Data Visualization

Lesser known intro

Amy Tzu-Yu Chen

2020-04-29

About Me

- UCLA'16, Statistics
- Data Scientist at System1 & Computational Linguistics MS Student at University of Washington
- Happy R user since STAT 20
- Find me at amy17519 @ Twitter, Github, and LinkedIn

Why is this a "lesser known" intro

You can easily find tutorials if you google "how to use "blah blah" visualization library"

(🤭 and just copy and paste code! It works!)

You can even build amazing graphs *without coding* using Tableau etc

(🤭 and they look nice! Nicer than my ggplots sometimes!)

🤔… However, having a deeper understanding on visualization tools and process is a great asset for data practitioners

🤩 Know behind-the-library design philosophy ➡ helps you understand a diverse range of graphics and powerful tools faster

🤩 Practice visualization process ➡ inform yourself, then educate your audience

Agenda

- Grammar of Graphics
- Strategy
- Visualization Process
    - Making *exploratory* Graph*s*
    - Making a *confirmatory* Graph
- Toolbox
- Resources

Grammar of Graphics

The Beauty of Grammar of Graphics

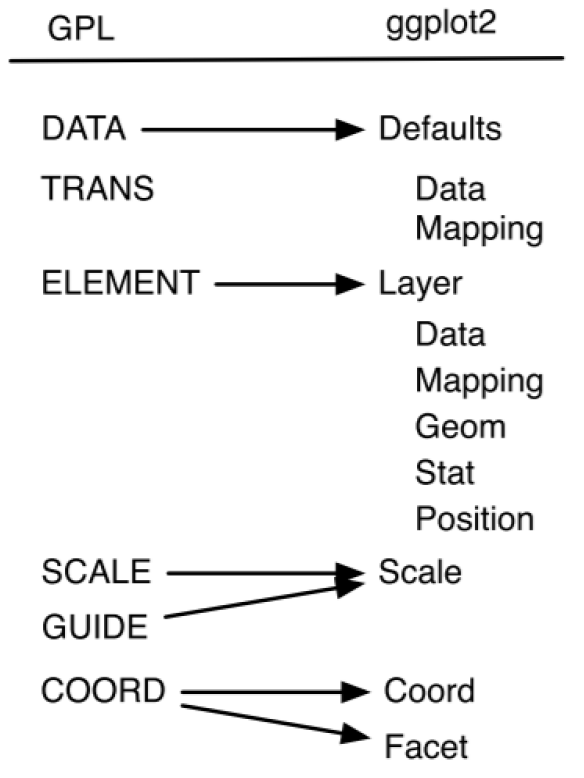
In languages, grammar keeps things in order.

If you know some grammar, you don't need to know all the vocabularies to speak.

