

# R PROGRAMMING

**for data visualization**

**Brooke Anderson**

Colorado State University

March 27, 2019

# Today's goals

LEARN principles of a key R plotting framework

UNDERSTAND what R can do for visualizations

KNOW what to do next to learn more

# Today's plan

**PLOT**

**MAP**

**INTERACT**

**REPORT**

**TIDY**

# PREREQUISITES

Setting up for success

[R vs. RStudio]

[R packages]

# Installing R packages

```
install.packages("readr")
```

Use the **install.packages** function to install an R package to your computer.

# Loading R packages

```
library("readr")
```

Use the **library** function to load an R package that is installed on your computer.



# Hello

my name is

<-

Assign an object a name with R's **gets arrow**

# Assignment with the gets arrow

You want to read in the “daily\_fatalities.csv” file, which is in the “data” subdirectory.

# Assignment with the gets arrow

Assign the filepath of this file to the R object named **fatalities\_files**.

Reference that object to read in the data and assign it to the R object named **daily\_fatalities**.

```
fatalities_file <- "data/daily_fatalities.csv"  
daily_fatalities <- read_csv(fatalities_file)
```

# Hurricane Irma



**NWS Key West** 

@NWSKeyWest

Follow



\*\*\*THIS IS AS REAL AS IT GETS\*\*\*

\*\*\*NOWHERE IN THE FLORIDA KEYS  
WILL BE SAFE\*\*\*

\*\*\*YOU STILL HAVE TIME TO  
EVACUATE\*\*\*

Please RT. [#Irma](#)

# Hurricane Irma

## Navy evacuates over 5,000 personnel from Florida base ahead of Hurricane Irma

Published time: 6 Sep, 2017 05:08

Edited time: 7 Sep, 2017 10:56

[Get short URL](#)



© Joe Skipper © Reuters

[← RESEARCH & DATA](#)

# Fatality Analysis Reporting System (FARS)

<https://www.nhtsa.gov/research-data/fatality-analysis-reporting-system-fars>

# Example data

daily\_fatalities

```
## # A tibble: 28 x 4
```

```
##   date          week weekday  fatalities
```

```
##   <date>        <dbl> <chr>         <dbl>
```

```
## 1 2017-08-27      35 Sunday           4
```

```
## 2 2017-08-28      35 Monday           5
```

```
## 3 2017-08-29      35 Tuesday           6
```

```
## 4 2017-08-30      35 Wednesday          6
```

```
## 5 2017-08-31      35 Thursday           6
```

```
## 6 2017-09-01      35 Friday            9
```

```
## 7 2017-09-02      35 Saturday           8
```

```
## 8 2017-09-03      36 Sunday          15
```

```
## 9 2017-09-04      36 Monday            7
```

```
## 10 2017-09-05     36 Tuesday           8
```

```
## # ... with 18 more rows
```

# PLOT

R's **ggplot2** framework for plotting



[Layering for ggplot]

# Spot the differences

# geoms and their aesthetics

# scales

# labels

# themes

```
irma_week_accs <- fl_accidents %>%  
  group_by(fips) %>%  
  summarize(fatals = sum(fatals))
```

```
irma_accs <- fl_accidents %>%  
  full_join(irma_week_accs, by = c("GEOID" = "fips")) %>%  
  mutate(fatals = ifelse(is.na(fatals), 0, fatalities))
```

# [Live coding example]

```
fl_accidents <- fl_accidents %>%  
  st_as_sf(coords = c("longitud", "latitude")) %>%  
  st_set_crs(st_crs(st_read(dsn, layer, ...)))
```

```
irma_track <- st_read("data/al112017_best_track",  
                      layer = "al112017_lin") %>%  
  st_transform(crs = st_crs(irma_accs))
```

