

Data Visualization with R & ggplot2

Karthik Ram

April 7, 2013

Some housekeeping

Install some packages

```
install.packages("ggplot2", dependencies = TRUE)
```

Download this PDF

github.com/karthikram/ggplot-lecture

PUBLIC **karthikram / ggplot-lecture** Pull Request Unwatch Star 0 Fork 0

Code Network Pull Requests 0 Issues 0 Wiki Graphs Settings

My lecture on ggplot at Cal (spring 2013) — [Read more](#)

Clone in Mac ZIP HTTP SSH Git Read-Only `git@github.com:karthikram/ggplot-lecture.git` Read+Write access

branch: master Files Commits Branches 1 Tags

ggplot-lecture / 16 commits

cropped slide image

karthikram authored 6 minutes ago latest commit f171851480

images an hour ago Added new beamer theme and some images [karthikram]

Base graphics

- Ugly, laborious, and verbose
- There are better ways to describe statistical visualizations.

Why ggplot2?

- Follows a grammar, just like any language.
- It defines basic components that make up a sentence. In this case, the grammar defines components in a plot.
- Grammar of graphics originally coined by Lee Wilkinson

Why ggplot2?

- Supports a continuum of expertise.
- Get started right away but with practice you can effortlessly build complex, publication quality figures.

Section 1

Basics

Some terminology

- **ggplot** - The main function where you specify the dataset and variables to plot
- **geoms** - geometric objects
 - `geom_point()`, `geom_bar()`, `geom_density()`, `geom_line()`, `geom_area()`
- **aes** - aesthetics
 - shape, transparency, color, fill.
- **scales** Define how your data will be plotted
 - *continuous*, *discrete*, *log*

Section 2

Assembling your first ggplot

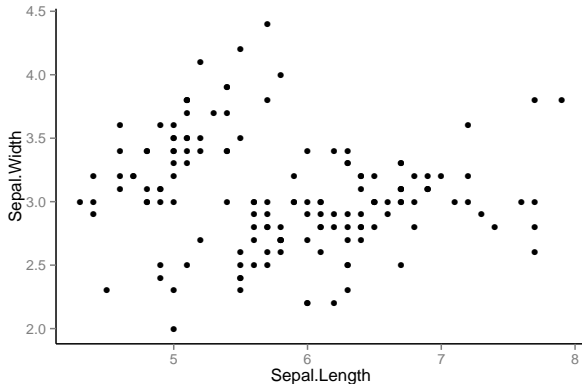
The iris dataset

```
head(iris)
```

##	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
## 1	5.1	3.5	1.4	0.2	setosa
## 2	4.9	3.0	1.4	0.2	setosa
## 3	4.7	3.2	1.3	0.2	setosa
## 4	4.6	3.1	1.5	0.2	setosa
## 5	5.0	3.6	1.4	0.2	setosa
## 6	5.4	3.9	1.7	0.4	setosa

Let's try an example

```
ggplot(data = iris, aes(x= Sepal.Length, y= Sepal.Width)) +  
geom_point()
```



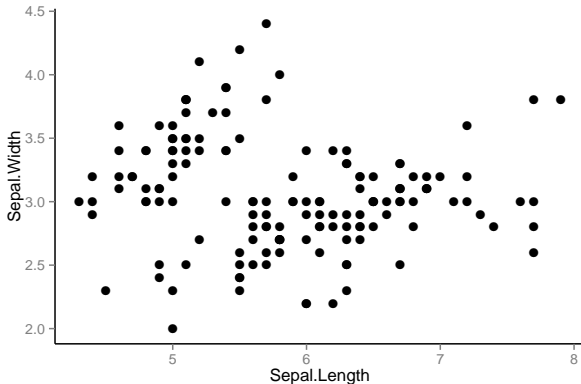
Basic structure

```
ggplot(data = iris, aes(x= Sepal.Length, y= Sepal.Width))  
  + geom_point()  
myplot <- ggplot(data = iris, aes(x= Sepal.Length, y= Sepal.Width))  
myplot + geom_point()
```

- Specify the data and variables inside the ggplot function.
- Anything else that goes in here becomes a global setting.
- Then add layers of geometric objects, statistical models, and panels.