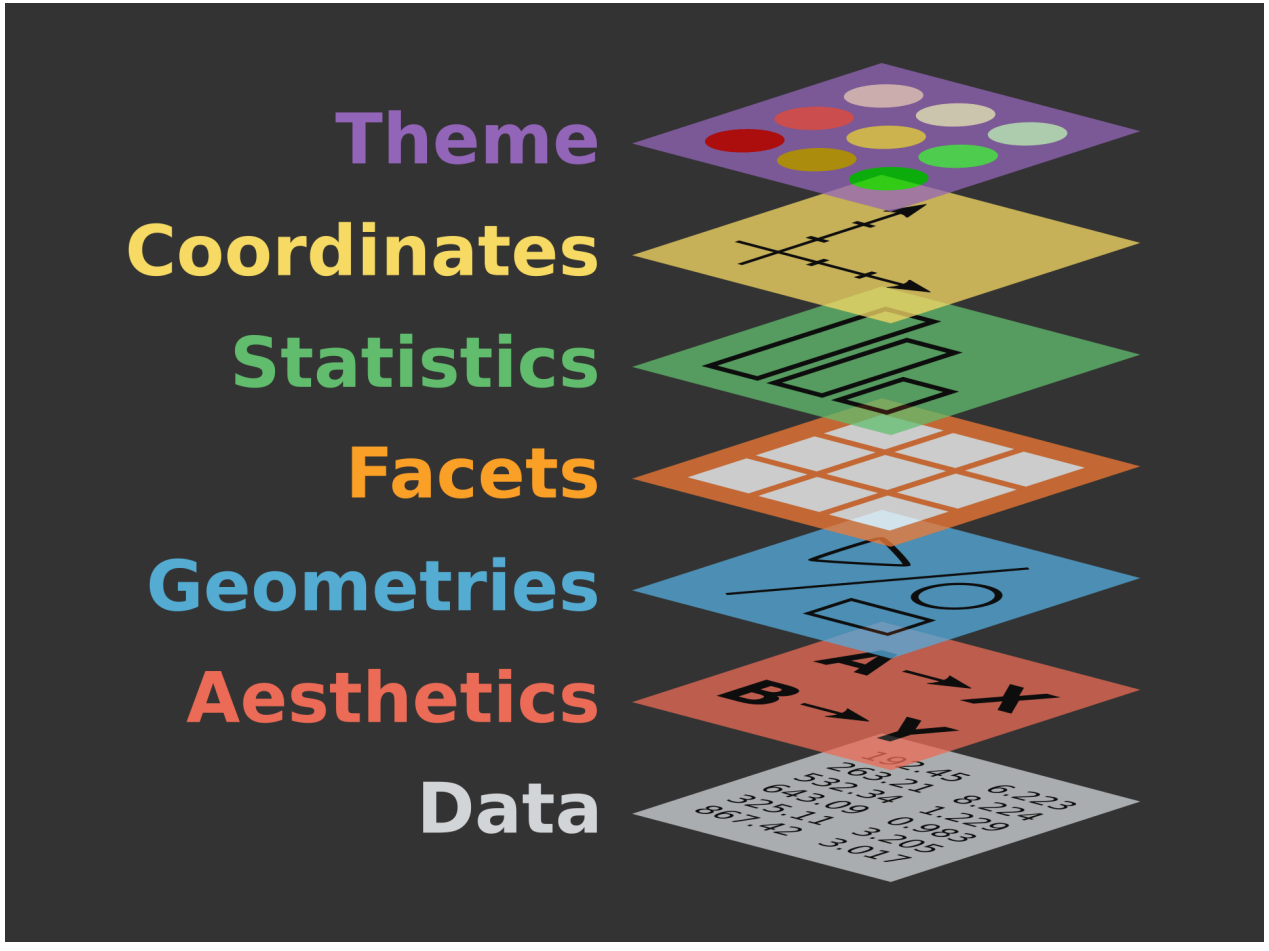
If you know some grammar of graphics, you don't need to know all coding syntax or graph types to make an informative graph

History - Grammar of Graphics

- Late 1990s, the concept was introduced by Leland Wilkinson. See The Grammar of Graphics 2nd Edition, 2005.
- 2000s: Hadley Wickham built the R visualization library ggplot2 based on grammar of graphics with modifications. He also published A Layered Grammar of Graphics, 2010.
- Many applications in visualization libraries/projects in different languages.



