Lecture 14: Oct 5, 2018

Tidy Data

- *Pipe Operator*
- *Tidy Data*
- *Tidy Transformation*
  - *Wide to Long*, *Long to Wide*, *Separate / Unite*

James Balamuta
STAT 385 @ UIUC

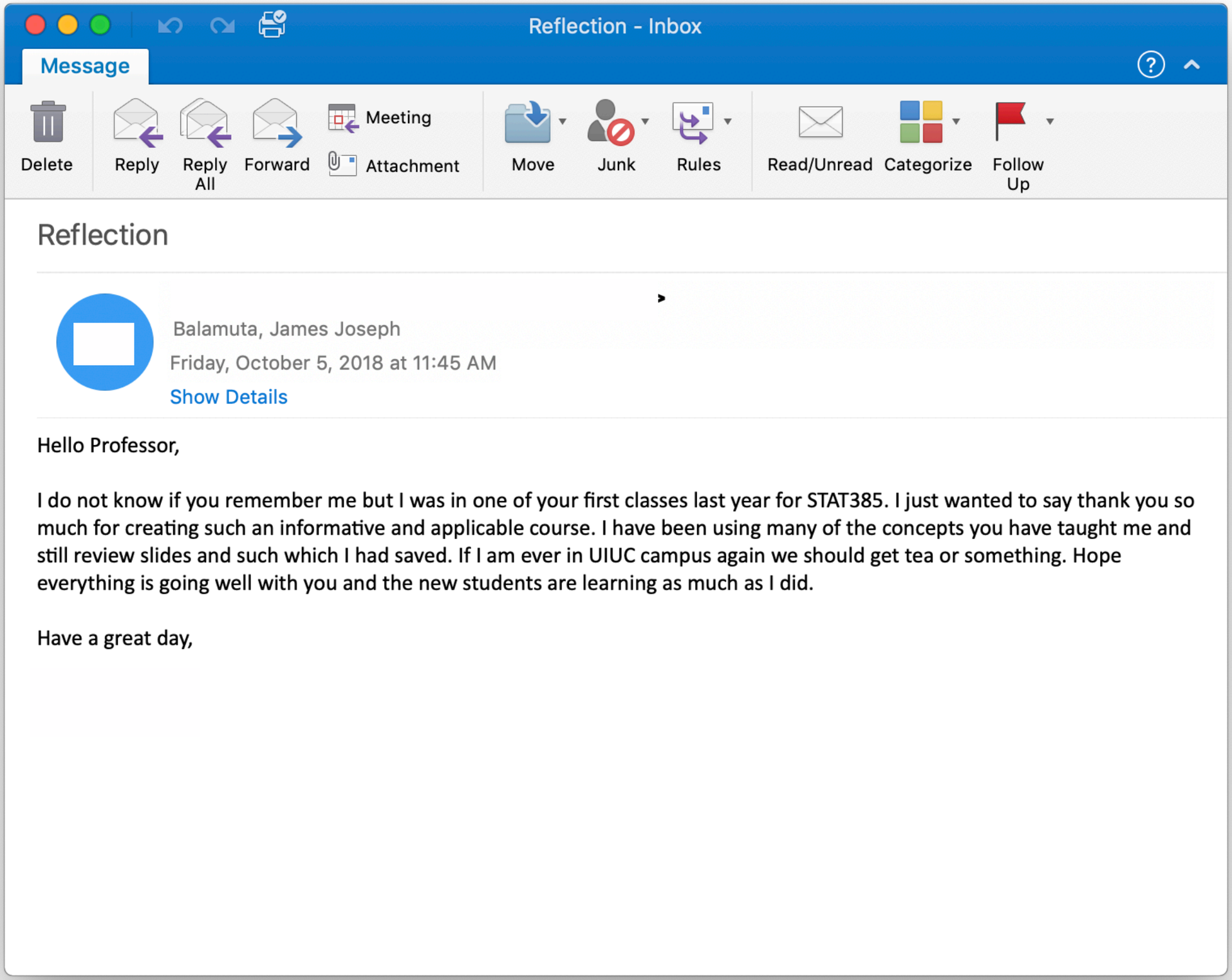
Announcements

- **hw05** is due **Friday, Oct 5th, 2018** at **6:00 PM**
- **John Lee's Office Hours** are now from **4 - 5 PM** on **WF**
- **Quiz 06** covers Week 5 contents @ CBTF.
  - Window: Oct 2nd - 4th
  - Sign up: https://cbtf.engr.illinois.edu/sched
- Want to review your homework or quiz grades?
  Schedule an appointment.
- Got caught using GitHub's web interface in hw01 or hw02?
  Let's chat.

Last Time

- **Exploratory Data Analysis**
  - Search for patterns within the data *before* modeling
  - Quantitative vs. Visual
- **Graphing in R**
  - Three different systems: Base R, lattice, **ggplot2**
- **ggplot2**
  - Grammar of graphics implementation in *R*
  - Powerful system for quickly constructing EDA graphics on data

Real World STAT 385 Alumni



Lecture Objectives

- Creating a workflow with the **Pipe operator.**
- Describe the three principles governing **Tidy Data**.
- Determine the operations required to **transform** a "messy" data set into a tidy one.

Pipe Operator

Definition:
Piping is the act of taking one value and immediately placing it into another function to form a flow of results.

Left Function
Transmitting function result
`rnorm(10)`

Pipe Operator
Facilitate moving left result to the function on right

Right Function
Receiving function result in first parameter
`abs(rnorm(10))`

```
rnorm(10) %>% abs()
```

%>% is read as "and, then"

Pipe Operator

… moving output from one function directly into another …

```
# install.packages("magrittr")
library("magrittr")
4 %>%          # Take the number four and, then
   sqrt()      # find the square root

# Same as
# sqrt(4)

c(7, 42,  1,  25) %>%   # Combine four elements and, then
   log() %>%            # take the natural log and, then
   round(2) %>%         # round to the second decimal and, then
   diff()               # take the difference between consecutive elements

# Same as
# diff(round(log(c(7,42,1,25)), 2))
```

Combining Multiple Steps

… options to work with multiple functions …

```r
# Embedded / Nested Functions
set.seed(821)
mean(rnorm(10))
# [1] -0.06009905

# Temporary Intermediate Variables
set.seed(821)
rand_nums = rnorm(10)
mean_nums = mean(rand_nums)

# Piped
# install.packages("magrittr")
library("magrittr")
set.seed(821)
rnorm(10) %>%   # Generate 10 random values from a normal and, then
   mean()       # take the mean.
```

Multi-step Problems

… ordering a Starbucks Drink with Mobile Ordering…

"Find drink, select store, order, go to store, and pickup drink."

