Outline

What I’m Covering:

- pros/cons different graphing packages
- ggplot and the grammar of graphics
- how to visualize summary stats and regression results
- basic spatial visualization
Outline

What I’m Covering:

- pros/cons different graphing packages
Outline

What I’m Covering:

- pros/cons different graphing packages
- ggplot and the grammar of graphics
Outline

What I’m Covering:

- pros/cons different graphing packages
- ggplot and the grammar of graphics
- how to visualize summary stats and regression results
Outline

What I’m Covering:

- pros/cons different graphing packages
- ggplot and the grammar of graphics
- how to visualize summary stats and regression results
- basic spatial visualization
Download code and slides at:

web.stanford.edu/~imalone/VAM.html
Choosing a Visualization Package

Toy Example: Plot number of democracies and autocracies over time.
Choosing a Visualization Package

Toy Example: Plot number of democracies and autocracies over time.

Options:

- plot (graphics)
Option 1: Plot

Code:

```r
plot(year, numdems, x, y
    # aesthetics stuff (color, plot type, size)
    col = "blue", type = "l", lwd=3,
    # main title and axes labels
    main = "Number of Democracies and Autocracies, 1800-2012",
    xlab = "Year",
    ylab = "Count")
lines(year, numauts, type = "l", col = "red", lwd =3)
legend("topleft", # location of legend
    legend=c("Democracies","Autocracies"), # legend labels/var
    col = c(4, 2), # colors for each,
    # legend colors default 1= black, 2 = red, 4=blue
    lty = c(1, 1),
    title="Regime Type") # Name of Legend
```
Number of Democracies and Autocracies, 1800–2012

Year
Count
Regime Type
Democracies
Autocracies
Choosing a Visualization Package

Toy Example.

Options:

- plot (graphics)
Choosing a Visualization Package

Toy Example.

Options:

- plot (graphics) *Overly simplistic.*
Choosing a Visualization Package

Toy Example.

Options:

- plot (graphics) Overly simplistic. Error interp.
Choosing a Visualization Package

Toy Example.

Options:

- plot (graphics) *Overly simplistic. Error interp. Limited online support.*
Choosing a Visualization Package

Toy Example.

Options:

- plot (graphics) *Overly simplistic. Error interp. Limited online support.*
- xyplot (lattice)
Step 1: Build First Layer for Number of Democracies.

```r
library(lattice)
layer1 = xyplot(numdems ~ year, #format: y ~ x
type = "l",
# add title
main = "Number of Democracies and Autocracies, 1800-2012",
col = "blue", lwd=1,
#legend
key=list(space="right", # location
# aesthetics (aes)
lines=list(col=c("red","blue"), lty=c(1,1), lwd=2),
#labels for each line
text=list(c("Autocracies","Democracies")))
```
Step 2: Build Second Layer for Number of Autocracies.

```r
layer2 = xyplot(numauts ~ year,
                 type = "l",
                 col = "red")
```
Step 3: Add the layers.

```r
#need extra package to put layers on top
suppressMessages(library(latticeExtra))
layer1 + layer2
```

![Number of Democracies and Autocracies, 1800–2012](image-url)
Options:

- plot (graphics) Overly simplistic. Error Interp. Limited online support.
- xyplot (lattice)
Options:

- plot (graphics) Overly simplistic. Error Interp. Limited online support.
- xyplot (lattice) Supplemental packages.
Options:

- **plot (graphics)**: Overly simplistic. Error Interp. Limited online support.
- **xyplot (lattice)**: Supplemental packages. Default look could be nicer.
Options:

- **plot (graphics)** Overly simplistic. Error Interp. Limited online support.
- **xyplot (lattice)** Supplemental packages. Default look could be nicer.
- **ggplot (ggplot2)**
Option 3: ggplot

```r
suppressMessages(library(ggplot2))

ggplot(data = NULL) +
  # line for dems and auts
  geom_line(aes(x=year, y = numdems, colour = "numdems")) +
  geom_line(aes(x=year, y = numauts, colour = "numauts")) +
  xlab("Year") + ylab("Count") +
  ggtitle("Number of Democracies and Autocracies, 1800-2012") +
  # legend aesthetics
  scale_color_manual(name = "Regime Type", # Name
                     labels = c(numdems="Democracies", numauts = "Autocracies"),
                     values=c(numdems=4,numauts=2))
```
Option 3: ggplot

Number of Democracies and Autocracies, 1800–2012

- **Regime Type**
  - Autocracies
  - Democracies

- Count
- Year
- 1800 1850 1900 1950 2000

Iris Malone
Advanced Data Visualization in R
November 6, 2015 13 / 68
Summary of Options:

- plot (graphics)
- xyplot (lattice)
- ggplot (ggplot2)
Why ggplot?