# MAP

R's **sf** framework for mapping



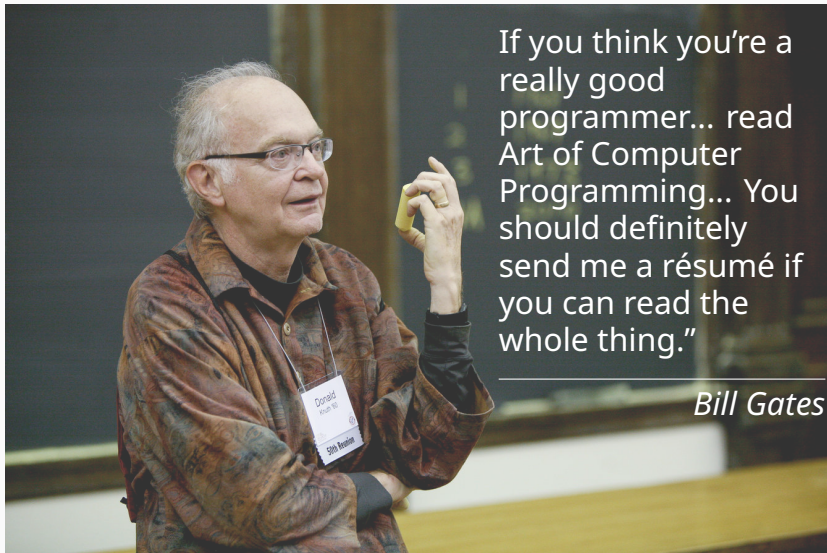
# INTERACT

R's **htmlwidgets** framework for interacting

# REPORT

R's **RMarkdown** framework for reporting

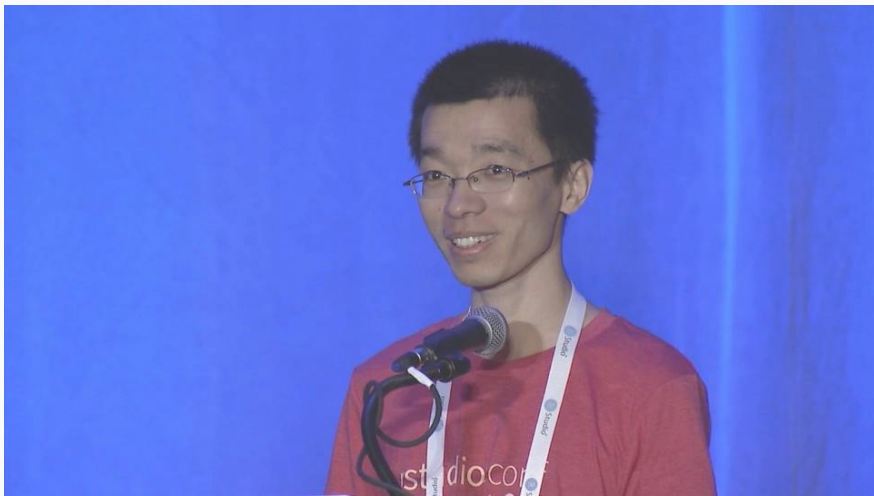
# Donald Knuth



If you think you're a really good programmer... read Art of Computer Programming... You should definitely send me a résumé if you can read the whole thing."

*Bill Gates*

# Yihui Xie



# WYSISYG

**What You See Is What You Get**

Text of the report, with Markdown **format markers**.

```
```{r}
```

```
number_one <- 1
```

```
number_one
```

```
```
```

More text, *also* with Markdown format markers.

And a list:

- \- Item 1

- \- Item 2

Text of the report, with Markdown **format markers**.

```
number_one <- 1  
number_one
```

```
## [1] 1
```

More text, *also* with Markdown format markers. And some items:

- Item 1
- Item 2

```
irma_week_accs <- fl_accidents %>%  
  group_by(fips) %>%  
  summarize(fatals = sum(fatals))
```

```
irma_accs <- fl_accidents %>%  
  full_join(irma_week_accs, by = c("GEOID" = "fips")) %>%  
  mutate(fatals = ifelse(is.na(fatals), 0, fatalities))
```

# [Live coding example]

```
fl_accidents <- fl_accidents %>%  
  st_as_sf(coords = c("longitud", "latitude")) %>%  
  st_set_crs(st_crs(st_read(dsn, layer, ...)))
```

```
irma_track <- st_read("data/al112017_best_track",  
                      layer = "al112017_lin") %>%  
  st_transform(crs = st_crs(irma_accs))
```



# TIDY

R's **tidyverse** framework for tidying

```
irma_week_accs <- fl_accidents %>%  
  group_by(fips) %>%  
  summarize(fatals = sum(fatals))
```

```
irma_accs <- fl_counties %>%  
  full_join(irma_week_accs, by = c("GEOID" = "fips")) %>%  
  mutate(fatals = ifelse(is_na(fatals), 0, fatalities))
```

# [Live coding example]

```
fl_accidents <- fl_accidents %>%  
  st_as_sf(coords = c("longitud", "latitude")) %>%  
  st_set_crs(st_crs(st_read(dsn, layer, ...)))
```

```
irma_track <- st_read("data/al112017_best_track",  
                     layer = "al112017_lin") %>%  
  st_transform(crs = st_crs(irma_accs))
```

The design is clean

The rules are simple

The code is extensible

# Open Source Fonts

**This is MONTSERRAT**

This is NOTO SANS

This is Lato (light)

This is inconsolata

THIS IS ALEGREYA SANS SMALL CAPS

# Color Palette



# **BIG BOLD TEXT**

but background color does not work

