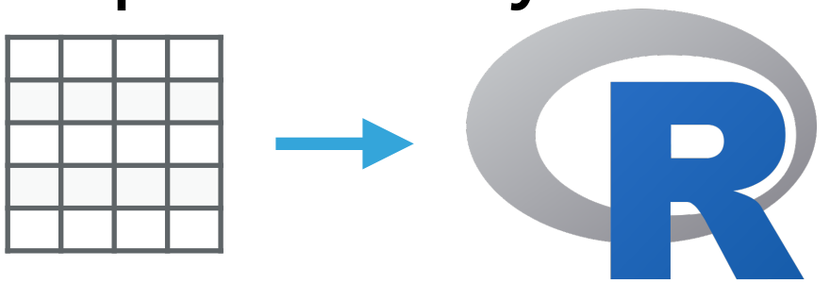
ASMS 2019
ANNUAL
CONFERENCE
WORKSHOP

SOCALBIOINFORMATICS.COM

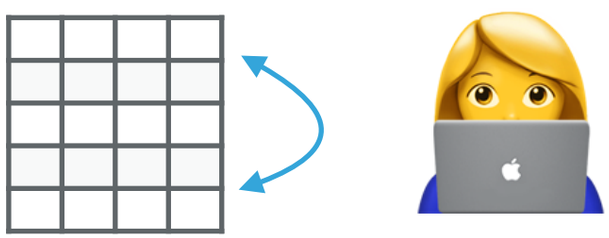
DATA MANIPULATION & ANALYSIS WITH R & THE TIDYVERSE

THREE MAIN TASKS TO MASTER

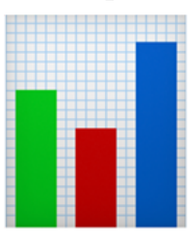
1. Import/read your data into R



2. Clean, tidy, manipulate, analyze your data



3. Export the output of your work



GETTING YOUR DATA INTO R

A large variety of file formats can be read with R

- Formatted text (e.g. csv, tsv):
  `read.csv` (base), `read_csv` (readr)

- Excel files
  `read.xlsx` (xlsx), `read_excel` (readxl)

- Lots of others
  jsonlite, xml2, httr, haven,
  Bioconductor packages (e.g. for mzML)

TIPS FOR BEGINNERS

- Formatted text files are easiest to start with (e.g. csv's)
- R will read in as a *data frame*
- Tabular data should have column names in the first row

Should be meaningful too

first_name	last_name	age	fav_food
Sara	Smith	42	Tacos
Cindy	Kline	24	Cake
John	Snow	32	Meat
Dany	Targaryen	36	Soup

- Check for crazy formatting that can mess things up; look for errors that pop-up

READING DATA EXAMPLE

1. Make an R project (see *Getting Started with R & RStudio*)

2. Place your input data *inside* your root project folder
   let's assume it's called: my_data.csv

3. Read your data into R

base R

```
dat <- read.csv("my_data.csv", stringsAsFactors = FALSE)

# do stuff with dat...
```

tidyverse

```
library(readr)
dat <- read_csv("my_data.csv")

# do stuff with dat...
```

Preferred way when working in the tidyverse

DATA MANIPULATION & ANALYSIS BASICS WITH THE TIDYVERSE

- Using the tidyverse is a great, modern way to learn R
- **tidyverse**: a set of R packages that revolve around the concept of tidy data and facilitate/streamline all steps of the data analysis process
- tidy data: a structured data format where
  - data is in a tabular format
  - rows represent the observational units
  - columns represent the variables being measured

DATA MANIPULATION & ANALYSIS BASICS WITH THE TIDYVERSE

A tidy data example

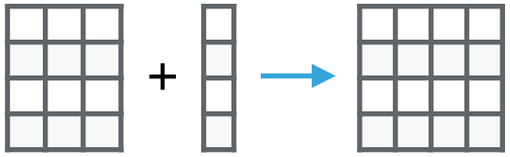
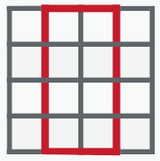
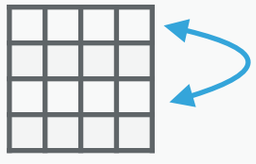
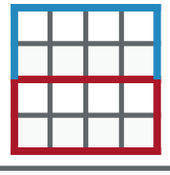
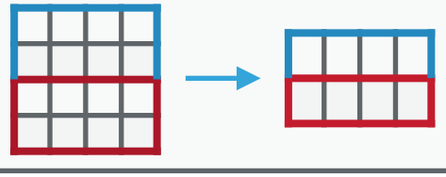
Each column is a variable being measured

