R Graphics with ggplot2

Clay Ford

Fall 2017

About ggplot2

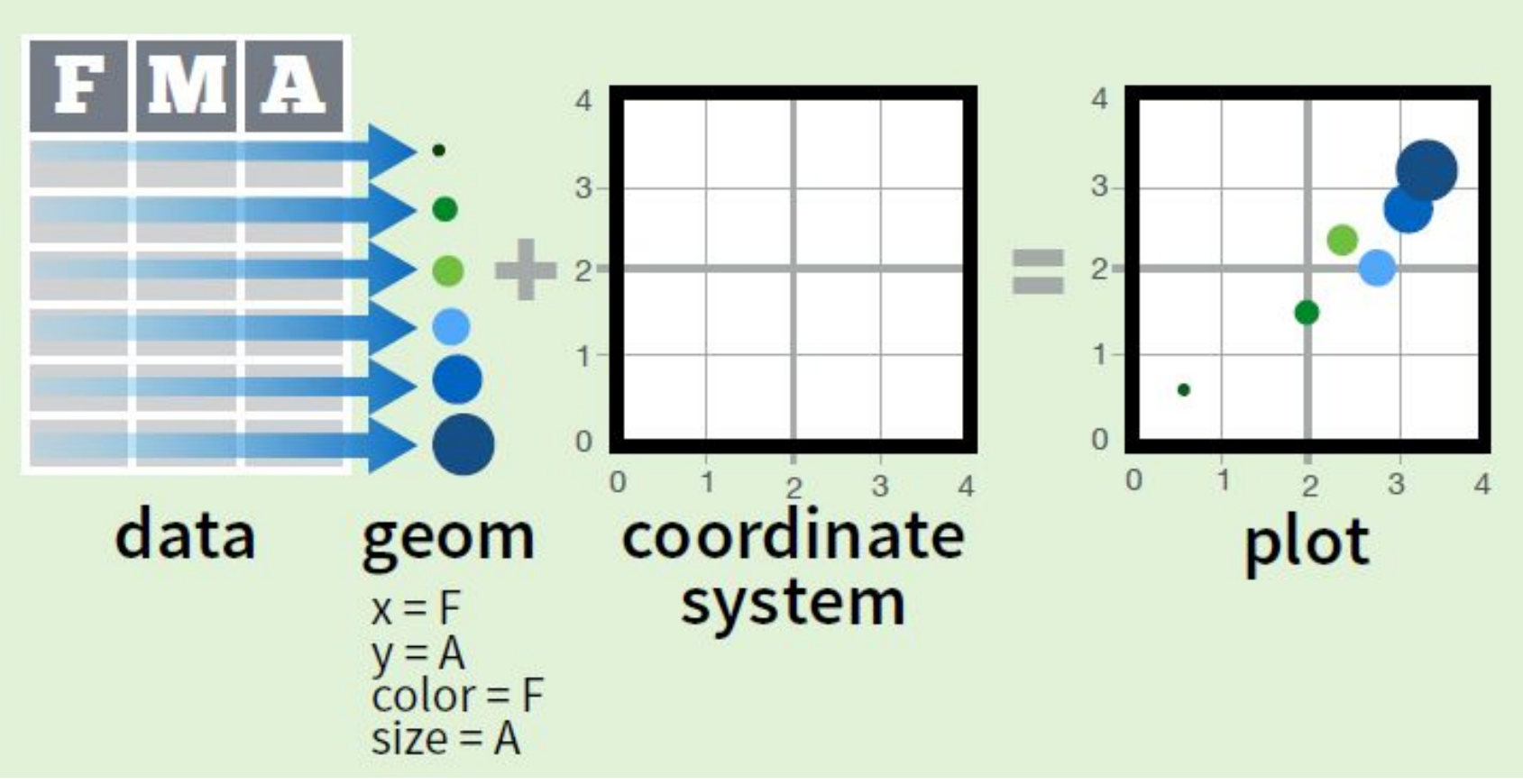
- Developed by Hadley Wickham in 2005.
- Implements the graphics scheme described in the book *The Grammar of Graphics* by Leland Wilkinson.
- Does not create interactive or 3D graphics.
- ggplot2 2.0 released in December 2015 with over 100 fixes and improvements; currently at version 2.2.1

The Grammar of Graphics

The *Grammar of Graphics* boiled down to 5 bullets, courtesy of Wickham (2016, p. 4):

- a statistical graphic is a mapping from data to **aes**thetic attributes (location, color, shape, size) of **geom**etric objects (points, lines, bars).
- the geometric objects are drawn in a specific **coord**inate system.
- **scale**s control the mapping from data to aesthetics and provide tools to read the plot (ie, axes and legends).
- the plot may also contain **stat**istical transformations of the data (means, medians, bins of data, trend lines).
- **facet**ing can be used to generate the same plot for different subsets of the data.

The Grammar of Graphics - illustration



github.com/rstudio/cheatsheets/raw/master/data-visualization-2.1.pdf

Basic ggplot2 syntax

Specify data, aesthetics and geometric shapes

`ggplot(data, aes(x=, y=, color=, shape=, size=)) +`
`geom_point()`, or `geom_histogram()`, or `geom_boxplot()`, etc.

- This combination is very effective for exploratory graphs.
- The data must be a data frame.
- The `aes()` function maps columns of the data frame to aesthetic properties of geometric shapes to be plotted.
- `ggplot()` defines the plot; the `geoms` show the data; layers are added with +
- Some examples should make this clear

