

Data Visualization in R with ggplot2::

Meg Hartwick, PhD

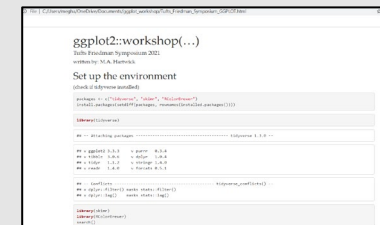
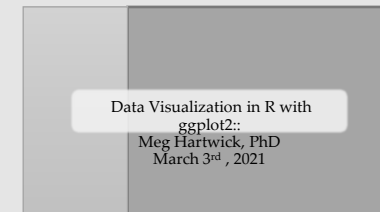
March 3rd, 2021

Workshop Overview

Materials Available at:

<https://github.com/meghartwick/ggplot2-Workshop>

- PowerPoint Slides
- HTML
 - <https://rpubs.com/meghartwick/733550>
- Notebook and Code



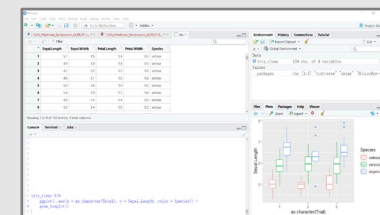
```
ggplot2:workshop(...)
Title: Data Visualization in R with ggplot2
Author: Meg Hartwick
Date: 2021-03-03
Output: pdf_document

Set up the environment
(should only be run once)

install.packages("tidyverse", "ggplot2", "rmarkdown")
library(tidyverse)
library(ggplot2)
library(rmarkdown)

# Run the following code to install the packages
# install.packages("tidyverse")
# install.packages("ggplot2")
# install.packages("rmarkdown")

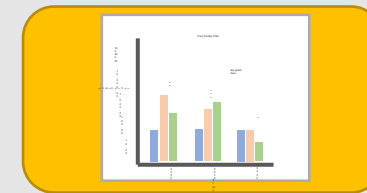
# Load the packages
library(tidyverse)
library(ggplot2)
library(rmarkdown)
```



Workshop Overview

Flow

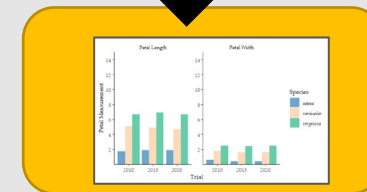
I. Foundation - Concept Review



II. Structure – ggplot2:: syntax

ggplot(...)

III. Application – Sketch to Story



Workshop Overview

Why do you usually make graphics?

1. For your own use.
2. Informal sharing with colleagues.
3. For formal presentations or publication.



Workshop Overview

How do you usually make graphics?

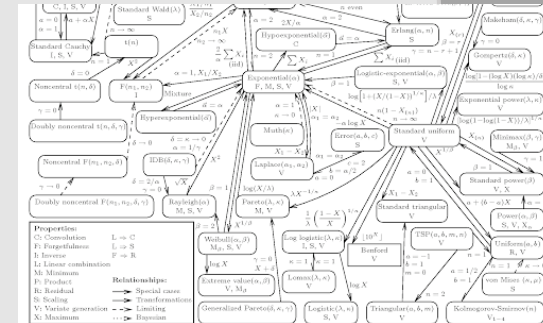
1. Quick and accessible (excel).
2. Statistical software (eg: JMP, SAS).
3. Tableau, base R, some ggplot2.
4. R, Python and/or others



Workshop Overview

What makes for a good graphic?

1. Really busy with every detail in text...



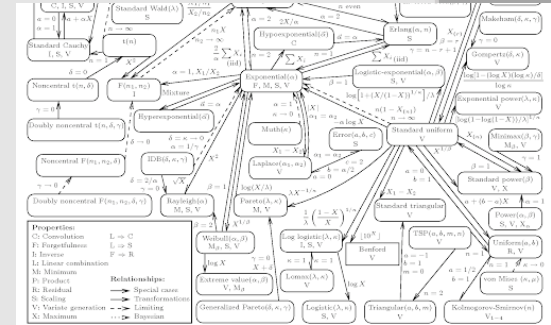
2. Flashy

3. A clear

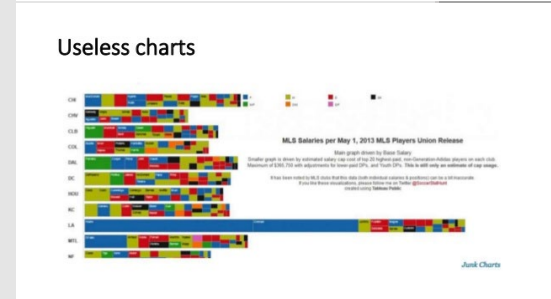
Workshop Overview

What makes for a good graphic?

1. Really busy with every detail in text...



2. Flashy plotting...

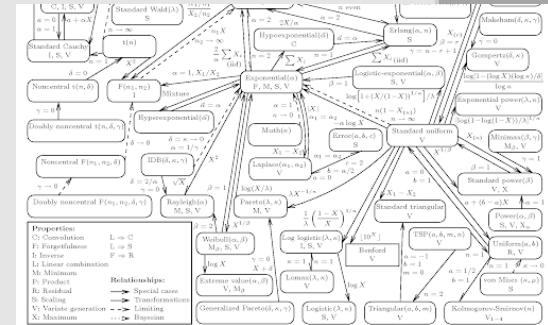


3. A clear

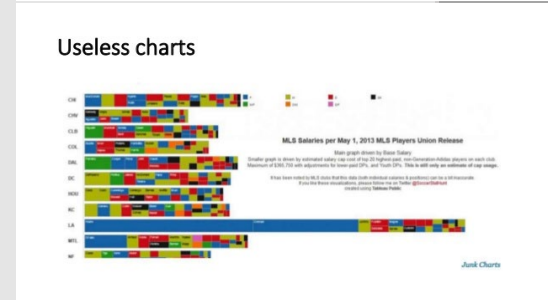
Workshop Overview

What makes for a good graphic?

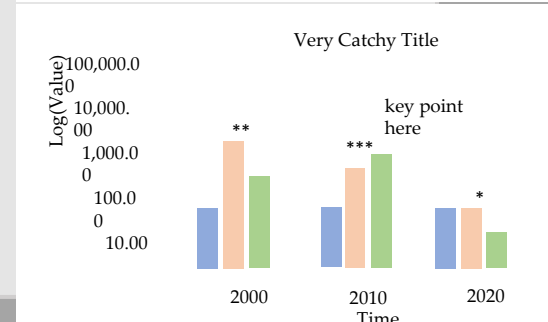
1. Really busy with every detail in text...



2. Flashy plotting...



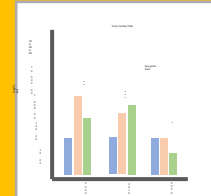
3. A clear message that tells a story...



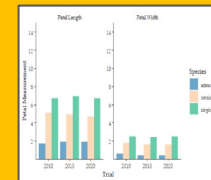
Workshop Overview

Today's Goal

- Understand how to develop graphics that:
 - Effectively tell a story
 - In ggplot2 (and some tidyverse)
 - Are refined or highly refined.



`ggplot(...)`



I. Foundation

1. Concepts

2. Tidyverse

3. ggplot2::

Foundations and Concepts Refresher

1. Data

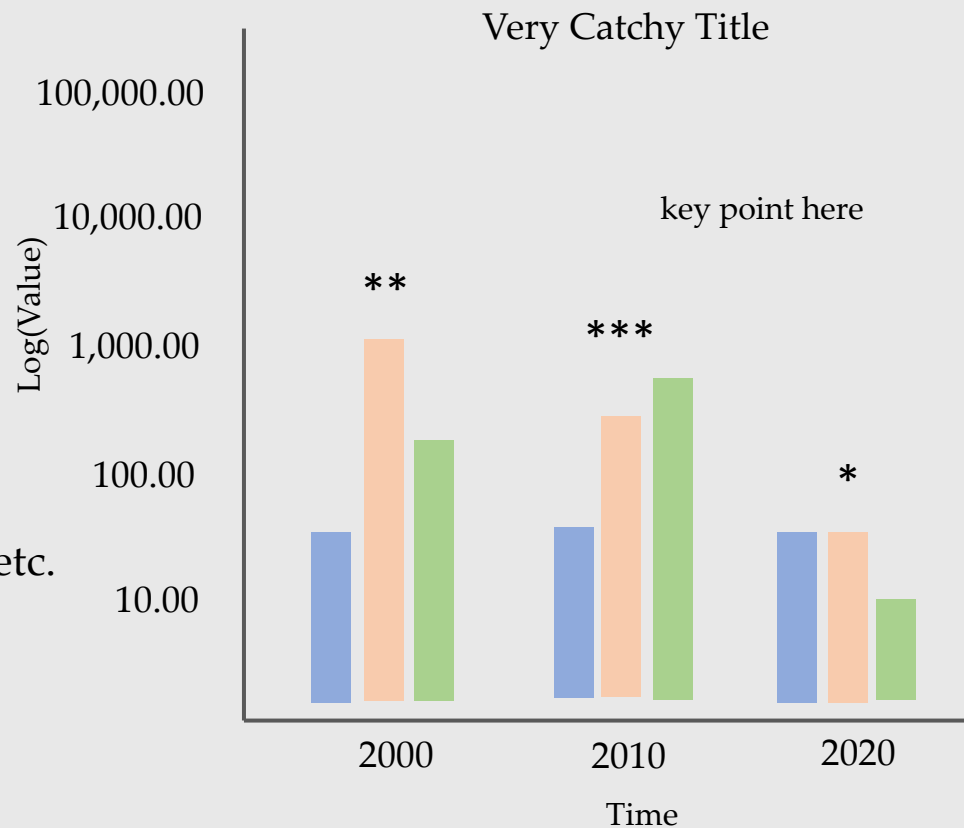
- Structure
 - Continuous
 - Discrete
- Format
 - Wide
 - Long

2. Graph Attributes

- Dimension
- Axes scale
- Symbols, color schemes etc.