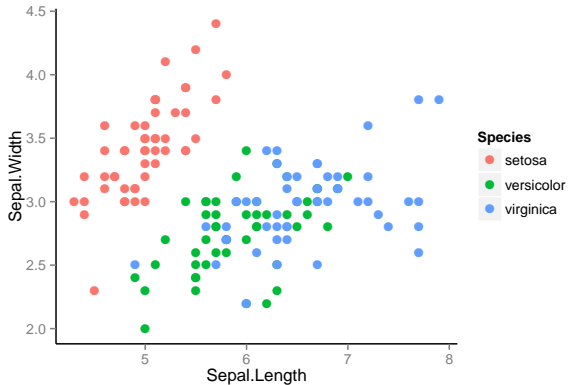
Increase the size of points

```
ggplot(data = iris, aes(x= Sepal.Length, y= Sepal.Width)) +  
geom_point(size = 3)
```



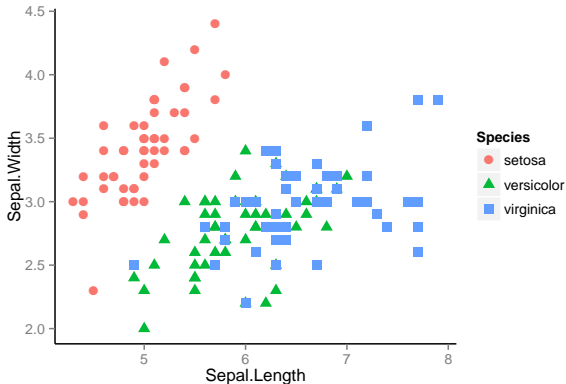
Add some color

```
ggplot(iris, aes(Sepal.Length, Sepal.Width, color = Species)) +  
geom_point(size = 3)
```



Also differentiate points by shape

```
ggplot(iris, aes(Sepal.Length, Sepal.Width, color = Species)) +  
geom_point(aes(shape = Species), size = 3)
```



Section 3

Exercise 1

Make the following plot.

Use the `airquality` dataset and generate a scatter plot of the year against `xxx`.

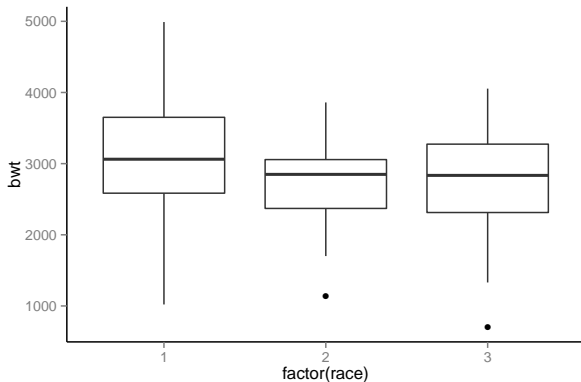
Then increase the size of the points.

Section 4

Box plots

See `geom_boxplot` for list of options

```
library(MASS)
ggplot(birthwt, aes(factor(race), bwt)) + geom_boxplot()
```

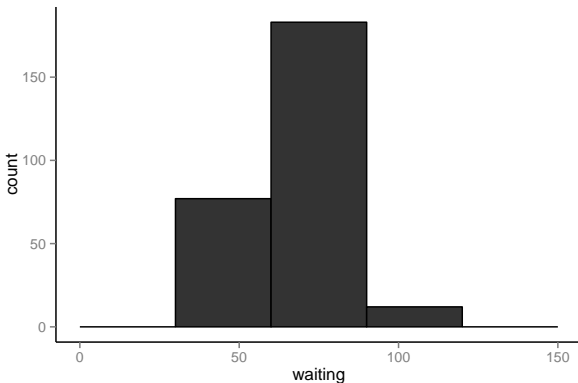


Section 5

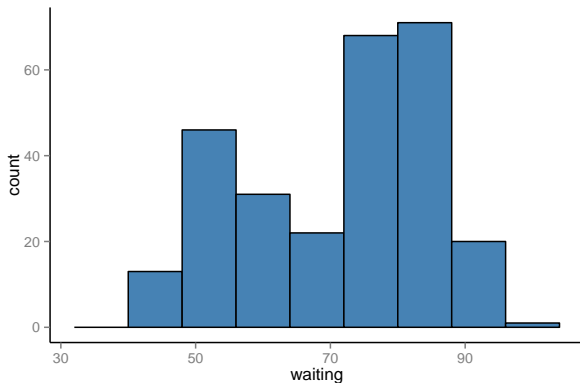
Histograms

See `geom_histogram` for list of options

```
h <- ggplot(faithful, aes(x=waiting))  
h + geom_histogram(binwidth=30, colour="black")
```



```
h <- ggplot(faithful, aes(x=waiting))  
h + geom_histogram(binwidth=8, fill="steelblue", colour="black")
```