Grammar of Graphics - Components of Grammar



Example Data

```
# df_measles comes from dataset dslabs::us_contagious_diseases
str(df_measles)
```

```
## Classes 'data.table' and 'data.frame':    3825 obs. of  6 variables:
##  $ disease        : Factor w/ 7 levels "Hepatitis A",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ state          : Factor w/ 51 levels "Alabama","Alaska",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ year           : num  1928 1929 1930 1931 1932 ...
##  $ weeks_reporting: num  52 49 52 49 41 51 52 49 40 49 ...
##  $ count          : num  8843 2959 4156 8934 270 ...
##  $ population     : num  2589923 2619131 2646248 2670818 2693027 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

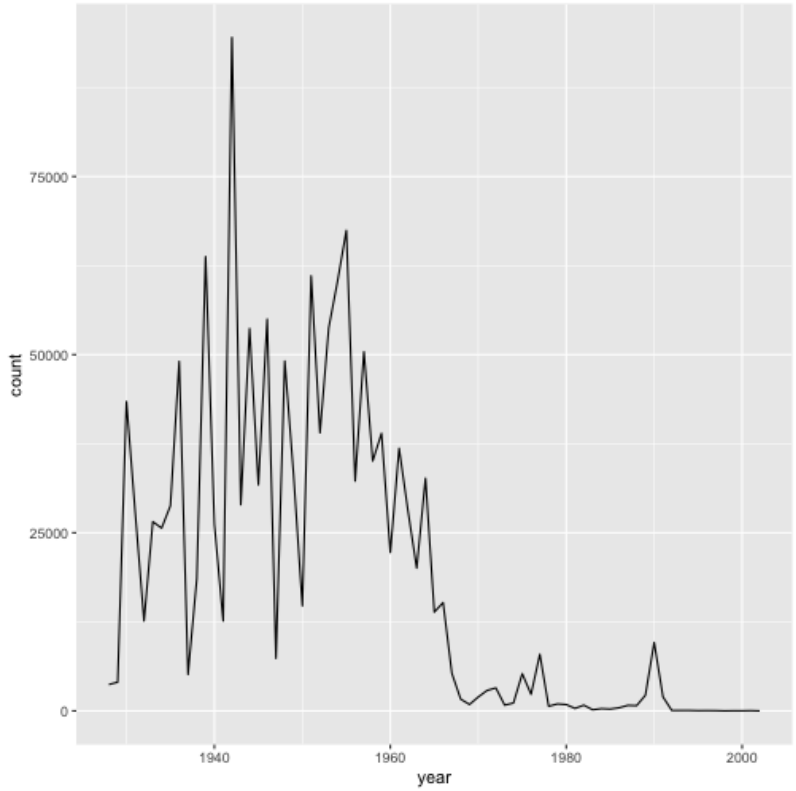
```
head(df_measles)
```

```
##    disease   state year weeks_reporting count population
## 1: Measles Alabama 1928              52  8843    2589923
## 2: Measles Alabama 1929              49  2959    2619131
## 3: Measles Alabama 1930              52  4156    2646248
## 4: Measles Alabama 1931              49  8934    2670818
## 5: Measles Alabama 1932              41   270    2693027
## 6: Measles Alabama 1933              51  1735    2713243
```

Grammar of Graphics

- Data
- Aesthestics
- Geometry
- Stats
- Facets
- Coordinate
- Theme

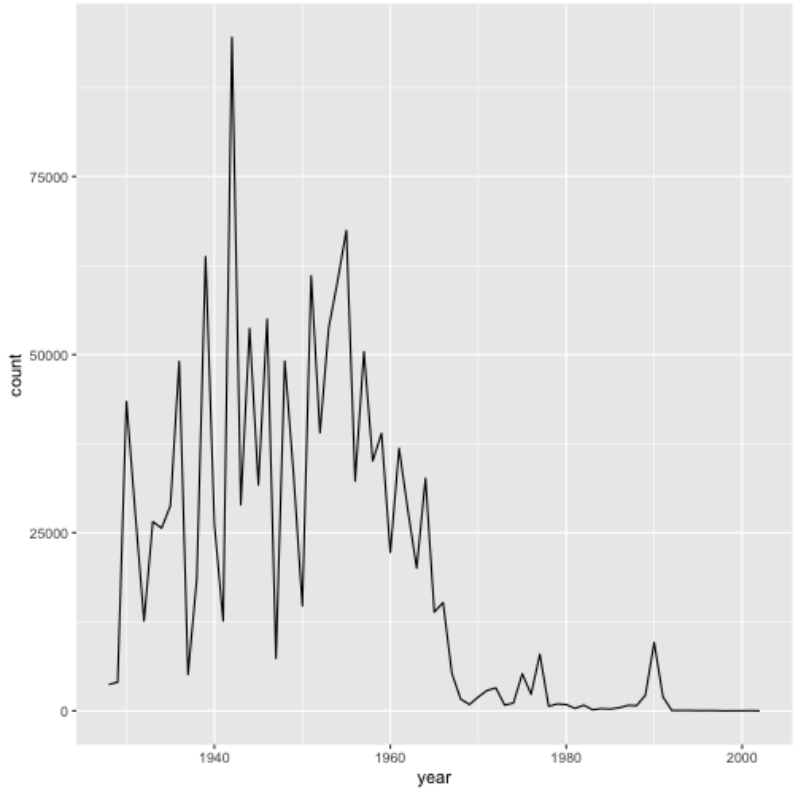
```
# Annual reported Measles cases in California
ggplot(data = CA_Measles, aes(x = year, y = count)) +
  geom_line()
```



Grammar of Graphics

- **Data**
- **Aesthetics**
- **Geometry**
- Stats
- Facets
- Coordinate
- Theme

