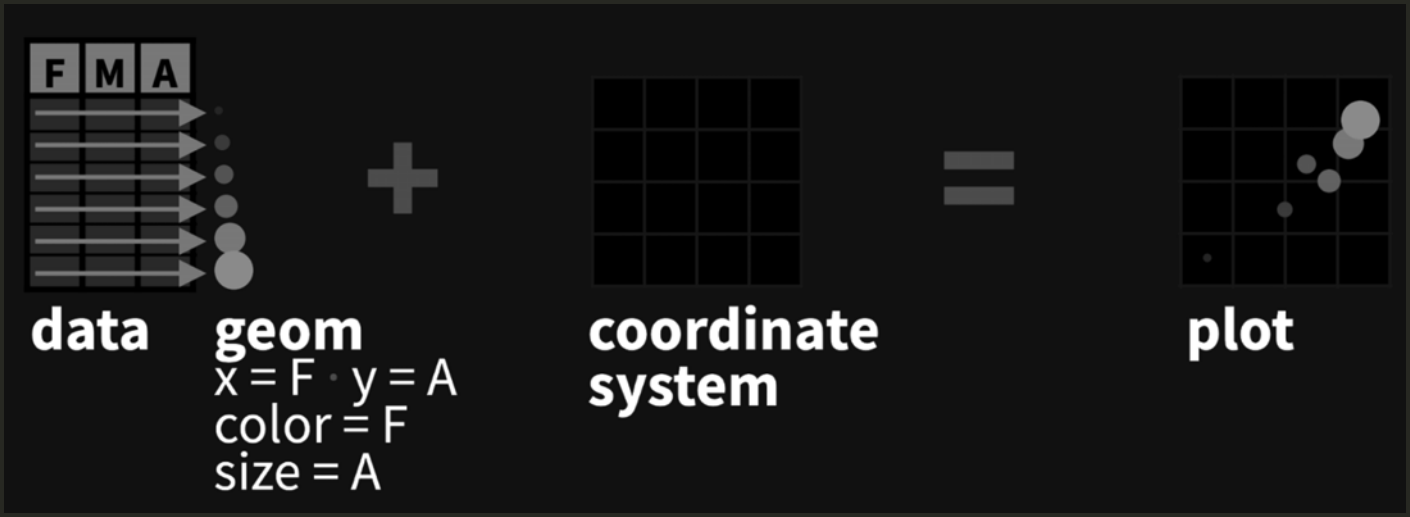

60
40
20
0
value
Bob
Gene
Linda
Louise
Tina
group
group
Bob
Gene
Linda
Louise
Tina


ggplot2:
VISUAL DATA EXPLORATION
Horst '18

ggplot2 is based on the grammar of graphics, the idea that you can build every graph from the same components: a data set, a coordinate system, and geoms—visual marks that represent data points



To display values, map variables in the data to visual properties of the geom (aesthetics) like size, color, and x and y locations.