Each row is a person
(the "unit" being measured)

first_name	last_name	age	fav_food
Sara	Smith	42	Tacos
Cindy	Kline	24	Cake
John	Snow	32	Meat
Dany	Targaryen	36	Soup

THE TIDYVERSE FOUNDATION: THE DPLYR PACKAGE

- Dplyr allows you to easily create powerful data analysis pipelines utilizing fundamental data manipulation *verbs*

`mutate`	Add a column	
`select`	Pick specific columns	
`filter`	Subset to specific rows	
`arrange`	Reorder/sort rows	
`group_by`	Group subsets	
`summarize`	Perform aggregated calculations	

THE TIDYVERSE FOUNDATION: THE DPLYR PACKAGE

- You can *pipe* together data, verbs, and other functions to create analysis pipelines that are easy to write and read

```
# dplyr examples using the 'storms' data set
library(dplyr)

# We'll use the dplyr built in 'storms' data set
storms

# Example dplyr pipeline
mean_stats <- storms %>%
  group_by(year) %>%
  summarize(mean_wind_speed = mean(wind),
            mean_air_pressure = mean(pressure)) %>%
  arrange(desc(mean_wind_speed))

mean_stats
```

This is the pipe operator takes left side as input into the right side

interactive example [code/01_dplyr_storms_example.R]

COMMUNICATE YOUR WORK

- After you've done your analysis, its time to *communicate* your work by (for example)
  - new, processed data set
  - summary results table
  - figures, plots
  - report

- R is particularly well suited for making all of these things

WRITING DATA EXAMPLE

1. Complete your analysis, with your output data in a *data frame* (e.g. mean_stats from the storm example)

2. Save your data frame as a .csv file
(easy to read with other tools, e.g. Excel)

base R

```
# Do your analysis here...  output data is 'final_data'

write.csv(final_data, "final_data.csv", row.names = FALSE)
```

tidyverse

```
library(readr)

# Do your analysis here...  output data is 'final_data'
dat <- write_csv(final_data, "final_data.csv")
```

