

# Linear Regression Models

## P8111

Lecture 03

Jeff Goldsmith  
January 26, 2016



THE DEPARTMENT OF  
**BIostatISTICS**



Columbia University  
**MAILMAN SCHOOL  
OF PUBLIC HEALTH**

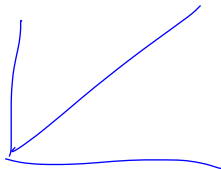
# Today's Lecture

- ggplot2

- R Markdown

# Graphics

- Plotting is one of the most important things you're going to do
- ✓ ■ Always (always, always) look at your data
- A *good* picture is worth 1,000 words; a bad picture is worth much less



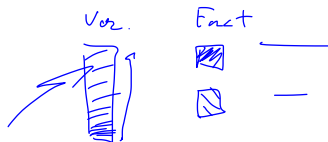
# Graphics in R

- base graphics are a thing – see e.g. `plot(x, y)`
- ✓ ■ lattice is also a thing
- We'll just focus on the ggplot system

# ggplot2

- Development lead by Hadley Wickham
  - ▶ Plays nicely with the dataframe-centric `dplyr` framework
- gg = “Grammar of Graphics”
  - ▶ Think verbs that perform actions on data

# Before we get started



- Time spent thinking about and organizing the data results in better graphs
- Graphs should be clear – useful legends, axis titles, informative (not superfluous) coloring / sizing / shading

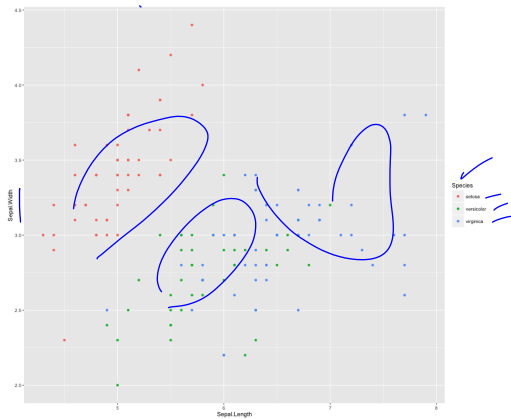
# Constructing a ggplot figure

- data: the dataframe you're using to construct your plot
- **aesthetic mappings**: connections between data and visual components (x and y, first; size, color, group, shape, etc)
- **layers**: how the data are actually shown (points, lines, boxplots, densities, smooths)

# Example

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

*(Handwritten blue arrows point from the underlined parts of the code to the corresponding elements in the plot below: 'ggplot' to the plot area, 'aes' to the legend, 'x' to the x-axis label, 'y' to the y-axis label, 'color' to the legend, 'geom\_point()' to the data points, and 'geom\_path()' to the blue loops.)*





# Some notes

- You can add multiple `geom`'s
- Each will inherit the global data and aesthetics unless you tell it to do something different
- Aesthetic mappings have reasonable default scales (e.g. colors); you can override these if you want
- Facetting can be a useful way to visualize data across factors

# Live coding

# Cheat Sheet

## Data Visualization with ggplot2 Cheat Sheet



### Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same few components: a **data** set, a set of **geoms** - visual marks that represent data points, and a **coordinate system**.



To display data values, map variables in the data set to aesthetic properties of the geom like **size**, **color**, and **x** and **y** locations.



Build a graph with **ggplot()**

**ggplot(data = mpg, aes(x = displ, y = hwy))**  
Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

**ggplot(data = mpg, aes(x = displ, y = hwy))**

Begin a plot that you finish by adding layers to. No defaults, but provides more control than **ggplot()**.

**ggplot(mpg, aes(hwy, cyl))**  
**geom\_point(aes(color = cyl))**  
**geom\_smooth(method = "lm")**  
**scale\_color\_manual()**  
**theme\_bw()**

Add a new layer to a plot with a **geom**, **id** or **stat**, **id** function. Each provides a geom, a set of aesthetic mappings, and a default stat and position adjustment.

**last\_plot()**

Returns the last plot

**ggsave("plot.png", width = 5, height = 5)**

Saves last plot as 5"x5" file named "plot.png" in working directory. Matches file type to file extension.