3. Key Elements

- Title
- Axes labels
- Embedded comments



I. Foundation

1. Concepts

2. Tidyverse

3. ggplot2::

Tidyverse

- Originally 'Hadleyverse'
- Collection of R packages with shared:
 - Philosophy
 - Structure
 - Syntax
- Package 'piping' (`%>%`)
 - Functions act as verbs
- Over 27 packages including:
 - *readr* – importing data
 - *dplyr* – data cleaning
 - *ggplot2* – data visualization
 - *lubridate* – working with dates
 - *stringr* – working with strings



I. Foundation

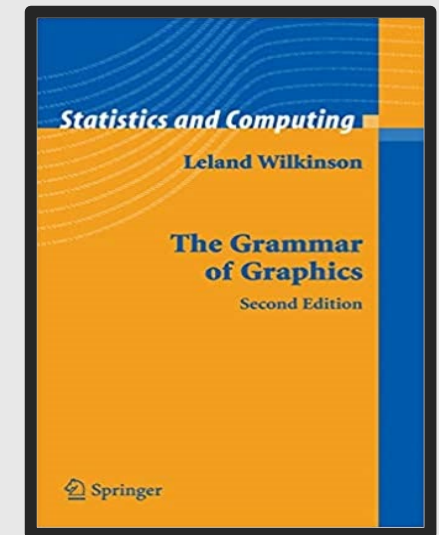
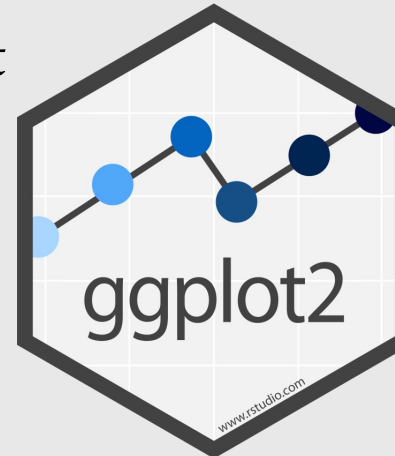
1. Concepts

2. Tidyverse

3. ggplot2::

ggplot2::

- Developed by Hadley Wickham
- 10+ years old
- Readability > base R plot
- Over 84 extensions
 - *ggtext*
 - *ggthemes*
 - *gganimate*
 - *esquisse*
- ‘Grammar of Graphics’
- 8 main class of functions
 - align with key elements and attributes of graphics communication



II. Structure

Foundations

1. Data

- Structure
 - Continuous
 - Discrete
- Format
 - Wide
 - Long

2. Graph attributes

- Dimension
- Axes scale
- Symbols, colors etc.

3. Key elements

- Title
- Axes labels
- Embedded comments

ggplot2:: Syntax

1. Data
2. Function
3. Coordinates

4. Mapping
5. Geometries
6. Scales
7. Facets

8. Themes

II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

5. Geometries

6. Scales

7. Facets

8. Themes

ggplot2::

- Slides
 - Code - How the element is called
 - Considerations – some assumptions to keep in mind
 - Arguments – How, when, what to use
- HTML
 - Follow along with the code and graphics
- Notebook and Code
 - Run code as source code or chunks

II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

5. Geometries

6. Scales

7. Facets

8. Themes

```
df %>%
```



II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

5. Geometries

6. Scales

7. Facets

8. Themes

df %>%

- Considerations

- Discrete or Continuous data for coordinates
- Discrete or Continuous metadata for mapping
- Missing Data etc.
- Tidyverse can be really useful here
 - data transformations
 - df reshaping
- Structure check
 - skimr::skim()

- Arguments

- Data can be piped to function or called within

`skimr::skim(df)`

`df %>% dplyr::filter() %>% dplyr::pivot()`

II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

5. Geometries

6. Scales

7. Facets

8. Themes

`df %>%`

- Considerations

- Discrete or Continuous data for coordinates
- Discrete or Continuous metadata for mapping
- Missing Data etc.
- Tidyverse can be really useful here
 - data transformations
 - df reshaping
- Structure check
 - `skimr::skim()`

Foundation: Graph Data

- Arguments

- Date can be piped to function or called within

`skimr::skim(df)`

`df %>% dplyr::filter() %>% dplyr::pivot()`

II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

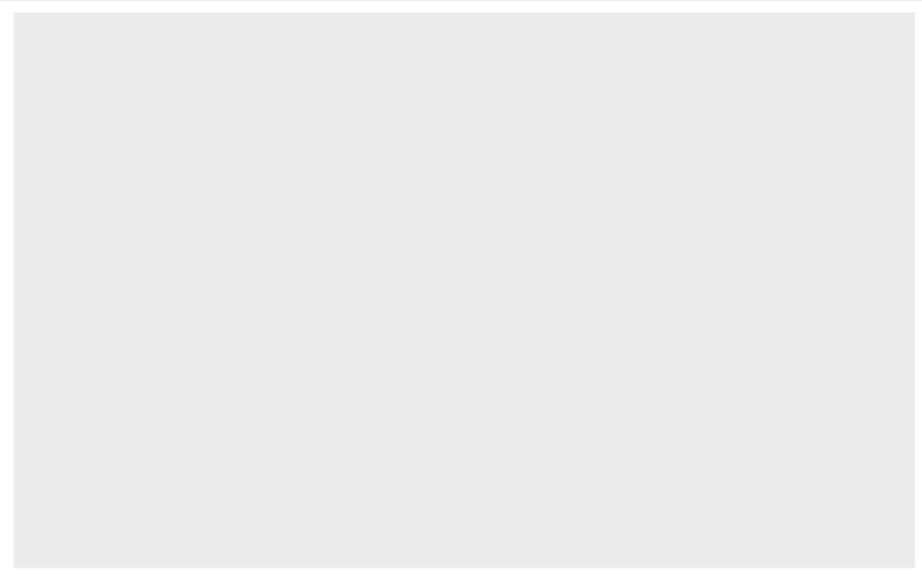
5. Geometries

6. Scales

7. Facets

8. Themes

```
df %>%  
ggplot(...) +
```



II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

5. Geometries

6. Scales

7. Facets

8. Themes

ggplot(...) +

- **Consideration**
 - Calls the plot object
 - Best placement of df for plot development
- **Arguments**
 - Data =
 - Coordinates =
 - Mapping =