PLOTTING DATA WITH GGPLOT2

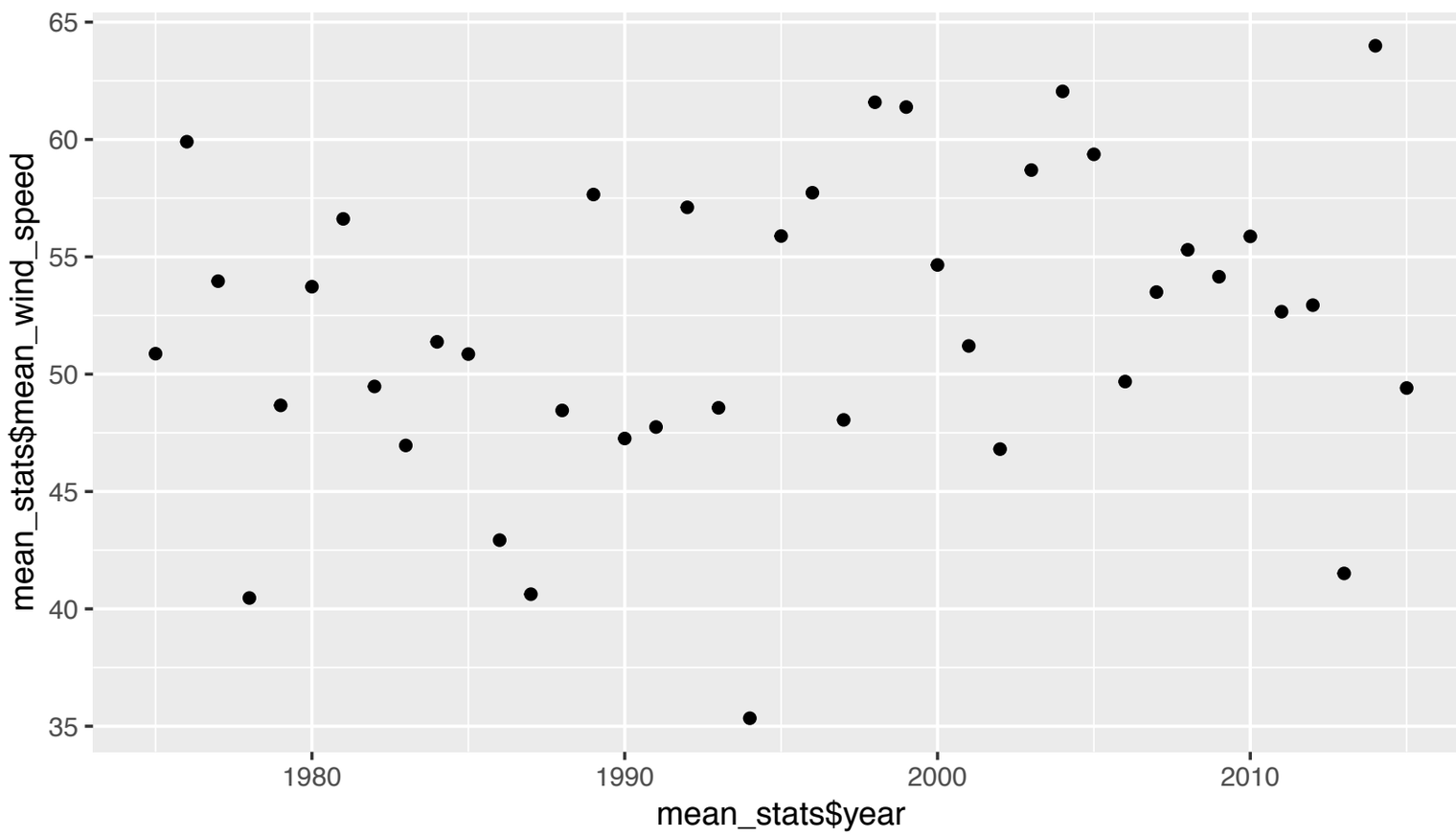
- Ggplot2 is an *amazing* plotting package
- There is a learning curve, but gives you lots of flexibility and power to make (almost) any (static) plot you want
- Allows you to map data (or data summaries) to visual elements, can be built up layer by layer
- Will take some time to learn, but it's absolutely worth it

GGPLOT2 FIRST STEPS: QPLOT

- `qplot` is a ggplot2 function for **q**uickly making **plot**s

```
library(ggplot2)

# ...the storms example code goes here
qplot(mean_stats$year, mean_stats$mean_wind_speed)
```