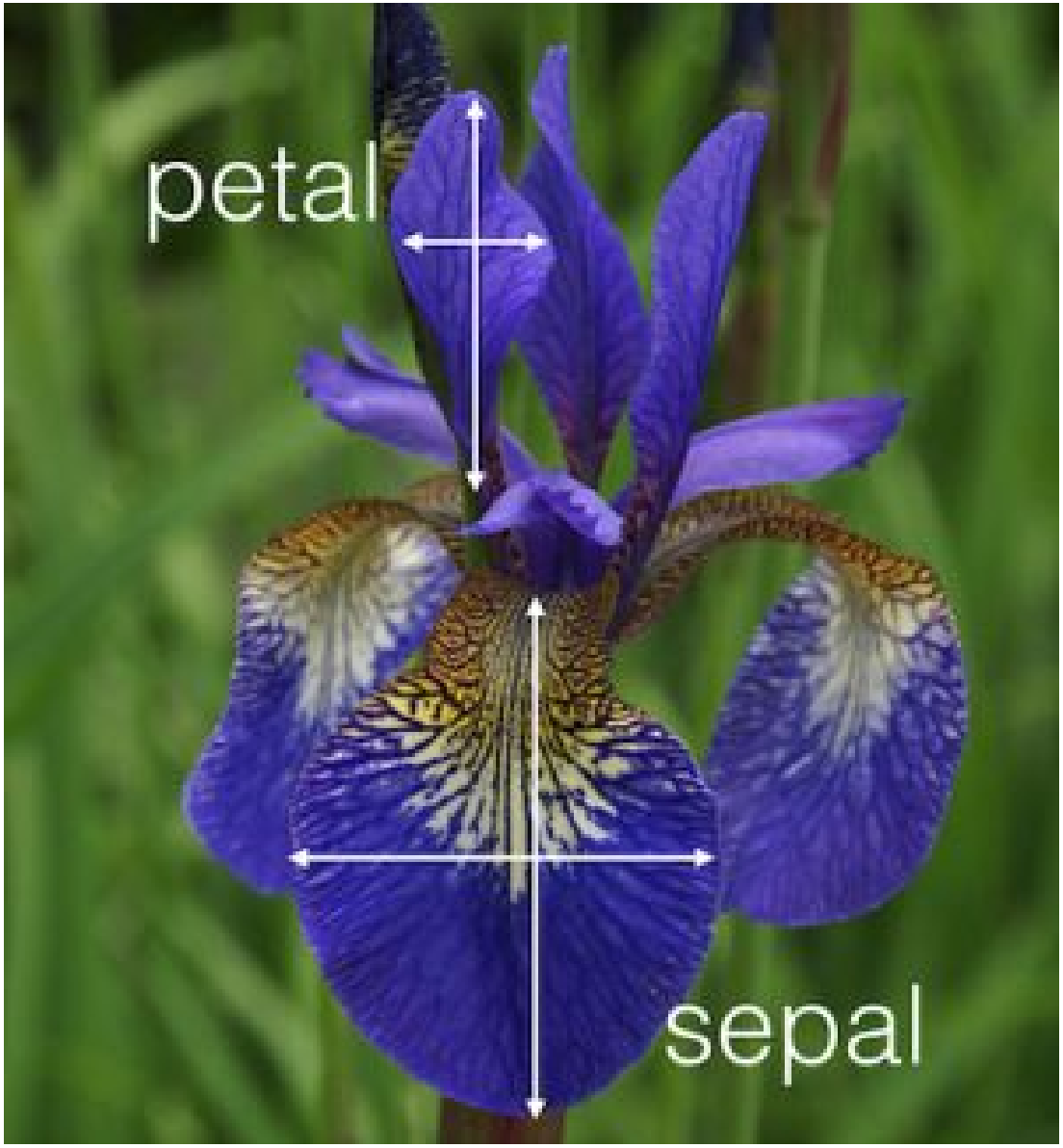
The iris data set

This is a famous data set from 1936, courtesy of Sir Ronald Fisher, that comes with R and is excellent for demonstrating ggplot2.

```
str(iris)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.
##  $ Species     : Factor w/ 3 levels "setosa","versicolor
```

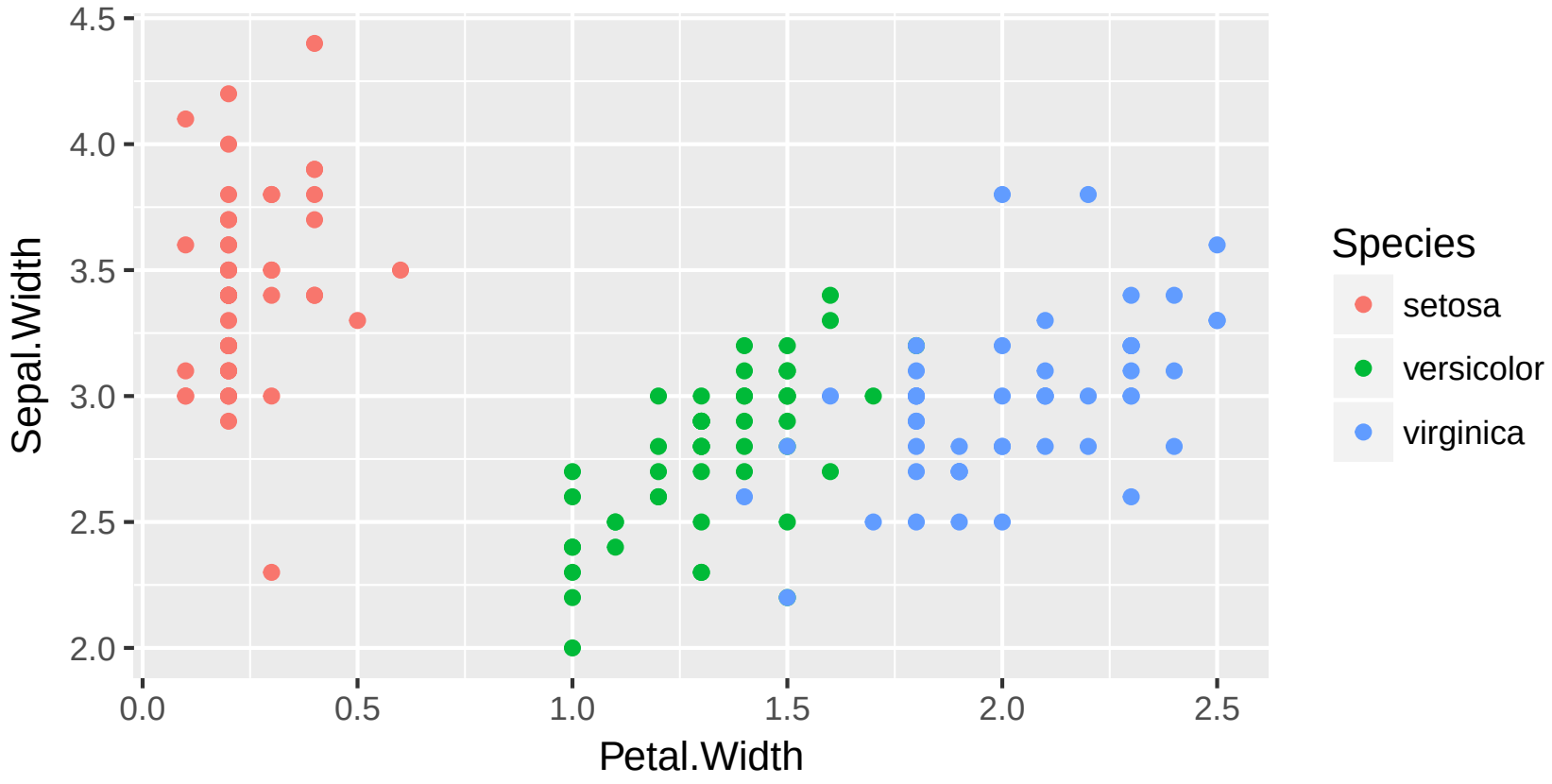
An iris



sebastianraschka.com/Images_old/2014_python_lda/

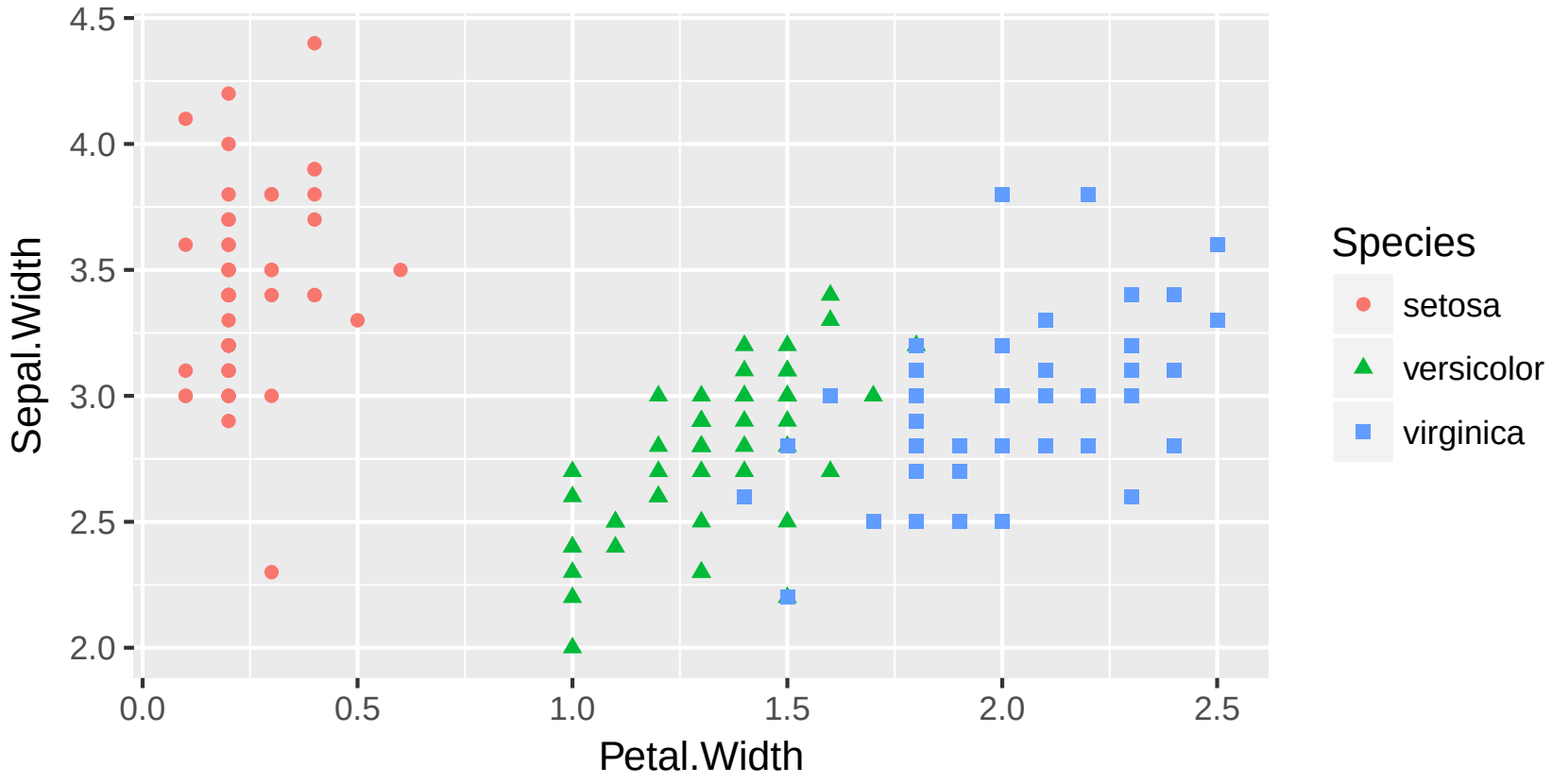
ggplot2 example - scatter plot coded by species

```
library(ggplot2) # once per session
ggplot(iris, aes(x=Petal.Width, y=Sepal.Width,
                 color=Species)) + geom_point()
```