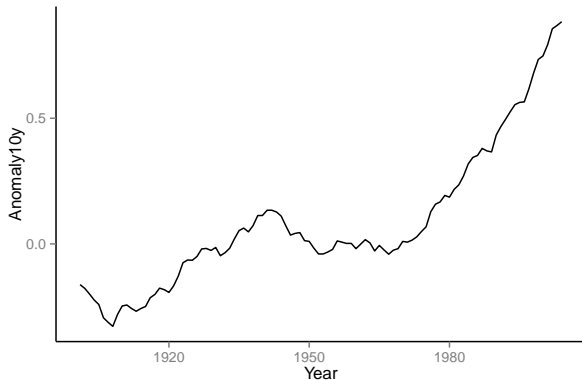


Section 6

Line plots

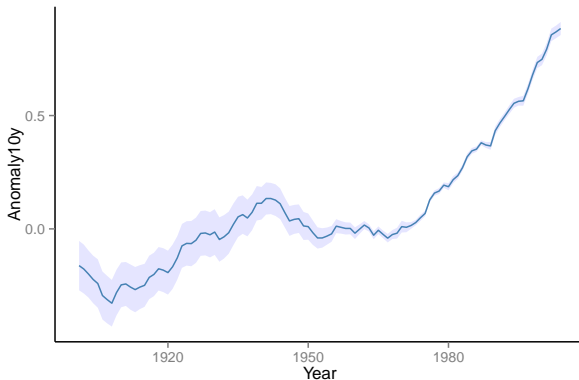
A simple line plot

```
climate <- read.csv("climate.csv", header = T)
ggplot(climate, aes(Year, Anomaly10y)) +
  geom_line()
```



We can also plot confidence regions

```
ggplot(climate, aes(Year, Anomaly10y)) +  
  geom_ribbon(aes(ymin = Anomaly10y - Unc10y,  
                ymax = Anomaly10y + Unc10y),  
            fill = "blue", alpha = .1) +  
  geom_line(color = "steelblue")
```



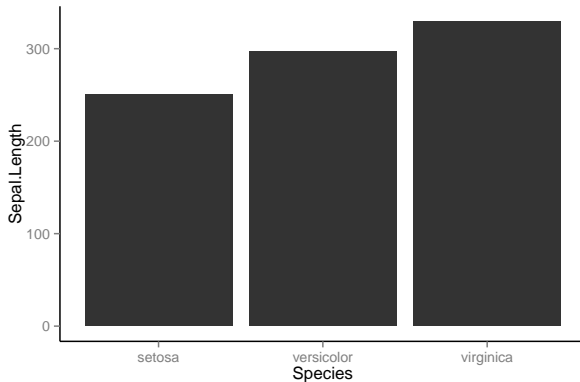
Exercise

- Modify the previous plot and change it such that there are three lines instead of one with a confidence band.
- Make the confidence bounds dashed red lines.