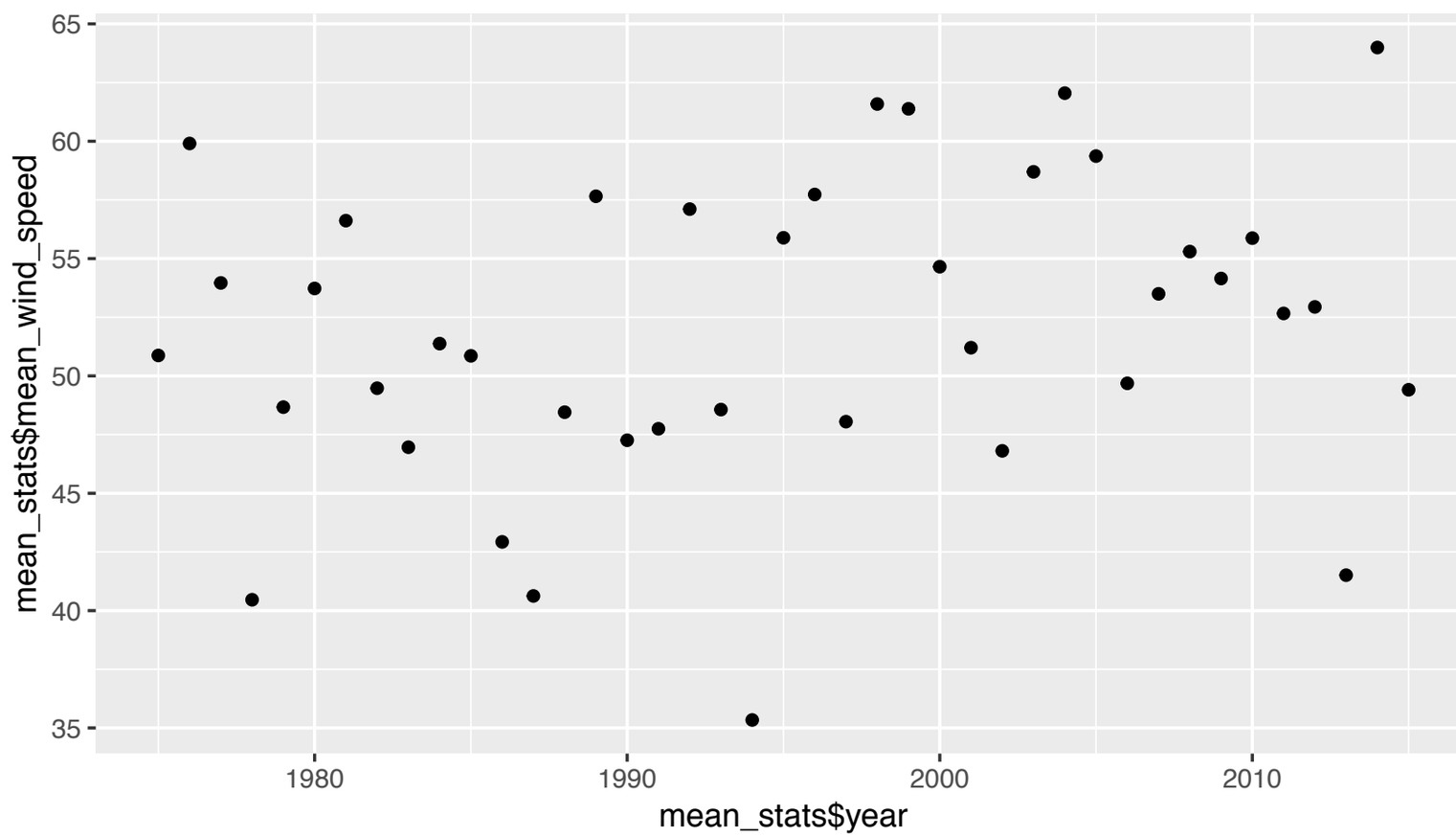


GGPLOT FIRST STEPS: QPLOT

- `qplot` makes a reasonable plot based on the input

Two continuous variables x & y gives a scatter plot

```
qplot(mean_stats$year, mean_stats$mean_wind_speed)
```