Embedded / Nested

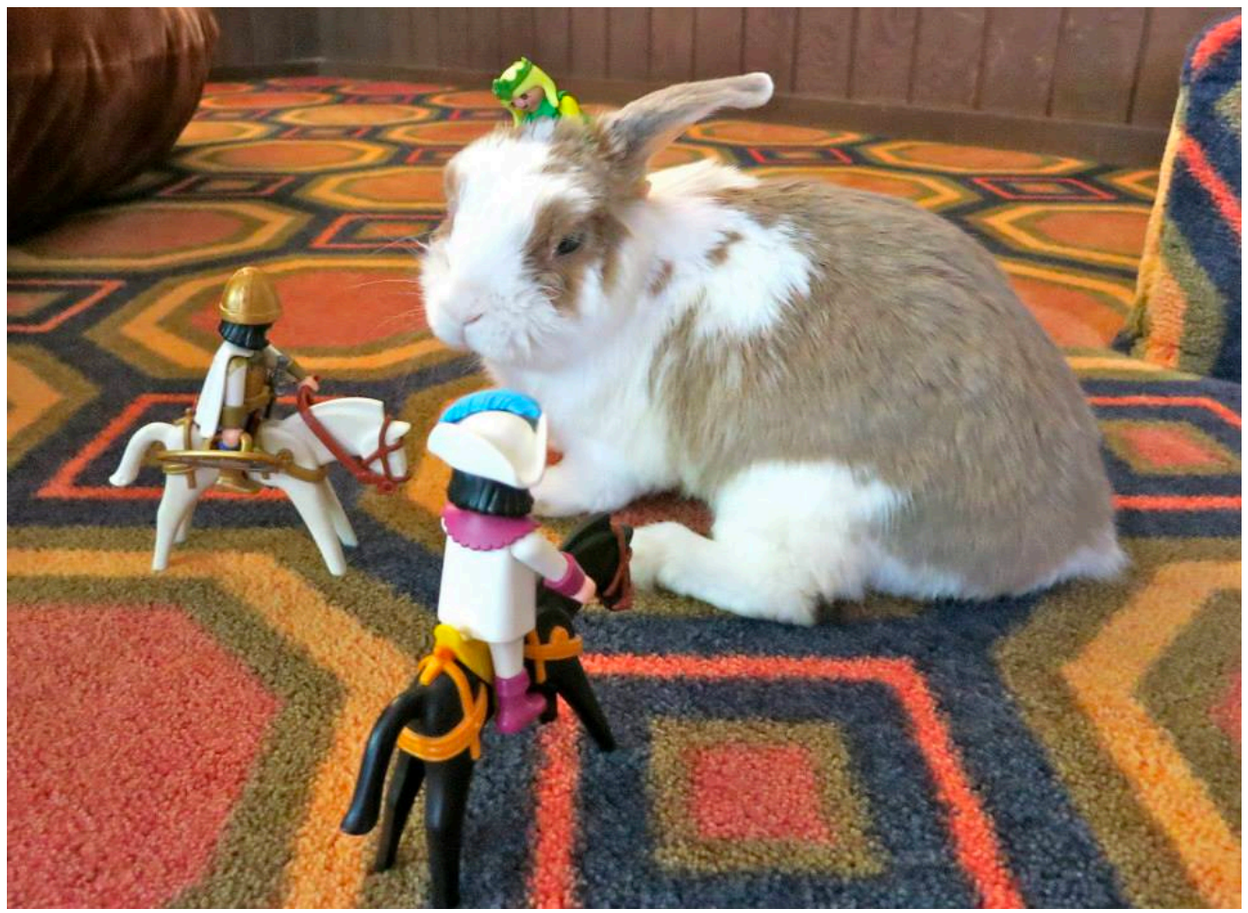
```
pickup(                                  # Step 5
  goto(                                  # Step 4
    order(                               # Step 3
      store(                             # Step 2
        drink("Java Chip Frap"),  # Step 1
        loc = "Green St.")
      )
    )
  )
```

Piped

```
drink("Java Chip Frap") %>%   # Step 1
  store(loc = "Green St.") %>% # Step 2
  order() %>%                  # Step 3
  goto() %>%                   # Step 4
  pickup()                     # Step 5
```

Bunny Foo Foo

… example of piping logic …



Family picture of S'more

[Hadley Wickham's Bunny Foo Foo Example](#)
UseR 2016 Keynote

```
# Create a Bunny
foo_foo = little_bunny()

bop_on(
    scoop_up(
      hop_through(foo_foo, forest),
       field_mouse
    ),
    head
)

# vs.

foo_foo %>%
 hop_through(forest) %>%
 scoop_up(field_mouse) %>%
  bop_on(head)
```

Argument Order

… piping to a parameter **other** than the *first* using a **period** (.)

```
subtract_vals = function(x, y) {
  x - y
}

x = 7; y = 4
subtract_vals(x, y)
# [1] 3

x %>% subtract_vals(y)   # Default, e.g. subtract_vals(x, y)
# [1] 3

x %>% subtract_vals( . , y) # Default, e.g. subtract_vals(x, y)
# [1] 3                     # ^ Period specifies where the value should go

y %>% subtract_vals(x, . ) # Pipe y to second argument, e.g. subtract_vals(x, y)
# [1] 3                     # ^ Period specifies where the value should go

x %>% subtract_vals(y, . ) # Pipe x to second argument, e.g. subtract_vals(y, x)
# [1] -3                    # ^ Period specifies where the value should go
```

Accessing Values

… representing objects with the period (.) in a pipe …

```r
my_df = data.frame(x = c(0, 1), y = c(2, 3))

my_df %>% .[["x"]]
          # ^ Period specifies where the data frame should go
# [1] 0 1

# Equivalent to
my_df$x
my_df[["x"]]
my_df[[1]]

# Nested period usage to retrieve the last column
my_df %>% .[[ncol(.)]]
          # ^        ^ Period specifies where the data frame should go
# [1] 2 3

# Equivalent to
my_df[[ncol(my_df)]]
```

Your Turn

Make the following embedded function calls "pipeable"

```
# install.packages("dplyr")
library("dplyr")
tail(filter(iris, Petal.Width > mean(Petal.Width)))
```

Write a pipe that provides the **sqrt** of **2+2**

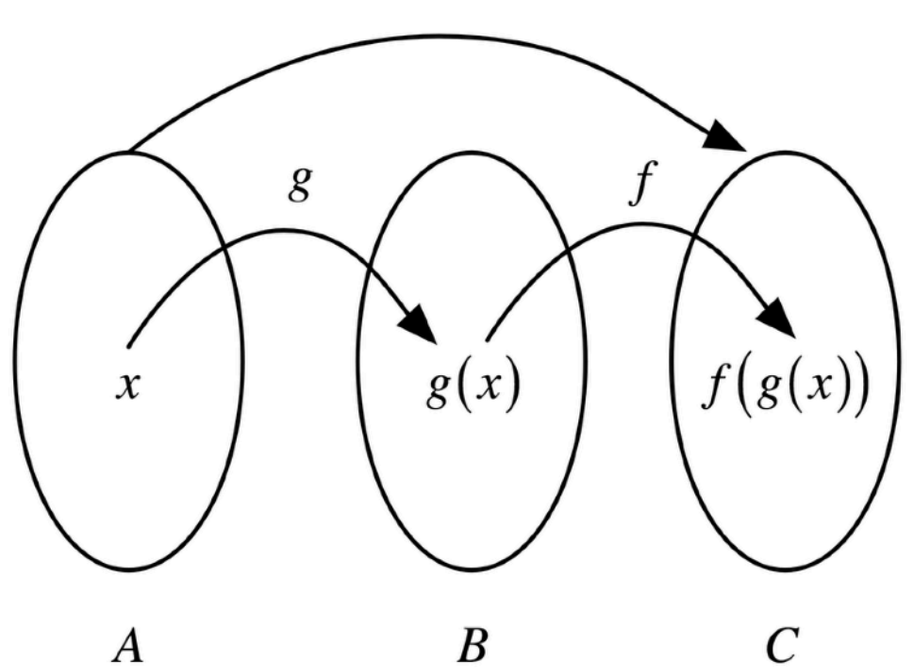
Mathematical Origins

… piping is the **composition of functions** …

Consider two functions $g : A \to B$ and $f : B \to C$.