- Used professionally
Why ggplot?

Example 1

Week 17 - QB Tiers

Peyton Manning
Nick Foles
Drew Brees
Cam Newton
Matthew Stafford
Jay Cutler
Andrew Luck
Tom Brady
Ben Roethlisberger
Philip Rivers
Russell Wilson
Colin Kaepernick
Andy Dalton
Ryan Tannehill
Ryan Fitzpatrick
Aaron Rodgers
Kirk Cousins
Alex Smith
Jason Campbell
Terelle Pryor
Tony Romo
Matt Ryan
Kyle Orton
Matt Cassel
Matt Flynn
Joe Flacco
Eli Manning
Carson Palmer
E.J. Manuel
Matt Schaub
Thaddeus Lewis
Matt McGloin
Why ggplot?

Example 2

Costs for Americans...

... have soared for education, child care and health care...

+40 pct. pts.
Change in prices relative to a 23% increase in prices for all items, 2005-2014

+20

College tuition and fees

-20

Child care/nursery school
Health care
Vehicle maintenance/repair
Food and beverages

-40

Housing
Personal care

-60

Clothing
New and used vehicles

-60

Cellphone service

-60

Toys

-60

Phonos and accessories
Refect prices unsubsidized by service providers

... and have plummeted for televisions, toys and phones, relative to other prices.
Why ggplot?

- Used professionally
Why ggplot?

- Used professionally
- Very pretty

Summary: Worth it.
Why ggplot?

- Used professionally
- Very pretty
- Easy to manipulate

Summary: Worth it.
Why ggplot?

- Used professionally
- Very pretty
- Easy to manipulate
- Great support online
Why ggplot?

- Used professionally
- Very pretty
- Easy to manipulate
- Great support online
- Knowledge transfers to other packages/languages (ggvis, Shiny, Python)

Summary: Worth it.

Iris Malone
Advanced Data Visualization in R
November 6, 2015 18 / 68
Why ggplot?

- Used professionally
- Very pretty
- Easy to manipulate
- Great support online
- Knowledge transfers to other packages/languages (ggvis, Shiny, Python)
- Steep Learning Curve
Why ggplot?

- Used professionally
- Very pretty
- Easy to manipulate
- Great support online
- Knowledge transfers to other packages/languages (ggvis, Shiny, Python)
- Steep Learning Curve
- Lots of syntax

Summary: Worth it.
Why ggplot?

- Used professionally
- Very pretty
- Easy to manipulate
- Great support online
- Knowledge transfers to other packages/languages (ggvis, Shiny, Python)
- Steep Learning Curve
- Lots of syntax
- Can be slow

Summary: Worth it.
Why ggplot?

- Used professionally
- Very pretty
- Easy to manipulate
- Great support online
- Knowledge transfers to other packages/languages (ggvis, Shiny, Python)
- Steep Learning Curve
- Lots of syntax
- Can be slow
- Defaults to weird colors

Summary: Worth it.
Why ggplot?

- Used professionally
- Very pretty
- Easy to manipulate
- Great support online
- Knowledge transfers to other packages/languages (ggvis, Shiny, Python)
- Steep Learning Curve
- Lots of syntax
- Can be slow
- Defaults to weird colors
- **Summary:** Worth it.
Following next few slides adapted from Samantha Tyner

- Based on Grammar of Graphics book by Leland Wilkinson hence ‘gg’
ggplot Syntax

Following next few slides adapted from Samantha Tyner

- Based on Grammar of Graphics book by Leland Wilkinson hence ‘gg’
- New Zealander Hadley Wickham → R
ggplot Syntax

Following next few slides adapted from Samantha Tyner

- Based on Grammar of Graphics book by Leland Wilkinson hence ‘gg’
- New Zealander Hadley Wickham → R
- Analogy: Think of parts of a plot like parts of a sentence
ggplot Syntax

Following next few slides adapted from Samantha Tyner

- Based on Grammar of Graphics book by Leland Wilkinson hence ‘gg’
- New Zealander Hadley Wickham → R
- Analogy: Think of parts of a plot like parts of a sentence
- Warning: qplot
ggplot Syntax

