dplyr

Jeff Allen

Dallas R Users Group
7/11/15

Code for talk: http://tres.tl/dplyrcode

@trestleJeff

My Background

- Computer Scientist
- First encountered R as a programming language (2007)
- Only later used it for data analysis
- Now a Software Engineer at RStudio (2013)

Your Background

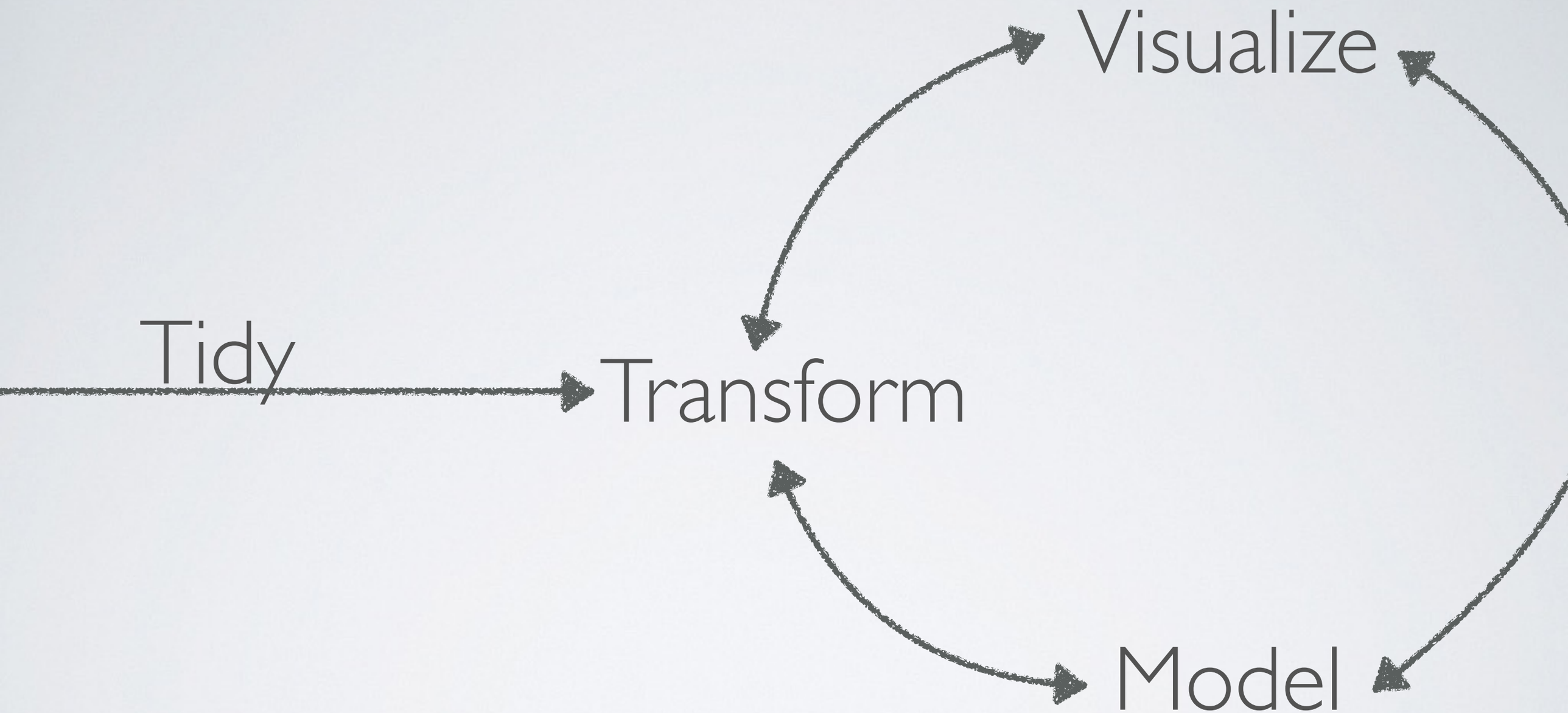
- New to R?
- Intermediate-Advanced R user?
- Used dplyr before?