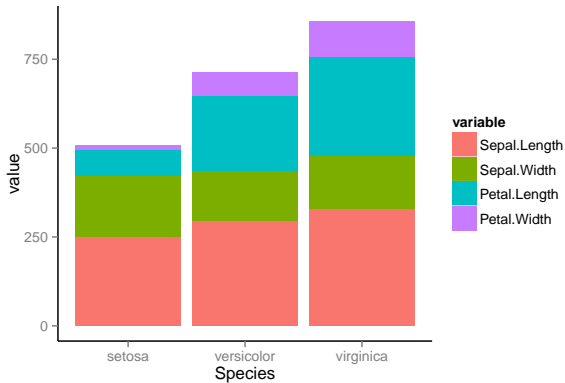
Section 7

Bar plots

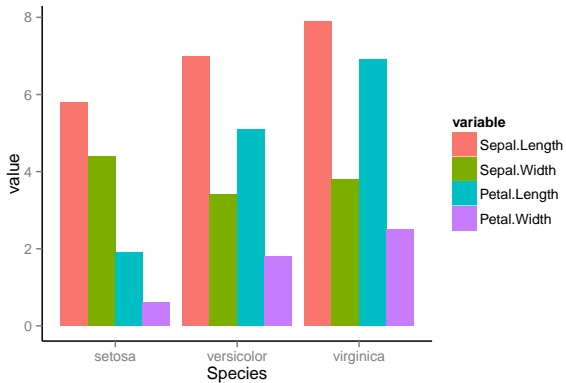
```
ggplot(iris, aes(Species, Sepal.Length)) +  
geom_bar(stat="identity")
```



```
df <- melt(iris, id.vars="Species")
ggplot(df, aes(Species, value, fill = variable)) +
  geom_bar(stat="identity")
```



```
df <- melt(iris, id.vars="Species")
ggplot(df, aes(Species, value, fill = variable)) +
  geom_bar(stat="identity", position = "dodge")
```



Section 8

plyr and reshape are key for using R

plyr and reshape

These two packages are the swiss army knives of R.

- plyr
 - ① dply
 - ② lply
 - ③ join
- reshape.
 - ① melt
 - ② dcast
 - ③ acast

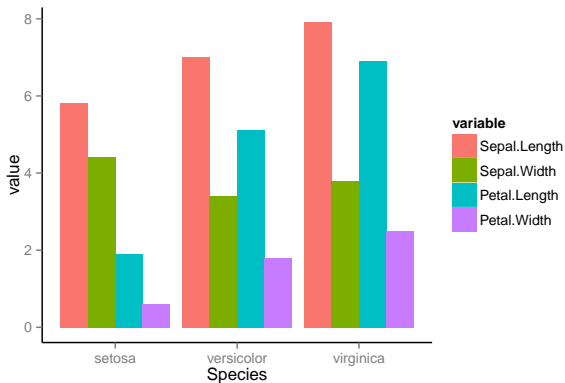

```
iris[1:2, ]
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1          3.5          1.4          0.2   setosa
## 2          4.9          3.0          1.4          0.2   setosa
```

```
df <- melt(iris, measured.vars="Species")
df[1:2, ]
```

```
##   Species    variable value
## 1   setosa Sepal.Length    5.1
## 2   setosa Sepal.Length    4.9
```

```
ggplot(df, aes(Species, value, fill = variable)) +  
geom_bar(stat="identity", position="dodge")
```



Section 9

Exercise 2

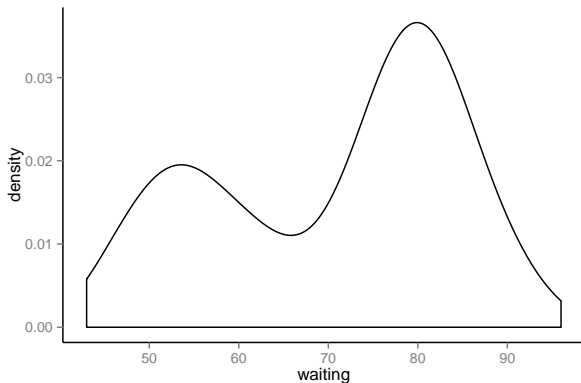
Take a dataset and make both a stacked bar chart and a dodged bar chart.

Section 10

Density Plots

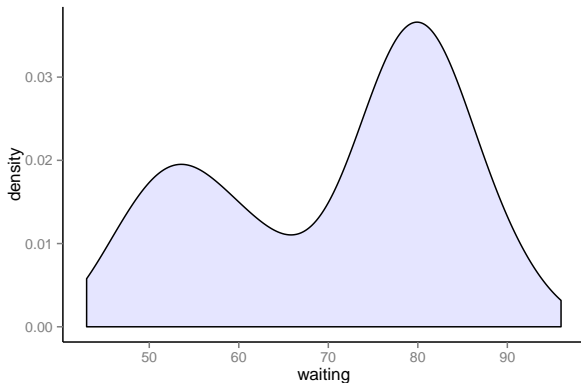
Density plots

```
ggplot(faithful, aes(waiting)) + geom_density()
```