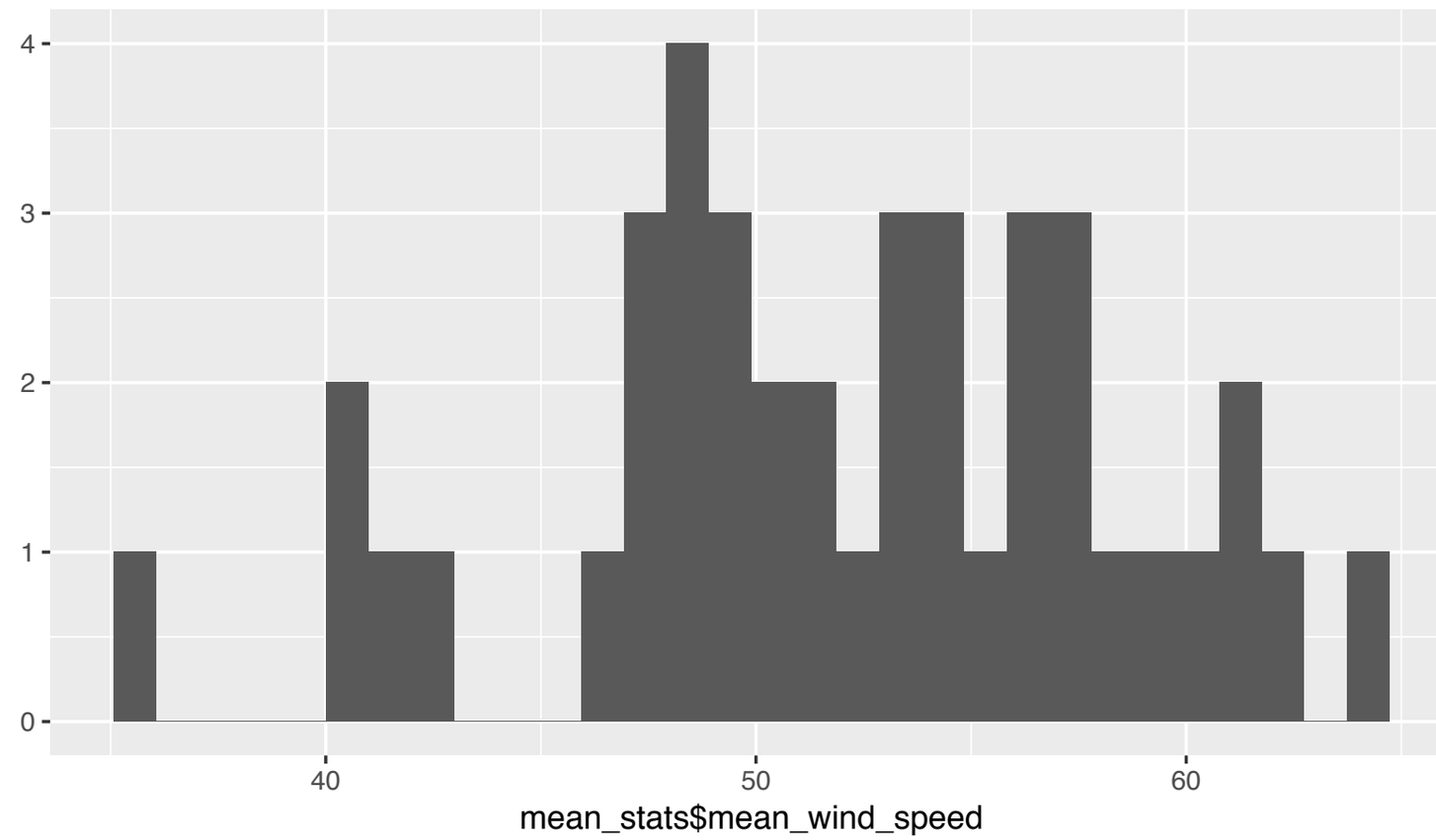


GGPLOT FIRST STEPS: QPLOT

- `qplot` makes a reasonable plot based on the input

One continuous variable x gives a histogram

```
qplot(mean_stats$mean_wind_speed)
```



THE REAL POWER COMES WITH THE GGPLOT FUNCTION

- The ggplot function give you complete flexibility in making just about any plot you want
- You'll need to
  - use *data* for input
  - specify which **geom**etric objects you'll use to represent the data (e.g. points, lines, boxes, etc.)
  - specify how the data map to the **geom**etric objects via **aes**thetics
  - potentially use **stat**istical summarizations of your data