These functions can be chained together by taking the output of one function and inserting it into the next, e.g. $f(g(x))$. Thus, the input g is x and the result $g(x)$ serves as the input for f .
The notation for this is $f \circ g$ (read as "f follows g").

$$(f \circ g)(x) = f\big(g(x)\big)$$



Tidy Data

Anna Karenina Principle*

… successful on each dimension or not at all …

"**Happy families** are all alike; **every unhappy family** is **unhappy** in its own way."

–Leo Tolstoy

"**Tidy datasets** are all alike but **every messy dataset** is **messy** in its own way."

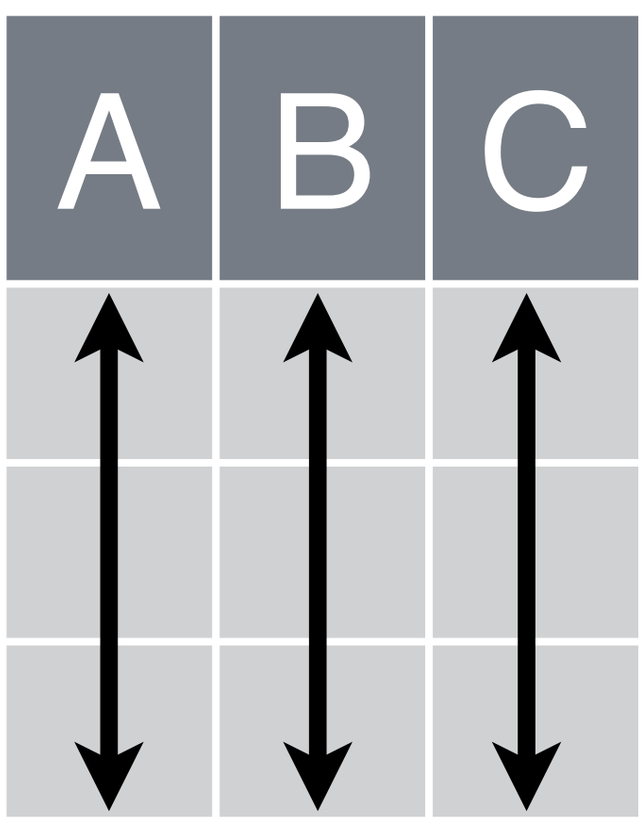
– Hadley Wickham

---

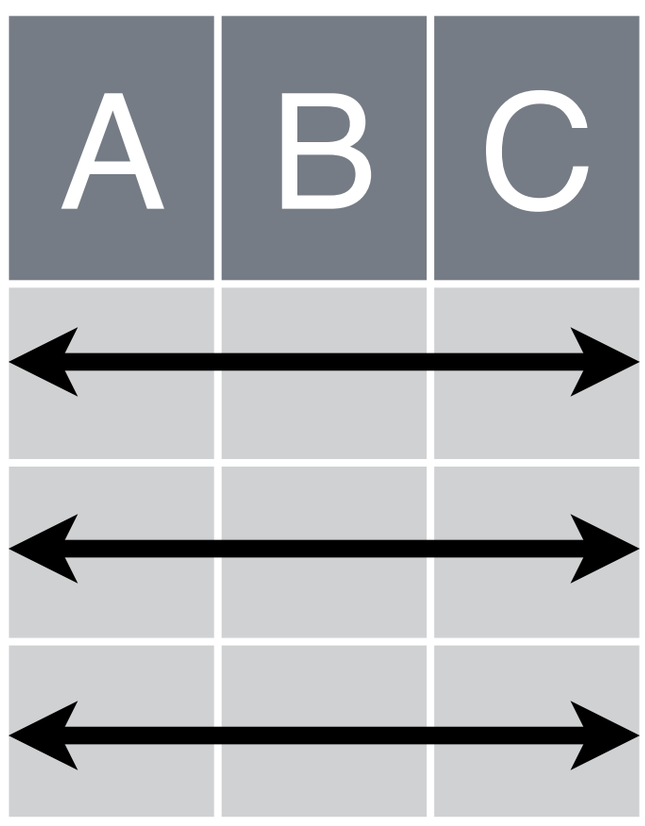
* Anna Karenina Principle guides how hypothesis tests are formed in a way that any hypothesis test has multiple ways for its null hypothesis to be violated but only one way for it to not be.

Tidy Data

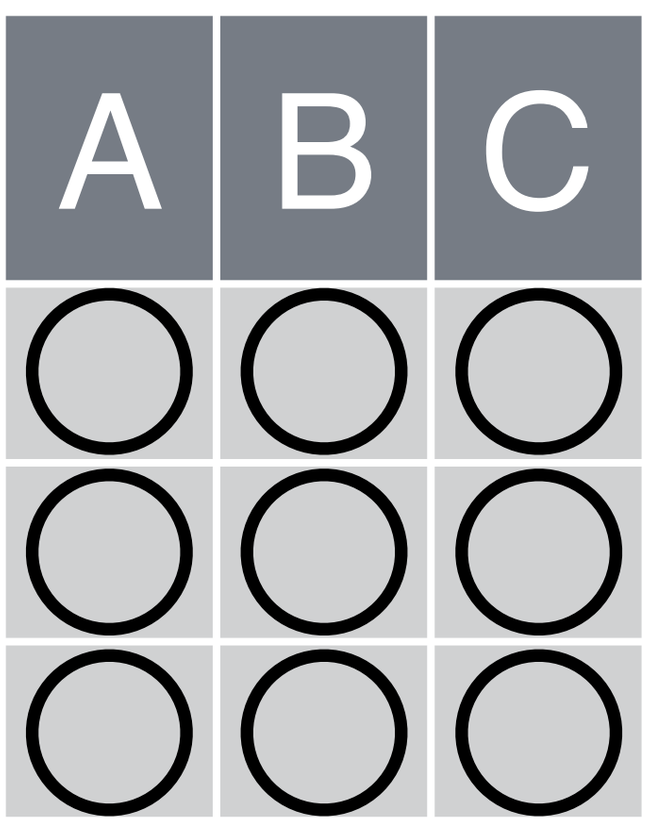
… facilitate data modeling, graphing, aggregation with structure …



Each **variable** must have its own **column**



Each **observation** must have its own **row**



Each **value** must have its own **cell**

Cobb's 3NF (Third Normal Form)

Tidy Data (Paper)

Messy

… three lurking variables …

Gender	Undergrad	Professional	Graduate
Men	18,345	352	7,173
Women	15,267	640	6,028
Unknown	12	0	9

3 x 4

enrolled_fa2017

Source: http://www.dmi.illinois.edu/stuenr/abstracts/FA17_ten.htm

Are you peeking?