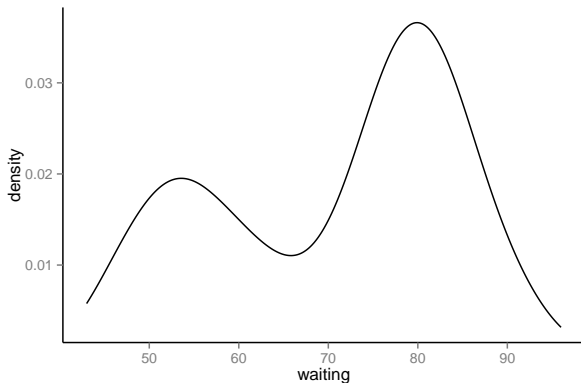


Density plots

```
ggplot(faithful, aes(waiting)) +  
geom_density(fill = "blue", alpha = 0.1)
```



```
ggplot(faithful, aes(waiting)) +  
geom_line(stat="density", fill="blue")
```



Section 11

Exercise 2

Section 12

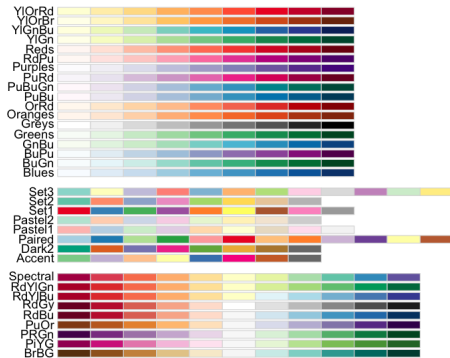
Mapping Variables to colors

Colors

```
aes(color = variable)
aes(color = "black")
# Or add it as a scale
scale_fill_manual(values = c("color1", "color2"))
```

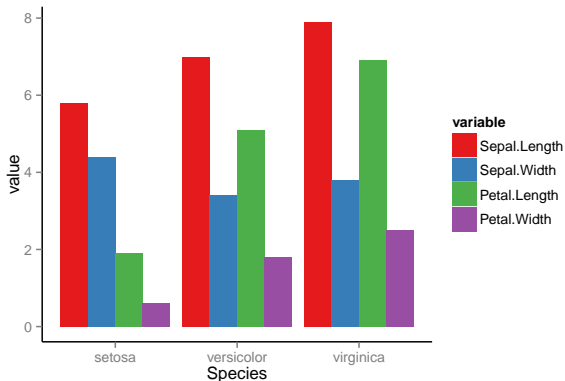
The RColorBrewer package

```
library(RColorBrewer)
display.brewer.all()
```



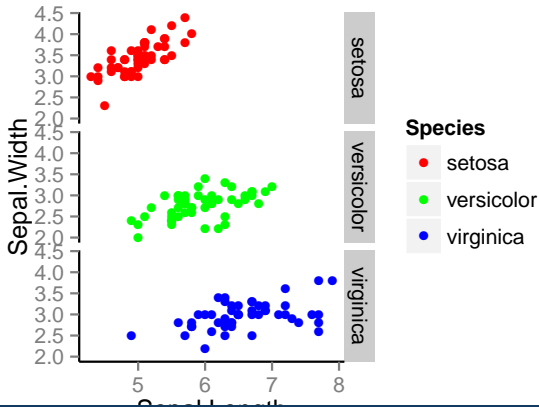
Using a color brewer palette

```
df <- melt(iris, id.vars="Species")
ggplot(df, aes(Species, value, fill = variable)) +
  geom_bar(stat="identity", position = "dodge") +
  scale_fill_brewer(palette="Set1")
```



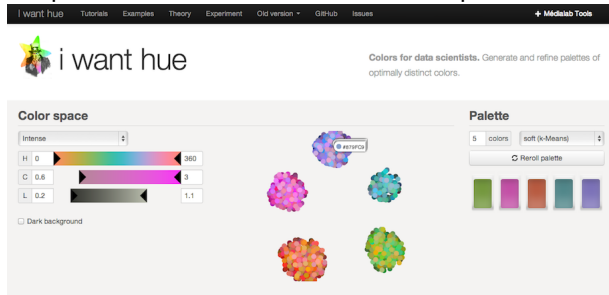
Manual color scale

```
ggplot(iris, aes(Sepal.Length, Sepal.Width, color = Species)) +  
geom_point() +  
facet_grid(Species ~ .) +  
scale_color_manual(values = c("red", "green", "blue"))
```



