Introduction to the tidyverse

Christopher Skovron
Northwestern University

July 16, 2018

Chris Skovron

- Data Science Scholar, Northwestern Institute on Complex Systems and Institute for Policy Research
- PhD Political Science, Michigan, 2017
- Using R since 2009
- I wish I'd learned the tidyverse first!

This workshop:

- Today, 9 - 12
- Wednesday, 9-12, same place
- Practice sessions, Today and Wednesday, 1:30pm, Mudd Library, Small Classroom, Room 2124

MATERIALS: https://github.com/cskovron/tidyverse-workshop

Prerequisites

```
install.packages(c('tidyverse','broom','infer','nycflights13'))
```

The tidyverse: Let's get organized

A new and changing set of tools

- An “opinionated” set of R packages
- A dogma about “tidy” data organization and code
- Rigid but the best current approach for a lot of reasons

Core packages

Core packages

- `ggplot2` - visualization
- `dplyr` - data wrangling
- `tidyr` - data cleaning and tidying
- `readr` - read in data
- `purrr` - functional programming
- `tibble` - tidy version of data frames
- `stringr` - strings and text
- `forcats` - better living through factors

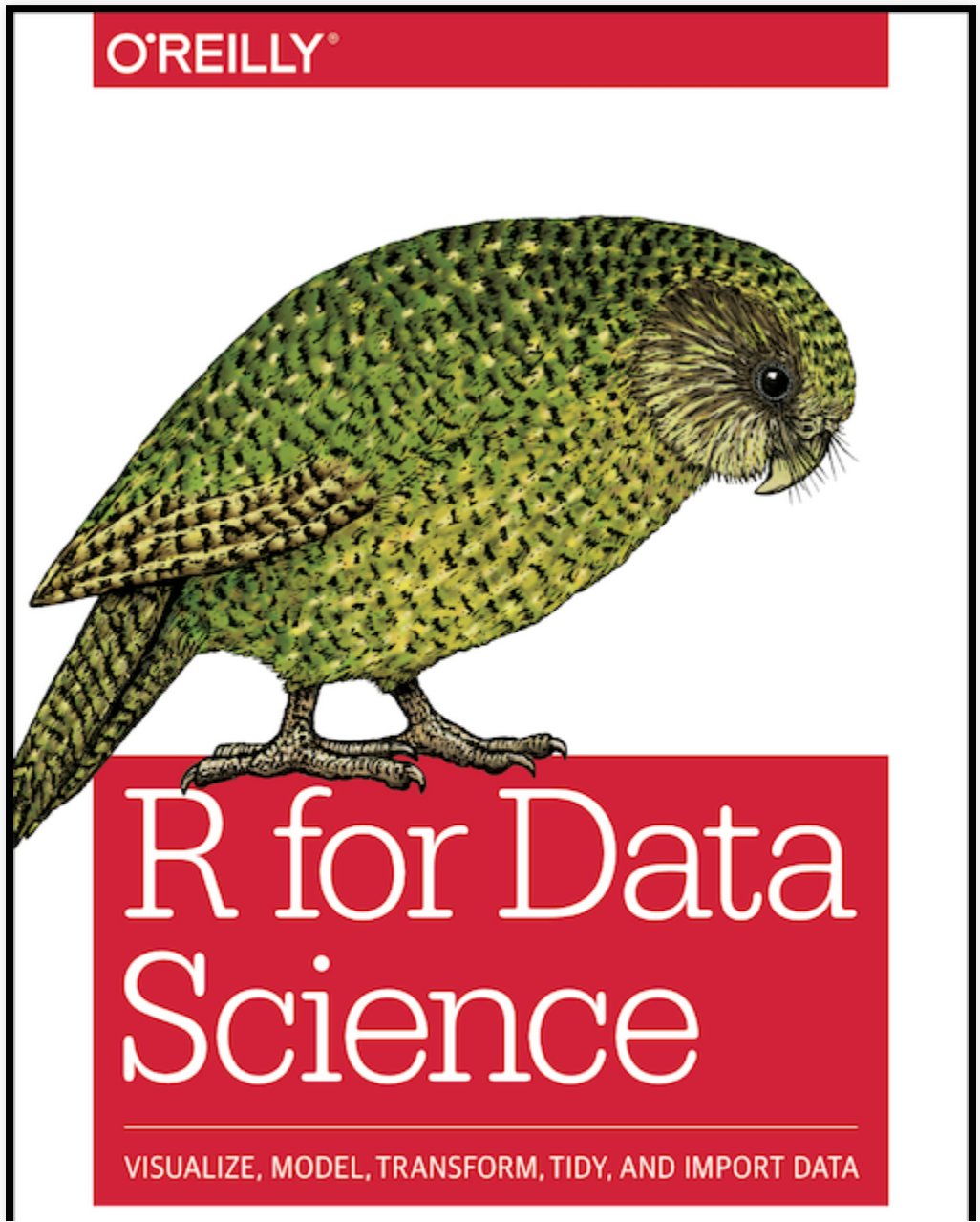
Others that play well with tidyverse (partial list)