Tidied Data

… reorganized data that can be manipulated quickly …

Year	Gender	Enrolled
Undergrad	Men	18,345
Undergrad	Women	15,267
Undergrad	Unknown	12
Professional	Men	352
Professional	Women	640
Professional	Unknown	0
Graduate	Men	7,173
Graduate	Women	6,028
Graduate	Unknown	9

9 x 3

enrolled_fa2017

Source: http://www.dmi.illinois.edu/stuenr/abstracts/FA17_ten.htm

"Messy Data"

… forms of messiness …

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

What kind of messiness was the enrollment table?

Year	Men	Women	Unknown
Undergrad	18,345	15,267	12
Professional	352	640	0
Graduate	7,173	6,028	9

3 x 4

enrolled_fa2017

Year	Gender	Enrolled
Undergrad	Men	18,345
Undergrad	Women	15,267
Undergrad	Unknown	12
Professional	Men	352
Professional	Women	640
Professional	Unknown	0
Graduate	Men	7,173
Graduate	Women	6,028
Graduate	Unknown	9

9 x 3

enrolled_fa2017_tidy

Source: http://www.dmi.illinois.edu/stuenr/abstracts/FA17_ten.htm

Alternative Names

... "long" and "wide" vs. "tidy" and "messy" ...

Long Data / Generally Tidy

weight <dbl>	Time <dbl>	Chick <ord>	Diet <fctr>
42	0	1	1
51	2	1	1
59	4	1	1
64	6	1	1
76	8	1	1
93	10	1	1

ChickWeight_long

Wide Data / Messy

Chick <ord>	Diet <fctr>	0 <dbl>	1 <dbl>	2 <dbl>
18	1	39	35	NA
16	1	41	45	49
15	1	41	49	56
13	1	41	48	43
9	1	42	51	59
20	1	41	47	54

ChickWeight_wide

---

* **Long Data** has each row as one response per subject and any variables for the subject that do not change over time or treatment will have the same value in all the rows.
** **Wide Data** has repeated responses or treatments of a subject in a single row with each response in its own column along with its properties.

Multiple Variables are Stored in **One** Column

… untidy, but useful …

country	year	column	cases
AD	2000	m014	0
AD	2000	m1524	0
AD	2000	m2534	1
AD	2000	m3544	0
AD	2000	m4554	0
AD	2000	m5564	0
AD	2000	m65	0
AE	2000	m014	2
AE	2000	m1524	4
AE	2000	m2534	4
AE	2000	m3544	6
AE	2000	m4554	5
AE	2000	m5564	12
AE	2000	m65	10
AE	2000	f014	3

country	year	sex	age	cases
AD	2000	m	0–14	0
AD	2000	m	15–24	0
AD	2000	m	25–34	1
AD	2000	m	35–44	0
AD	2000	m	45–54	0
AD	2000	m	55–64	0
AD	2000	m	65+	0
AE	2000	m	0–14	2
AE	2000	m	15–24	4
AE	2000	m	25–34	4
AE	2000	m	35–44	6
AE	2000	m	45–54	5
AE	2000	m	55–64	12
AE	2000	m	65+	10
AE	2000	f	0-14	3

Variables are Stored in **Both** Rows and Columns

... untidy, and problematic

V1

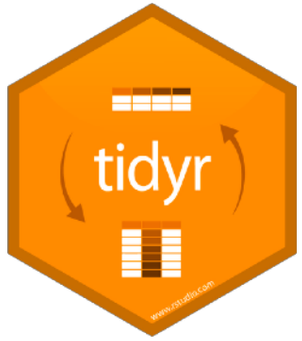
id	year	month	element	d1	d2	d3	d4	d5	d6	d7	d8
MX17004	2010	1	tmax	—	—	—	—	—	—	—	—
MX17004	2010	1	tmin	—	—	—	—	—	—	—	—
MX17004	2010	2	tmax	—	27.3	24.1	—	—	—	—	—
MX17004	2010	2	tmin	—	14.4	14.4	—	—	—	—	—
MX17004	2010	3	tmax	—	—	—	—	32.1	—	—	—
MX17004	2010	3	tmin	—	—	—	—	14.2	—	—	—
MX17004	2010	4	tmax	—	—	—	—	—	—	—	—
MX17004	2010	4	tmin	—	—	—	—	—	—	—	—
MX17004	2010	5	tmax	—	—	—	—	—	—	—	—
MX17004	2010	5	tmin	—	—	—	—	—	—	—	—

V2

id	date	element	value
MX17004	2010-01-30	tmax	27.8
MX17004	2010-01-30	tmin	14.5
MX17004	2010-02-02	tmax	27.3
MX17004	2010-02-02	tmin	14.4
MX17004	2010-02-03	tmax	24.1
MX17004	2010-02-03	tmin	14.4
MX17004	2010-02-11	tmax	29.7
MX17004	2010-02-11	tmin	13.4
MX17004	2010-02-23	tmax	29.9
MX17004	2010-02-23	tmin	10.7

id	date	tmax	tmin
MX17004	2010-01-30	27.8	14.5
MX17004	2010-02-02	27.3	14.4
MX17004	2010-02-03	24.1	14.4
MX17004	2010-02-11	29.7	13.4
MX17004	2010-02-23	29.9	10.7
MX17004	2010-03-05	32.1	14.2
MX17004	2010-03-10	34.5	16.8
MX17004	2010-03-16	31.1	17.6
MX17004	2010-04-27	36.3	16.7
MX17004	2010-05-27	33.2	18.2