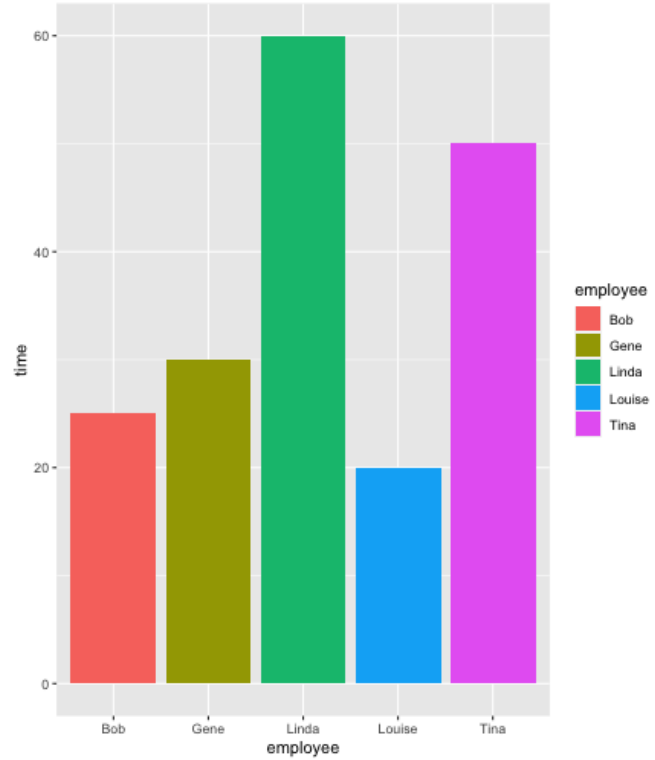


You only need two lines of code, really.

The rest is just extra customization.

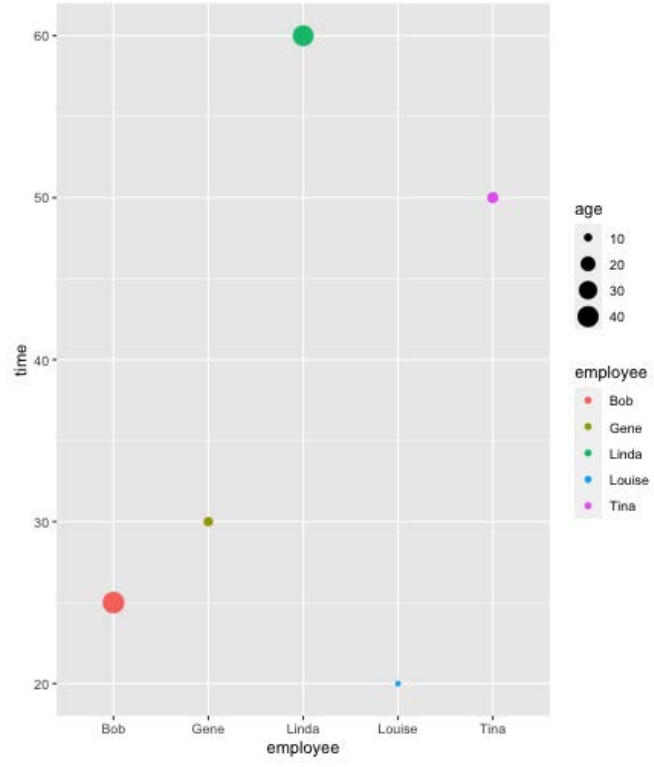
```
burgers

ggplot(burgers, aes(x=employee, y=time, fill=employee)) +
  geom_col(position="stack")
```

```
  employee time age interest where
1      Bob   25  42  cooking front
2     Gene   30  11    music front
3    Linda   60  39     wine front
4   Louise   20   9    chaos front
5     Tina   50  13   horses front
```