## Geoms - Use a geom to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

### One Variable

**Continuous**  
**a = ggplot(mpg, aes(hwy))**  
**geom\_area(aes(fill = "bin"))**  
**geom\_density2d()**  
**geom\_histogram(bins = 30)**  
**geom\_point()**  
**geom\_dotplot()**  
**geom\_freqpoly()**  
**geom\_histogram(bins = 30)**  
**geom\_histogram(aes(x = displ, y = hwy))**

### Discrete

**b = ggplot(mpg, aes(class))**  
**geom\_bar()**

### Graphical Primitives

**c = ggplot(mpg, aes(long, lat))**  
**geom\_polygon(aes(group = group))**  
**d = ggplot(economics, aes(date, unemploy))**  
**geom\_path()**  
**geom\_ribbon(aes(ymin = unemploy - 900, ymax = unemploy + 900))**  
**e = ggplot(seals, aes(x = long, y = lat))**  
**geom\_segment(aes(xend = long + delta\_long, yend = lat + delta\_lat))**  
**f = ggplot(rects, aes(x = long, y = lat, xmin = long - delta\_x, xmax = long + delta\_x, ymin = lat - delta\_y, ymax = lat + delta\_y))**

### Two Variables

**Continuous X, Continuous Y**  
**f = ggplot(mpg, aes(cty, hwy))**  
**geom\_blank()**  
**geom\_jitter()**  
**geom\_point()**  
**geom\_quantile()**  
**geom\_rug(sides = "tl")**  
**geom\_smooth(model = lm)**  
**geom\_text(aes(label = class))**

### Continuous Bivariate Distribution

**g = ggplot(movies, aes(year, rating))**  
**geom\_bin2d(binwidth = c(20, 0.5))**  
**geom\_density2d()**  
**geom\_hex()**

### Continuous Function

**j = ggplot(economics, aes(date, unemploy))**  
**geom\_area()**  
**geom\_line()**  
**geom\_step(direction = "hv")**

### Visualizing error

**k = ggplot(d, aes(grip, flt, ymin = flt - se, ymax = flt + se))**

**Discrete X, Continuous Y**  
**g = ggplot(mpg, aes(class, hwy))**  
**geom\_bar(stat = "identity")**  
**geom\_boxplot()**  
**geom\_violin()**

**l = ggplot(mpg, aes(class, hwy))**  
**geom\_crossbar(latten = 2)**  
**geom\_errorbar()**  
**geom\_linerange()**  
**geom\_pointrange()**

### Discrete X, Discrete Y

**h = ggplot(diamonds, aes(cut, color))**  
**geom\_jitter()**

**Maps**  
**data = data.frame(murder = USArrests\$Murd, state = tolower(row.names(USArrests)))**  
**map = map\_data("state")**  
**l = ggplot(data, aes(f1 = murder))**  
**geom\_map(map\_id = state, map = map) + expand\_limits()**

### Three Variables

**m = ggplot(seals, aes(long2 = delta\_lat2))**  
**geom\_contour(aes(z = z))**

**n = ggplot(raster(aes(f1 = z, f2 = 0.5, f3 = 0.5, interpolate = FALSE)))**  
**geom\_raster(aes(f1 = z, f2 = 0.5, f3 = 0.5, interpolate = FALSE))**  
**geom\_title(aes(f1 = z))**

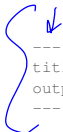
# R Markdown

- How you present your results is important
- Reproducibility matters – both to ensure reasonable results and to make your life easier
- R Markdown helps you package both your analysis (code) and presentation (text) in a single document

# R Markdown

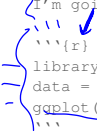
- A “Markdown” language is a lightweight syntax that can be easily converted to HTML or another format
- R Markdown lets you combine formatted text with code chunks
- Having text and code in the same place, and having the combined output be user-friendly, is huge for your workflow

# R Markdown Example



```
---  
title: "A First R Markdown Document"  
output: html_document  
---
```

I'm going to sample from a normal distribution and draw a density plot.



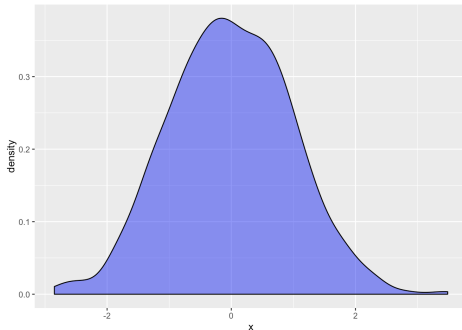
```
```{r}  
library(ggplot2)  
data = data.frame(x = rnorm(1000))  
ggplot(data, aes(x = x)) + geom_density(fill = "blue", alpha = .5)  
```
```

# R Markdown Example

## A First R Markdown Document

I'm going to sample from a normal distribution and draw a density plot.

```
library(ggplot2)
data = data.frame(x = rnorm(1000))
ggplot(data, aes(x = x)) + geom_density(fill = "blue", alpha = .5)
```



# R Markdown Tips

- You can control what is shown in code chunk options
  - Generally, you should show only what you need to
- You can control some important behaviors using code chunk options
- You can access objects created in a code chunk later – in another code chunk or inline.
- You can export directly to PDF
- You can include nicely-formatted equations in a . . . . .