- `broom` - tidy model outputs
- `lubridate` - work with dates and times
- `httr` - scrape web APIs
- `rvest` - web scraping

The tidyverse is constantly evolving

- It's an official RStudio product, so watch their website
- Easy install and load:
- `install.packages('tidyverse')`
- `library(tidyverse)`

The holy text of the tidyverse



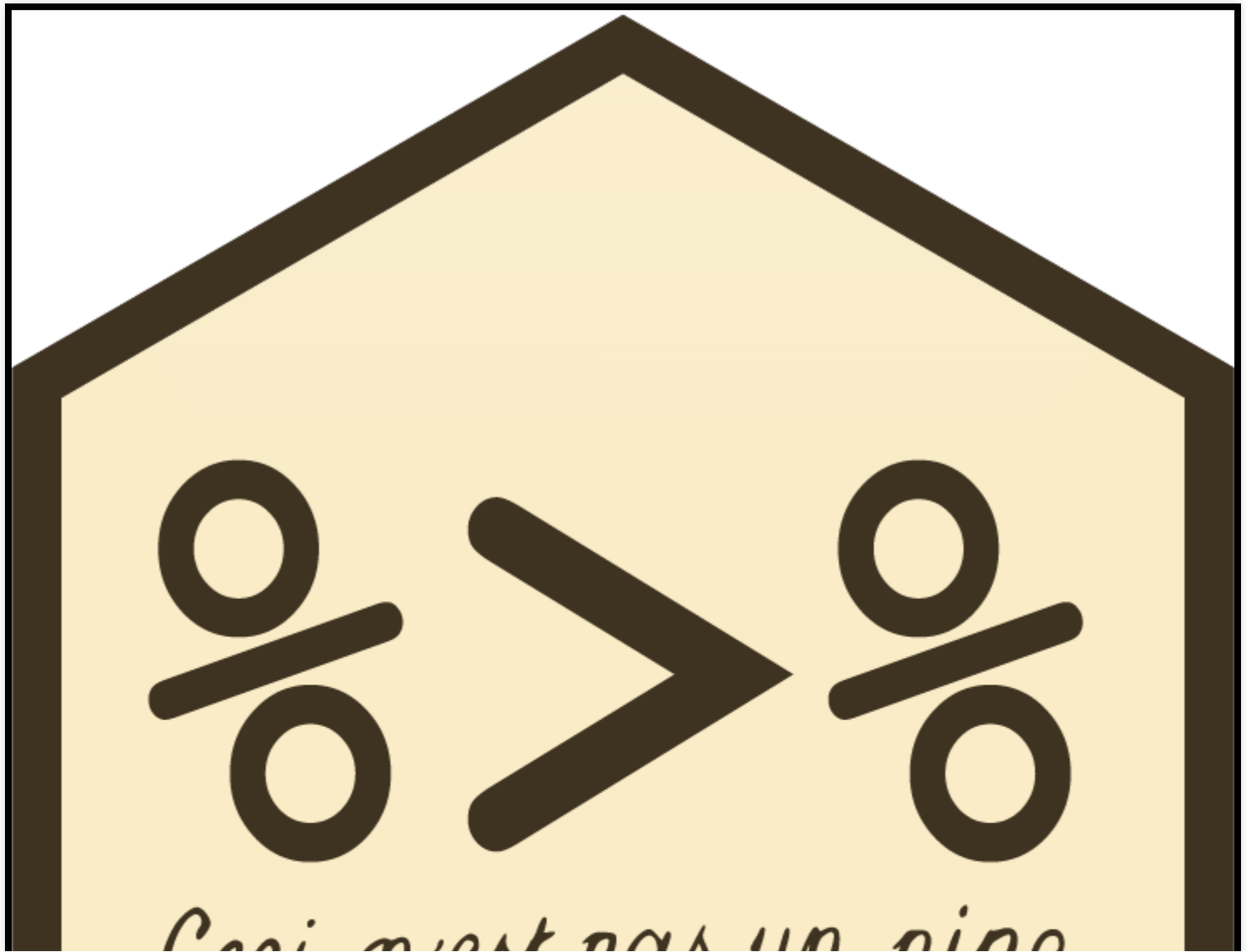
R for Data Science is freely available online. I encourage you to read all of it. We will only cover what I think are the most important chapters in these two days. I've blatantly stolen major parts of the workshop from this book. It's great!