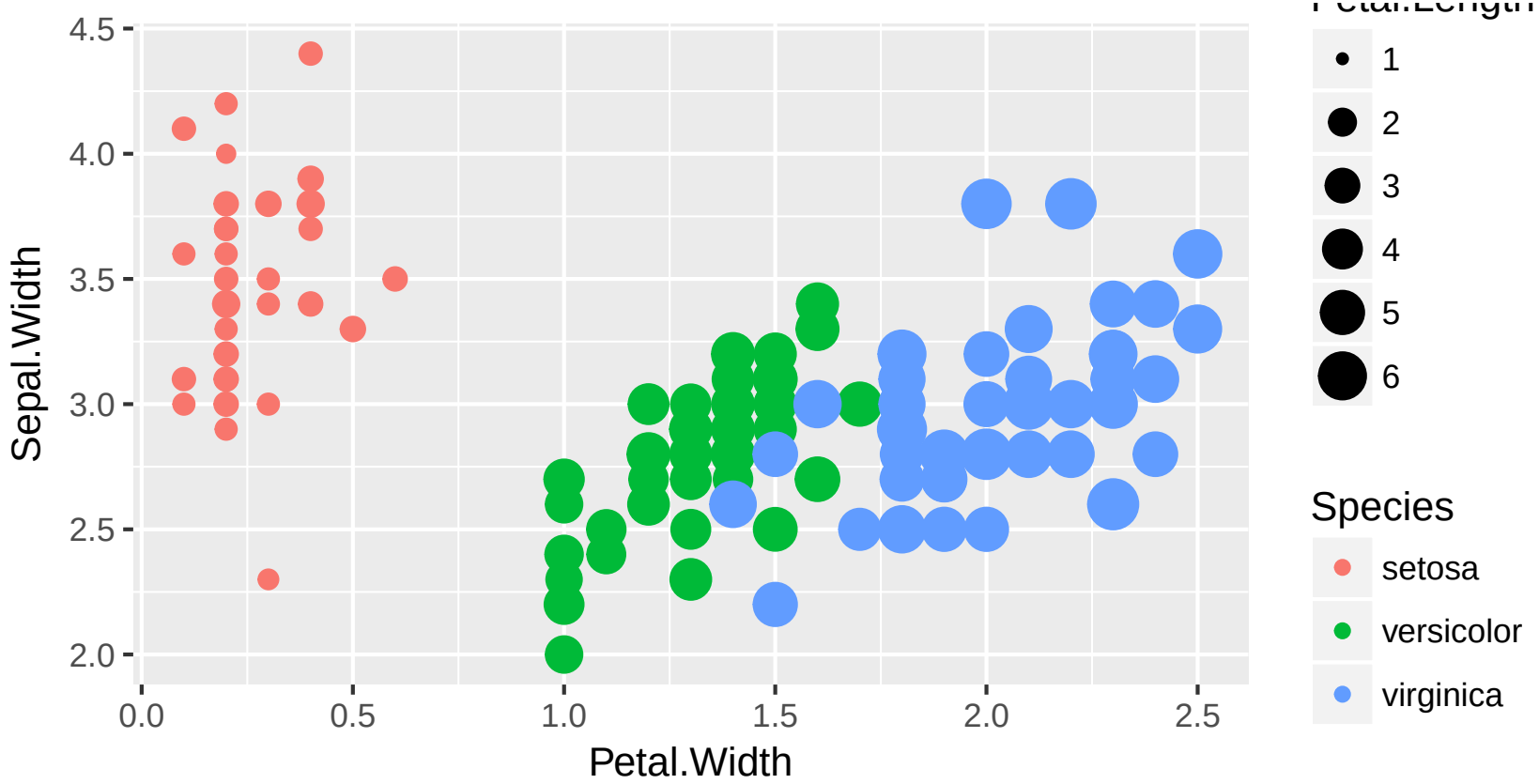
ggplot2 example - scatter plot coded by species

```
ggplot(iris, aes(x=Petal.Width, y=Sepal.Width,
                 color=Species, shape=Species)) +
  geom_point()
```



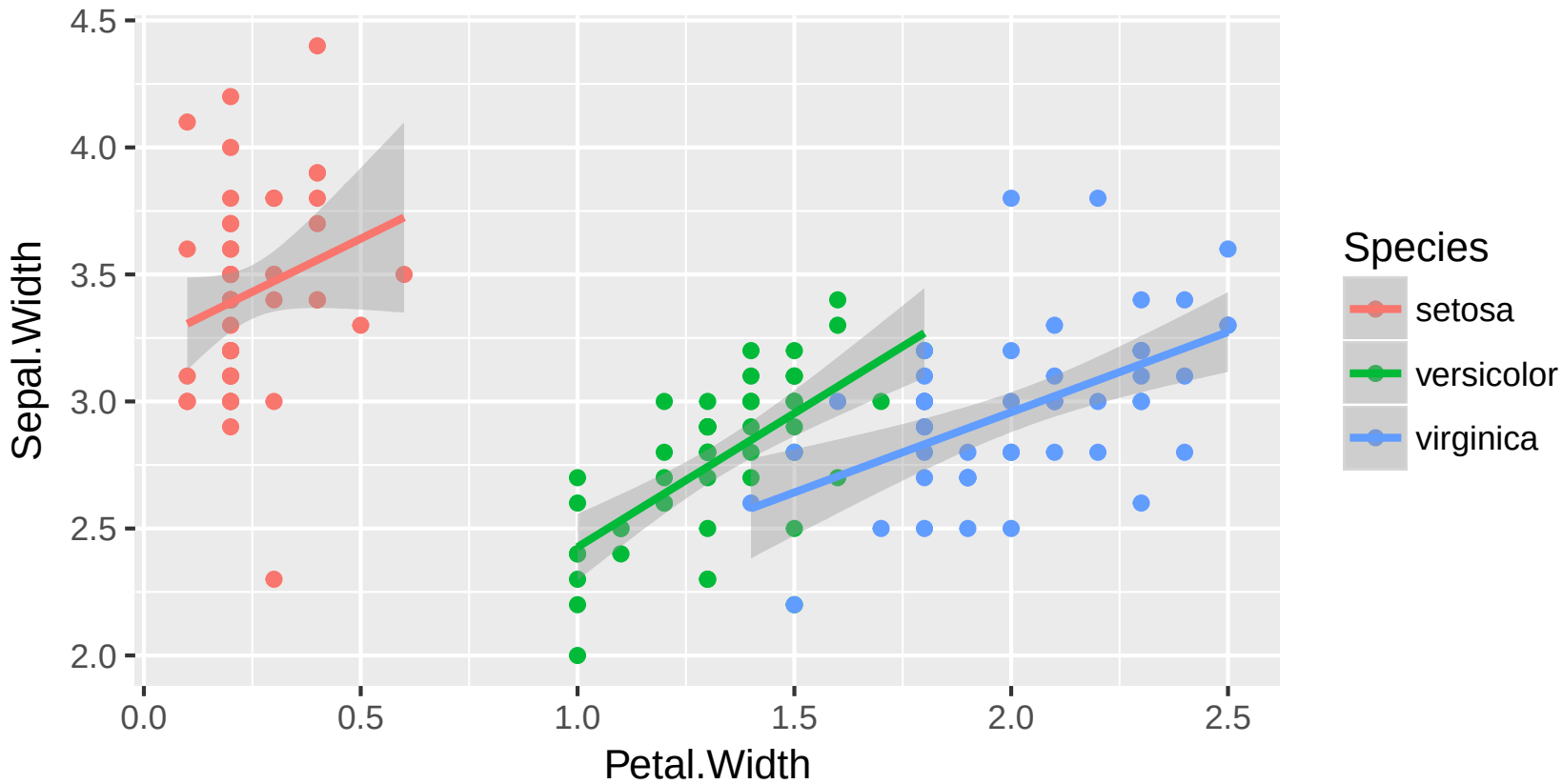
ggplot2 example - scatter plot coded by species, size by Petal.Length

```
ggplot(iris, aes(x=Petal.Width, y=Sepal.Width,
                 color=Species, size=Petal.Length)) +
  geom_point()
```