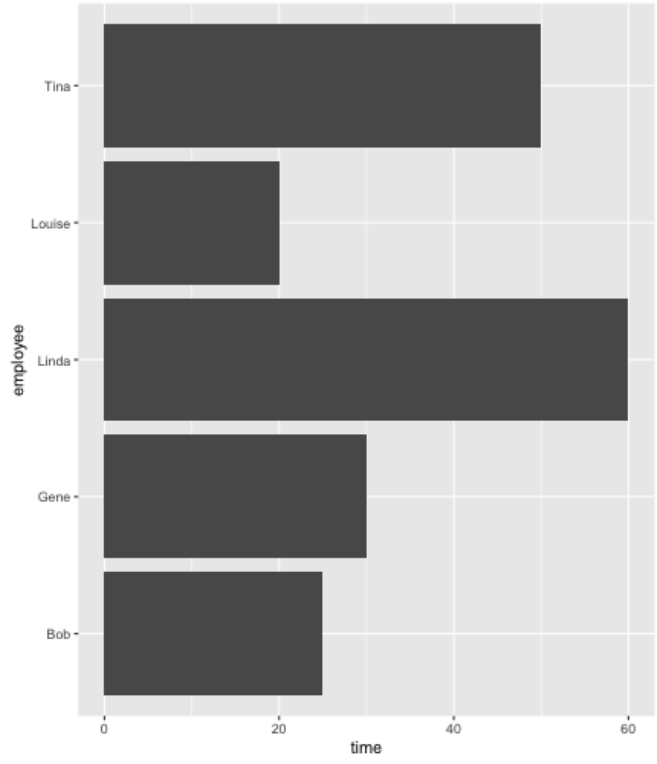


Layers

```
  ggplot(burgers) +
    geom_point(aes(x=employee, y=time, fill=employee, color=emp
```



```
ggplot(burgers) +
  geom_col(aes(x=time, y=employee), stat="identity")
```



```
disney <- read_csv("data/disney_movies_total_gross.csv")

glimpse(disney)
```

```
Rows: 579
Columns: 6
$ movie_title              <chr> "Snow White and the Seven Dwarfs", "Pinocchio…
$ release_date             <chr> "Dec 21, 1937", "Feb 9, 1940", "Nov 13, 1940"…
$ genre                    <chr> "Musical", "Adventure", "Musical", "Adventure…
$ MPAA_rating              <chr> "G", "G", "G", "G", "G", NA, "G", NA, "G", NA…
$ total_gross              <chr> "$184,925,485", "$84,300,000", "$83,320,000",…
$ inflation_adjusted_gross <chr> "$5,228,953,251", "$2,188,229,052", "$2,187,0…
```

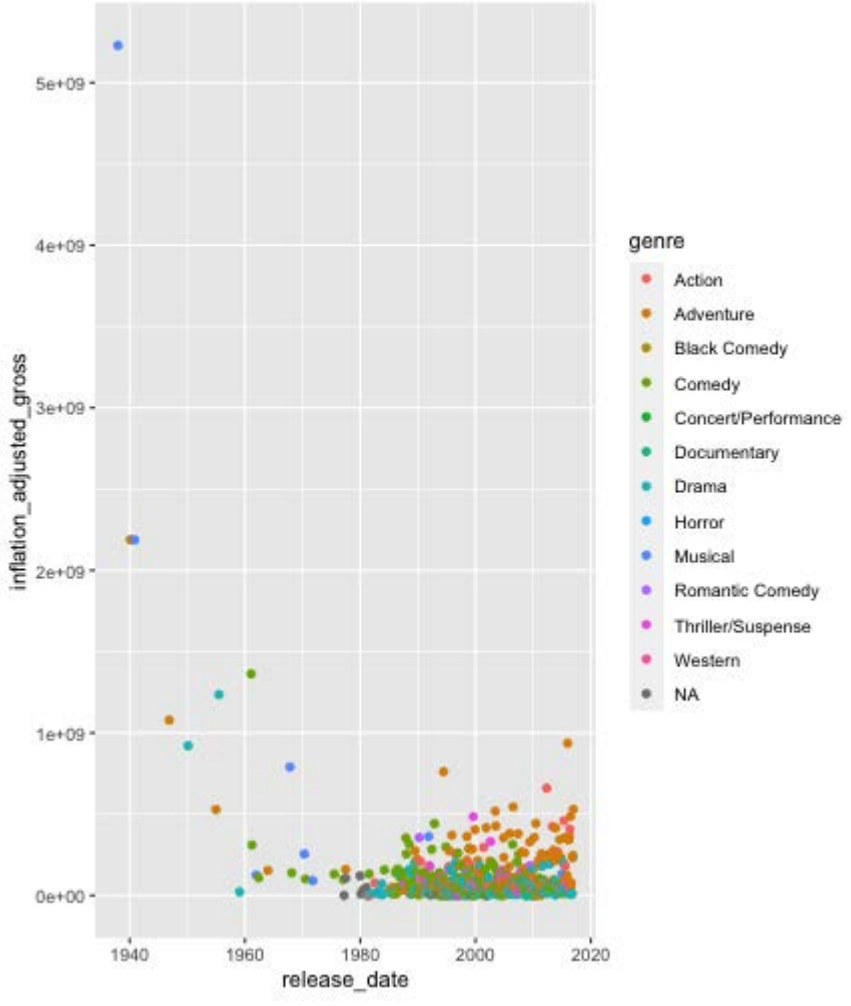
```
disney <- read_csv("data/disney_movies_total_gross.csv")

glimpse(disney)
```

```
Rows: 579
Columns: 6
$ movie_title              <chr> "Snow White and the Seven Dwarfs", "Pinocchio…
$ release_date             <chr> "Dec 21, 1937", "Feb 9, 1940", "Nov 13, 1940"…
$ genre                    <chr> "Musical", "Adventure", "Musical", "Adventure…
$ MPAA_rating              <chr> "G", "G", "G", "G", "G", NA, "G", NA, "G", NA…
$ total_gross              <chr> "$184,925,485", "$84,300,000", "$83,320,000",…
$ inflation_adjusted_gross <chr> "$5,228,953,251", "$2,188,229,052", "$2,187,0…
```