- Noun → Data

```r
ggplot(data = df)
```
ggplot Syntax

- Noun → Data

\[ \text{ggplot(data} = \text{df}) \]

- Verb → “geom_” + Plot Type

\[ \text{ggplot(data} = \text{df)} + \text{geom_bar()} \]

<table>
<thead>
<tr>
<th>geom Type</th>
<th>Table 1: Geom Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>abline</td>
<td>area</td>
</tr>
<tr>
<td>errorbar</td>
<td>histogram</td>
</tr>
<tr>
<td>point</td>
<td>ribbon</td>
</tr>
<tr>
<td>blank</td>
<td>density</td>
</tr>
<tr>
<td>polygon</td>
<td>quantile</td>
</tr>
</tbody>
</table>
ggplot Syntax

- Noun → Data

```r
ggplot(data = df)
```

- Verb → “geom_” + Plot Type

```r
ggplot(data = df) + geom_bar()
```

- Adjectives → Aesthetics (“aes”) (x, y, fill, colour, linetype)

```r
ggplot(data = df, aes(x=categorical.var, fill=group.var)) + geom_bar()
```
### ggplot Syntax

**Adjectives → Aesthetics (“aes”)**

<table>
<thead>
<tr>
<th>Function</th>
<th>Colour</th>
<th>Size</th>
<th>Linetype</th>
<th>Alpha</th>
<th>Fill</th>
<th>Weight</th>
<th>Shape</th>
<th>Width</th>
<th>Height</th>
<th>Angle</th>
<th>Hjust</th>
<th>Vjust</th>
<th>Falpha</th>
</tr>
</thead>
<tbody>
<tr>
<td>abline</td>
<td>black</td>
<td>0.5</td>
<td></td>
<td>yes</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>area</td>
<td>yes</td>
<td>0.5</td>
<td></td>
<td>yes</td>
<td>grey20</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>bar</td>
<td>yes</td>
<td>0.5</td>
<td></td>
<td>yes</td>
<td>grey20</td>
<td>1</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>bin2d</td>
<td>yes</td>
<td>0.5</td>
<td></td>
<td>yes</td>
<td>grey60</td>
<td>1</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>boxplot</td>
<td>grey20</td>
<td>0.5</td>
<td>solid</td>
<td>yes</td>
<td>white</td>
<td>1</td>
<td>16</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>contour</td>
<td>#3366FF</td>
<td>0.5</td>
<td></td>
<td>yes</td>
<td>--</td>
<td>1</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>crossbar</td>
<td>black</td>
<td>0.5</td>
<td></td>
<td>yes</td>
<td>yes</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>density</td>
<td>black</td>
<td>0.5</td>
<td></td>
<td>yes</td>
<td>yes</td>
<td>1</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>density2d</td>
<td>#3366FF</td>
<td>0.5</td>
<td></td>
<td>yes</td>
<td>--</td>
<td>1</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>errorbar</td>
<td>black</td>
<td>0.5</td>
<td></td>
<td>yes</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>0.5</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>errorbarh</td>
<td>black</td>
<td>0.5</td>
<td></td>
<td>yes</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>0.5</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>freepoly</td>
<td>black</td>
<td>0.5</td>
<td></td>
<td>yes</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>hex</td>
<td>yes</td>
<td>0.5</td>
<td></td>
<td>--</td>
<td>grey50</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>hline</td>
<td>black</td>
<td>0.5</td>
<td></td>
<td>yes</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>llinerange</td>
<td>black</td>
<td>0.5</td>
<td></td>
<td>yes</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>path</td>
<td>black</td>
<td>0.5</td>
<td></td>
<td>yes</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>point</td>
<td>black</td>
<td>2</td>
<td></td>
<td>yes</td>
<td>yes</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>15</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>pointrange</td>
<td>black</td>
<td>0.5</td>
<td></td>
<td>yes</td>
<td>yes</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>16</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>polygon</td>
<td>NA</td>
<td>0.5</td>
<td></td>
<td>yes</td>
<td>grey20</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>quantile</td>
<td>#3366FF</td>
<td>0.5</td>
<td></td>
<td>yes</td>
<td>--</td>
<td>1</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>raster</td>
<td>--</td>
<td>--</td>
<td></td>
<td>--</td>
<td>yes</td>
<td>grey20</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>rect</td>
<td>yes</td>
<td>0.5</td>
<td></td>
<td>yes</td>
<td>grey20</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>ribbon</td>
<td>yes</td>
<td>0.5</td>
<td></td>
<td>yes</td>
<td>grey20</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>rug</td>
<td>black</td>
<td>0.5</td>
<td></td>
<td>yes</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>segment</td>
<td>black</td>
<td>0.5</td>
<td></td>
<td>yes</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>smooth</td>
<td>#3366FF</td>
<td>0.5</td>
<td></td>
<td>0.4</td>
<td>grey60</td>
<td>1</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>step</td>
<td>black</td>
<td>0.5</td>
<td></td>
<td>yes</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>text</td>
<td>black</td>
<td>5</td>
<td></td>
<td>yes</td>
<td>--</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>tile</td>
<td>yes</td>
<td>0.1</td>
<td></td>
<td>yes</td>
<td>grey20</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>violin</td>
<td>grey20</td>
<td>0.5</td>
<td>solid</td>
<td>yes</td>
<td>white</td>
<td>1</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>vline</td>
<td>black</td>
<td>0.5</td>
<td>solid</td>
<td>yes</td>
<td>white</td>
<td>1</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>
ggplot Syntax. Aesthetics Sidebar.