Tidy Transformation



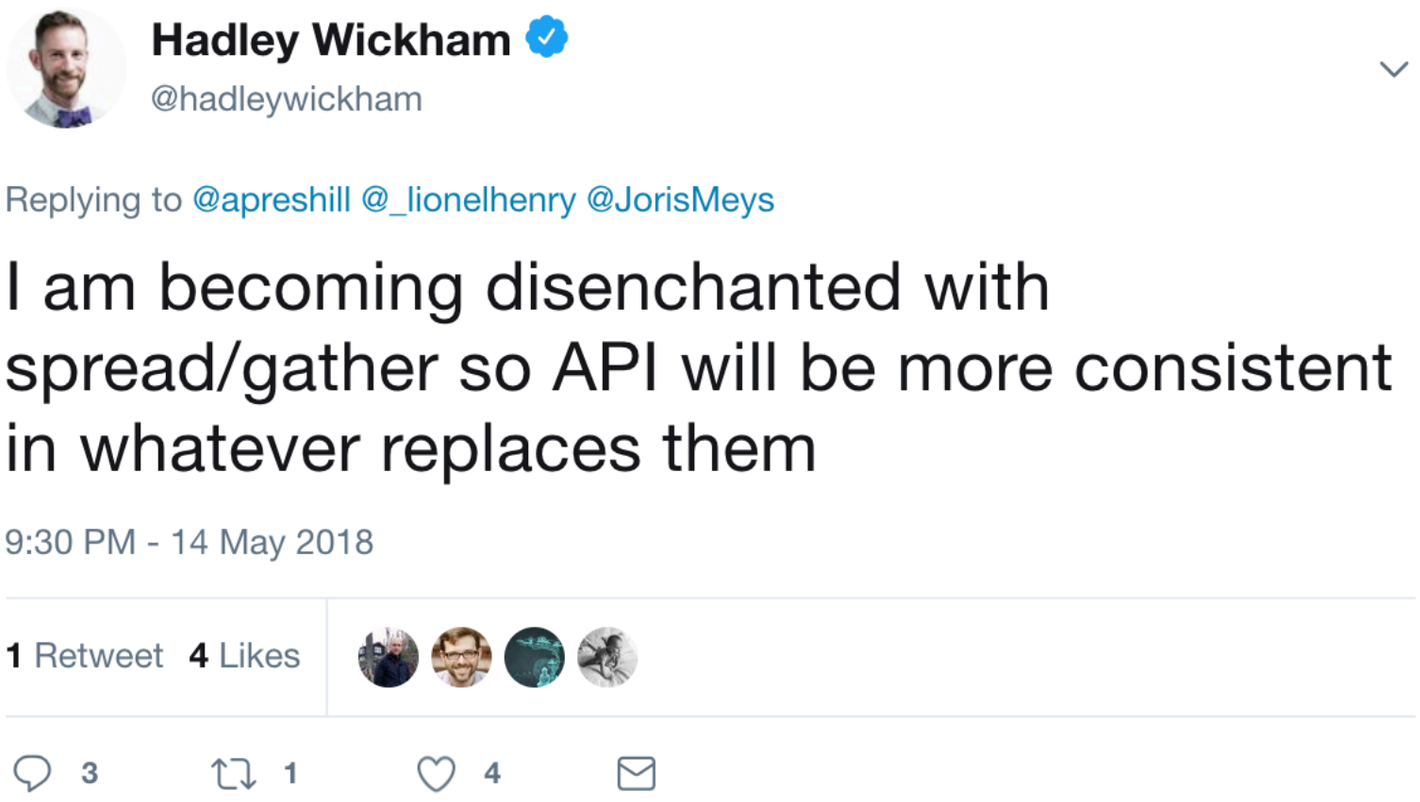
… tidying data …

```
install.packages("tidyr")
library("tidyr")
```

Function	Description
gather(data, key, value, …)	Convert from **wide** to **long-form**
spread(data, key, value, …)	Convert from **long-form** to **wide-form**
separate(data, col, into, sep)	Split apart a column into other columns
separate_rows(data, … , sep)	Split apart a column into rows
unite(data, col, …)	Collapse columns into one column
drop_na(data, …)	Remove all missing values
fill(data, …)	Fill or impute values for missing data

🚨🚨🚨 Warning 🚨🚨🚨

… functions may change …



https://twitter.com/hadleywickham/status/996201030860746752

Tidying in Action

… moving from wide to long and back again …

wide

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

Source

wide

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

gather(wide, key, val, x:y)

spread(long, key, val)

long

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

Recall

Key and Value

… focusing on semistructured …

```
password: nottelling1
```

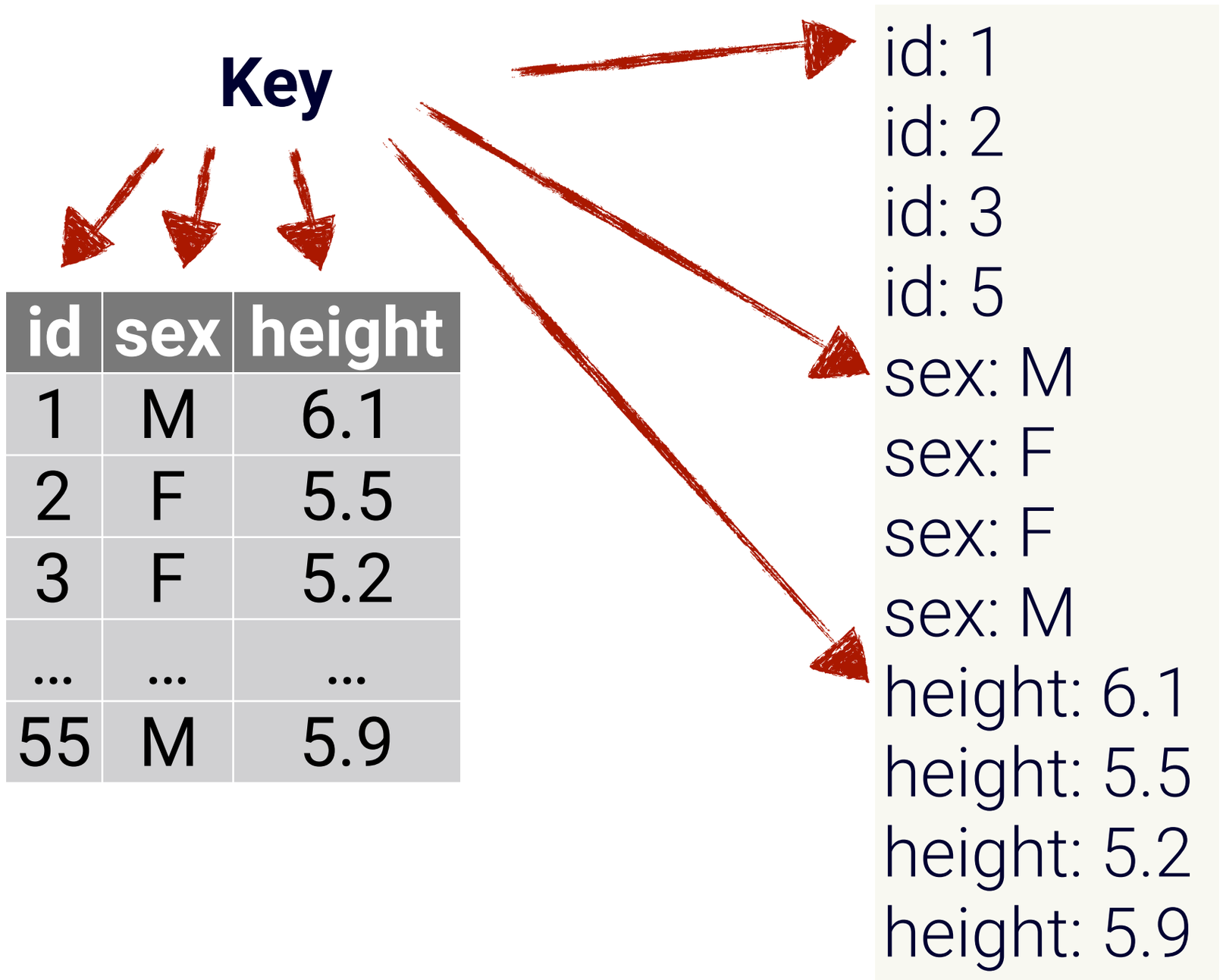
Key
Descriptor for Information

Separator
Indicate split between Key and Value

Value
Actual Data

Representing Data

… where key and values comes into play ..



