ggPlot2

Walid Gharib

adapted from Geoffrey Fucile

SIB
Swiss Institute of Bioinformatics

Why use ggplot2?

- High level of abstraction for plot specification

- Often looks nicer and is more flexible

- Many users, active mailing lists

- Consistent underlying "grammar of graphics"
  - L. Wilkinson 2005, Springer

ggplot2 – grammar of graphics

- Specify "building blocks" of a plot:
  - Data **(must be data frame)** → noun
  - Aesthetic mapping → adjectives
  - Geometric object → verb
  - Statistical transformation → verb
  - Position adjustment → preposition
  - Many others…

- All components combine to make a layer → sentence

- Can place layers on top of each other → paragraph

ggPlot2 – Building Plots Iteratively

> data(iris)

> head(iris)

- We would like to plot the Sepal length against the Petal length to see if there is any correlation

> ggplot(data = iris, aes(x = Sepal.Length , y=Petal.Length))

Why is the plot empty?

ggPlot2 – Building Plots Iteratively

- Add geoms, graphical representations of the data using points, lines, bars, etc

> ggplot(data = iris, aes(x = Sepal.Length , y=Petal.Length)) + **geom_point()**

- The + in ggplot2 is useful as it allows you to modify existing ggplot objects (clear plots first):

> surveys_plot <- ggplot(data = iris, aes(x = Sepal.Length , y=Petal.Length))

> surveys_plot + geom_point()

ggPlot2 – Building Plots Iteratively

- Add transparency (alpha) to avoid overplotting:

```
> ggplot(data = iris, aes(x = Sepal.Length , y=Petal.Length)) +
geom_point(alpha = 0.1)
```

- Add colors:

```
> ggplot(data = iris, aes(x = Sepal.Length , y=Petal.Length)) +
geom_point(alpha = 0.1, color = "blue")
```

```
> ggplot(data = iris, aes(x = Sepal.Length , y=Petal.Length)) +
geom_point(alpha = 0.1, aes(color = species_id))
```

ggPlot2 – Boxplots

- Visualize the distribution of weights by species

> ggplot(data = iris, aes(x = Sepal.Length , y=Petal.Length)) + geom_boxplot()

- Add points to the boxplot to visualize number of measurements and their distribution:

> ggplot(data = iris, aes(x = Sepal.Length , y=Petal.Length)) + geom_boxplot(alpha = 0, color = "tomato") +

 geom_jitter(alpha = 0.3, color = "tomato")