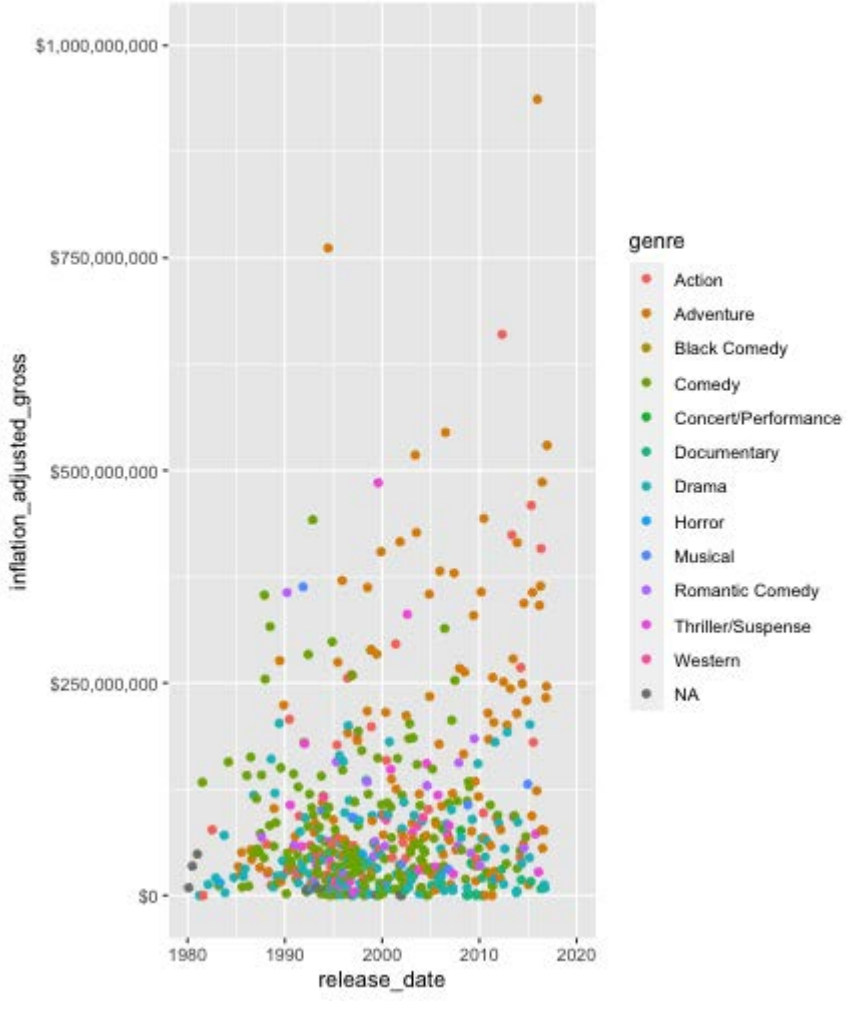
```
disney <- disney %>%
  mutate(release_date=mdy(release_date),
         total_gross=parse_number(total_gross),
         inflation_adjusted_gross=parse_number(inflation_adjusted_gross))

glimpse(disney)
```

```
Rows: 579
Columns: 6
$ movie_title              <chr> "Snow White and the Seven Dwarfs", "Pinocchio…
$ release_date             <date> 1937-12-21, 1940-02-09, 1940-11-13, 1946-11-…
$ genre                    <chr> "Musical", "Adventure", "Musical", "Adventure…
$ MPAA_rating              <chr> "G", "G", "G", "G", "G", NA, "G", NA, "G", NA…
$ total_gross              <dbl> 184925485, 84300000, 83320000, 65000000, 8500…
$ inflation_adjusted_gross <dbl> 5228953251, 2188229052, 2187090808, 107851057…
```