id	sex	height
1	M	6.1
2	F	5.5
3	F	5.2
…	…	…
55	M	5.9

Example Transform

… gather and spread operations …

```
enrolled_fa2017_long = gather(enrolled_fa2017_wide,
                              key = "Gender", value = "Enrolled", Men:Unknown)
```

Year	Men	Women	Unknown
Undergrad	18,345	15,267	12
Professional	352	640	0
Graduate	7,173	6,028	9

enrolled_fa2017_wide

Year	Gender	Enrolled
Undergrad	Men	18,345
Undergrad	Women	15,267
Undergrad	Unknown	12
Professional	Men	352
Professional	Women	640
Professional	Unknown	0
Graduate	Men	7,173
Graduate	Women	6,028
Graduate	Unknown	9

enrolled_fa2017_long

```
enrolled_fa2017_wide = spread(enrolled_fa2017_long,  key = "Gender", value = "Enrolled")
```

Source: http://www.dmi.illinois.edu/stuenr/abstracts/FA17_ten.htm

Wide to Long

```
enrolled_fa2017_long = gather(enrolled_fa2017_wide,
                              key = "Gender", value = "Enrolled",
                              Men:Unknown)
```

Year	Men	Women	Unknown
Undergrad	18,345	15,267	12
Professional	352	640	0
Graduate	7,173	6,028	9

3 x 4

enrolled_fa2017_wide

Year	key: Gender	value: Enrolled
Undergrad	Men	18,345
Undergrad	Women	15,267
Undergrad	Unknown	12
Professional	Men	352
Professional	Women	640
Professional	Unknown	0
Graduate	Men	7,173
Graduate	Women	6,028
Graduate	Unknown	9

9 x 3

enrolled_fa2017_long

Source: http://www.dmi.illinois.edu/stuenr/abstracts/FA17_ten.htm

```
enrolled_fa2017_long = gather(enrolled_fa2017_wide,
                              key = "Gender", value = "Enrolled",
                              Men:Unknown)
```

Year	Men	Women	Unknown
Undergrad	18,345	15,267	12
Professional	352	640	0
Graduate	7,173	6,028	9

3 x 4

enrolled_fa2017_wide

Year	key: Gender	value: Enrolled
Undergrad	Men	18,345

enrolled_fa2017_long

Source: http://www.dmi.illinois.edu/stuenr/abstracts/FA17_ten.htm

```
enrolled_fa2017_long = gather(enrolled_fa2017_wide,
                              key = "Gender", value = "Enrolled",
                              Men:Unknown)
```

Year	Men	Women	Unknown
Undergrad	18,345	15,267	12
Professional	352	640	0
Graduate	7,173	6,028	9

3 x 4

enrolled_fa2017_wide

key **value**

Year	Gender	Enrolled
Undergrad	Men	18,345
Professional	Men	352

enrolled_fa2017_long

Source: http://www.dmi.illinois.edu/stuenr/abstracts/FA17_ten.htm

```
enrolled_fa2017_long = gather(enrolled_fa2017_wide,
                              key = "Gender", value = "Enrolled",
                              Men:Unknown)
```

Year	Men	Women	Unknown
Undergrad	18,345	15,267	12
Professional	352	640	0
Graduate	7,173	6,028	9

3 x 4

enrolled_fa2017_wide

key **value**

Year	Gender	Enrolled
Undergrad	Men	18,345
Professional	Men	352
Graduate	Men	7,173

enrolled_fa2017_long

Source: http://www.dmi.illinois.edu/stuenr/abstracts/FA17_ten.htm

```
enrolled_fa2017_long = gather(enrolled_fa2017_wide,
                              key = "Gender", value = "Enrolled",
                              Men:Unknown)
```

Year	Men	Women	Unknown
Undergrad	18,345	15,267	12
Professional	352	640	0
Graduate	7,173	6,028	9

3 x 4

enrolled_fa2017_wide

key **value**

Year	Gender	Enrolled
Undergrad	Men	18,345
Professional	Men	352
Graduate	Men	7,173
Undergrad	Women	15,267

enrolled_fa2017_long

Source: http://www.dmi.illinois.edu/stuenr/abstracts/FA17_ten.htm

```
enrolled_fa2017_long = gather(enrolled_fa2017_wide,
                              key = "Gender", value = "Enrolled",
                              Men:Unknown)
```

Year	Men	Women	Unknown
Undergrad	18,345	15,267	12
Professional	352	640	0
Graduate	7,173	6,028	9

3 x 4

enrolled_fa2017_wide

key **value**

Year	Gender	Enrolled
Undergrad	Men	18,345
Professional	Men	352
Graduate	Men	7,173
Undergrad	Women	15,267
Professional	Women	640

enrolled_fa2017_long

Source: http://www.dmi.illinois.edu/stuenr/abstracts/FA17_ten.htm

```
enrolled_fa2017_long = gather(enrolled_fa2017_wide,
                              key = "Gender", value = "Enrolled",
                              Men:Unknown)
```

Year	Men	Women	Unknown
Undergrad	18,345	15,267	12
Professional	352	640	0
Graduate	7,173	6,028	9

3 x 4

enrolled_fa2017_wide

key **value**

Year	Gender	Enrolled
Undergrad	Men	18,345
Professional	Men	352
Graduate	Men	7,173
Undergrad	Women	15,267
Professional	Women	640
Graduate	Women	6,028

enrolled_fa2017_long

Source: http://www.dmi.illinois.edu/stuenr/abstracts/FA17_ten.htm

```
enrolled_fa2017_long = gather(enrolled_fa2017_wide,
                              key = "Gender", value = "Enrolled",
                              Men:Unknown)
```

Year	Men	Women	Unknown
Undergrad	18,345	15,267	12
Professional	352	640	0
Graduate	7,173	6,028	9

3 x 4

enrolled_fa2017_wide

Year	key: Gender	value: Enrolled
Undergrad	Men	18,345
Professional	Men	352
Graduate	Men	7,173
Undergrad	Women	15,267
Professional	Women	640
Graduate	Women	6,028
Undergrad	Unknown	12

enrolled_fa2017_long

Source: http://www.dmi.illinois.edu/stuenr/abstracts/FA17_ten.htm

```
enrolled_fa2017_long = gather(enrolled_fa2017_wide,
                              key = "Gender", value = "Enrolled",
                              Men:Unknown)
```

Year	Men	Women	Unknown
Undergrad	18,345	15,267	12
Professional	352	640	0
Graduate	7,173	6,028	9

enrolled_fa2017_wide

3 x 4

Year	key: Gender	value: Enrolled
Undergrad	Men	18,345
Professional	Men	352
Graduate	Men	7,173
Undergrad	Women	15,267
Professional	Women	640
Graduate	Women	6,028
Undergrad	Unknown	12
Professional	Unknown	0

enrolled_fa2017_long

Source: http://www.dmi.illinois.edu/stuenr/abstracts/FA17_ten.htm

```
enrolled_fa2017_long = gather(enrolled_fa2017_wide,
                              key = "Gender", value = "Enrolled",
                              Men:Unknown)
```