`ggplot(df, aes()) + ...`

`ggplot(df) +`

`ggplot() +`

`df %>% ggplot(., aes()) +`

II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

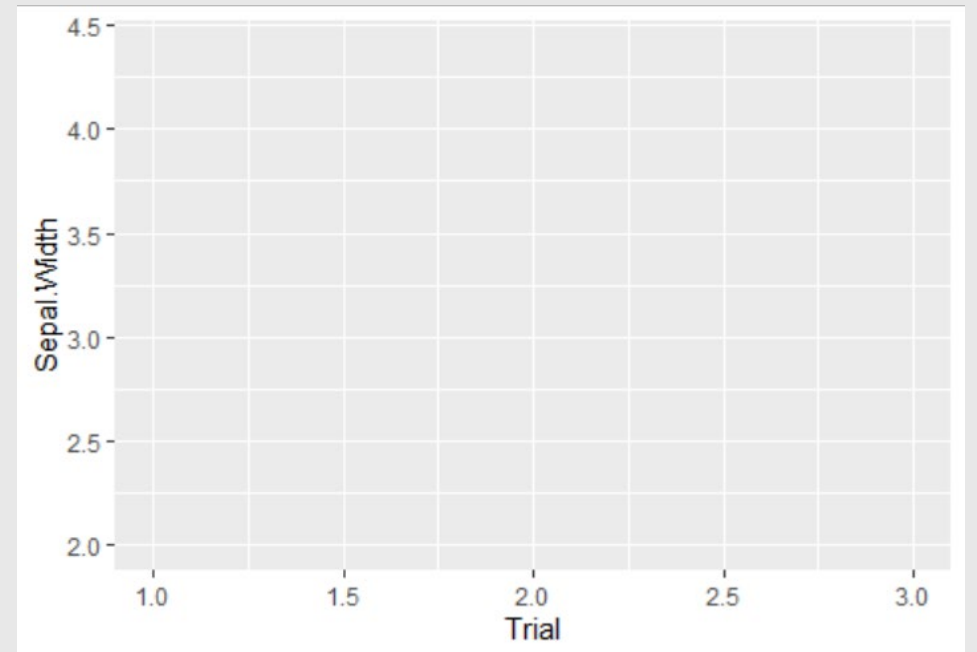
5. Geometries

6. Scales

7. Facets

8. Themes

```
df %>%  
ggplot(., aes(x = x, y = y, ...)) +
```



II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

5. Geometries

6. Scales

7. Facets

8. Themes

```
ggplot(., aes(x = x, y = y, ...)) +
```

- Considerations
 - Data positions for plot
 - Not necessary to specify here, but must be supplied in plot layers
 - Continuous Data
 - Coordinate according to the data
 - Discrete Data
 - At 1, 2, 3 etc. on axis
 - Alphabetical for character class
 - Level for factors class

II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

5. Geometries

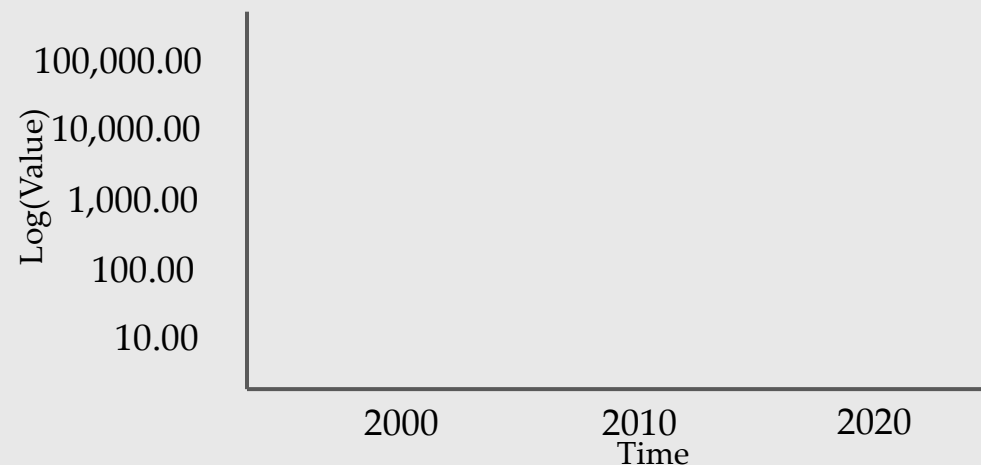
6. Scales

7. Facets

8. Themes

```
ggplot(., aes(x = x, y = y, ...)) +
```

- Arguments
 - x =
 - y =
 - positional, not necessary to specify
 - for geometries that use count, y not accepted



II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

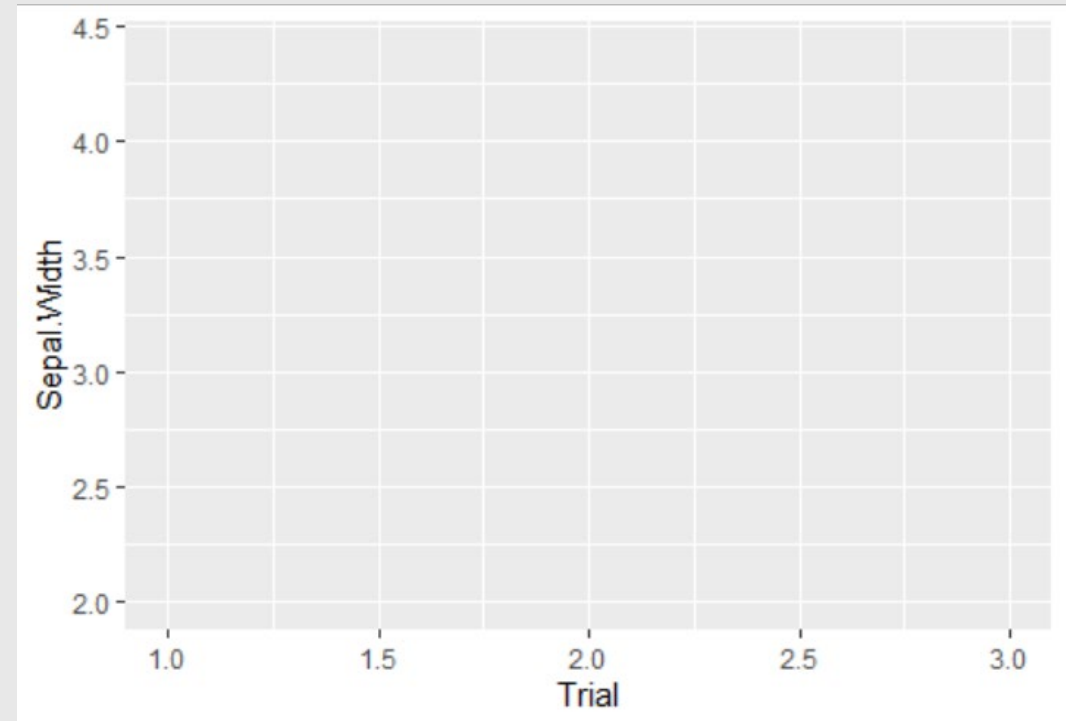
5. Geometries

6. Scales

7. Facets

8. Themes

```
df %>%  
ggplot(., aes(x = x, y = y, *= var1) +
```



II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

5. Geometries

6. Scales

7. Facets

8. Themes

```
ggplot(., aes(x = x, y = y, * = var1)) +
```

- Considerations

- Variables are mapped to visual properties (aesthetics)
- Choosing the aesthetic (*)
 - Color
 - Fill
 - Size
 - Shape
 - Linetype
 - Transparency
- Is the Continuous or Discrete?
- What are you mapping to?



II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

5. Geometries

6. Scales

7. Facets

8. Themes

```
ggplot(., aes(x = x, y = y, * = var1)) +
```

- Arguments
 - Mappings can be set in ggplot()
 - ggplot(.,aes(color = var1))
 - Mappings can be set in individual layers
 - geom_point(.,aes(color = var1))
 - What is outside of the mapping will be interpreted literally
 - geom_point(.,aes(color = 'var1'))
 - Levels of the mapping are set in scales, else:
 - (.,aes(), color = 'black')