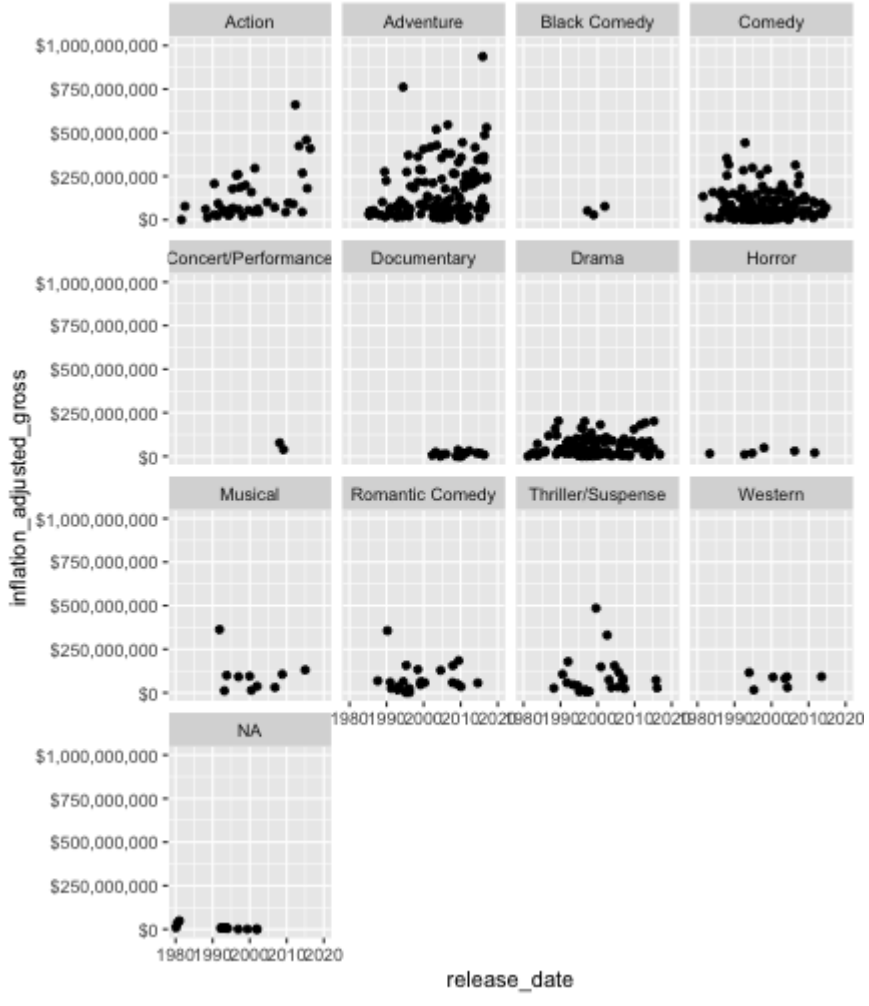
```
disney %>%
  ggplot(aes(x=release_date,
             y=inflation_adjusted_gross,
             color=genre)) +
  geom_point()
```