Year	Men	Women	Unknown
Undergrad	18,345	15,267	12
Professional	352	640	0
Graduate	7,173	6,028	9

enrolled_fa2017_wide

3 x 4

Year	key: Gender	value: Enrolled
Undergrad	Men	18,345
Professional	Men	352
Graduate	Men	7,173
Undergrad	Women	15,267
Professional	Women	640
Graduate	Women	6,028
Undergrad	Unknown	12
Professional	Unknown	0
Graduate	Unknown	9

enrolled_fa2017_long

Source: http://www.dmi.illinois.edu/stuenr/abstracts/FA17_ten.htm

```
enrolled_fa2017_long = gather(enrolled_fa2017_wide,
                              key = "Gender", value = "Enrolled",
                              Men:Unknown)
```

Year	Men	Women	Unknown
Undergrad	18,345	15,267	12
Professional	352	640	0
Graduate	7,173	6,028	9

3 x 4

enrolled_fa2017_wide

key **value**

Year	Gender	Enrolled
Undergrad	Men	18,345
Professional	Men	352
Graduate	Men	7,173
Undergrad	Women	15,267
Professional	Women	640
Graduate	Women	6,028
Undergrad	Unknown	12
Professional	Unknown	0
Graduate	Unknown	9

9 x 3

enrolled_fa2017_long

Source: http://www.dmi.illinois.edu/stuenr/abstracts/FA17_ten.htm

Long to Wide

```
enrolled_fa2017_wide = spread(enrolled_fa2017_long,
                              key = "Gender", value = "Enrolled")
```

Year	Men	Women	Unknown
Undergrad	18,345	15,267	12
Professional	352	640	0
Graduate	7,173	6,028	9

3 x 4

enrolled_fa2017_wide

key **value**

Year	Gender	Enrolled
Undergrad	Men	18,345
Professional	Men	352
Graduate	Men	7,173
Undergrad	Women	15,267
Professional	Women	640
Graduate	Women	6,028
Undergrad	Unknown	12
Professional	Unknown	0
Graduate	Unknown	9

enrolled_fa2017_long

9 x 3

Source: http://www.dmi.illinois.edu/stuenr/abstracts/FA17_ten.htm

```
enrolled_fa2017_wide = spread(enrolled_fa2017_long,
                              key = "Gender", value = "Enrolled")
```

Year	Men	Women	Unknown
Undergrad	18,345		
Professional	352		
Graduate	7,173		

3 x 4

enrolled_fa2017_wide

	key	value
Year	Gender	Enrolled
Undergrad	Men	18,345
Professional	Men	352
Graduate	Men	7,173
Undergrad	Women	15,267
Professional	Women	640
Graduate	Women	6,028
Undergrad	Unknown	12
Professional	Unknown	0
Graduate	Unknown	9

enrolled_fa2017_long

9 x 3

Source: http://www.dmi.illinois.edu/stuenr/abstracts/FA17_ten.htm

```
enrolled_fa2017_wide = spread(enrolled_fa2017_long,
                              key = "Gender", value = "Enrolled")
```

Year	Men	Women	Unknown
Undergrad	18,345	15,267	
Professional	352	640	
Graduate	7,173	6,028	

3 x 4

enrolled_fa2017_wide

key **value**

Year	Gender	Enrolled
Undergrad	Men	18,345
Professional	Men	352
Graduate	Men	7,173
Undergrad	Women	15,267
Professional	Women	640
Graduate	Women	6,028
Undergrad	Unknown	12
Professional	Unknown	0
Graduate	Unknown	9

enrolled_fa2017_long 9 x 3

Source: http://www.dmi.illinois.edu/stuenr/abstracts/FA17_ten.htm

```
enrolled_fa2017_wide = spread(enrolled_fa2017_long,
                              key = "Gender", value = "Enrolled")
```

Year	Men	Women	Unknown
Undergrad	18,345	15,267	12
Professional	352	640	0
Graduate	7,173	6,028	9

3 x 4

enrolled_fa2017_wide

key **value**

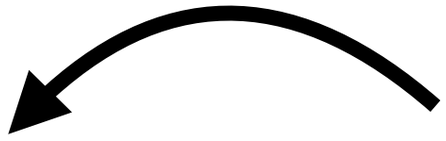
Year	Gender	Enrolled
Undergrad	Men	18,345
Professional	Men	352
Graduate	Men	7,173
Undergrad	Women	15,267
Professional	Women	640
Graduate	Women	6,028
Undergrad	Unknown	12
Professional	Unknown	0
Graduate	Unknown	9

enrolled_fa2017_long

9 x 3

Source: http://www.dmi.illinois.edu/stuenr/abstracts/FA17_ten.htm

```
enrolled_fa2017_wide = spread(enrolled_fa2017_long,
                              key = "Gender", value = "Enrolled")
```



Year	Men	Women	Unknown
Undergrad	18,345	15,267	12
Professional	352	640	0
Graduate	7,173	6,028	9

3 x 4

enrolled_fa2017_wide

key **value**

Year	Gender	Enrolled
Undergrad	Men	18,345
Professional	Men	352
Graduate	Men	7,173
Undergrad	Women	15,267
Professional	Women	640
Graduate	Women	6,028
Undergrad	Unknown	12
Professional	Unknown	0
Graduate	Unknown	9

9 x 3

enrolled_fa2017_long

Source: http://www.dmi.illinois.edu/stuenr/abstracts/FA17_ten.htm

Your Turn

1. Convert **mtcars** to a **long**-form data set

```
#        model type   value
# 1 AMC Javelin  mpg  15.200
# 2 AMC Javelin  cyl   8.000
# 3 AMC Javelin disp 304.000
# 4 AMC Javelin   hp 150.000
# 5 AMC Javelin drat   3.150
# 6 AMC Javelin   wt   3.435

  # Move the rowname to a variable name inside the data set.
mtcars$model = rownames(mtcars)
```