II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

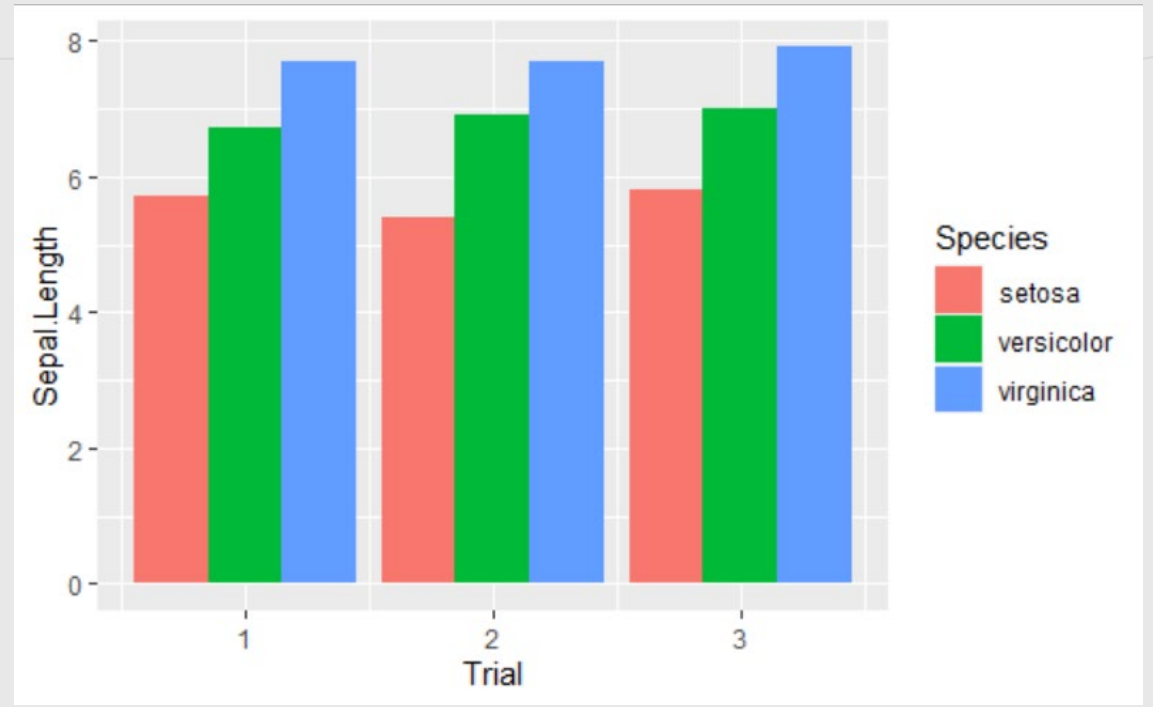
5. Geometries

6. Scales

7. Facets

8. Themes

```
df %>%  
ggplot(., aes(x = x, y = y, * = var1))+  
geom_*(...) +
```



II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

5. Geometries

6. Scales

7. Facets

8. Themes

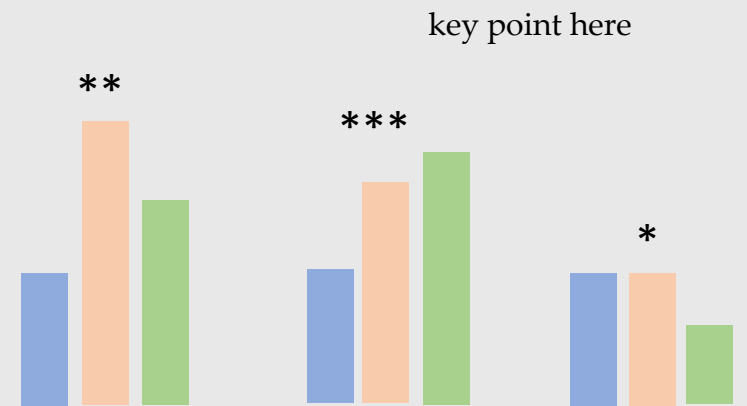
`geom_*(...)` +

- Considerations

- What kind of visual representations (*) are the best approach?

- Types of Geometry

- Reference Lines
- Barcharts
- Dots and points
- Boxplots
- Heatmaps
- Maps
- Density
- Polygons
- Jitters
- Error Bars
- Text and Labels



II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

5. Geometries

6. Scales

7. Facets

8. Themes

`geom_*(...)` +

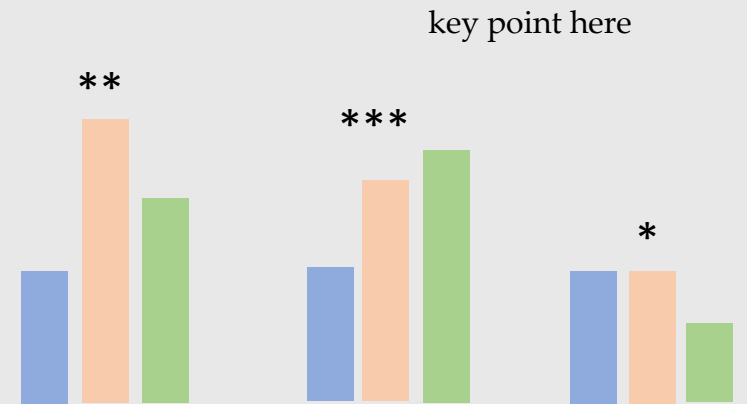
- Considerations

- What kind of visual representations (*) are the best approach?

- Types of Geometry

- Reference Lines
- Barcharts
- Dots and points
- Boxplots
- Heatmaps
- Maps
- Density
- Polygons
- Jitters
- Error Bars
- Text and Labels

Foundation: Graph Attributes



II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

5. Geometries

6. Scales

7. Facets

8. Themes

`geom_*(...)` +

- Arguments

- Over 40 individual geoms

- `geom_*()`

- `geom_point()`
- `geom_hist()`
- `geom_bar()`
- `geom_col()`
- `geom_tile()`

- Can build individual mappings within `geoms_*()`

- Can add multiple geometries as annotation layers

- Know the defaults

`geom_*(mapping =, data =, stat =, position =)`



II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

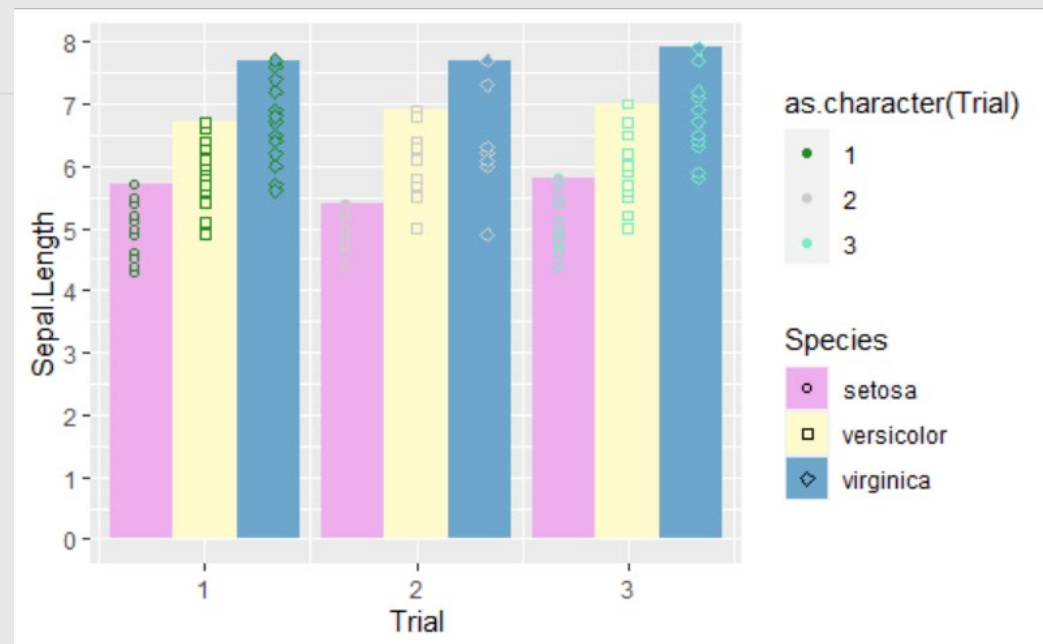
5. Geometries

6. Scales

7. Facets

8. Themes

```
df %>%  
ggplot(., aes(x = x, y = y, * = var1))+  
geom_*(...) +  
scale_*_*(...) +
```



II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

5. Geometries

6. Scales

7. Facets

8. Themes

`scale_*_*(...) +`

- **Concept**
 - Specify Data and Mappings
 - Set arguments for coordinates (*)
 - x
 - y
 - Set arguments for mappings (*)
 - color
 - fill
 - alpha
 - linetype
 - ... and many more
 - Different calls for discrete and continuous data types
 - Commonly used defaults are prebuilt

II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

5. Geometries

6. Scales

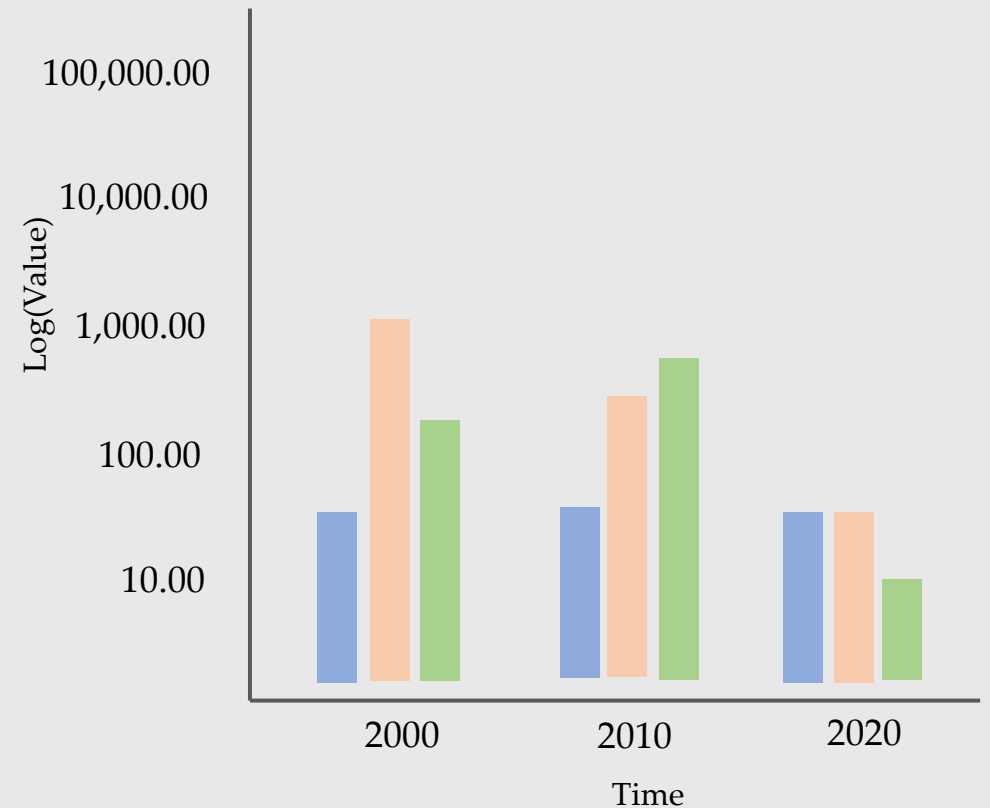
7. Facets

8. Themes

`scale_*_*(...) +`

- Arguments

```
scale_*_*(  
  name = ,  
  breaks = ,  
  values = ,  
  labels = ,  
  limits = ,  
  trans = ,  
  guide = ,  
  position = ,  
  ....)
```