Histograms – probability densities

> ggplot(data=iris, aes(x=Sepal.Length, **..density..**)) + geom_histogram(**bins=20,** colour="black", fill ="yellow") + geom_density(color="blue") + ggtitle("Sepal.Length distribution")
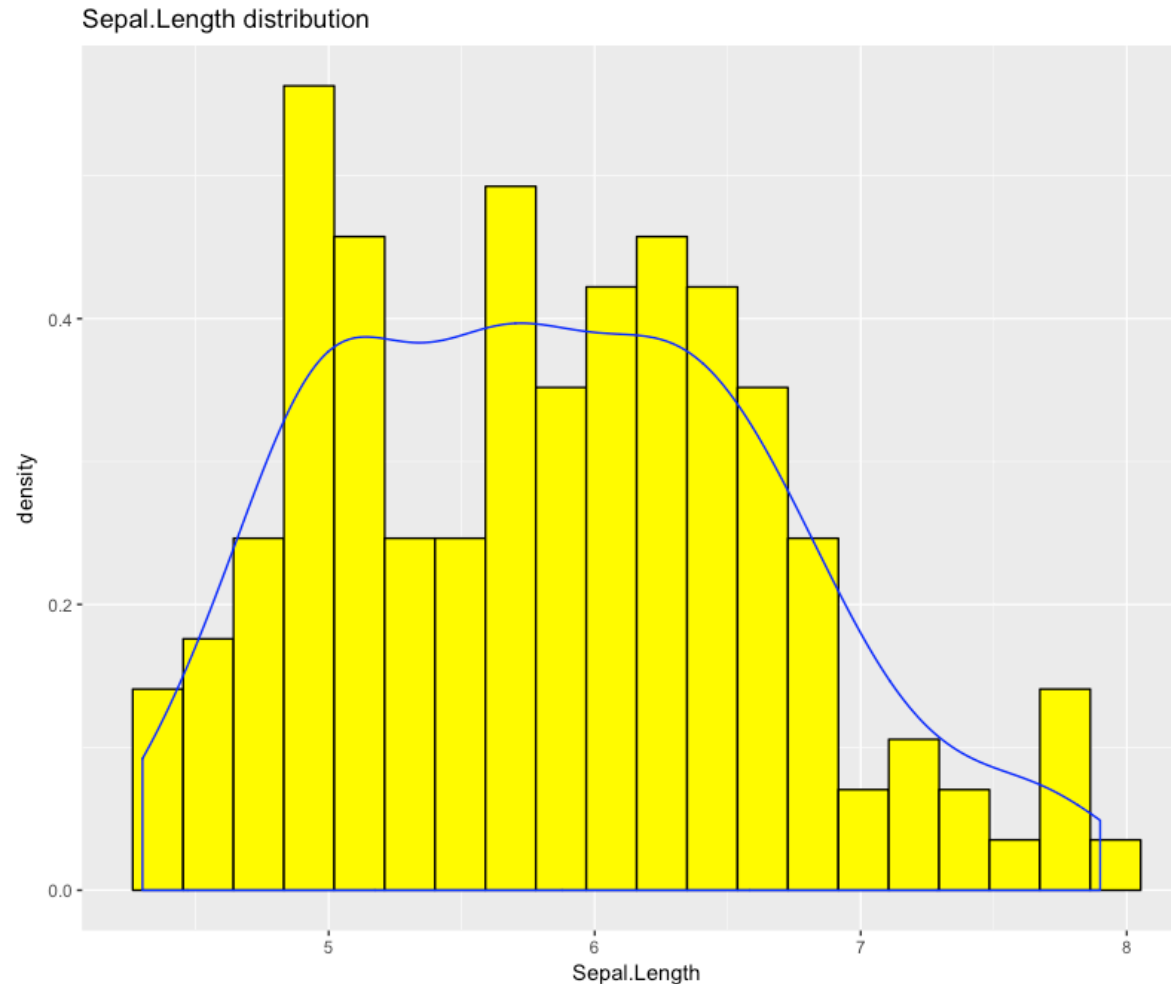
ggPlot2 – grouping and lm() integration