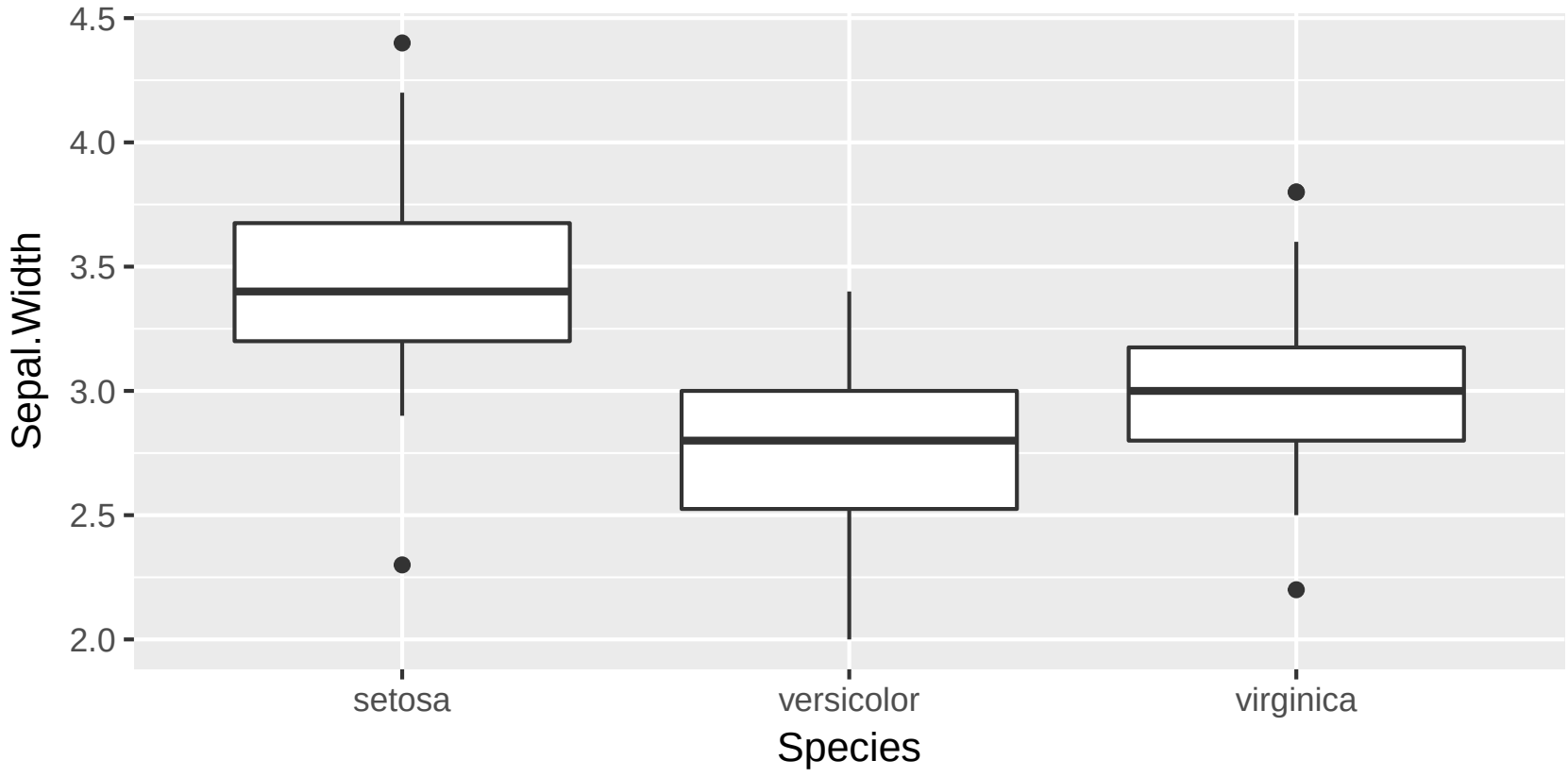
ggplot2 example - add multiple geoms (points and smooth line)

```
ggplot(iris, aes(x=Petal.Width, y=Sepal.Width,
                 color=Species)) +
  geom_point() + geom_smooth(method="lm")
```



ggplot2 example - boxplot (statistical transformation)

```
ggplot(iris, aes(x=Species, y=Sepal.Width)) +
  geom_boxplot()
```



Moving beyond ggplot + geoms

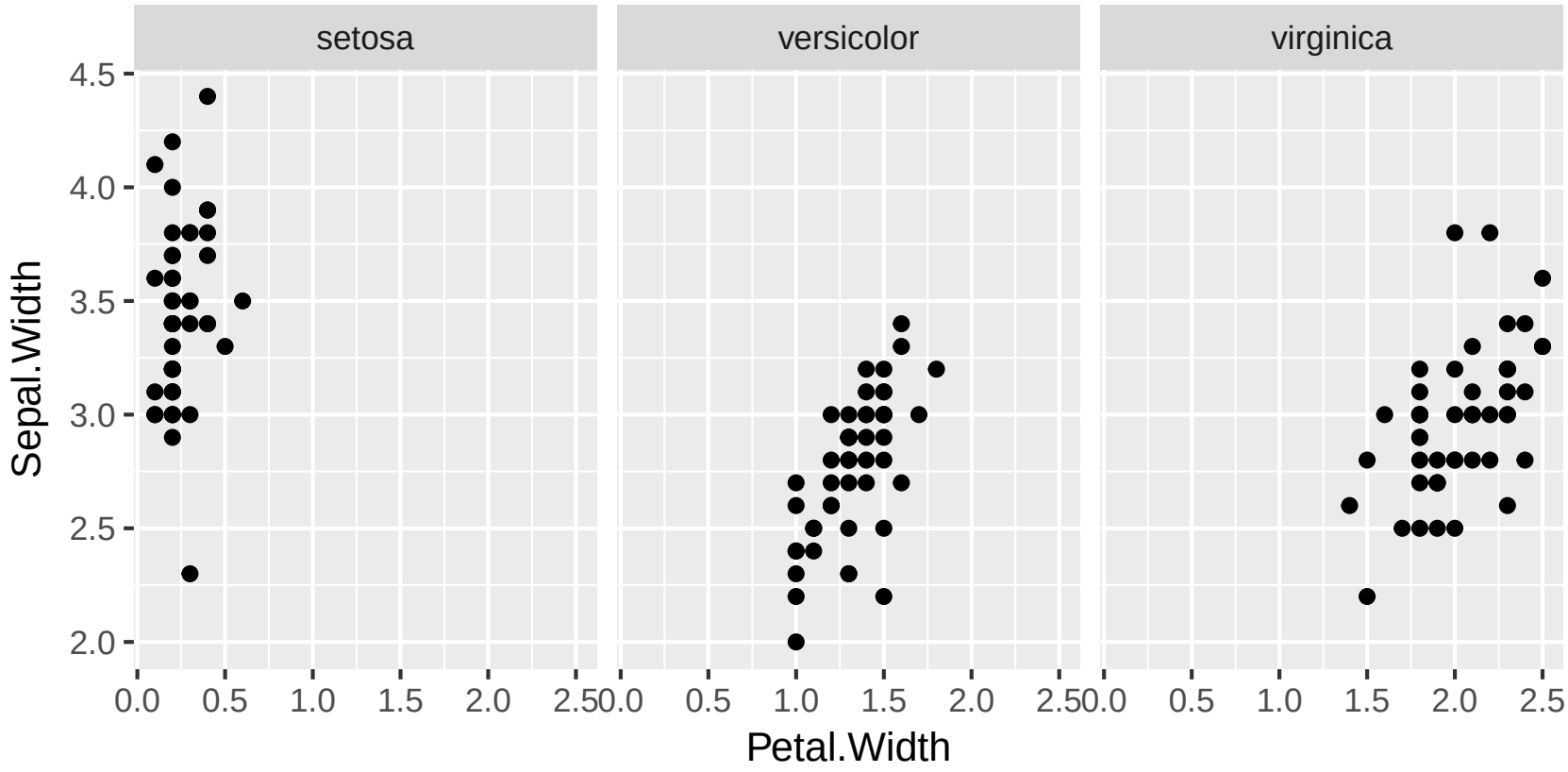
- A natural next step in exploratory graphing is to create plots of subsets of data. These are called facets in ggplot2.
- Use `facet_wrap()` if you want to facet by one variable and have ggplot2 control the layout. Example:
  - `+ facet_wrap( ~ var)`
- Use `facet_grid()` if you want to facet by one and/or two variables and control layout yourself.