II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

5. Geometries

6. Scales

7. Facets

8. Themes

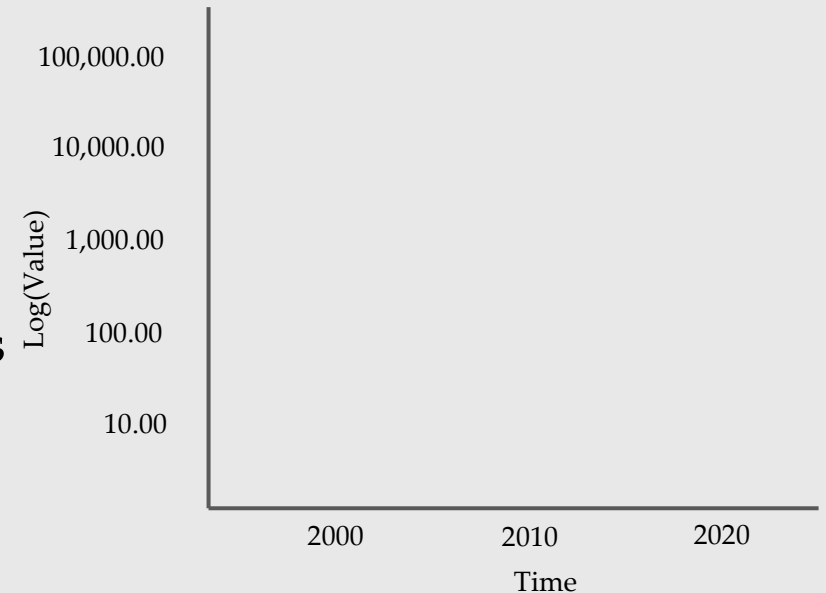
`scale_*_*(...) +`

- Standard Axis Scales

- `scale_x_continuous(...)`
- `scale_y_continuous(...)`
- `scale_x_discrete(...)`
- `scale_y_discrete(...)`

- Pre-Built Custom Axis Scales

- `scale_*_log10(...)`
- `scale_*_reverse(...)`
- `scale_*_sqrt(...)`
- `scale_*_datetime(...)`
- ... and many more



II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

5. Geometries

6. Scales

7. Facets

8. Themes

`scale_*_*(...) +`

- Standard Alpha Scales
 - `scale_alpha_continuous(...)`
 - `scale_alpha_discrete(...)`
- Standard Shape Scales
 - `scale_shape_continuous(...)`
 - `scale_shape_discrete(...)`
- Standard Linetype Scales
 - `scale_linetype_continuous(...)`
 - `scale_linetype_discrete(...)`
- ... and many more Standard
- ... and many more Pre-Built



II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

5. Geometries

6. Scales

7. Facets

8. Themes

`scale_*_*(...) +`

- Standard Color Scales
 - `scale_color_continuous(...)`
 - `scale_fill_continuous(...)`
 - `scale_color_manual(...)`
 - `scale_fill_manual(...)`
- Pre-Built Custom Color Scales
 - `scale_*_brewer(...)`
 - `scale_*_gradient(...)`
 - `scale_*_gradientn(...)`
 - `scale_*_viridis(...)`
 - ... and many more



II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

5. Geometries

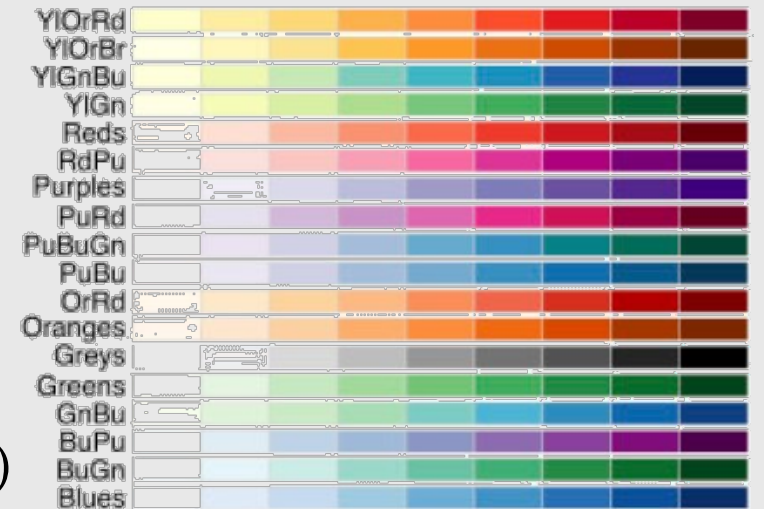
6. Scales

7. Facets

8. Themes

`scale_*_*(...) +`

- Standard Scale – specify colors
 - `scale_color_manual(values = c('red', 'green', 'blue'))`
- Pre-Built Custom Color Scales
 - `scale_*_brewer(`
 - `type = ,`
 - `palette = ,`
 - `direction = ,`
 - `aesthetics =)`
 - `scale_color_brewer(palette = 'set2')`



II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

5. Geometries

6. Scales

7. Facets

8. Themes

```
df %>%  
ggplot(., aes(x = x, y = y, * = var1)) +  
geom_*(....) +  
scale_*_*(...) +  
facet_*(...)
```



II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

5. Geometries

6. Scales

7. Facets

8. Themes

`facet_*(...)` +

- **Concept**
 - Highlight levels in data through multiple panels
 - `facet_wrap(...)`
 - creates a ribbon of levels
 - `facet_grid(...)`
 - creates a matrix of rows and columns of variable combinations
- **Arguments**
 - `nrow =`,
 - `ncol =`,
 - `scales =`,
 - `shrink =`,
 - `labeller =`,
 - `strip.position =`,
 - `...)`

II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

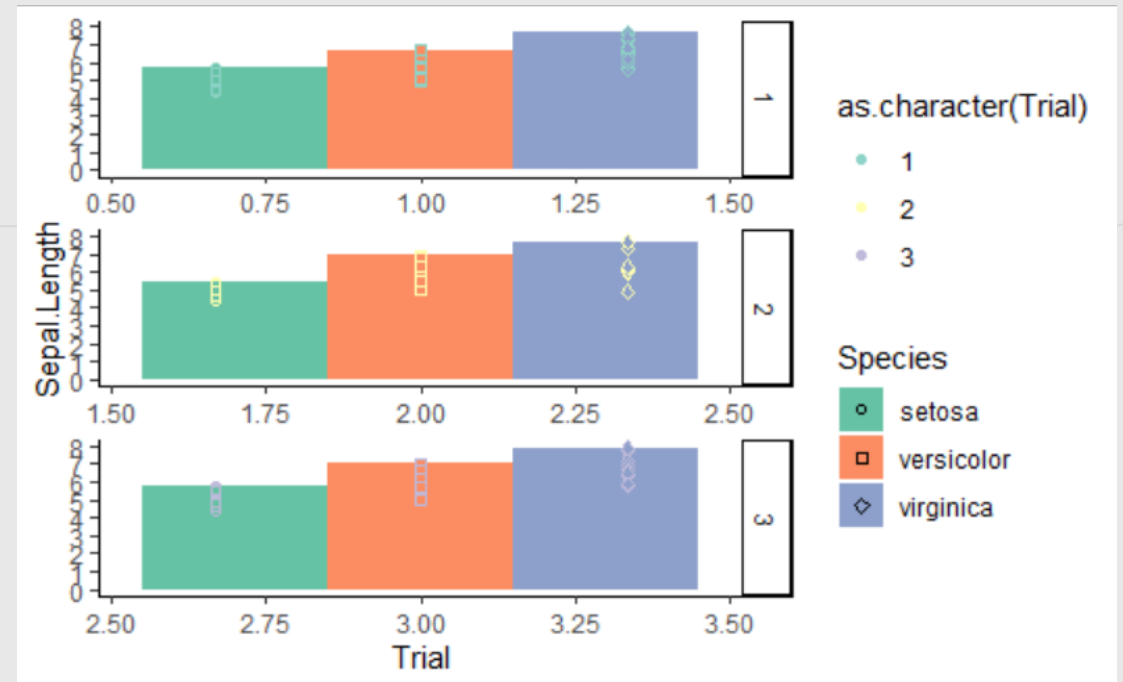
5. Geometries

6. Scales

7. Facets

8. Themes

```
df %>%  
  ggplot(., aes(x = x, y = y, * = var1)) +  
  geom_*(...) +  
  scale_*_*(...) +  
  facet_*(...) +  
  theme(...)
```



II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

5. Geometries

6. Scales

7. Facets

8. Themes

theme(...)