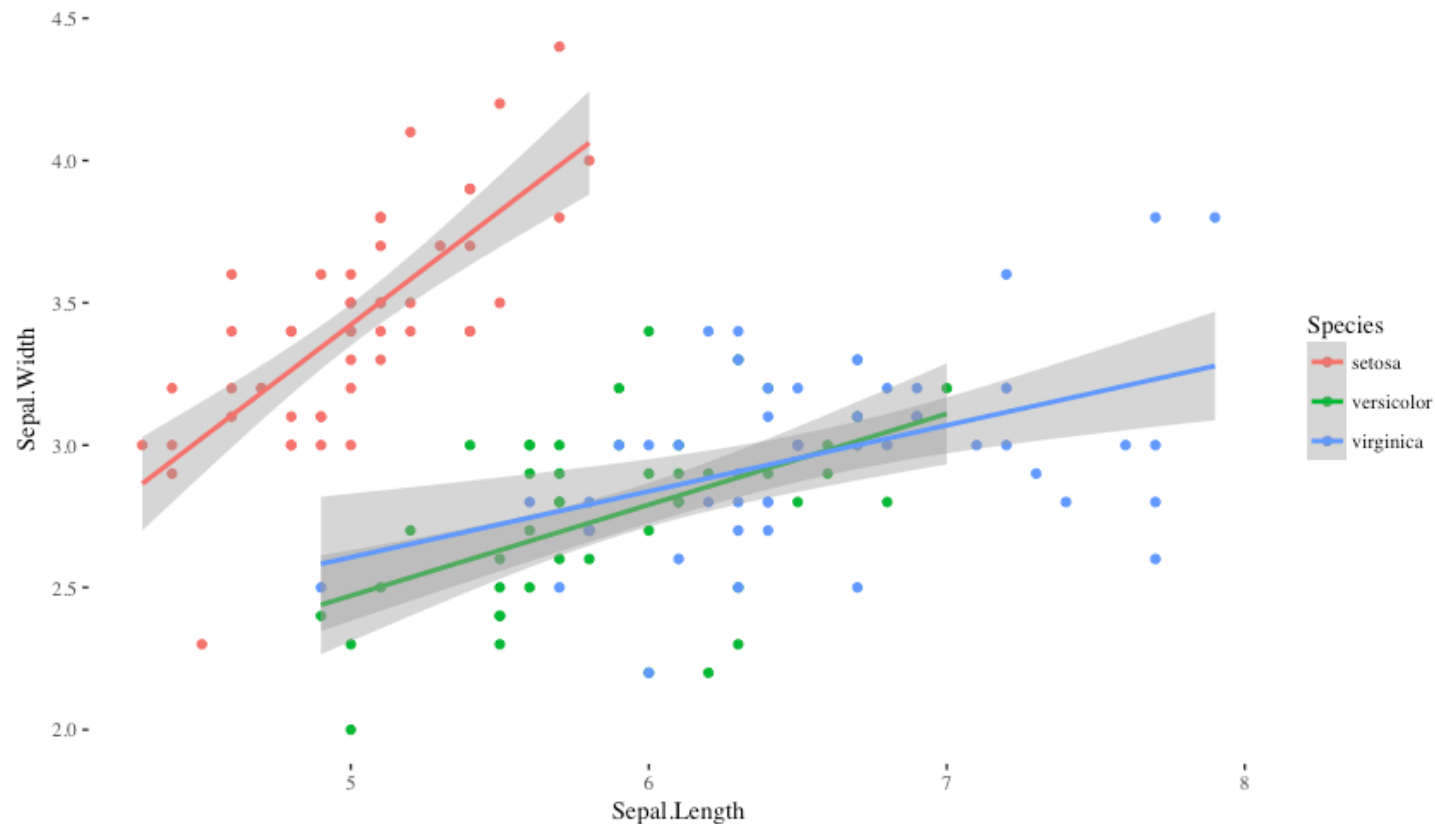> ggplot(iris, aes(x=Sepal.Length, y=Petal.Length**, group=Species**)) + geom_point() + stat_smooth(method="lm")