# Live coding

# R Markdown Cheat Sheet

learn more at [rmarkdown.rstudio.com](http://rmarkdown.rstudio.com)

rmarkdown 0.2.50 Updated: 8/14



## 1. Workflow

- **I. Open** - Open a file that uses the .Rend extension.

iii. **Write** - Write content with the goal to use it Markdown syntax.

- iii. **Embed** - Embed R code that creates output to include in the report.

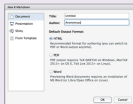
iv. **Render** - Replace R code with its output and transform the report into a slideshow, pdf, html or ms Word file.



## 2. Open File

Start by saving a text file with the extension `.Rmd`, or open an RStudio Rmd template

- In the menu bar, click **File ► New File ► R Markdown...**
- A window will open. Select the class of output you would like to make with your Rmd file
- Select the specific type of output to make with the radio buttons (you can change this later)
- Click OK



#### 4. Choose Output

Write a YAML header that explains what type of document to build from your R Markdown file.

## YAML

A YAML header is a set of key: value pairs at the start of your file. Begin and end the header with a line of three dashes [ - - - ]

```

title: "Untitled"
author: "Anonymous"
output: html_document

```

This is the start of my report. The above is metadata saved in a YAML header.

The RStudio template writes the YAML header for you

The output value determines which type of file R will build from your .Rmd file (in Step 6)

```
output: html_document ***** html file (web page)
```

```
output: pdf_document ..... pdf document
```

output: word document..... Microsoft Word .docx

```
output: beamer_presentation..... beamer slideshow (pdf)
```

```
output: loslides presentation..... loslides slideshow (html)
```

REStudio® is a trademark of RStudio, Inc. • [www.rstudio.com](http://www.rstudio.com) • [info@rstudio.com](mailto:info@rstudio.com) • 844-448-1212 • [rstudio.com](http://rstudio.com)

### 3. Markdown

Next, write your report in plain text. Use markdown syntax to describe how to format text in the final report.

## syntax

```
Plain text
End a line with two spaces
*italics* and _italics_
**bold** and __bold__
superscript^2^
--strikethrough--
[link](www.rstudio.com)
```

```
# Header 1
## Header 2
### Header 3
#### Header 4
##### Header 5
##### Header 6
```

```
endash: --
endash: ---
ellipsis: ...
inline equation: SA = \pi*r^{2}$
image: 

horizontal rule (or slide break)
```

```
***
> block quote
```

- \* unordered list
- \* item 2
  - + sub-item 1
  - + sub-item 2

```
1. ordered list
2. item 2
    + sub-item 1
    + sub-item 2
```

| Table Header | Second Header |
|--------------|---------------|
| Table Cell   | Cell 2        |
| Cell 3       | Cell 4        |

becomes

Plain text  
End a line with two spaces to start a new paragraph.  
*Italic and italic*  
**bold and bold**  
superscript<sup>2</sup>  
~~strikethrough~~  
[link](#)

## Header 1

## Header 2

### Header 3

#### Header 4

## Hewlett 5

```

enddash: -
enddash: -
ellipse: ...
inline equation:  $\hat{A} = x + y$ 

```



horizontal rule for slide break

block quote

- unordered list
- item 2
  - sub-item 1
  - sub-item 2

1. ordered list
2. item 2
  - sub-item 1
  - sub-item 2

| Table Header | Second Header |
|--------------|---------------|
| Table Cell   | Cell 2        |
| Cell 3       | Cell 4        |

# Today's big ideas

- Intro to `ggplot2`
- Intro to R Markdown

- 
- [google.com](https://www.google.com); [stackoverflow](https://stackoverflow.com)
  - `ggplot2` Cheat Sheet
  - The `ggplot2` book on GitHub by Hadley
  - STAT 545 “Intro to `ggplot2`”, “R Markdown”
  - Exploratory Data Analysis with R (The `ggplot2` Plotting System)