Examples:

+ `facet_grid(. ~ var1)` - facets in columns
+ `facet_grid(var1 ~ .)` - facets in rows
+ `facet_grid(var1 ~ var2)` - facets in rows and columns

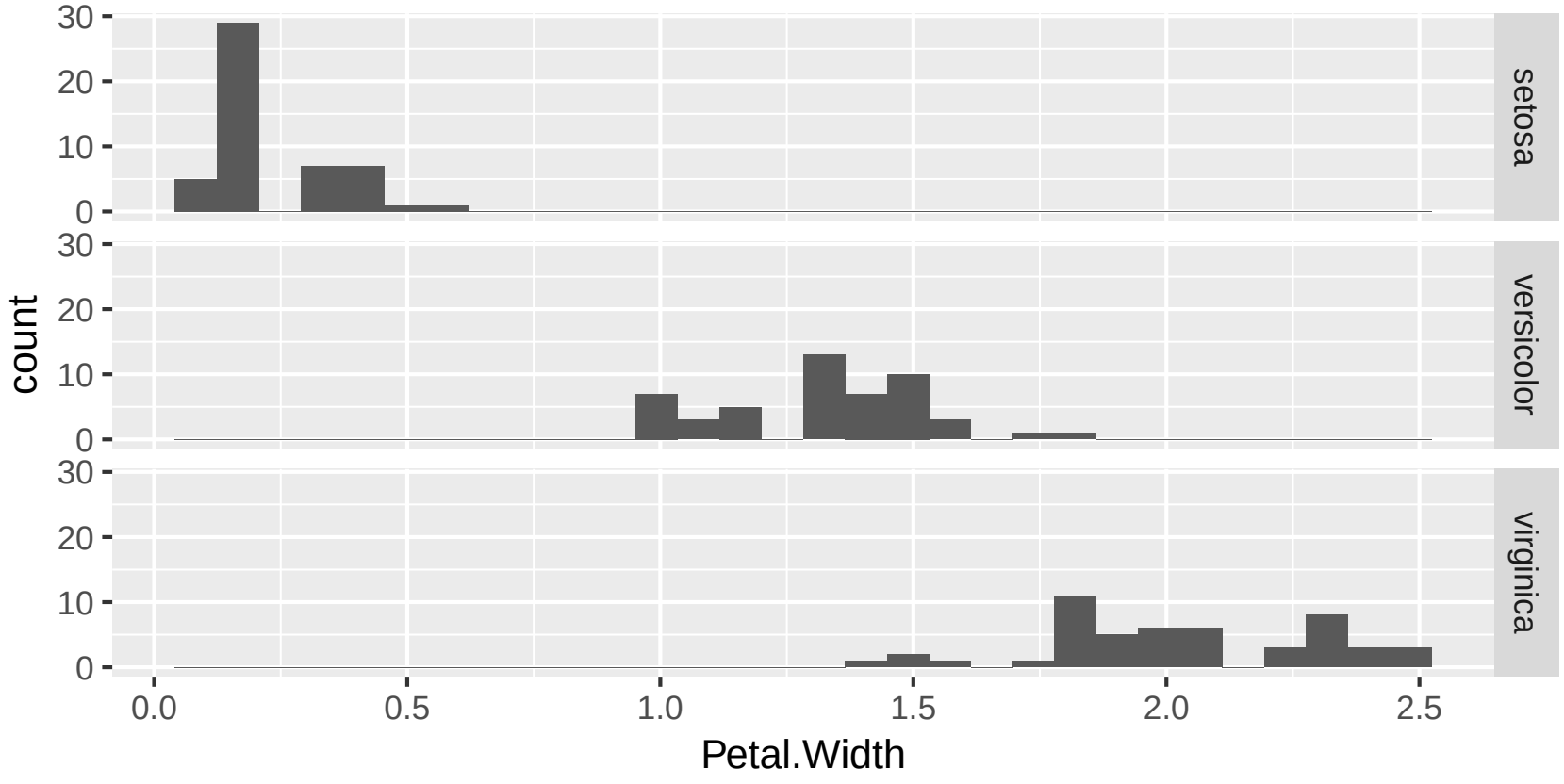
ggplot2 example - facet_wrap

```
ggplot(iris, aes(x=Petal.Width, y=Sepal.Width)) +
  geom_point() + facet_wrap(~ Species)
```



ggplot2 example - facet_grid (histograms)

```
ggplot(iris, aes(x=Petal.Width)) +
  geom_histogram() + facet_grid(Species ~ .)
```