```
# Annual reported Measles cases in California
ggplot(data = CA_Measles, aes(x = year, y = count)) +
  geom_line()
```



Grammar of Graphics

- **Data**
- **Aesthestics**
- **Geometry**
- Stats
- Facets
- Coordinate
- Theme

Data, Aesthestics(for input data), and Geometry are required to make a minimal graph

```
  ggplot(aes(x = year, y = count)) +
   geom_line()
  ## Error: `data` must be a data frame.... 🤔
```

```
ggplot(data = CA_Measles) +
  geom_line()
## Error in order(c(1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L,
## argument 3 is not a vector
```

```
ggplot(data = CA_Measles, aes(x = year, y = count)) +
  geom_line()
# No error, but you will get an empty ggplot canvas
```

Grammar of Graphics

- **Data**
- **Aesthestics**
- **Geometry**
- Stats
- Facets
- Coordinate
- Theme

Data, Aesthestics(for input data), and Geometry are required to make a minimal graph

```
ggplot(aes(x = year, y = count)) +
  geom_line()
## Error: `data` must be a data frame.... 🤔
```

```
  ggplot(data = CA_Measles) +
   geom_line()
  ## Error in order(c(1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L
  ## argument 3 is not a vector
```

```
ggplot(data = CA_Measles, aes(x = year, y = count)) +
  geom_line()
# No error, but you will get an empty ggplot canvas
```

Grammar of Graphics

- **Data**
- **Aesthestics**
- **Geometry**
- Stats
- Facets
- Coordinate
- Theme

Data, Aesthestics(for input data), and Geometry are required to make a minimal graph

```
ggplot(aes(x = year, y = count)) +
  geom_line()
## Error: `data` must be a data frame.... 🤔

ggplot(data = CA_Measles) +
 geom_line()
## Error in order(c(1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L,
## argument 3 is not a vector
```