Note. Difference between fill, colour, and placement.

Default.

```r
ggplot(data = NULL, aes(x=numdems)) + geom_bar()
```

## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
Fill.

```r
ggplot(data = NULL, aes(x=numdems)) + geom_bar(fill = "red")
```

## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' if appropriate.
ggplot Syntax. Aesthetics Sidebar.

Colour.

```r
ggplot(data = NULL, aes(x=numdems)) + geom_bar(colour="red")
```

## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' for.

---

**Iris Malone**  
*Advanced Data Visualization in R*  
**November 6, 2015**  
25 / 68
ggplot Syntax. Aesthetics Sidebar.

Fill and Colour.

```r
ggplot(data = NULL, aes(x=numdems)) + geom_bar(fill="white", colour="red")
```

```r
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```

```r
##
```

Iris Malone

Advanced Data Visualization in R

November 6, 2015 26 / 68
ggplot Syntax. Aesthetics Sidebar

Defined inside the aesthetics argument. Ack!

```r
ggplot(data = NULL) +
    geom_bar(aes(x=numdems, colour = "red"))
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```

```
# stat_bin: binwidth defaulted to range/30. Use 'binwidth = x'
```

Iris Malone
Advanced Data Visualization in R
November 6, 2015 27 / 68
Defined inside the aesthetics argument.

```r
ggplot(data = NULL) +
  geom_bar(aes(x=numdems,
               fill=factor(I(year)<1950)))
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to 
```

```r
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' 
```
ggplot Syntax

- Adverb $\rightarrow$ stat (e.g. identity, bin)

```r
ggplot(data = df, aes(x=categorical.var, fill=group.var)) + geom_bar(stat = "bin")
```
ggplot Syntax

- **Adverb → stat** (e.g. identity, bin)

```r
ggplot(data = df, aes(x=categorical.var, fill=group.var)) + geom_bar(stat = "bin")
```

- **Preposition → position** (e.g. fill, dodge, identity)

```r
ggplot(data = df, aes(x=categorical.var, fill=group.var)) + geom_bar(stat="bin", position = "identity", binwidth=5)
```
Fearon and Laitin (2003). Let’s suppose we would like to get a feel for their data by first just looking at the number of civil wars in their dataset as a function of two variables: (1) region and (2) time.

```r
library(foreign)
df = read.dta("repdata.dta")
#subset data so 1 obs/civil war
dfonset = subset(df, df$onset == 1)
```

D.V. Civil War Onset

I.V. Ethnic Fractionalization, Mountainous Terrain, Oil, New State, Others
ggplot Syntax. Example. Civil War by Region and Decade

```r
p = ggplot(data = dfonset, aes(x=decade, fill = region)) + geom_bar(position = "identity", binwidth=5)
p
```

![Graph showing the number of civil wars by region and decade](image)
ggplot Syntax. Example. Civil War by Region and Decade

```r
p = ggplot(data = dfonset, aes(x=decade, fill = region)) + geom_bar(position = "dodge", binwidth=5)
```