- Concept
 - Encompasses ALL the options of `ggplot2::plot 'Elements'`
 - 4 main modifiers
 - `line`: all line elements
 - `rect`: all rectangular elements
 - `text`: all text elements
 - `title`: all title elements (including: plot, axes, legends..)
 - Pre-built themes available
 - `theme_bw(...)`
 - `theme_grey(...)`
 - `ggthemes::`
 - a package with a wider selection of Pre-Built themes

II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

5. Geometries

6. Scales

7. Facets

8. Themes

theme(...)

- Concept
 - Encompasses ALL the options of `ggplot2::plot 'Elements'`
 - 4 main modifiers
 - line: all line elements
 - rect: all rectangular elements
 - text: all text elements
 - title: all title elements (including: plot, axes, legends..)
 - Pre-built themes available
 - `theme_bw(...)`
 - `theme_grey(...)`
 - `ggthemes::`
 - a package with a wider selection of Pre-Built themes

Foundation: Graph Elements

II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

5. Geometries

6. Scales

7. Facets

8. Themes

theme(...) +

- Argument
 - The main modifiers:
 - theme(
 - text* = element_text(),
 - panel* = element_rect(),
 - axis* = element_line(),
 - title* = element_text()
 - where * = ...

II. Structure

1. Data

2. Function

3. Coordinates

4. Mapping

5. Geometries

6. Scales

7. Facets

8. Themes

theme(...) +

```
theme(  
  aspect.ratio,  
  axis.title,  
  axis.title.x,  
  axis.title.x.top,  
  axis.title.x.bottom,  
  axis.title.y,  
  axis.title.y.left,  
  axis.title.y.right,  
  axis.text,  
  axis.text.x,  
  axis.text.x.top,  
  axis.text.x.bottom,  
  axis.text.y,  
  axis.text.y.left,  
  axis.text.y.right,  
  axis.ticks,  
  axis.ticks.x,  
  axis.ticks.x.top,  
  axis.ticks.x.bottom,  
  axis.ticks.y,  
  axis.ticks.y.left,  
  axis.ticks.y.right,  
  axis.ticks.length,  
  axis.ticks.length.x,  
  axis.ticks.length.x.top,  
  axis.ticks.length.x.bottom,  
  axis.ticks.length.y,  
  axis.ticks.length.y.left,  
  axis.ticks.length.y.right,  
  axis.line,  
  axis.line.x,  
  axis.line.x.top,  
  axis.line.x.bottom,  
  axis.line.y,  
  axis.line.y.left,  
  axis.line.y.right,  
  legend.background,  
  legend.margin,  
  legend.spacing,  
  legend.spacing.x,  
  legend.spacing.y,  
  legend.key,  
  legend.key.size,  
  legend.key.height,  
  legend.key.width,  
  legend.text,  
  legend.text.align,  
  legend.title,  
  legend.title.align,  
  legend.position,  
  legend.direction,  
  legend.justification,  
  legend.box,  
  legend.box.just,  
  legend.box.margin,  
  legend.box.background,  
  legend.box.spacing,  
  panel.background,  
  panel.border,  
  panel.spacing,  
  panel.spacing.x,  
  panel.spacing.y,  
  panel.grid,  
  panel.grid.major,  
  panel.grid.minor,  
  panel.grid.major.x,  
  panel.grid.major.y,  
  panel.grid.minor.x,  
  panel.grid.minor.y,  
  panel.ontop,  
  plot.background,  
  plot.title,  
  plot.title.position,  
  plot.subtitle,  
  plot.caption,  
  plot.caption.position,  
  plot.tag,  
  plot.tag.position,  
  plot.margin,  
  strip.background,  
  strip.background.x,  
  axis.ticks.length.x.bottom,  
  strip.background.x,  
  strip.background.y,  
  strip.placement,  
  strip.text,  
  strip.text.x,  
  strip.text.y,  
  strip.switch.pad.grid,  
  strip.switch.pad.wrap,  
  ...)
```

III. Application



Sketch

Story

III. Application



- Facetted Width and Length
- Bargraph
- Trial axis breaks as '2010', '2015', '2020'
- Y axis at 2, 4, 6, 8
- Choose a different palette for species
- Ditch the points, and accompanying scale
- Dodge position
- Species in italics
- No grey in facet
- New font

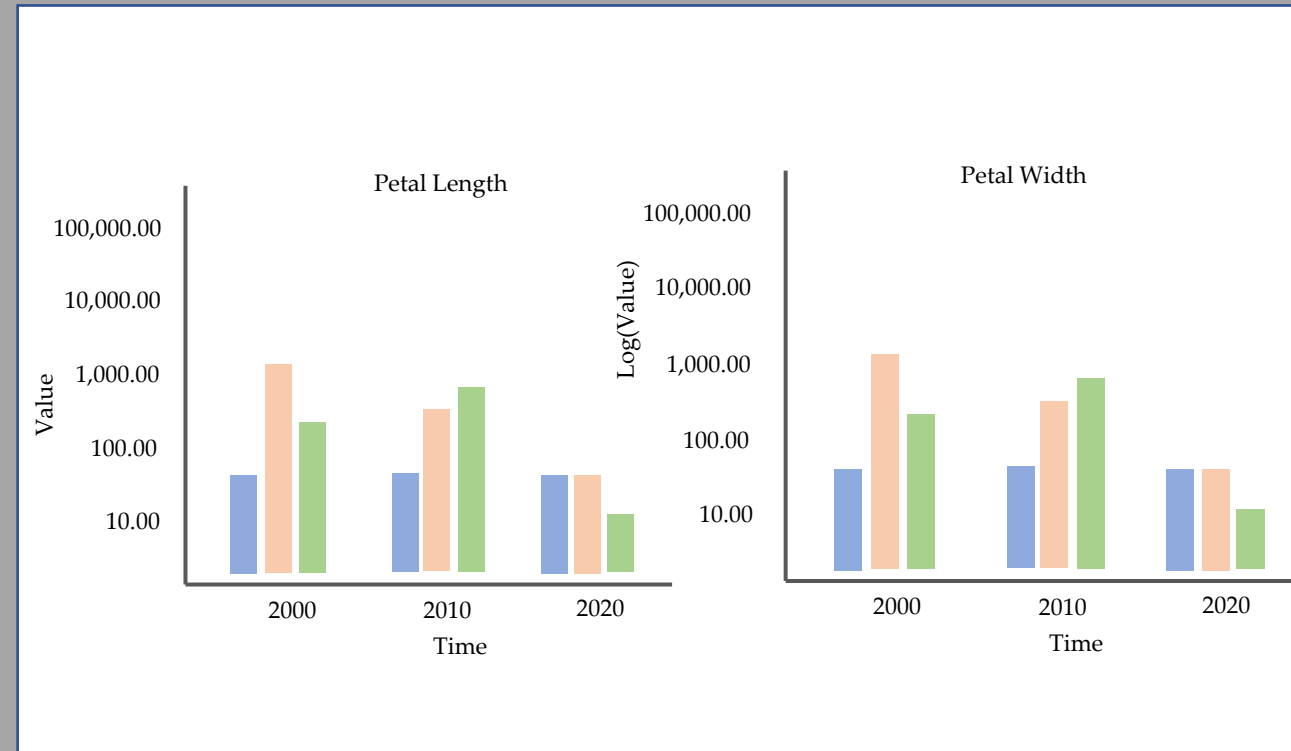
Sketch



Story

III. Application

- Facetted Width and Length
- Bargraph
- Trial axis breaks as '2010', 2015', '2020'
- Y axis at 2, 4, 6, 8
- Choose a different palette for species
- Ditch the points, and accompanying scale
- Dodge position
- Species in italics
- No grey in facet
- New font