This workshop is a partial walkthrough of R4DS

- Adding in additional lessons on some of my favorite tidy analysis tasks
- Focus on `dplyr` for the first day, for most users the most important package to learn
- Will sprinklin in some `ggplot2` on the second day, but it's the oldest and most well-known tidy package and has its own workshop. Learn it!

Verbs and pipes

This wacky thing is the pipe.




Ceci n'est pas un piège.

The pipe

The pipe chains together commands so that you can do a bunch of manipulations in order to a data frame without calling it repeatedly or nesting many parentheses

Those are gnarly characters to type, but fortunately RStudio has a keyboard shortcut, CMD+SHIFT+M (CTRL+SHIFT+M on Windows) to quickly type a pipe for you.

The pipe

A good mantra for effectively using the pipe in your workflow is:

"Dataframe first, dataframe once"

Ugly code without the pipe

If you don't use the pipe, your code, *even if you use dplyr*, can end up with a horrid-looking and unintiutive structure of nested parentheses and objects called in an unintuitive order:

```
car_data <-
  transform(aggregate(. ~ cyl,
                      data = subset(mtcars, hp > 100),
                      FUN = function(x) round(mean(x, 2))),
            kpl = mpg*0.4251)
```

Better code with the pipe

```
car_data <-
  mtcars %>%
  subset(hp > 100) %>%
  aggregate(. ~ cyl, data = ., FUN = . %>% mean %>% round(2)) %>%
  transform(kpl = mpg %>% multiply_by(0.4251)) %>%
  print
```

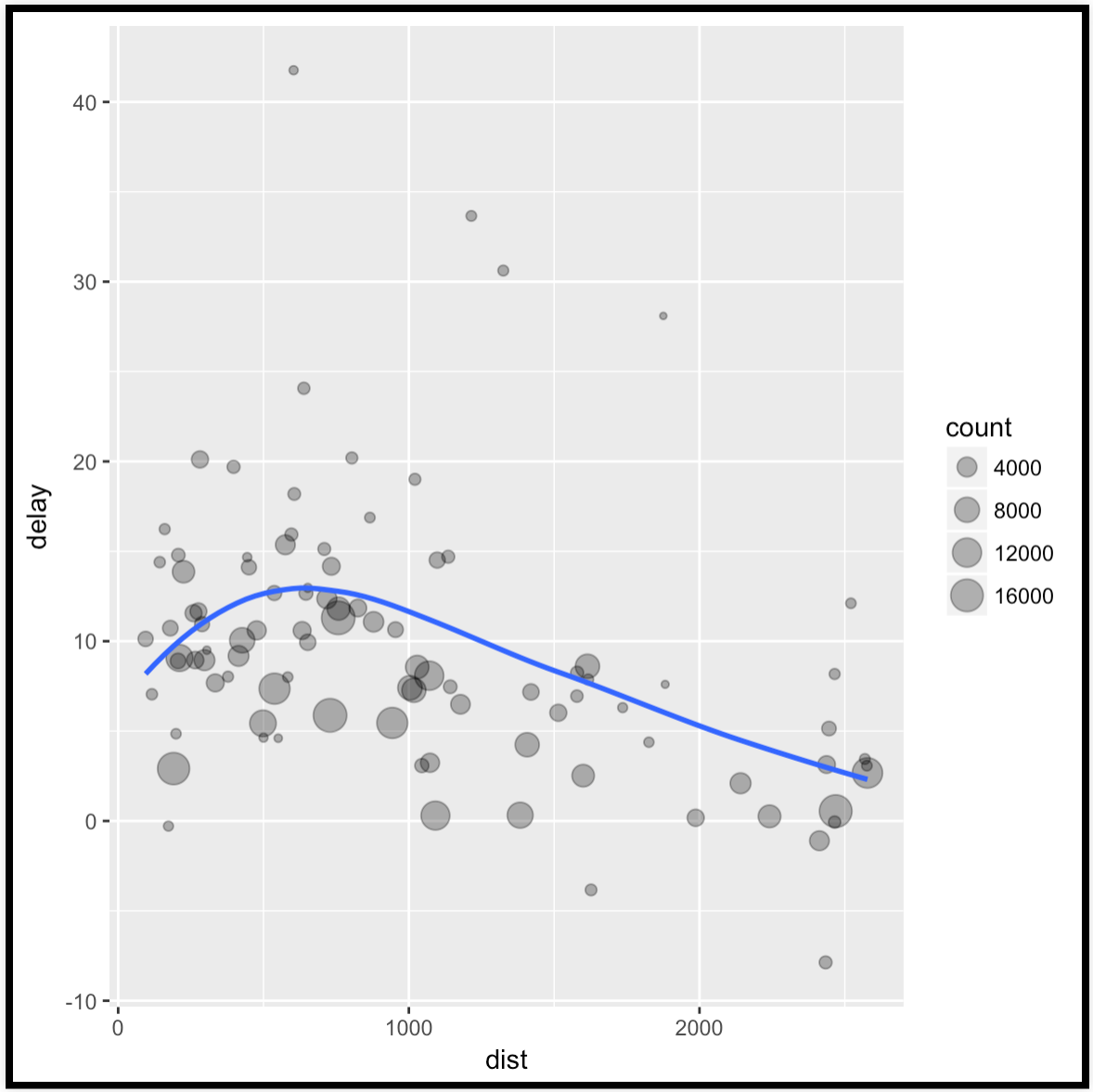
Unpiped operation

```
by_dest <- group_by(flights, dest)
delay <- summarise(by_dest,
  count = n(),
  dist = mean(distance, na.rm = TRUE),
  delay = mean(arr_delay, na.rm = TRUE)
)
delay <- filter(delay, count > 20, dest != "HNL")
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