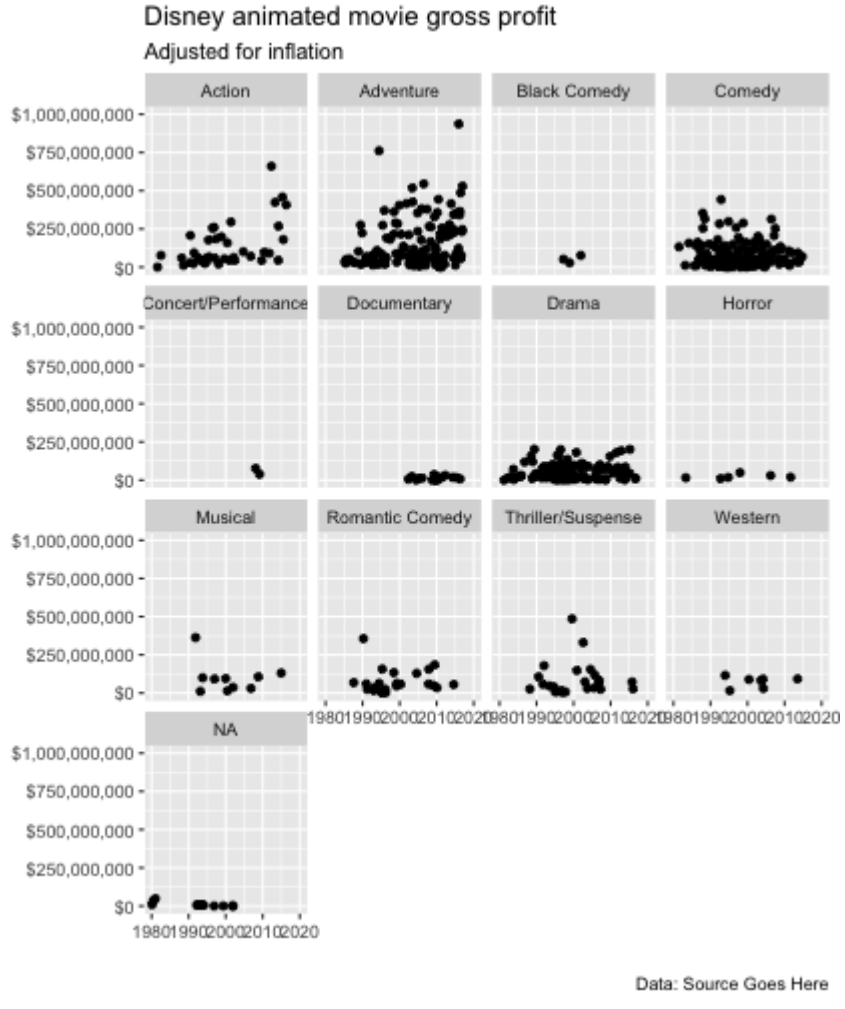
```
disney %>%
  ggplot(aes(x=release_date,
             y=inflation_adjusted_gross,
             color=genre)) +
  geom_point() +
  scale_x_date(limits=c(ymd("1980-01-01"),
                        ymd("2020-01-01")
                   labels=scales::date_format
  scale_y_continuous(limits=c(0, 1000000000),
                   labels=scales::dollar_form
```