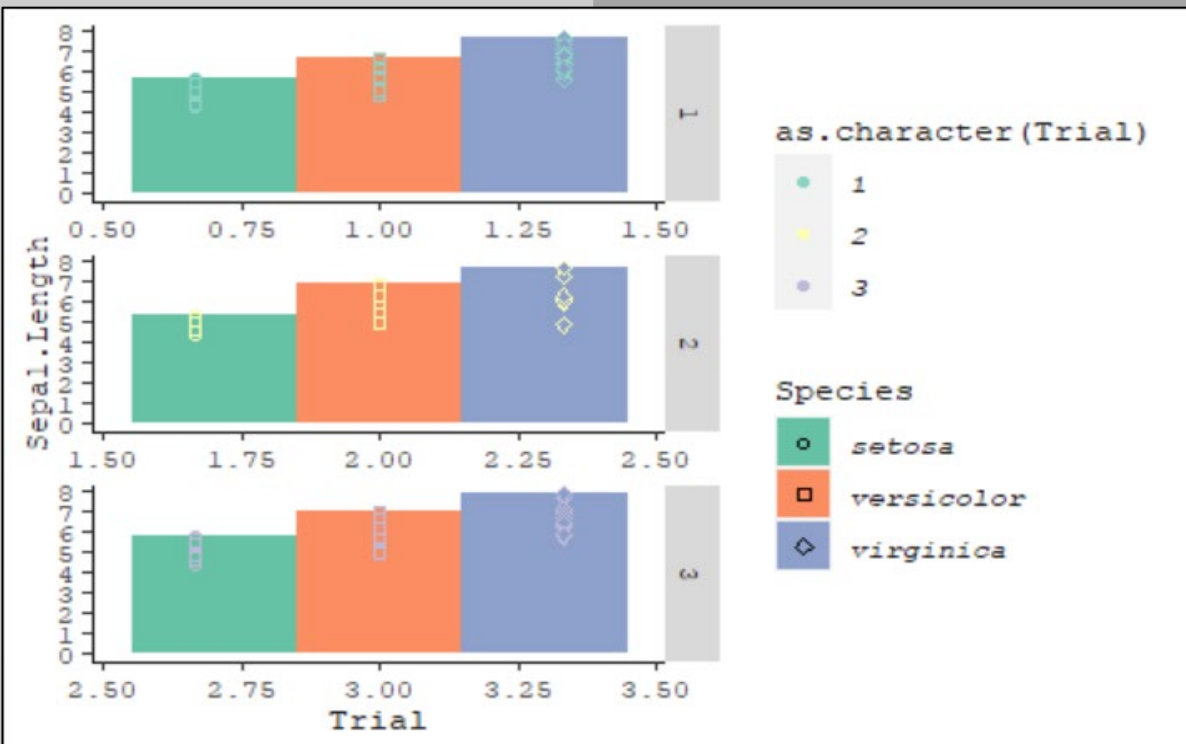


Sketch

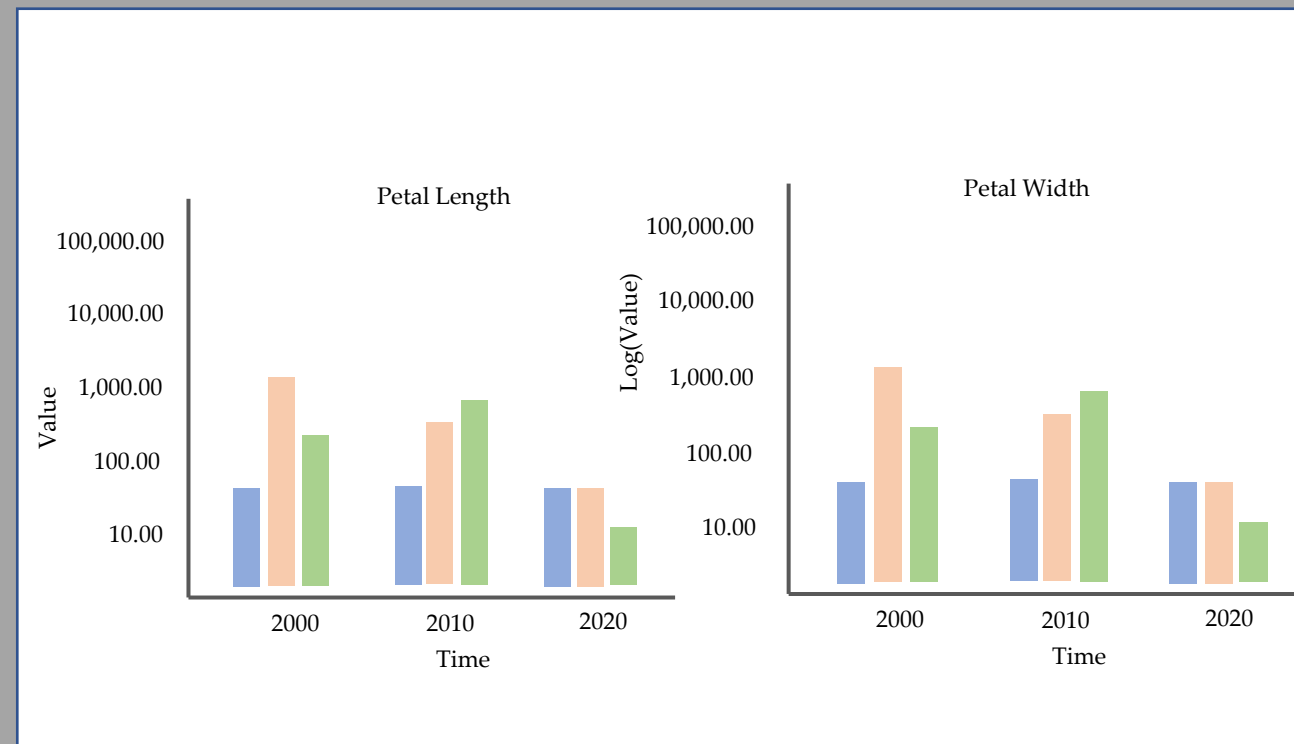


Story

III. Application

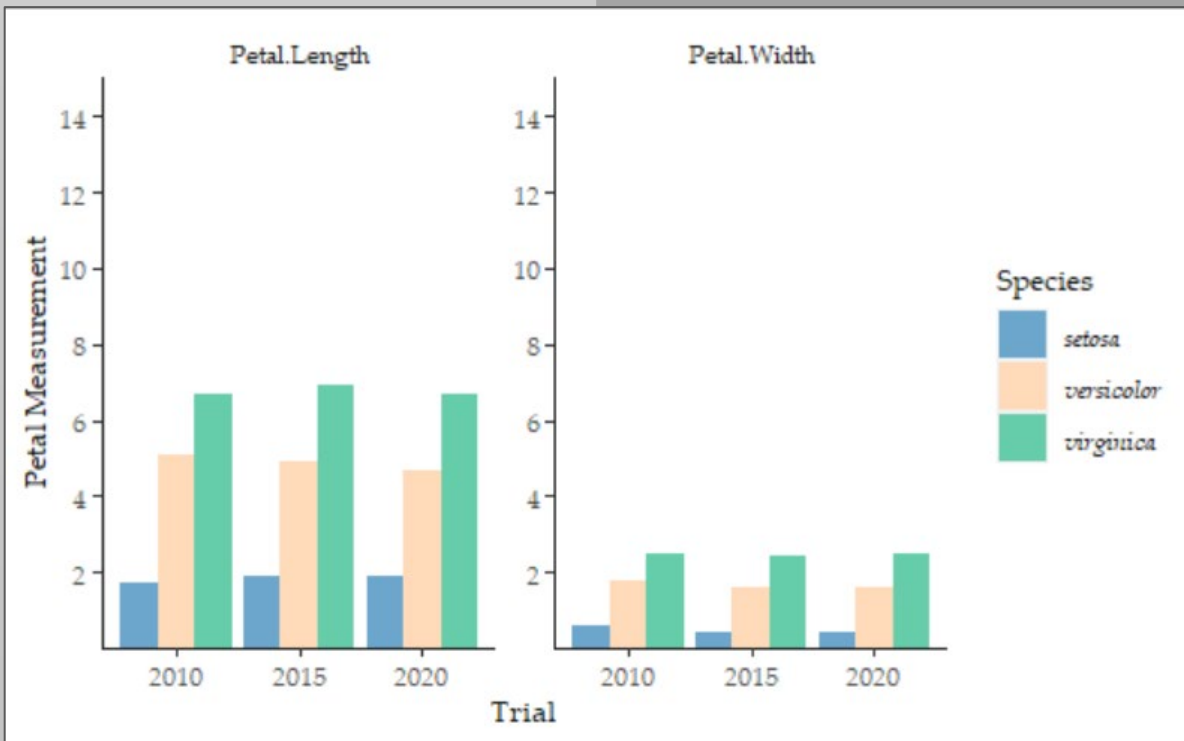


Sketch

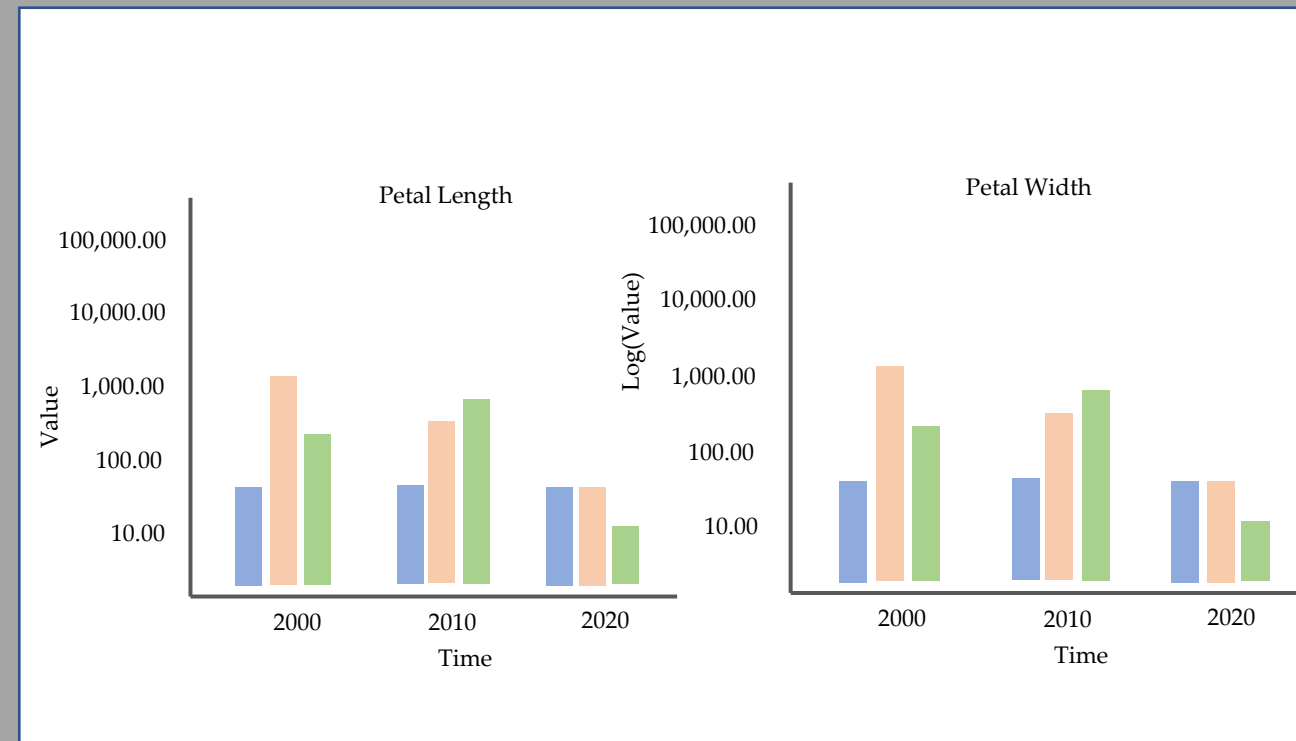


Story

III. Application

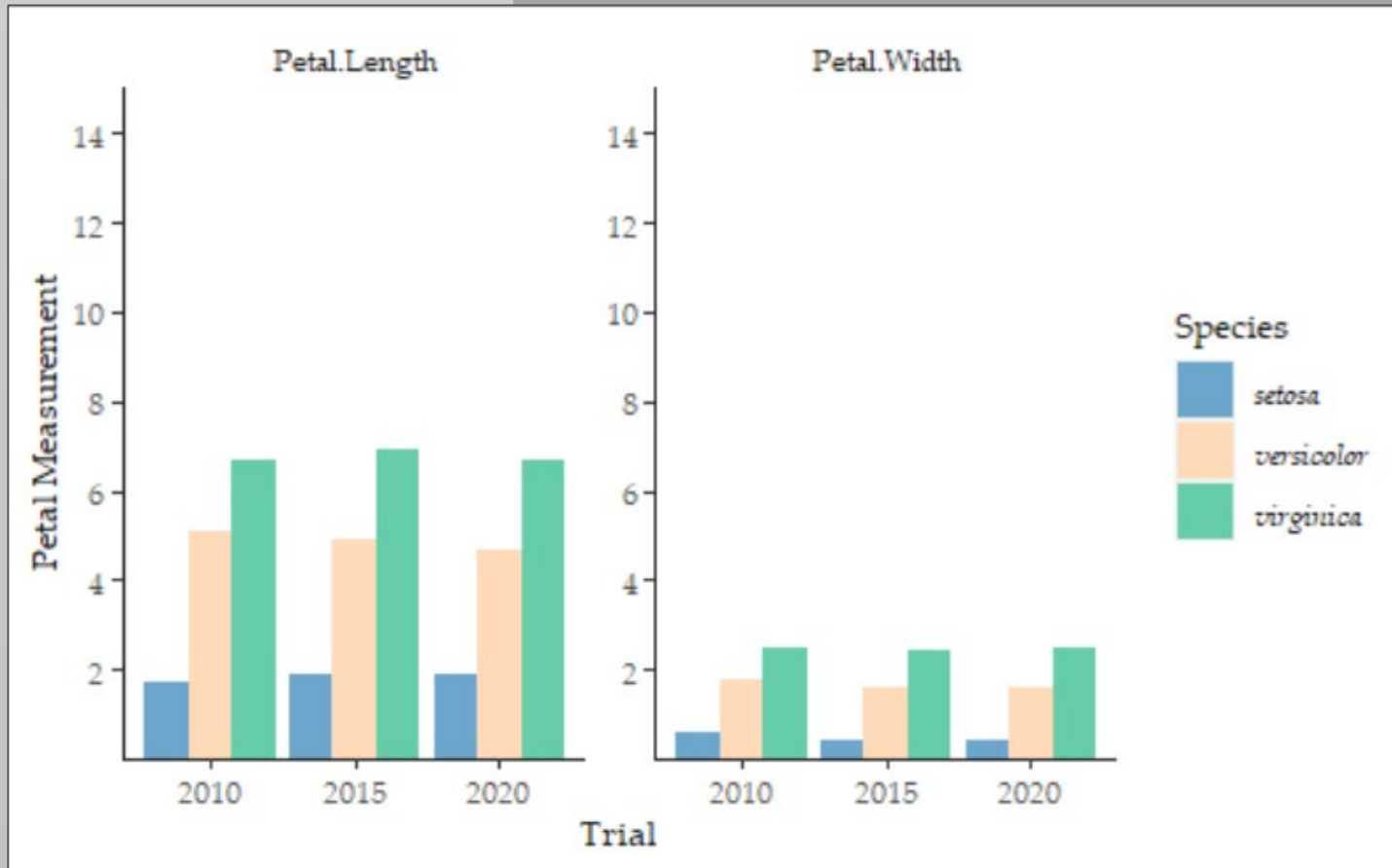


Polished



Story

III. Application



This is just one round of editing!
What else would you change???

Polished

Conclusions

Troubleshooting

- Class of data: counts or continuous data?
- Layer, layer, layer, order matters
- What are the arguments within the function
 - Change the default or consider an alternate geom/function
- Are you invoking a discrete call on continuous data?
- Did you set color when you meant fill?
- Did you specify the multiple arguments for the same item?
 - The last one will be what is seen, check your code.
- Is the default of the function to use a count transform?

Conclusions

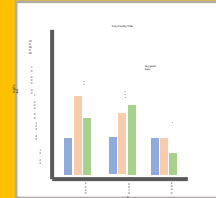
Resources for working in R

- ggplot2.tidyverse.org
- ggplot2-book.org/
- www.r-graph-gallery.com/
- tidytuesday podcast and webpage
- Esquisse and Colors Add-Ins
- Thomas lin Pedersen – ggplot2 – two-part series
- Stackoverflow

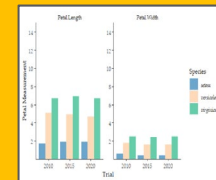
Conclusions

Today's Goal

- Understand how to develop graphics that:
 - Effectively tell a story
 - In ggplot2 (and some tidyverse)
 - Are refined or highly refined.



`ggplot(...)`



- Github: github.com/meghartwick/
- LinkedIn: [/meghan-hartwick-83291551/](https://www.linkedin.com/company/meghan-hartwick-83291551/)
- Twitter: [@HartwickMeghan](https://twitter.com/HartwickMeghan)

'The question in R is not if it can be done, but how.'

Questions?