EXAMPLES WITH GGPLOT

- Let's rewrite the qplot code using ggplot

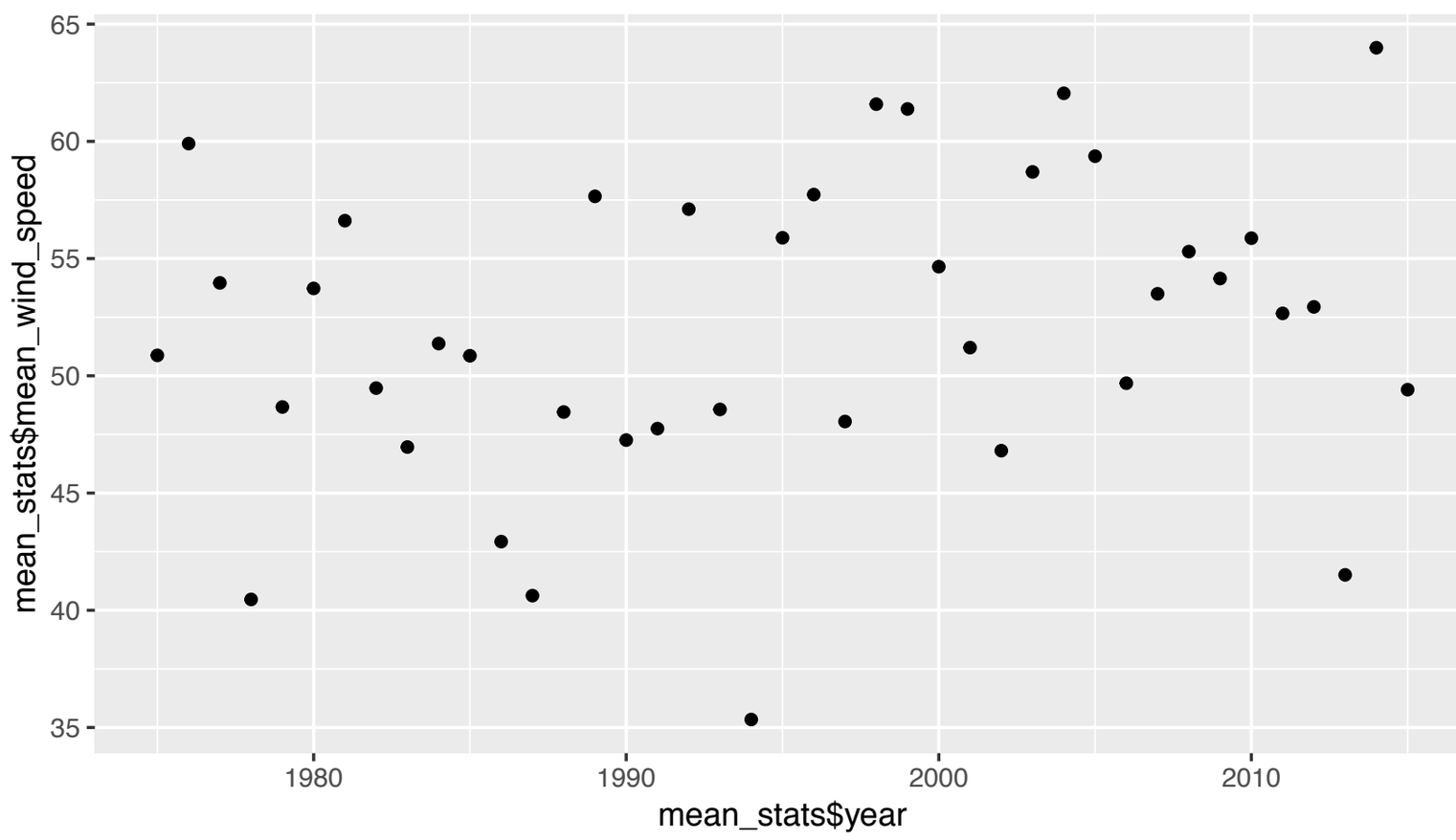
input data

mapping variables to x & y

add a layer

```
ggplot(mean_stats, aes(x = year, y = mean_wind_speed)) +
  geom_point()
```