R Consortium

- Support R Core with development and finances
- Organized by Linux Foundation
- New R-forge, documentation, etc.
- https://www.r-consortium.org/

dplyr

- Open-source R package
- From Hadley Wickham (ggplot2, plyr, devtools, …)
- Grammar of data manipulation
  - Operates on data.frames



http://www.londonr.org/Presentations/Hadley%20Wickham%20-%20bigr-data-londonr.pdf

Tidy
tidyr

dplyr
Transform

Visualize
ggplot2
ggvis

Model
The rest of R…

http://www.londonr.org/Presentations/Hadley%20Wickham%20-%20bigr-data-londonr.pdf

Motivation

- Unified syntax, captures 90% of data transformation tasks
- Consistent interface (great for "piping")
- **Performance** (up to 100x in certain cases)
- More to come…

Data Intake

- At simplest: A special data.frame
- All the same properties of a data.frame
- `tbl_df(myDataFrame)`

Fundamental verbs

- **select** - subset columns
- **filter** - subset rows
- **mutate** - add new columns
- **arrange** - re-order rows
- **summarize** - reduce to single row
- **group_by** - "bin" data

Fundamental verbs

- **select** - subset columns
- **filter** - subset rows
- **mutate** - add new columns
- **arrange** - re-order rows
- **summarize** - reduce to single row
- **group_by** - "bin" data

select

- Take a subset of columns
- Use column names without quotes
- "-" to exclude a variable
- `starts_with(), ends_with(), matches(), …`

Fundamental verbs

- **select** - subset columns
- **filter** - subset rows
- **mutate** - add new columns
- **arrange** - re-order rows
- **summarize** - reduce to single row
- **group_by** - "bin" data

filter

- Take a subset of rows

- Use regular R Boolean vector logic

Fundamental verbs

- **select** - subset columns
- **filter** - subset rows
- **mutate** - add new columns
- **arrange** - re-order rows
- **summarize** - reduce to single row
- **group_by** - "bin" data

mutate

- Add new columns

- Potentially based on existing columns

A Brief Interruption

- Pipes offer an alternative syntax to nest functions


```
baz(
    foo(a=1),
    b=2
)
```

==

```
foo(a=1) %>%
  baz(b=2)
```

- Comes from the magrittr package

```
tumble_after(
  broke(
    fell_down(
      fetch(
        went_up(jack_jill, "hill"),
        "water"),
      jack
    ),
    "crown"),
  "jill"
)
```

```
jack_jill %>%
  went_up("hill") %>%
  fetch("water") %>%
  fell_down("jack") %>%
broke("crown") %>%
tumble_after("jill")
```

Fundamental verbs

- **select** - subset columns
- **filter** - subset rows
- **mutate** - add new columns
- **arrange** - re-order rows
- **summarize** - reduce to single row
- **group_by** - "bin" data

arrange

- Sort rows

- Use `desc()` to sort in decrementing order

Fundamental verbs

- **select** - subset columns
- **filter** - subset rows
- **mutate** - add new columns
- **arrange** - re-order rows
- **summarize** - reduce to single row
- **group_by** - "bin" data

summarize

- Aggregate data into a single row

- Provide a summarization function for each column you want to keep

- Special functions like `n()` to get the count

Fundamental verbs

- **select** - subset columns
- **filter** - subset rows
- **mutate** - add new columns
- **arrange** - re-order rows
- **summarize** - reduce to single row
- **group_by** - "bin" data

group_by

- Bin data into independent sets
- By itself, doesn't change the data
- Perform further actions — such as `summarize()` — independently on each group

nycflights13

Joins

- Bind data from two tables together
- left_join(), right_join(), inner_join(), full_join(), …
- Concatenates columns together for rows that have corresponding keys

Joins

User	Age	Dept
joe	41	QA
kim	39	IT
steve	32	IT

Dept	Room#
IT	307
QA	410

Joins

User	Age	Dept
joe	41	QA
kim	39	IT
steve	32	IT

Dept	Room#
IT	307
QA	410

Joins

User	Age	Dept
joe	41	QA
kim	39	IT
steve	32	IT

Dept	Room#
IT	307
QA	410

User	Age	Dept	Room#
joe	41	QA	410
kim	39	IT	307
steve	32	IT	307

Join Key Collisions

User	Age	Dept
joe	41	QA
kim	39	IT
steve	32	IT

Dept	Room#
IT	307
QA	410

User	Age	Dept
joe	41	QA
kim	39	IT
steve	32	IT

Dept	Room#	Age
IT	307	15
QA	410	7

User	Age	Dept
joe	41	QA
kim	39	IT
steve	32	IT

Dept	Room#	Age
IT	307	15
QA	410	7

User	Age	Dept	Room#

User	Age	Dept
joe	41	QA
kim	39	IT
steve	32	IT

Dept	Room#	Age
IT	307	15
QA	410	7

by="Dept"

User	Age	Dept	Room#
joe	41	QA	410
kim	39	IT	307
steve	32	IT	307

Data Sources

- ✔ Local data.frame or data.table
- ☐ Local SQLite database
- ☐ Remote MySQL/PostgreSQL database
- ☐ Google BigQuery, Amazon RedShift, MonetDB

dplyr + MySQL

- dplyr views MySQL as just another data source
- `translate_sql()` does the behind-the-scenes magic
  - Converts what it can to a SQL query
  - Runs everything else locally in R

Lazy Evaluation

- dplyr avoids executing queries until it absolutely has to

- Use `explain()` to ask the RDBMS about the execution plan for this query.