```
  ggplot(data = CA_Measles, aes(x = year, y = count)) +
   geom_line()
  # No error, but you will get an empty ggplot canvas
```

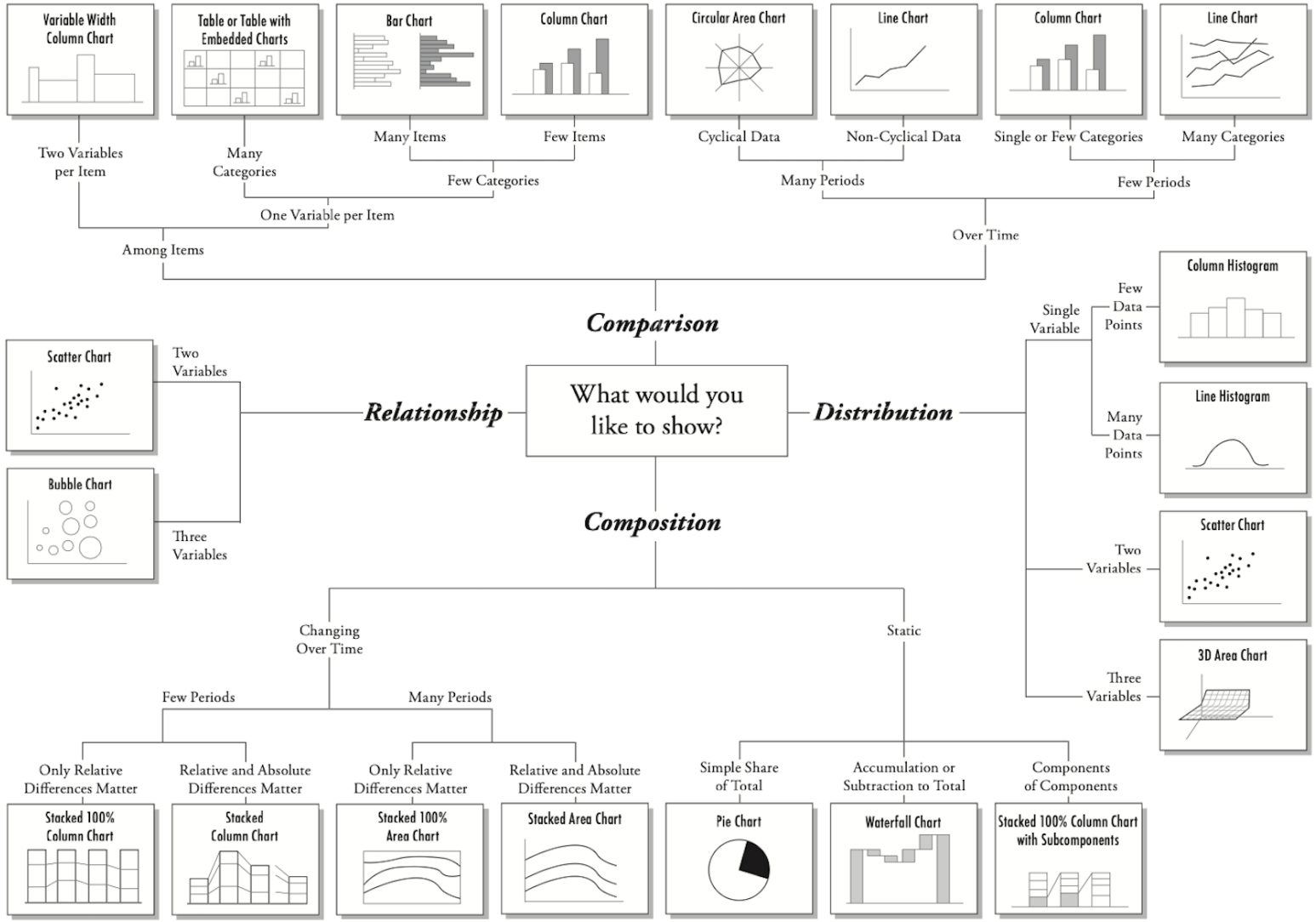
Different Library, Similar Syntax, Same Basic Components

```
library(plotly)
plot_ly(CA_Measles, x = ~year, y = ~count, type = 'scatter', mode = 'lines')