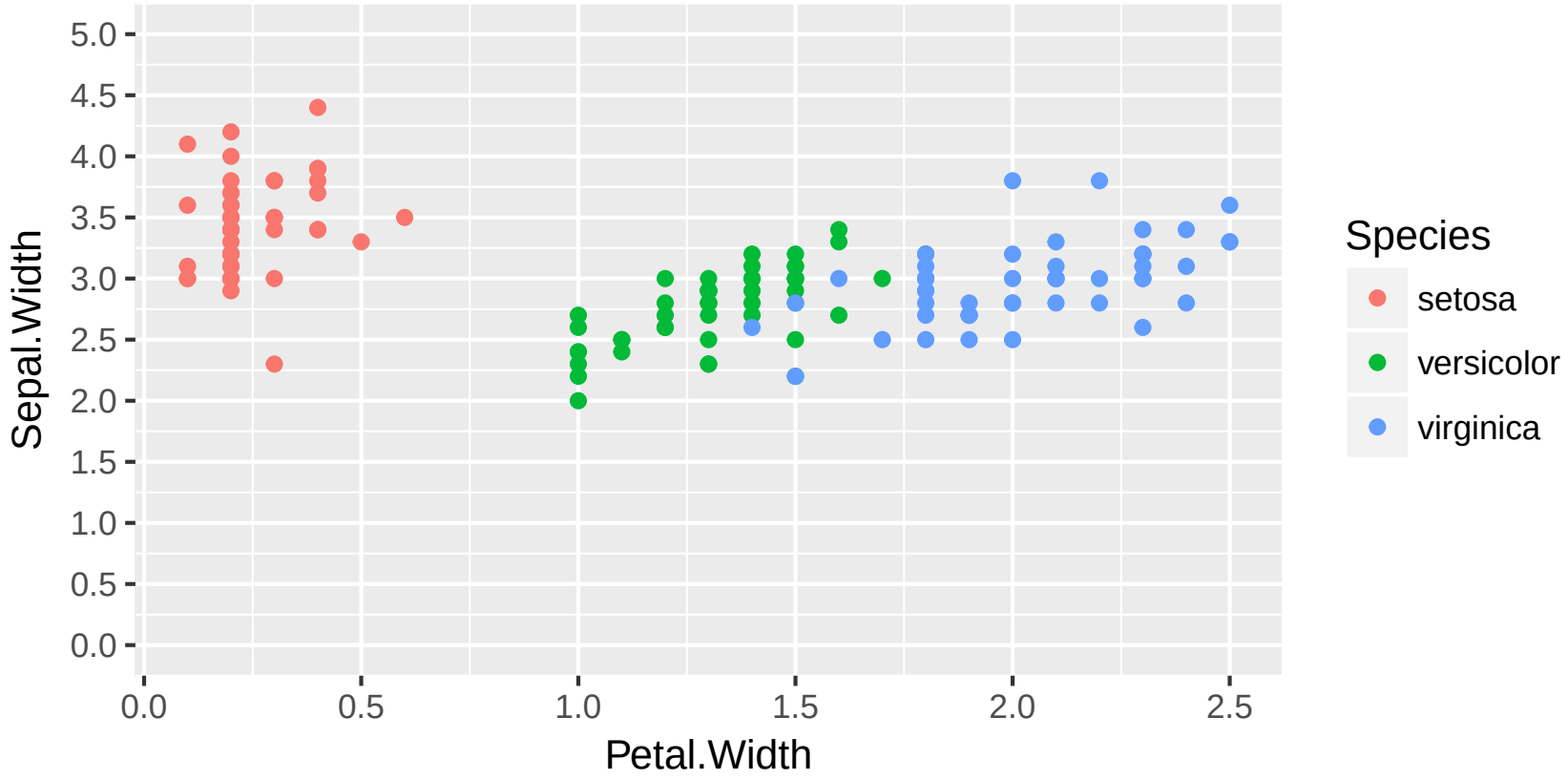


Customizing scales

- Scales control the mapping from data to aesthetics and provide tools to read the plot (ie, axes and legends).
- Every aesthetic has a default scale. To modify a scale, use a `scale` function.
- All scale functions have a common naming scheme: `scale` _ name of aesthetic _ name of scale
- Examples: `scale_y_continuous`, `scale_color_discrete`, `scale_fill_manual`
- Heads up: The documentation for `ggplot2` scale functions will frequently use functions from the `scales` package (also by Wickham)!

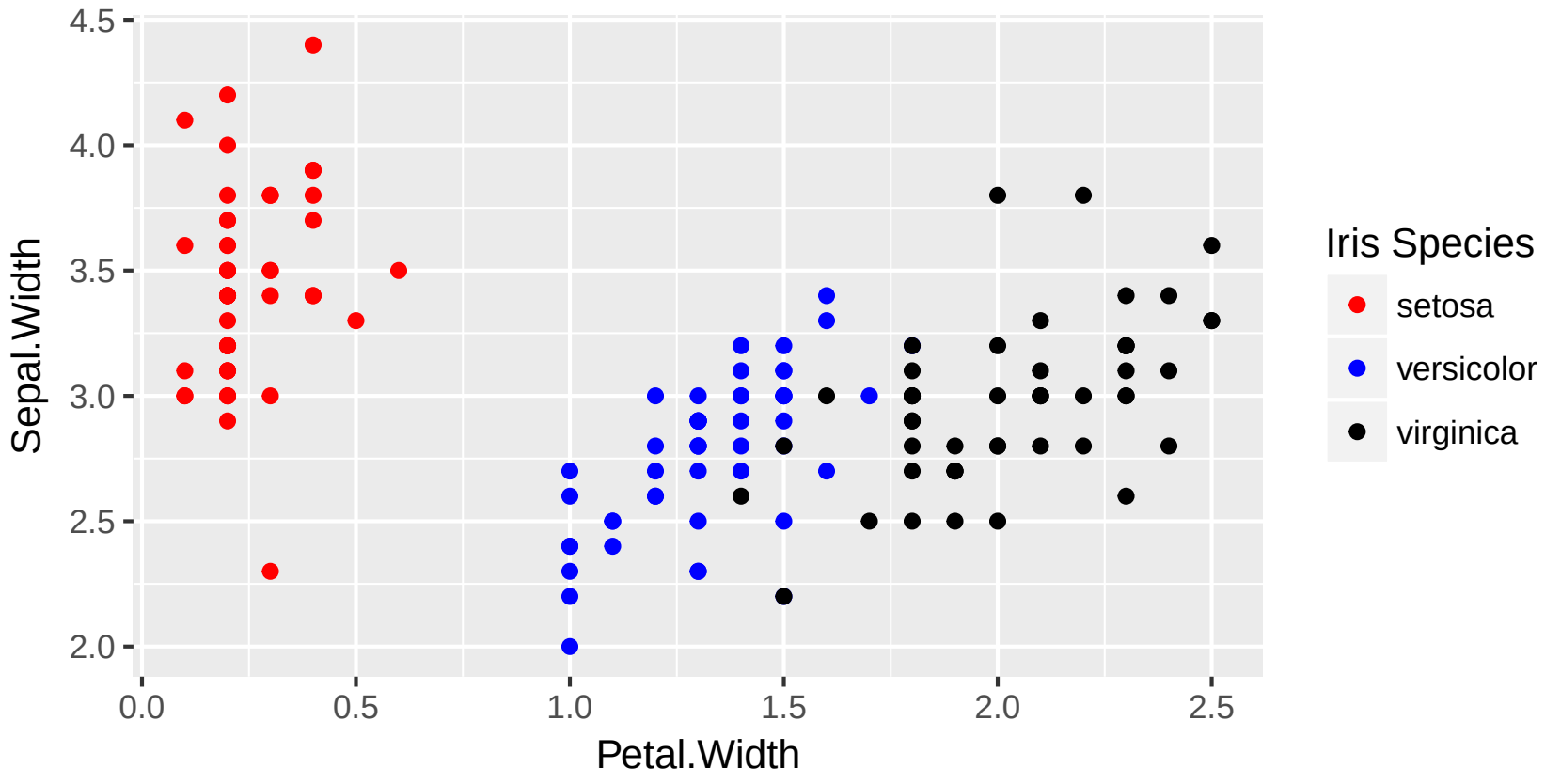
ggplot2 example - update scale for y-axis

```
ggplot(iris, aes(x=Petal.Width, y=Sepal.Width,
                 color=Species)) + geom_point() +
  scale_y_continuous(limits=c(0,5), breaks=seq(0,5,0.5))
```



ggplot2 example - update scale for color

```
ggplot(iris, aes(x=Petal.Width, y=Sepal.Width,
                 color=Species)) + geom_point() +
  scale_color_manual(name="Iris Species",
                     values=c("red","blue","black"))
```