Refer to a color chart for beautiful visualizations

<http://tools.medialab.sciences-po.fr/iwanthue/>

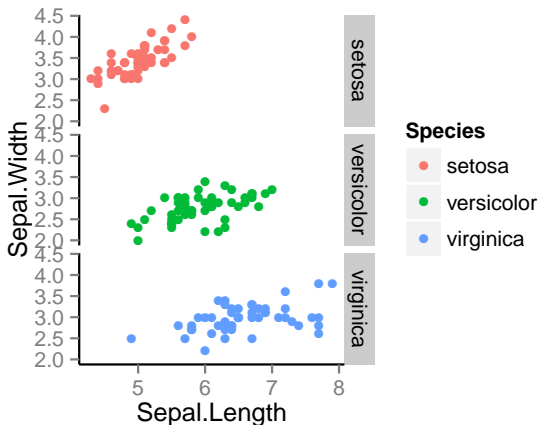


Section 13

Faceting

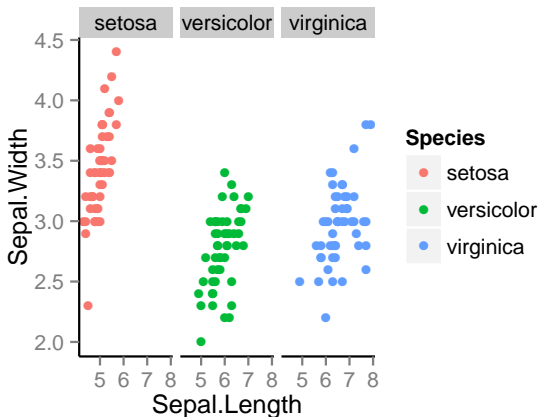
Faceting along columns

```
ggplot(iris, aes(Sepal.Length, Sepal.Width, color = Species)) +  
geom_point() +  
facet_grid(Species ~ .)
```



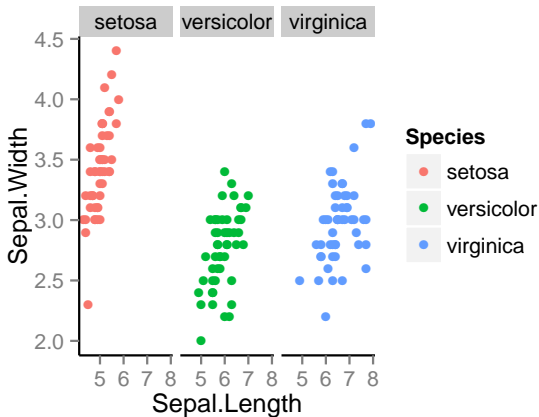
and along rows

```
ggplot(iris, aes(Sepal.Length, Sepal.Width, color = Species)) +  
geom_point() +  
facet_grid(. ~ Species)
```



Colors

```
ggplot(iris, aes(Sepal.Length, Sepal.Width, color = Species)) + geom_point(  
  facet_wrap(~Species)
```

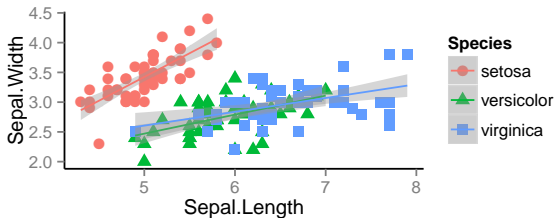


Section 14

Plotting models

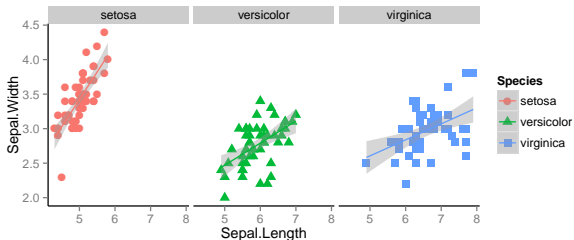
Add statistics to your plot1

```
ggplot(iris, aes(Sepal.Length, Sepal.Width, color = Species)) +  
geom_point(aes(shape = Species), size = 3) +  
geom_smooth(method="lm")
```