library(highcharter)
hchart(CA_Measles, 'line', hcaes(x = year, y = count)
```

Strategy

Chart Suggestions—A Thought-Starter



Bottomline

- Focus on showing data patterns using an appropriate ~~fancy~~ graph
- *Informativeness* >> Clarity >> Aesthestics
- Data visualization could be subjective, but

> The greatest value of a picture is when it forces us to notice what we never expected to see
>
> -- *John W. Tukey*

-- as opposed to what we wanted to confirm.

Visualization Process

Making Exploratory Graphs

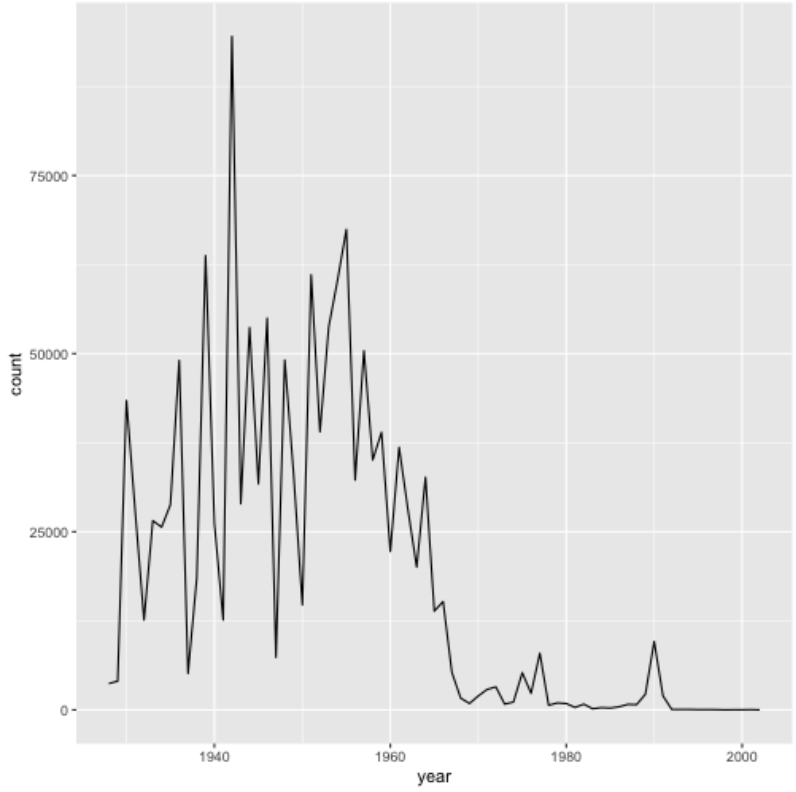
> to be able to say that we looked one layer deeper, and found nothing, is a definite step forward -- though not as far as to be able to say that we looked deeper and found thus-and-suck
>
> -- *John W. Tukey*

- Make LOTS of exploratory graphs, and only present those that can convince yourself and guide the audience through your data analysis
- In this stage, we only care about *informativeness*! We will worry about Clarity and Aesthestics in next statge.

Making Exploratory Graphs - Measles

- **Data**
- **Aesthetics**
- **Geometry**
- Stats
- Facets
- Coordinate
- Theme

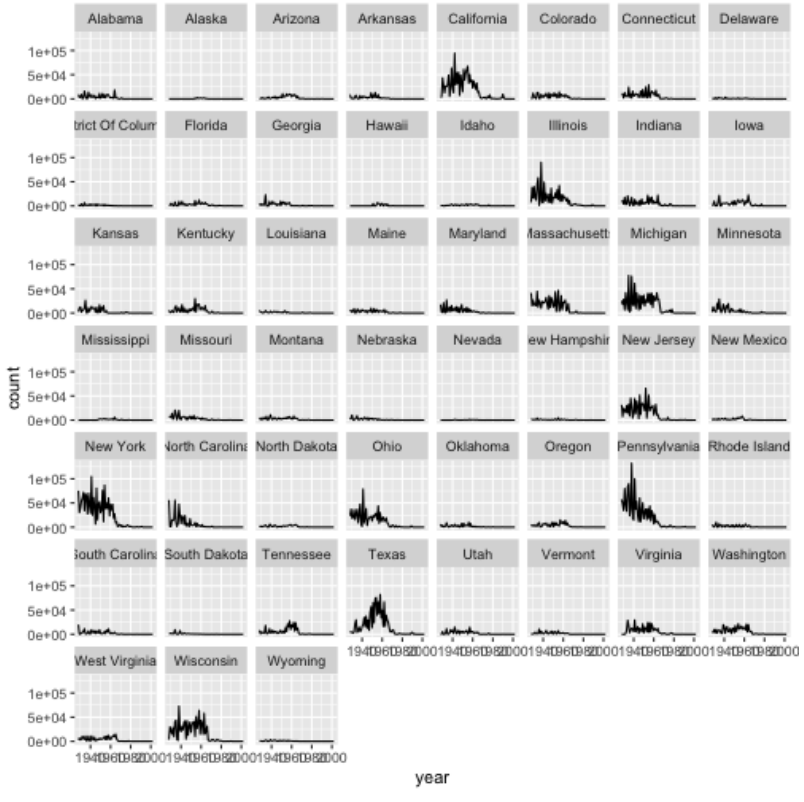
```
ggplot(data = CA_Measles, aes(x = year, y = count)) +
  geom_line()
```



Making Exploratory Graphs - Measles

- **Data**
- **Aesthetics**
- **Geometry**
- Stats
- **Facets**
- Coordinate
- Theme