plotting as points

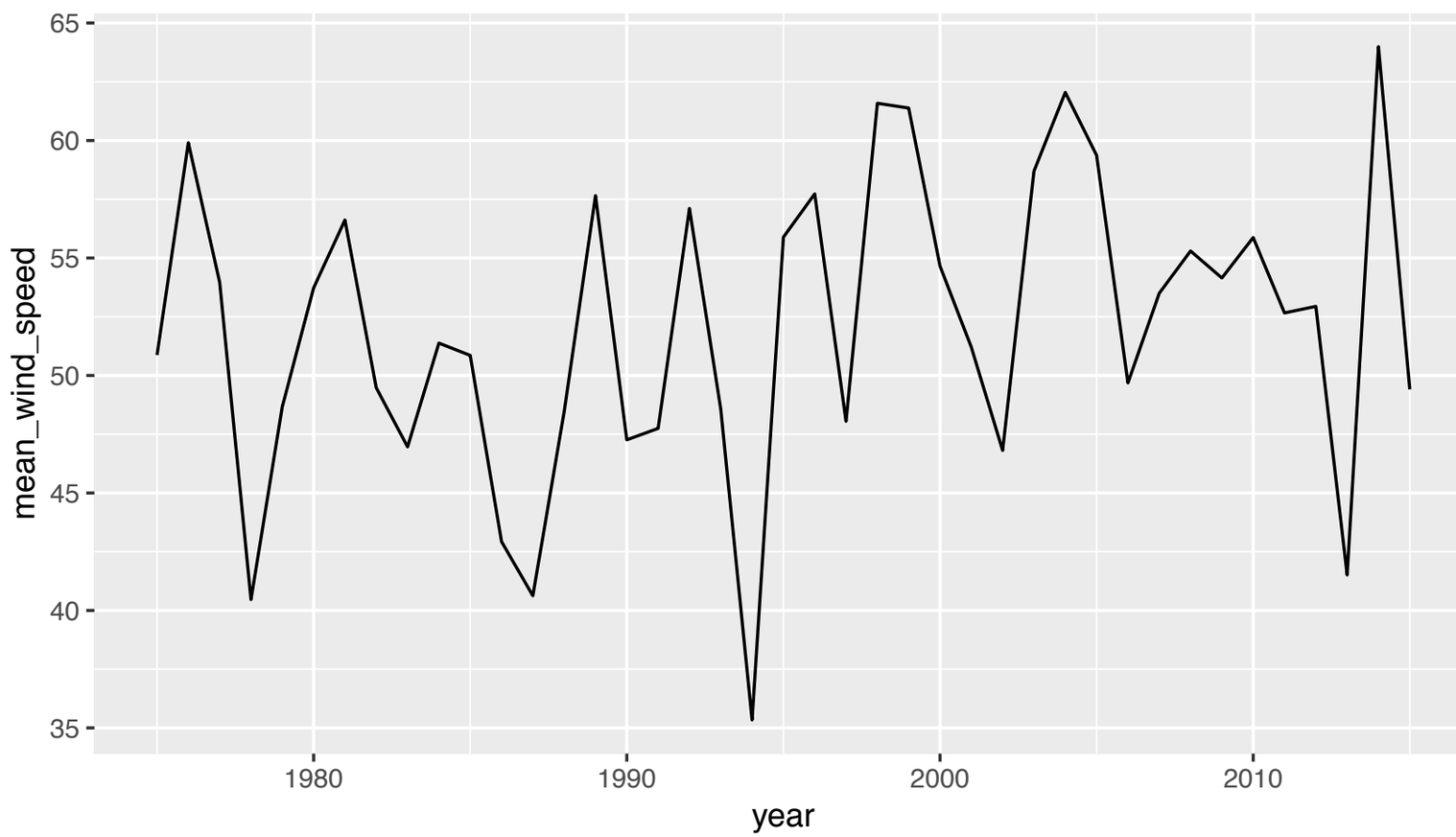


EXAMPLES WITH GGPLOT

- How about lines instead?

```
ggplot(mean_stats, aes(x = year, y = mean_wind_speed)) +
  geom_line()
```