```
# It looks like delays increase with distance up to ~750 miles
# and then decrease. Maybe as flights get longer there's more
# ability to make up delays in the air?
ggplot(data = delay, mapping = aes(x = dist, y = delay)) +
  geom_point(aes(size = count), alpha = 1/3) +
  geom_smooth(se = FALSE)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```


40
30
20
10
0
-10
delay
0
1000
2000
dist
count
4000
8000
12000
16000

Break down your work into steps

There are three steps to prepare this data:

1. Group flights by destination.
2. Summarise to compute distance, average delay, and number of flights.
3. Filter to remove noisy points and Honolulu airport, which is almost twice as far away as the next closest airport.

Cleaner code with the pipe

There's another way to tackle the same problem with the pipe, `%>%`:

```
delays <- flights %>%
  group_by(dest) %>%
  summarise(
    count = n(),
    dist = mean(distance, na.rm = TRUE),
    delay = mean(arr_delay, na.rm = TRUE)
  ) %>%
  filter(count > 20, dest != "HNL")
```

Pipe into `ggplot()`

Hopefully you're either already familiar with `ggplot2` or are taking the workshop, because that's the one tidyverse package we won't explore in enough detail. You can use the pipe to pass data from a piped set of say, `dplyr` commands into a call to `ggplot()`. Just replace the data argument of the call to `ggplot` with a `.`.

```
g = ncs %>%
  group_by(party,ideo_relative_to_legislature) %>%
  tally  %>%
  group_by(party) %>%
  mutate(pct=(100*n)/sum(n)) %>%
  filter(!is.na(party)) %>%
  filter(!is.na(ideo_relative_to_legislature)) %>%
  ggplot(., aes(x=ideo_relative_to_legislature, y = pct))+geom_col()+face
  theme_bw()+
  xlab('Candidate perception of self relative to legislature') +
  ylab('Percent of party respondents') +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("Candidates' perceptions of their ideology relative to the legi
```

Outline of the rest of the workshop:

- Today: data transformation with dplyr
- Today/Wednesday: wrangle and tidy data with tibbles, tidyr and readr
- Wednesday: an incomplete introduction to working with strings, dates, and factors
- Finishing up: Tidying your modeling with infer and broom

Two sets of exercises

- I pillaged all the exercises from the relevant chapters of R for Data Science. We'll do a few of these during lectures, but they can mostly be done in the afternoon practice sessions or on your own time. Find them in the 'r4dsexercises' directory of the github repo.
- RStudio has a separate "Master the Tidyverse" class that has a lot of exercises on various aspects of the tidyverse. Some of the exercises are on parts of the tidyverse we won't cover, but you should be able to get most of them. These are in the master-tidyverse-exercises folder of the repo. Solutions are there too, don't peek!

Follow along

I’m primarily working off of the slides. They are available as HTML and PDF, but you may find it useful to open the .Rmd files that the slides are written in and follow along. You’ll be able to run the code chunks and to see how the slides were written. There are some additional lecture notes in the corelecturenotes folder of the github repo, but these are a little thin, and I’d suggest you go straight to R for Data Science for further reading.