```
ggplot(data = df_measles, aes(x = year, y = count)) +
  geom_line() +
  facet_wrap(~state)
```



Making Exploratory Graphs - Measles

- **Data**
- **Aesthestics**
- **Geometry**
- Stats
- Facets
- Coordinate
- Theme

```
df_avg_pop <- df_measles[, .(mean_pop = mean(population, na.rm = TRUE)), stat
ggplot(data = df_avg_pop, aes(x = state, y = mean_pop)) +
  geom_col()
```

Making Exploratory Graphs - Measles

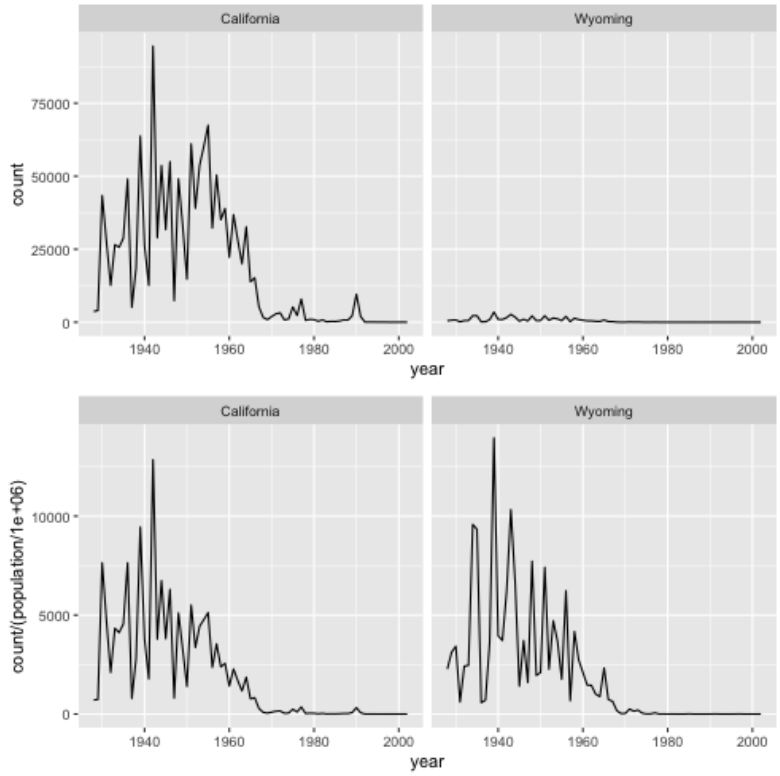
- **Data**
- **Aesthestics**
- **Geometry**
- Stats
- **Facets**
- Coordinate
- Theme

```
ca_wy_noweight <- ggplot(df_measles[state %in% c("California", "Wyoming")],
                         aes(x = year, y = count)) +
                  geom_line() +
                  facet_wrap(~state)
ca_wy_weighted <- ggplot(df_measles[state %in% c("California", "Wyoming")],
                         aes(x = year, y = count / (population / 1000000))) +
                  geom_line() +
                  facet_wrap(~state)
```

Making Exploratory Graphs - Measles

- **Data**
- **Aesthestics**
- **Geometry**
- Stats
- **Facets**
- Coordinate
- Theme

```
library(patchwork)
ca_wy_noweight / ca_wy_weighted
```



Making Exploratory Graphs - Measles

- **Data**
- **Aesthestics**
- **Geometry**
- Stats
- **Facets**
- Coordinate
- Theme