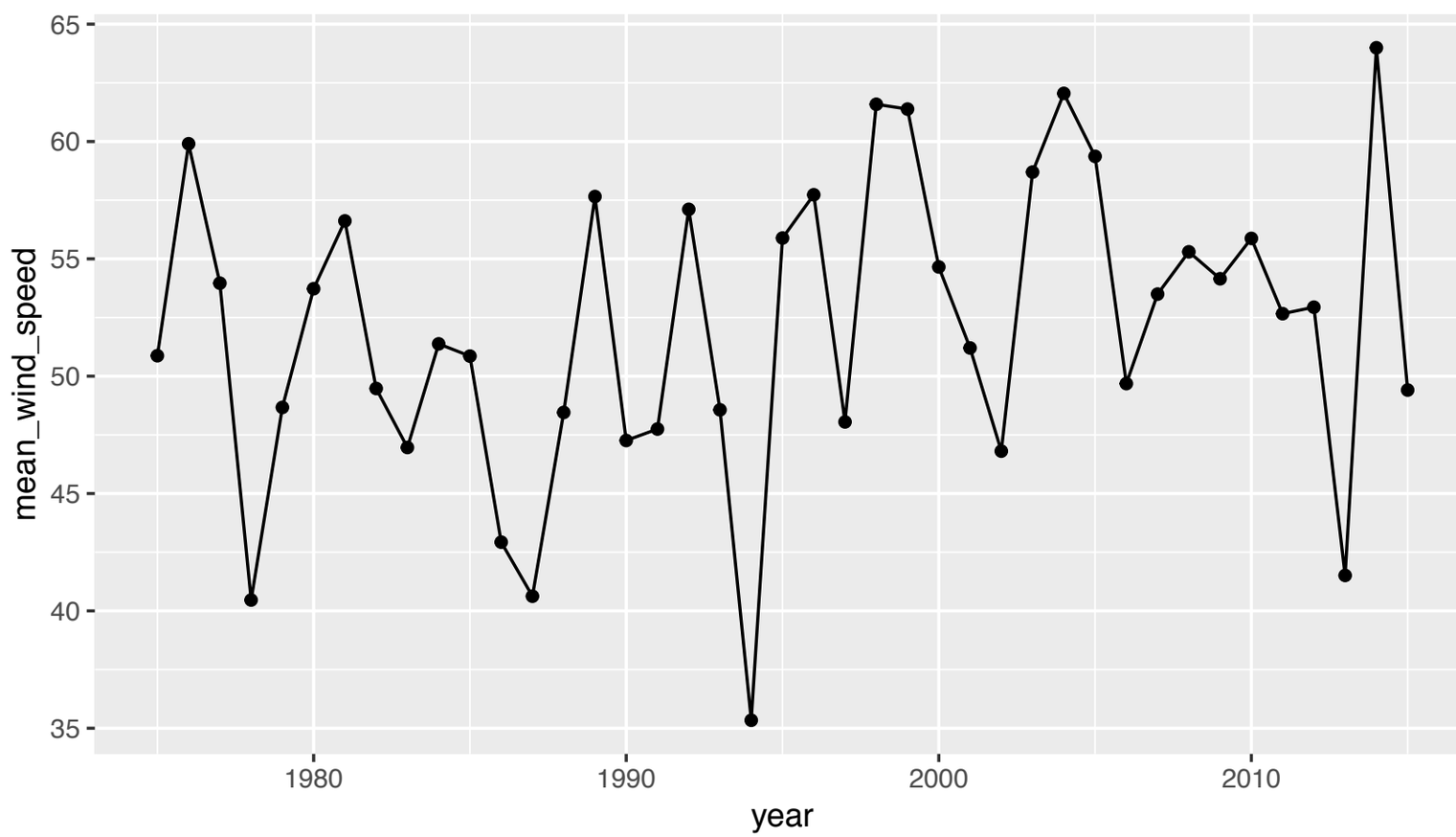
only difference



EXAMPLES WITH GGPLOT

- Or both?

```
ggplot(mean_stats, aes(x = year, y = mean_wind_speed)) +
  geom_point() +
  geom_line()
```



interactive example [02_ggplot2_storms_example.R]

RMARKDOWN IS GREAT FOR REPORTING

- RMarkdown is a combination of R + Markdown
- Allows you to write traditional text along with R code
- Can embed analysis code into a report document so the code lives with the explanatory text
- Don't have time to cover it but check out https://bookdown.org/yihui/rmarkdown/
- You can make:
  reports, presentations (slides), books, websites, posters, …