![Graph showing civil wars by region and decade with bar charts for each decade from 1960 to 1990.]
Add Title.

```r
p = p +
  ggtitle("Civil Wars by Space and Time")
p
```
ggplot Syntax: Axes

Add Axes.

\[
p = p + xlab("Decade") + ylab("Civil War Frequency")
\]

\[p\]
ggplot Syntax: Theme

Change Theme.

```r
p = p + theme_bw()
p
```
ggplot Syntax: Theme

Change Theme.

```r
p = p + theme_classic() #looks like plot!
p
```
What’s great about ggplot is you can customize your own! This is what’s going on, under the hood, for theme_classic for example:

```r
p = p + theme(
  panel.grid.major = element_blank(), #grid lines
  panel.grid.minor = element_blank(),
  panel.border = element_blank(), #border
  panel.background = element_blank(), #background
  axis.line = element_line(colour = "black"), #change axes line
  axis.text.x = element_text(colour = "black"),
  axis.text.y = element_text(colour = "black"))
```
ggplot Syntax: Color.

Change Colors. Reds! Use `scale_fill_manual` for bins, `scale_colour_manual` for lines, points, and `scale_linetype_manual` otherwise.

```r
rhg_cols = c("#771C19","#AA3929","#E25033","#F27314",
              "#F8A31B","#E2C59F","#556670","#000000")
p = p + scale_fill_manual(values = rhg_cols)
p```

Civil Wars by Space and Time

- western democracies and japan
- e. europe and the former soviet union
- asia
- n. africa and the middle east
- sub-saharan africa
- latin america and the caribbean
ggplot Syntax: Color.

Or choose others! Blues!

```r
#default brewer colors
p = p +
  scale_fill_brewer()
```

```
## Scale for 'fill' is already present. Adding another scale for 'fill', which will replace the existing scale.
```

```r
p
```

Civil Wars by Space and Time

<table>
<thead>
<tr>
<th>region</th>
</tr>
</thead>
<tbody>
<tr>
<td>western democracies and japan</td>
</tr>
<tr>
<td>e. europe and the former soviet union</td>
</tr>
<tr>
<td>asia</td>
</tr>
<tr>
<td>n. africa and the middle east</td>
</tr>
<tr>
<td>sub-saharan africa</td>
</tr>
<tr>
<td>latin america and the caribbean</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1960</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1970</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1980</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1990</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
ggplot Syntax: Color.

```r
#default grey
p = p + scale_fill_grey()
```

## Scale for 'fill' is already present. Adding another scale for 'fill', which will replace the existing scale.

```
p
```

---

**Civil Wars by Space and Time**

- **western democracies and japan**
- **e. europe and the former soviet union**
- **asia**
- **n. africa and the middle east**
- **sub-saharan africa**
- **latin america and the caribbean**

---

Iris Malone

Advanced Data Visualization in R

November 6, 2015 40 / 68
ggplot Syntax: Legend.

Change Legend Labels (or Colors)

```r
p = p +
    scale_fill_manual(name ="My Super Awesome Legend Title",
                       values=c("darkblue", "darkred", "darkgreen",
                                 "grey", "darkorange", "purple"),
                       labels=c("Western Dems and Japan", "Former USSR", "Asia",
                                 "North Africa and \n Middle East", "Sub-Saharan Africa",
                                 "Latin America"))
```

## Scale for 'fill' is already present. Adding another scale for
ggplot Syntax: Legend.

Why define fill (or color or linetype) inside aes? Keep track of variables!

Recall:

```r
ggplot(data = NULL, aes(x = year)) +
  geom_line(aes(y = numdems, colour = "numdems")) +
  geom_line(aes(y = numauts, colour = "numauts")) +
  scale_color_manual(name = "Regime Type",
                     labels = c(numdems="Democracies", numauts = "Autocracies"),
                     values=c(numdems="blue",numauts="red"))
```
ggplot Syntax: Legend.

Note: p is ggplot object with 2 components in aes: x = decade and fill = region

```r
p = p +
    scale_fill_manual(name = "My Super Awesome Legend Title",
                      values = c("darkblue", "darkred", "darkgreen",
                                  "grey", "darkorange", "purple"),
                      labels = c("Western Dems and Japan", "Former USSR", "Asia",
                                  "North Africa and \n Middle East", "Sub-Saharan Africa",
                                  "Latin America"))
```

```r
## Scale for 'fill' is already present. Adding another scale for
```
Civil Wars by Space and Time

- Western Dems and Japan
- Former USSR
- Asia
- North Africa and Middle East
- Sub-Saharan Africa
- Latin America

Legend:
- My Super Awesome Legend Title
ggplot Syntax: Legend.

Change the position of the legend.