```
ggplot(data = df_measles, aes(x = year, y = count / (population / 1000000)))
  geom_line() +
  facet_wrap(~state)
```

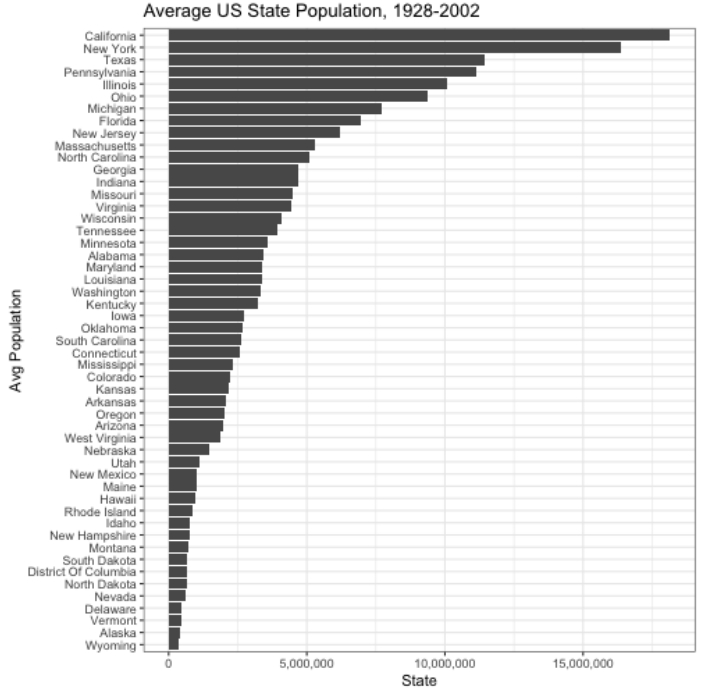
Making a Confirmatory Graph

- Few exploratory graphs need to become confirmatory graphs
- Key findings or evidence in your data analysis that help draw conclusions or inform modeling decisions
- Now we have the information we want to share, we can work on Clarity and Aesthestics

Making a Confirmatory Graph -- Measles

- **Data**
- **Aesthestics**
- **Geometry**
- Stats
- **Facets**
- **Coordinate**
- **Theme**

```
ggplot(data = df_avg_pop, aes(x = reorder(state, mean_pop), y = mean_pop)) +
  geom_col() + coord_flip() +
  ggtitle("Average US State Population, 1928-2002") +
  scale_y_continuous(labels = scales::comma) +
  xlab("Avg Population") + ylab("State") +
  theme_bw()
```



Making a Confirmatory Graph -- Measles

- **Data**
- **Aesthestics**
- **Geometry**
- Stats
- **Facets**
- **Coordinate**
- **Theme**

```
ggplot(data = df_measles, aes(x = year, y = count / (population / 1000000)))
  geom_line() + facet_wrap(~state) +
  ggtitle("Measle Cases per Million People by State, 1928-2002") +
  scale_y_continuous(labels = scales::comma) +
  xlab("State") + ylab("Cases/1m Population") +
  theme_bw()
```

Toolbox

Visualization Toolbox

- Lots of ggplot extensions
  - patchwork - arrange and stitch graphs together
  - gganimate - make animated ggplots
  - ggdendro - make dendrogram in ggplot
  - ggrepel - display labels nicely
  - ggradar - radar chart
  - ggmap - draw maps
  - cowplot - arrange graphs to be publication ready
  - ggiraph - make ggplot interactive
  - ggfacet - facet on a map
- Color Palettes
  - r-color-palettes
  - Wes Anderson Palettes
  - html color codes - if you really want to customize

Visualization Toolbox

- Highcharter
- Dygraph
- Plotly
- leaflet - Interactive maps
- Altair - Can show distribution in the highlighted region - Python only

Resources

Tutorials, Videos, Books, and Paper

- Liz Sander - Telling stories with data using the grammar of graphics
- Hadley Wickham - A Layered Grammar of Graphics
- Thomas Lin Pedersen - ggplot2 Workshop (video, 4.5hr tutorials with latest dev updates)
- John W. Tukey - Exploratory Data Analysis, Preface
- Dipanjan (DJ) Sarkar - A Comprehensive Guide to the Grammar of Graphics for Effective Visualization of Multi-dimensional Data

Thanks!

Slides created using the R package **xaringan**.