> ggplot(iris, aes(x=Sepal.Length, y=Petal.Length, group=Species, **color = Species**)) + geom_point() + stat_smooth(method="lm")
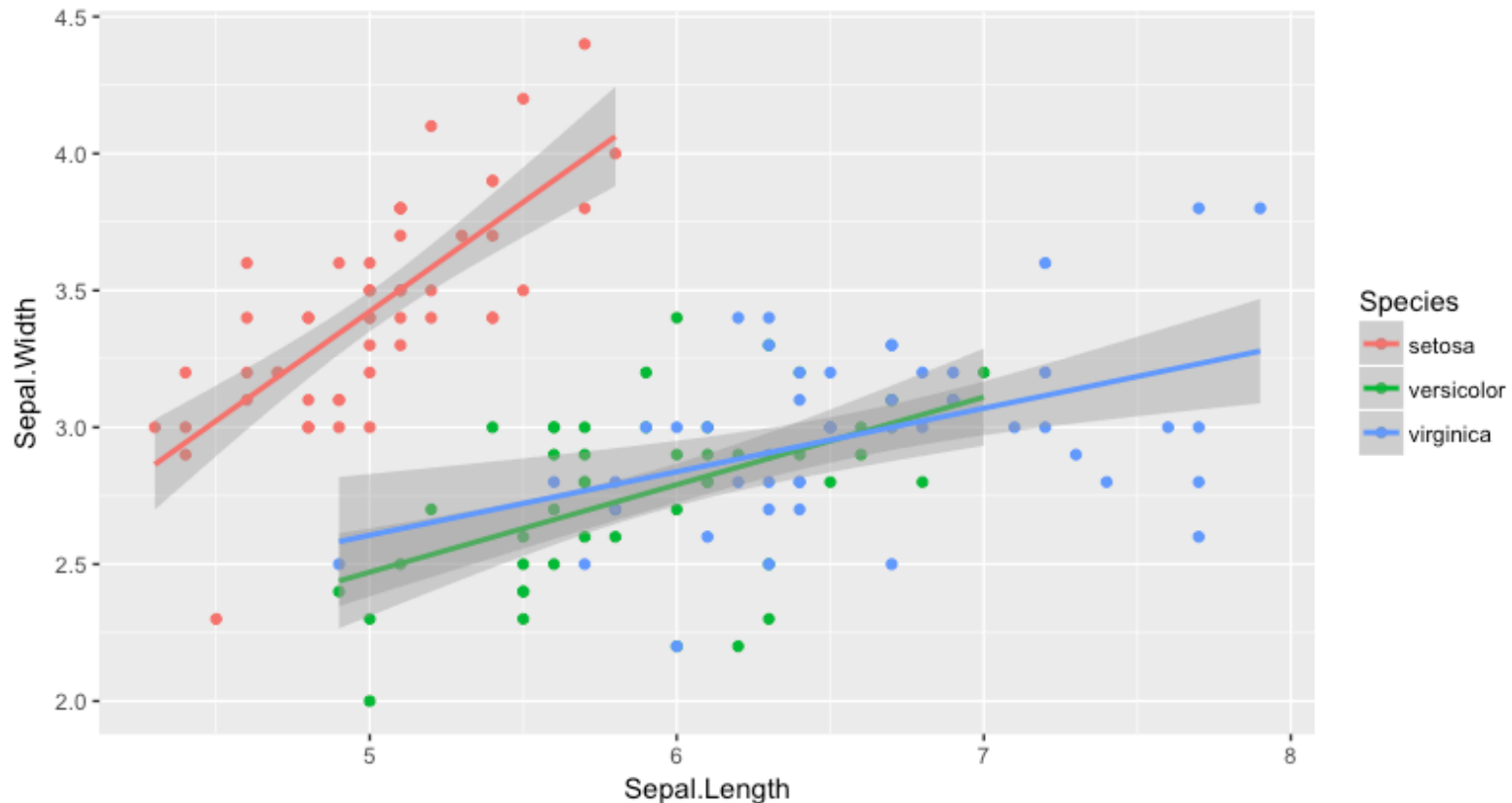
ggPlot2 - Themes

> library("ggthemes")   **# https://github.com/jrnold/ggthemes**

> ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width, color=Species)) +
geom_point() + stat_smooth(method="lm") **+ theme_tufte()**