2. Transform **mtcars_long** back to a **wide**-form data.

Separate / Unite

Splitting Data

… breakdown Location into Latitude and Longitude …

Multiple variables are stored in one column

```
cities %>% separate(loc, c("lat", "lng"), sep = ",")
```

city <chr>	pop <dbl>	iso3 <chr	province <chr>	loc <chr>
Houston	4053287	USA	Texas	29.81997438, -95.33997929
Miami	2983947	USA	Florida	25.7876107, -80.22410608
Atlanta	2464454	USA	Georgia	33.83001385, -84.39994938
Chicago	5915976	USA	Illinois	41.82999066, -87.75005497
Los Angeles	8097410	USA	California	33.98997825, -118.1799805
Washington, D.C.	2445216	USA	District of Columbia	38.89954938, -77.00941858
New York	13524139	USA	New York	40.74997906, -73.98001693

lat <chr>	lng <chr>
29.81997438	-95.33997929
25.7876107	-80.22410608
33.83001385	-84.39994938
41.82999066	-87.75005497
33.98997825	-118.1799805
38.89954938	-77.00941858
40.74997906	-73.98001693

cities

Source Data

Latitude Longitude

Uniting Data

… combining Latitude and Longitude into Location…

```
cities_split %>% unite(loc, c("lat", "lng"), sep = ",")
```

city <chr>	pop <dbl>	iso3 <chr	province <chr>	lat <chr>	lng <chr>
Houston	4053287	USA	Texas	29.81997438	-95.33997929
Miami	2983947	USA	Florida	25.7876107	-80.22410608
Atlanta	2464454	USA	Georgia	33.83001385	-84.39994938
Chicago	5915976	USA	Illinois	41.82999066	-87.75005497
Los Angeles	8097410	USA	California	33.98997825	-118.1799805
Washington, D.C.	2445216	USA	District of Columbia	38.89954938	-77.00941858
New York	13524139	USA	New York	40.74997906	-73.98001693

loc <chr>
29.81997438, -95.33997929
25.7876107, -80.22410608
33.83001385, -84.39994938
41.82999066, -87.75005497
33.98997825, -118.1799805
38.89954938, -77.00941858
40.74997906, -73.98001693

cities_split

Source Data

Your Turn

1. Load the **who** data set found in the **tidyr**
   - who: World Health Organization
   - Data is from 2014 tuberculosis report
   - http://www.who.int/tb/country/data/download/en/
2. Is the **who** data tidy?
   If not, what principles are being violated?
3. How should the **who** data be tidied?

Resources

Data Import Cheatsheet

… second page …

Tibbles - an enhanced data frame

The **tibble** package provides a new S3 class for storing tabular data, the tibble. Tibbles inherit the data frame class, but improve three behaviors:

- **Subsetting** - [always returns a new tibble, [[and $ always return a vector.
- **No partial matching** - You must use full column names when subsetting
- **Display** - When you print a tibble, R provides a concise view of the data that fits on one screen

```
# A tibble: 234 × 6
   manufacturer      model displ
          <chr>      <chr> <dbl>
1          audi         a4   1.8
2          audi         a4   1.8
3          audi         a4   2.0
4          audi         a4   2.0
5          audi         a4   2.8
6          audi         a4   2.8
7          audi         a4   3.1
8          audi a4 quattro   1.8
9          audi a4 quattro   1.8
10         audi a4 quattro   2.0
# ... with 224 more rows, and 3
#   more variables: year <int>,
#   cyl <int>, trans <chr>
```

tibble display

A large table to display

```
156 1999   6   auto(l4)
157 1999   6   auto(l4)
158 2008   6   auto(l4)
159 2008   8   auto(s4)
160 1999   4 manual(m5)
161 1999   4   auto(l4)
162 2008   4 manual(m5)
163 2008   4 manual(m5)
164 2008   4   auto(l4)
165 2008   4   auto(l4)
166 1999   4   auto(l4)
 [ reached getOption("max.print")
-- omitted 68 rows ]
```

data frame display

- Control the default appearance with options:

  options(tibble.print_max = n, tibble.print_min = m, tibble.width = Inf**)**

- View full data set with **View()** or **glimpse()**
- Revert to data frame with **as.data.frame()**

Tidy Data with tidyr

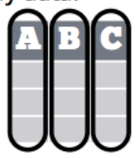
Tidy data is a way to organize tabular data. It provides a consistent data structure across packages.

A table is tidy if:

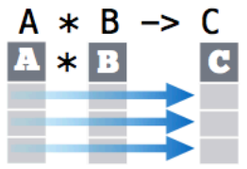
Each **variable** is in its own **column**

Each **observation**, or **case**, is in its own **row**

Tidy data:



Makes variables easy to access as vectors



Preserves cases during vectorized operations

Reshape Data - change the layout of values in a table

Use **gather()** and **spread()** to reorganize the values of a table into a new layout.

gather(data, key, value, ..., na.rm = FALSE, convert = FALSE, factor_key = FALSE**)**

gather() moves column names into a **key** column, gathering the column values into a single **value** column.

table4a

country	1999	2000
A	0.7K	2K
B	37K	80K
C	212K	213K

→

country	year	cases
A	1999	0.7K
B	1999	37K
C	1999	212K
A	2000	2K
B	2000	80K
C	2000	213K

key value

gather(table4a, `1999`, `2000`, key = "year", value = "cases")

spread(data, key, value, fill = NA, convert = FALSE, drop = TRUE, sep = NULL**)**

spread() moves the unique values of a **key** column into the column names, spreading the values of a **value** column across the new columns.

table2

country	year	type	count
A	1999	cases	0.7K
A	1999	pop	19M
A	2000	cases	2K
A	2000	pop	20M
B	1999	cases	37K
B	1999	pop	172M
B	2000	cases	80K
B	2000	pop	174M
C	1999	cases	212K
C	1999	pop	1T
C	2000	cases	213K
C	2000	pop	1T

→

country	year	cases	pop
A	1999	0.7K	19M
A	2000	2K	20M
B	1999	37K	172M
B	2000	80K	174M
C	1999	212K	1T
C	2000	213K	1T

key value

spread(table2, type, count)

Split Cells

Use these functions to split or combine cells into individual, isolated values.

separate(data, col, into, sep = "[^[:alnum:]]+", remove = TRUE, convert = FALSE, extra = "warn", fill = "warn", ...**)**

Separate each cell in a column to make several columns.

table3

country	year	rate
A	1999	0.7K/19M
A	2000	2K/20M
B	1999	37K/172M
B	2000	80K/174M
C	1999	212K/1T
C	2000	213K/1T

→

country	year	cases	pop
A	1999	0.7K	19M
A	2000	2K	20M
B	1999	37K	172
B	2000	80K	174
C	1999	212K	1T
C	2000	213K	1T

separate(table3, rate, into = c("cases", "pop"))

separate_rows(data, ..., sep = "[^[:alnum:].]+", convert = FALSE**)**

Separate each cell in a column to make several rows. Also **separate_rows_()**.

table3

country	year	rate
A	1999	0.7K/19M
A	2000	2K/20M
B	1999	37K/172M
B	2000	80K/174M
C	1999	212K/1T
C	2000	213K/1T

→

country	year	rate
A	1999	0.7K
A	1999	19M
A	2000	2K
A	2000	20M
B	1999	37K
B	1999	172M
B	2000	80K

https://github.com/rstudio/cheatsheets/raw/master/data-import.pdf

Recap

- **Tidy Data**
  - Each **variable** must have its own **column**
  - Each **observation** must have its own **row**
  - Each **value** must have its own **cell**

- **Long vs. Wide**
  - **Long Data** has each row as one response per subject and any variables for the subject that do not change over time or treatment will have the same value in all the rows. Use **gather()**.
  - **Wide Data** has repeated responses or treatments of a subject in a single row with each response in its own column along with its properties. Use **spread()**.

Acknowledgements

- RStudio's Data Import Cheatsheet Styling

- Hadley Wickham's R4DS Chapter 12: Tidy Data and Tidy Data Paper