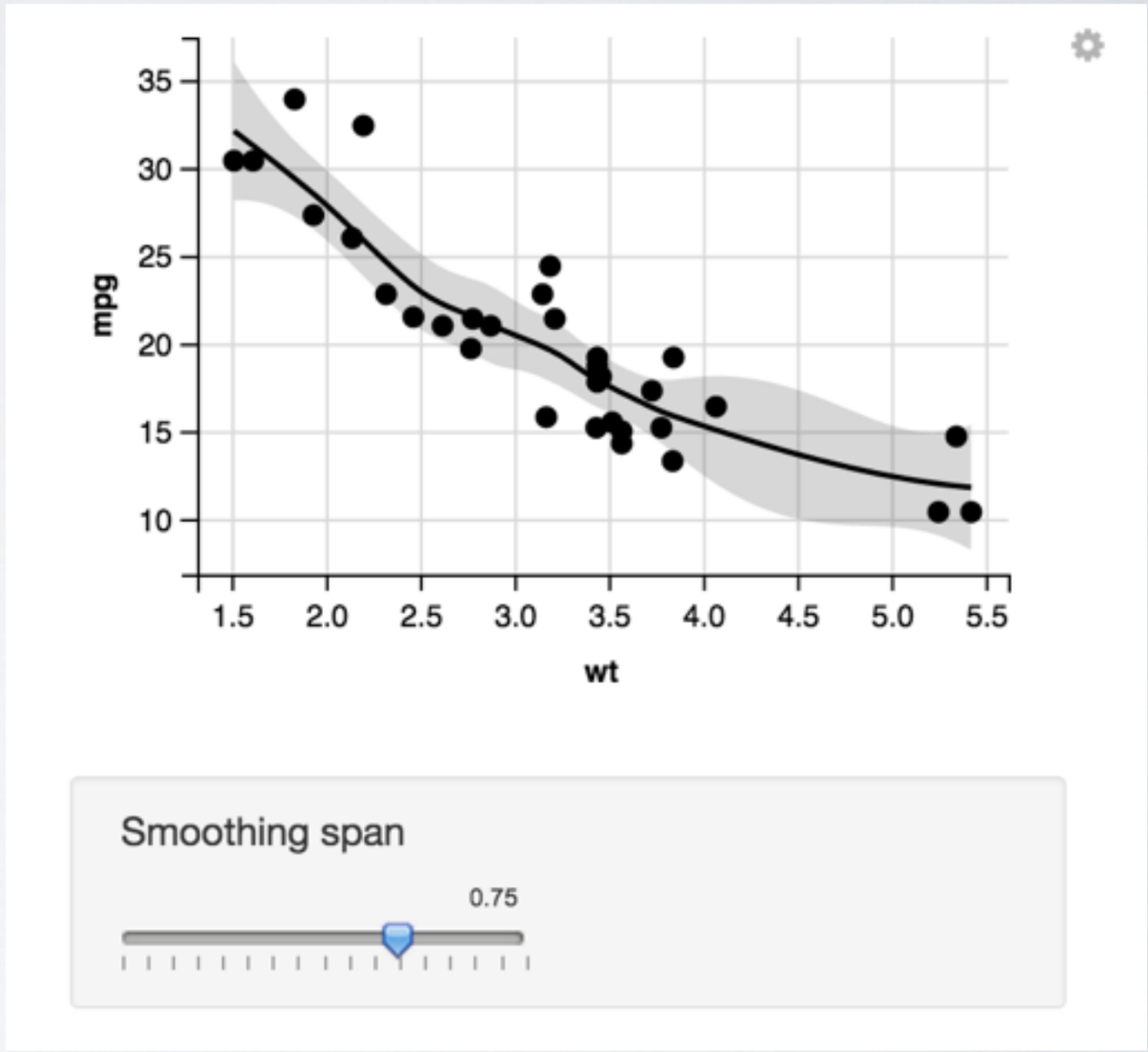
- Use `collect()` to force evaluation

What's Next?

ggvis

ggvis

- Successor to ggplot2
- Same "grammar of graphics." Updated syntax
- *Of* the Web — runs in a browser
- Built-in reactivity
- Pipeable, like dplyr
- http://ggvis.rstudio.com/



leaflet

leaflet

- R package for creating interactive maps
- A new major release recently
- Trivial to use