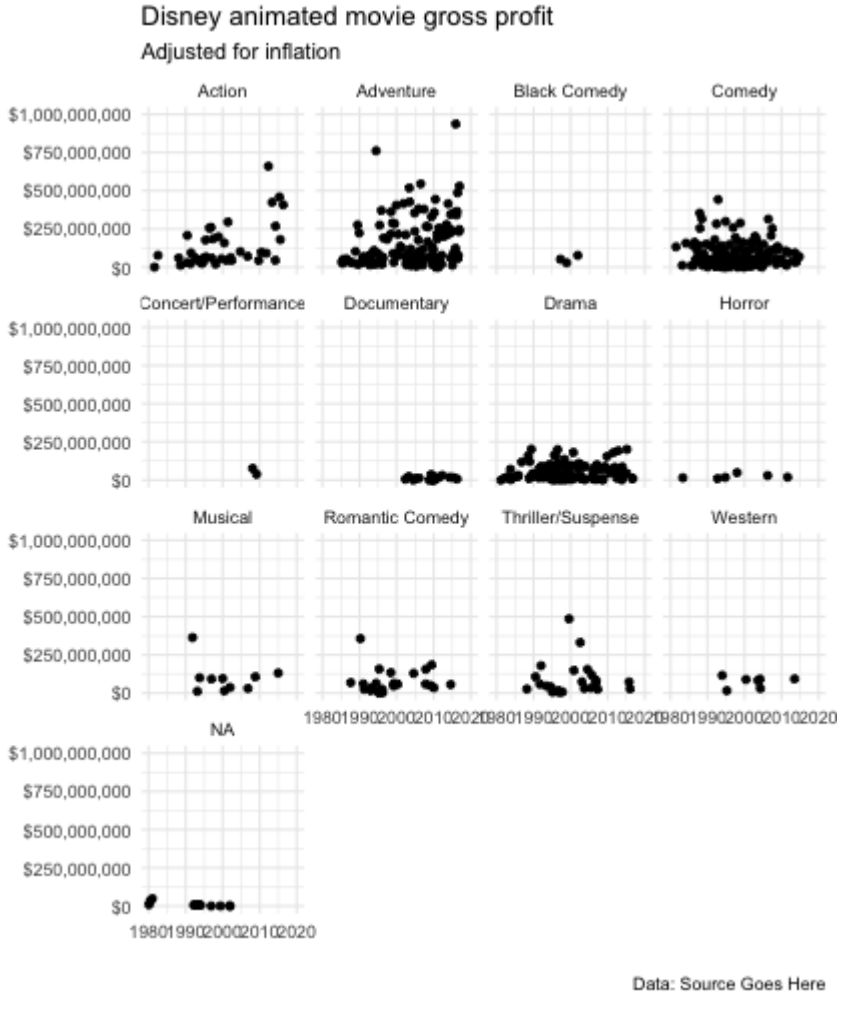
```
disney %>%
  ggplot(aes(x=release_date,
             y=inflation_adjusted_gross)) +
  geom_point() +
  scale_x_date(limits=c(ymd("1980-01-01"),
                        ymd("2020-01-01")
               labels=scales::date_format
  scale_y_continuous(limits=c(0, 1000000000),
                     labels=scales::dollar_form
  facet_wrap(~genre)
```



```
disney %>%
  ggplot(aes(x=release_date,
             y=inflation_adjusted_gross)) +
  geom_point() +
  scale_x_date(limits=c(ymd("1980-01-01"),
                        ymd("2020-01-01")
                labels=scales::date_format
  scale_y_continuous(limits=c(0, 1000000000),
                labels=scales::dollar_form
  facet_wrap(~genre) +
  labs(title="Disney animated movie gross profi
  labs(subtitle="Adjusted for inflation") +
  labs(y="", x="") +
  labs(caption="Data: Source Goes Here")
```



```
disney %>%
  ggplot(aes(x=release_date,
             y=inflation_adjusted_gross)) +
  geom_point() +
  scale_x_date(limits=c(ymd("1980-01-01"),
                        ymd("2020-01-01")
                   labels=scales::date_format
  scale_y_continuous(limits=c(0, 1000000000),
                   labels=scales::dollar_form
  facet_wrap(~genre) +
  labs(title="Disney animated movie gross profi
  labs(subtitle="Adjusted for inflation") +
  labs(y="", x="") +
  labs(caption="Data: Source Goes Here") +
  theme(strip.background = element_rect(colour
  theme(legend.key = element_rect(fill = "white
  theme_minimal()
```




ggplot2:
Build a data
MASTERPIECE
THEMES
gEOMS
aes
fill
color
ggAnimate
Plotly
HORST '18