Add statistics to your plot2

```
ggplot(iris, aes(Sepal.Length, Sepal.Width, color = Species)) +  
  geom_point(aes(shape = Species), size = 3) +  
  geom_smooth(method="lm") +  
  facet_grid(. ~ Species)
```



Section 15

Themes

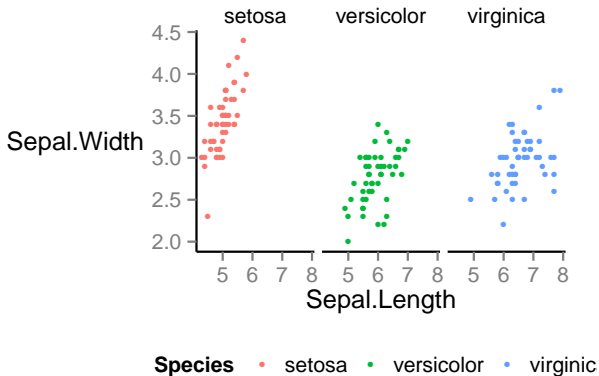
Adding themes

```
theme()  
# see ?theme() for more options
```


A themed plot

```
ggplot(iris, aes(Sepal.Length, Sepal.Width, color = Species)) +  
geom_point(size = 1.2, shape = 16) +  
facet_wrap( ~ Species) +  
theme(legend.key = element_rect(fill = NA),  
legend.position = "bottom",  
strip.background = element_rect(fill = NA),  
axis.title.y = element_text(angle = 0))
```

Adding themes



Section 16

Create functions to automate your plotting

Commonly used scales

```
my_custom_plot <- function(df) {  
  ggplot(df, ...)   
}
```

Section 17

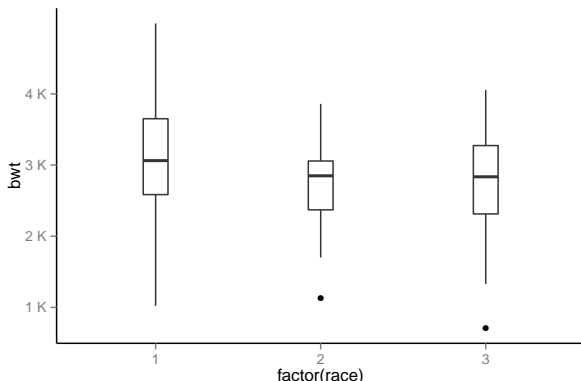
Scales

Commonly used scales

```
scale_fill_discrete(), scale_colour_discrete()  
scale_fill_hue(), scale_color_hue()  
scale_fill_manual(), scale_color_manual()  
scale_fill_brewer(), scale_color_brewer()  
scale_linetype(), scale_shape_manual()
```

Adding a continuous scale

```
library(MASS)
ggplot(birthwt, aes(factor(race), bwt)) +
  geom_boxplot(width=.2) +
  scale_y_continuous(labels = (paste0(1:4, " K")),
    breaks=seq(1000, 4000, by = 1000))
```



Another example

```
# Assign the plot to an object
dd <- ggplot(iris, aes(Sepal.Length, Sepal.Width, color = Species)) +
  geom_point(size=4, shape=16) +
  facet_grid(. ~Species)
# Now add a scale
dd +
  scale_y_continuous(breaks=seq(2, 8, by = 1),
    labels=paste0(2:8, " cm"))
```


Section 18

Publication quality figures

- If the plot is on your screen

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

- If your plot is assigned to an object

```
ggsave(plot1, "~/path/to/figure/filename.png")
```

- Specify a size

```
ggsave("/path/to/figure/filename.png", width = 6, height = 4)
```

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

```
ggsave("/path/to/figure/filename.eps")
```

```
ggsave("/path/to/figure/filename.jpg")
```

```
ggsave("/path/to/figure/filename.pdf")
```

Further help

- Practice
- Read the docs (either locally in R or at <http://docs.ggplot2.org>)
- Work together