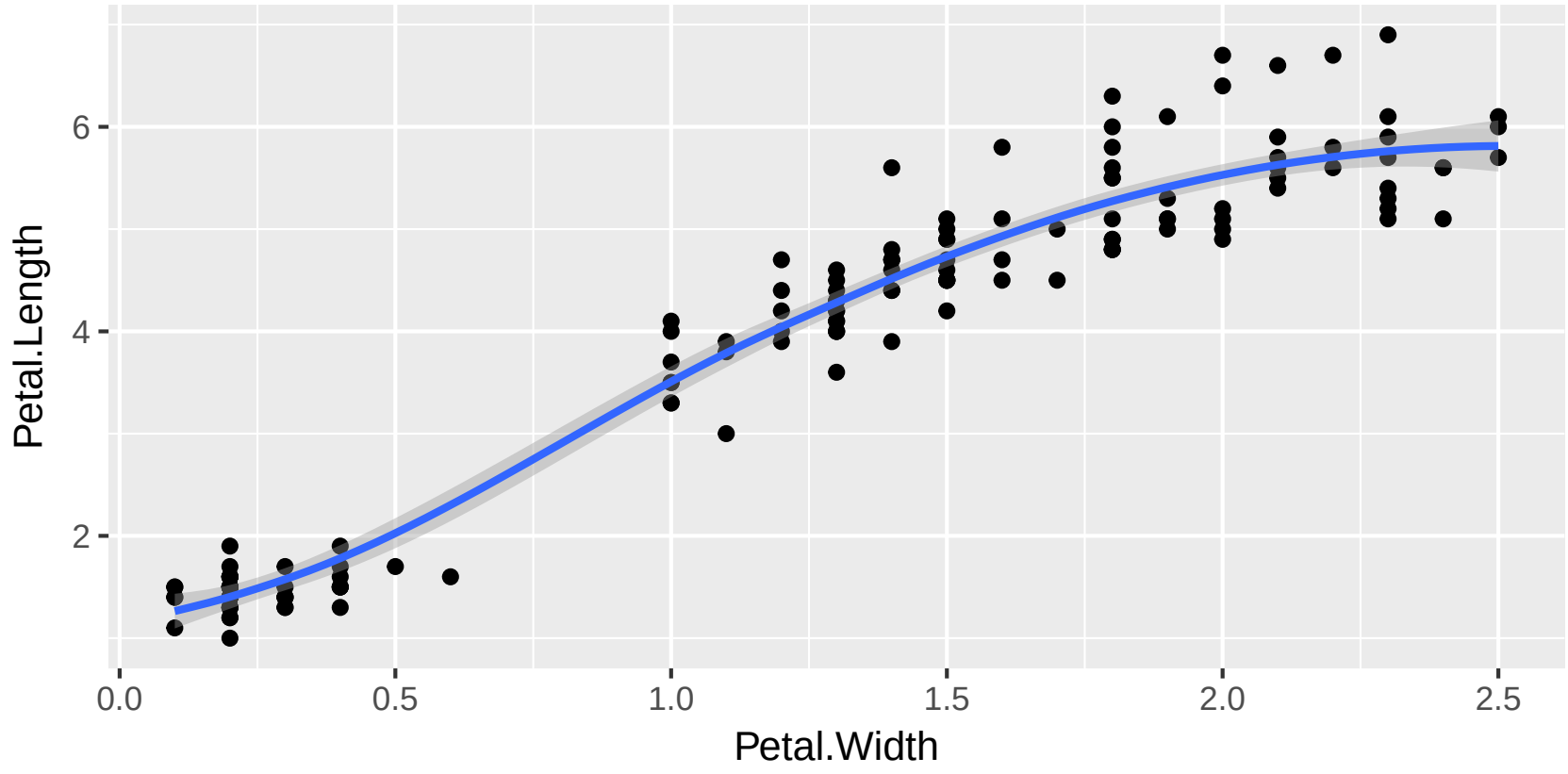


stat functions

- All `geoms` perform a default statistical transformation.
- For example, `geom_histogram()` bins the data before plotting. `geom_smooth()` fits a line through the data according to a specified method.
- In some cases the transformation is the "identity", which just means plot the raw data. For example, `geom_point()`
- These transformations are done by `stat` functions. The naming scheme is `stat_` followed by the name of the transformation. For example, `stat_bin`, `stat_smooth`, `stat_boxplot`
- **Every geom has a default stat, every stat has a default geom.**

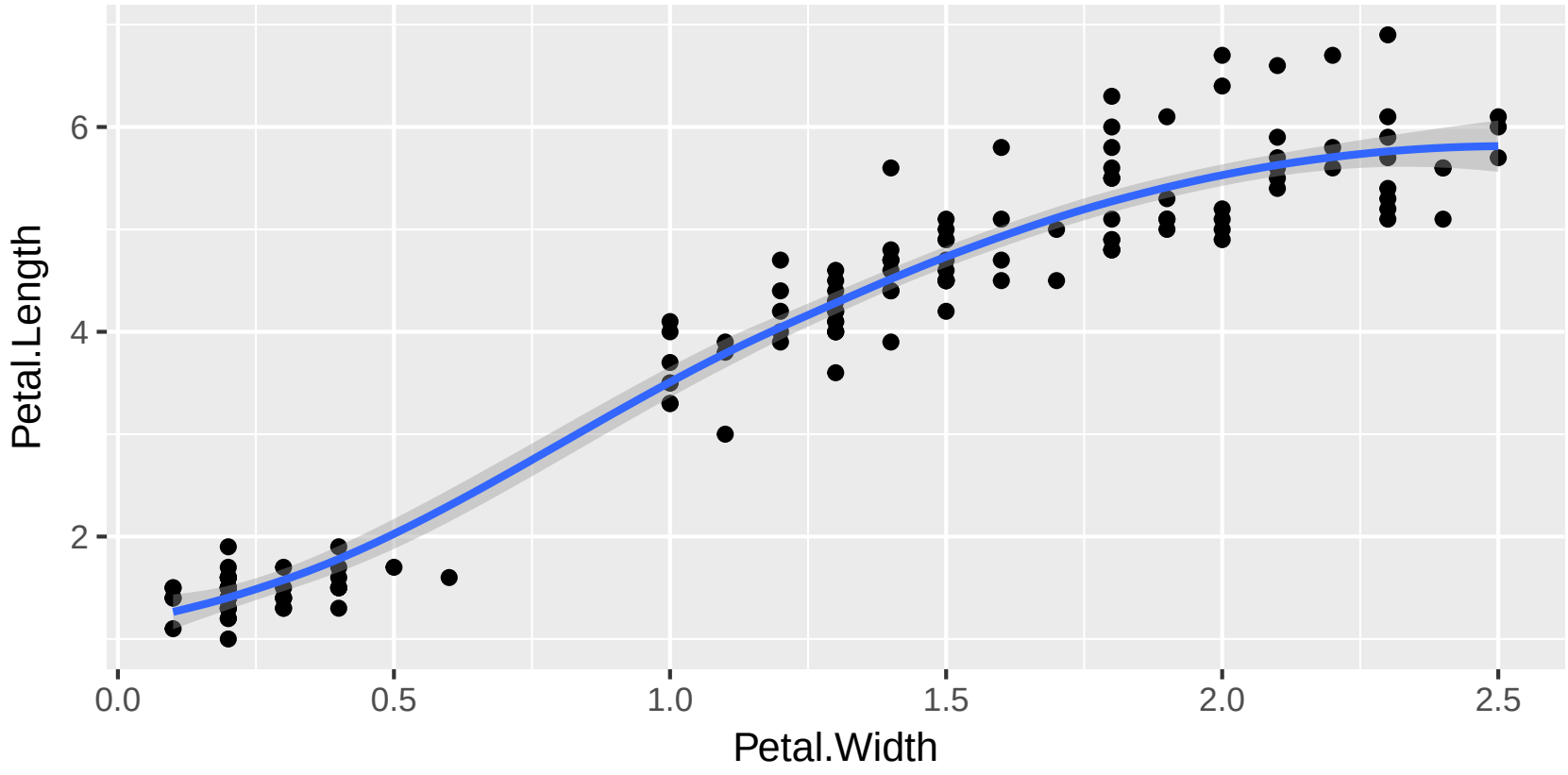
ggplot2 example - geom using default stat

```
ggplot(iris, aes(x=Petal.Width, y=Petal.Length)) +
  geom_point() + geom_smooth()
```



ggplot2 example - stat using default geom

```
ggplot(iris, aes(x=Petal.Width, y=Petal.Length)) +
  geom_point() + stat_smooth()
```



Why should I care about `stat` versus `geom`?