ggPlot2 Global Formatting

- The par() function of base R does not apply to ggPlot2
- A convenient package for global formatting of ggPlot2 figures is "cowplot"

```
> library("cowplot")


> a <- ggplot(data1, aes()) + geom_point()
> b <- ggplot(data2, aes()) + geom_point()
> c <- ggplot(data3, aes()) + geom_point()
> plot_grid(a, b, c, align='h', ncol=3)
```
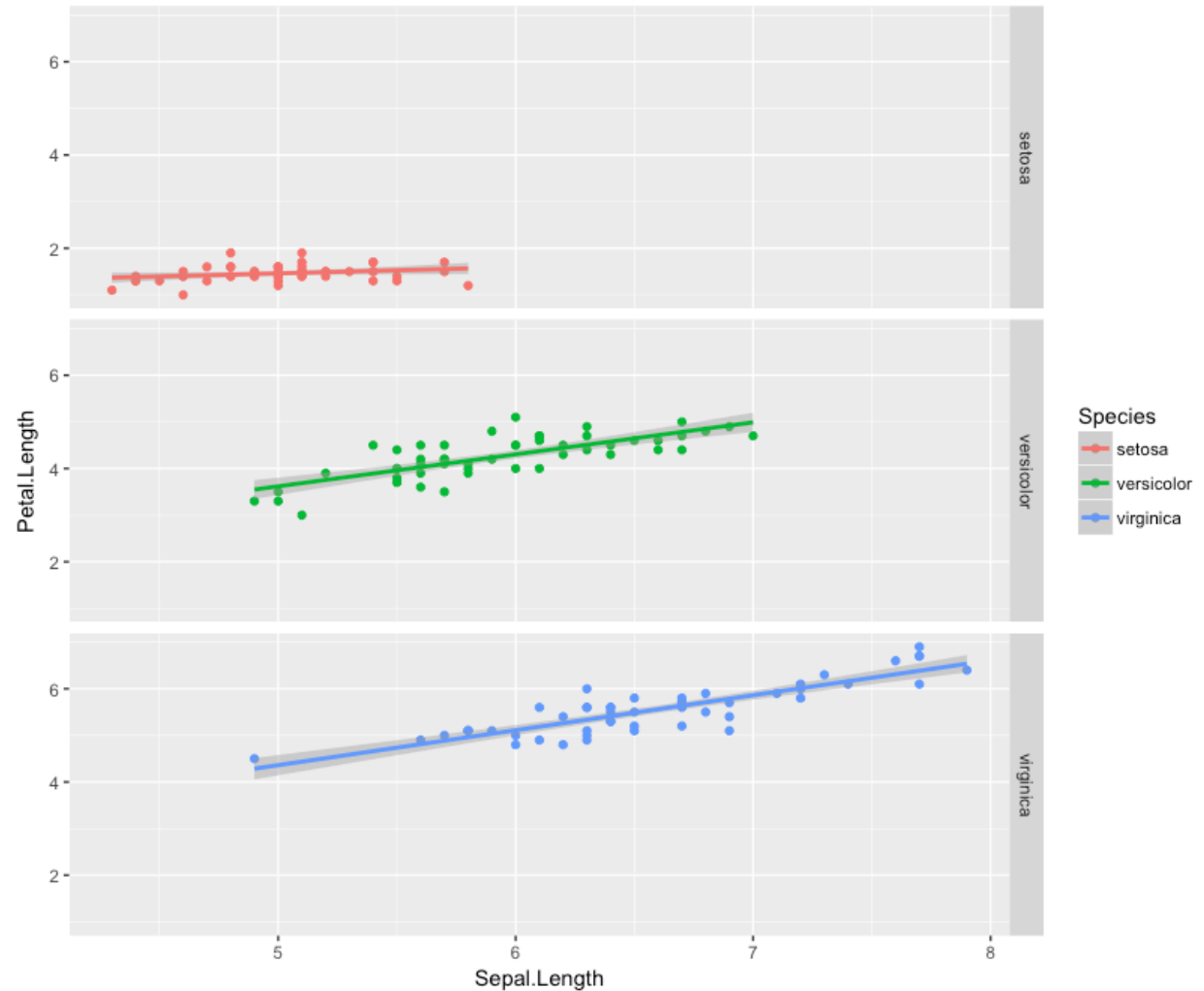
ggPlot2 – Divide figure via facets

➢ ggplot(iris, aes(x=Sepal.Length, y=Petal.Length, group=Species, color=Species)) +
  geom_point() +
  stat_smooth(method="lm") +
  facet_grid(Species ~ .)

Saving ggplots

- If the plot is on your screen

>ggsave("~/path/to/figure/filename.png")

- If your plot is assigned to an object

>ggsave(plot1, file = "~/path/to/figure/filename.png")

- Specify a size

>ggsave(plot1,file = " ~/path/to/figure/filename.png", width = 6, height =4)

- or any format (pdf, png, eps, svg, jpg)

>ggsave(plot1,file = "/path/to/figure/filename.eps")

Read the docs!

- Documentation locally in R or at http://docs.ggplot2.org/current/