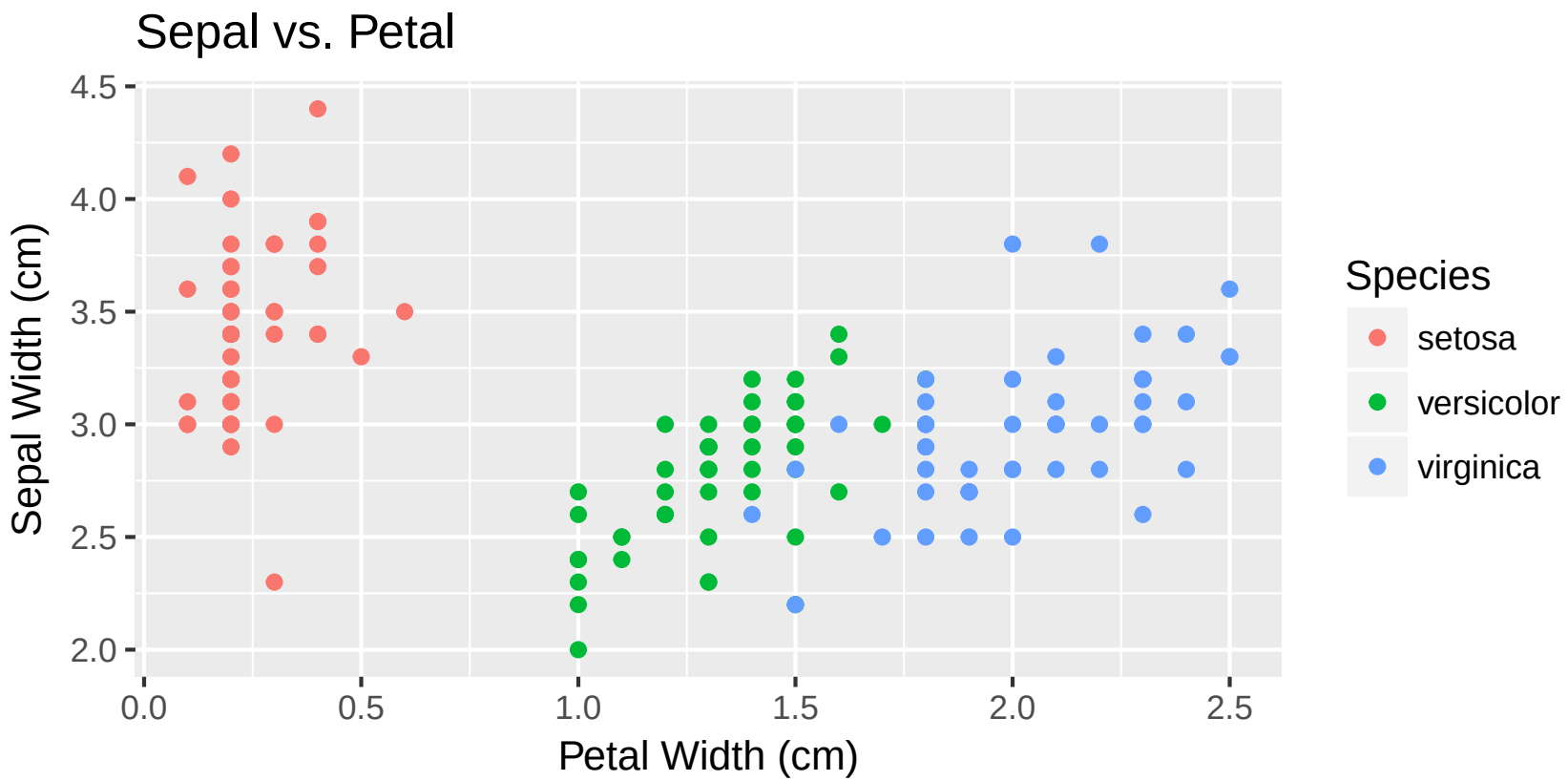
- **The stat and geom functions can use each other's arguments.**
- When consulting the documentation for a particular `geom` you'll notice there is also documentation for an associated `stat`, and vice versa. (an exception is `geom_point` and `stat_identity`.)
- Understanding how geoms and statistical transformations work together in ggplot2 can help you master the syntax faster.

Update themes and labels

- The default ggplot2 theme is excellent. It follows the advice of several landmark papers regarding statistics and visual perception. (Wickham 2016, p. 176)
- However you can change the theme using ggplot2's themeing system. To date, there are seven built-in themes: `theme_gray` (*default*), `theme_bw`, `theme_linedraw`, `theme_light`, `theme_dark`, `theme_minimal`, `theme_classic`
- You can also update axis labels and titles using the `labs` function.

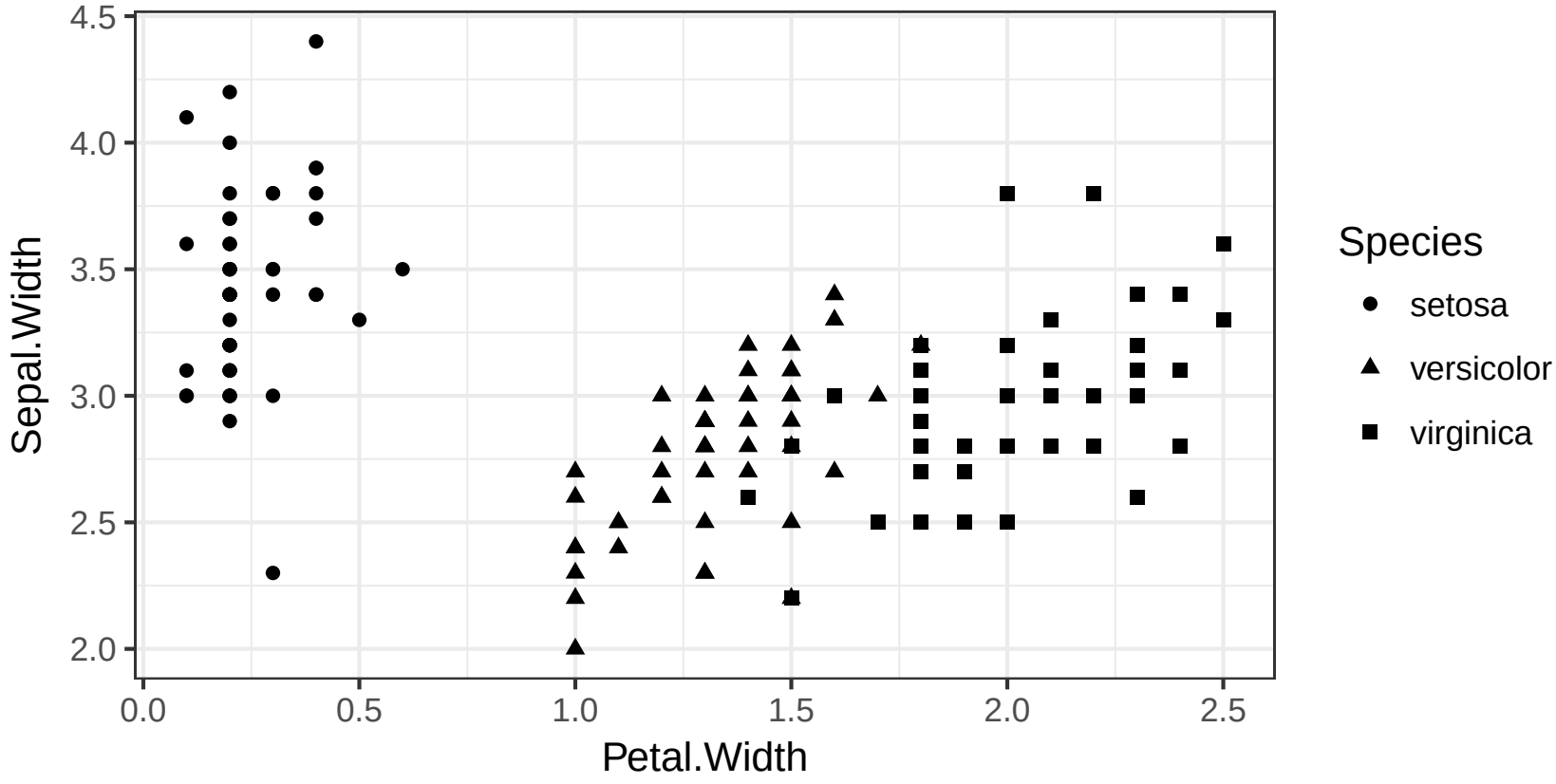
ggplot2 example - update labels

```
ggplot(iris, aes(x=Petal.Width, y=Sepal.Width,
                 color=Species)) + geom_point() +
  labs(title="Sepal vs. Petal",
       x="Petal Width (cm)", y="Sepal Width (cm)")
```



ggplot2 example - change theme

```
ggplot(iris, aes(x=Petal.Width, y=Sepal.Width,
                 shape=Species)) + geom_point() +
  theme_bw()
```



ggplot2 - some tips

- Can do a lot with `ggplot(data, aes()) + geom`!
- Data must be a data frame (not a matrix or collection of vectors)
- The `ggplot2` documentation has many good examples
- Prepare to invest some time if you want master ggplot2; the RStudio ggplot2 cheat sheet can help.

Let's go to R!

References and further study

- Chang, W. (2013), *R Graphics Cookbook*, O'Reilly.
- Wickham, H. (2016), *ggplot2: Elegant Graphics for Data Analysis* (2nd ed), Springer.
- Wickham, H. and Grolemund G. (2017), *R for Data Science*. O'Reilly. `http://r4ds.had.co.nz/`

ggplot2 cheat sheet

`https://github.com/rstudio/cheatsheets/raw/master/data-visualization-2.1.pdf`

Cookbook for R - Graphs

`http://www.cookbook-r.com/Graphs/`

Official ggplot2 web site

`http://ggplot2.org/`

Thanks for coming

- For help and advice with your data analysis: `statlab@virginia.edu`
- Sign up for more workshops or see past workshops: `http://data.library.virginia.edu/training/`
- Register for the Research Data Services newsletter to be notified of new workshops: `http://data.library.virginia.edu/newsletters/`