```r
p = p + theme(legend.position="top")
p
```
ggplot Syntax: Legend.

Change the position of the legend.

```r
# Position legend in graph, where x,y is 0,0 (bottom left)  
# to 1,1 (top right)

p = p + theme(legend.position=c(0.10, .8))

p
```

![Civil Wars by Space and Time chart]

- **Western Dems and Japan**
- **Former USSR**
- **Asia**
- **North Africa and Middle East**
- **Sub-Saharan Africa**
- **Latin America**
Remove the legend.

\[
p = p + \text{theme(legend.position="none")}
\]

\[
p
\]
ggplot Syntax: Annotation.

Suppose you want to add a label. For example, what’s the one western democracy with a civil war in the 1960s?

```r
#it's the UK vs the IRA
p = p + annotate("text", label = "UK vs IRA",
               x = 1959, y = 2, size = 6, colour = "black")
p
```

Iris Malone
Advanced Data Visualization in R
November 6, 2015 48 / 68
Suppose you want a separate barplot for every decade.

```r
ggplot(data = dfonset, aes(x=decade, group = region, 
    fill = factor(region))) + geom_bar(stat="bin", position = "dodge", binwidth=5) + facet_wrap(~region) #var you want separate plots by
```
ggplot Syntax: Saving results.

Option 1.

```r
pdf("nameoffile.pdf", width=12, height = 5)
p
dev.off()
```

```
## pdf
## 2
```

Option 2.

```r
ggsave(p, file="nameoffile.pdf", width=12, height=5)
```
Visualizing Regression Results

- Coef Plots
- Not covered here: marginal effects, predicted outcomes
\[ m1 = \text{glm}(\text{onset} \sim \text{warl} + \text{gdpenl} + \text{lpopl1} + \text{lmtnest} + \text{ncontig} + \text{Oil} + \text{nwstate} + \text{instab} + \text{polity2l} + \text{ethfrac} + \text{relfrac}, \text{data} = \text{df}, \text{family} = \text{"binomial"}) \]

\text{library(coefplot)}

\text{coefplot(m1)}
Coefplot: Our Own Function

Adapted from Stat Bandit

```r
#Format the data
coeffplot.gg = function(model, data){
    # data is a data frame with 4 columns
    # data$names gives variable names
    # data$modelcoef gives center point
    # data$ylo gives lower limits
    # data$yhi gives upper limits
    modelcoef = summary(model)$coefficients[1:length(model$coefficients), 1]
    modelse = summary(model)$coefficients[1:length(model$coefficients), 2]
    ylo = modelcoef - qt(.975, nrow(data))*(modelse)
    yhi = modelcoef + qt(.975, nrow(data))*(modelse)
    names = names(m1$coefficients)
    dfplot = data.frame(names, modelcoef, modelse, ylo, yhi)
    # ...
}
```
Define the plot

cofplot.gg = function(model, data){
  # ... 
  # define plot
  library(ggplot2)
  p = ggplot(dfplot, aes(x=names,
                          y=modelcoef,
                          ymin=ylo, ymax=yhi))
  + geom_pointrange(colour=ifelse(ylo < 0 & yhi > 0, 
                                 "red", "blue"))
  + theme_bw() + coord_flip()
  + geom_hline(aes(x=0), lty=2)
  + xlab('Variable') + ylab('')
  return(p)
}
Evaluate the function

\texttt{coefplot.gg(m1, df)}
Spatial Visualization

Packages:

- rworldmap
- maps
- ggmap

\[
\text{suppressMessages}\left(\text{library}\left(\text{maps}\right)\right)
\text{suppressMessages}\left(\text{library}\left(\text{ggmap}\right)\right)
\text{suppressMessages}\left(\text{library}\left(\text{mapproj}\right)\right)
\]
Adapted from Mahbubul Majumder.

```r
dfworldmap = map_data("world")

ggplot() + geom_polygon(aes(x=long, y=lat, group=group),
                        fill="grey65",
                        data=dfworldmap) + theme_bw()
```

Iris Malone
Advanced Data Visualization in R
November 6, 2015
Chloropleth maps

Suppose we want to map different levels of a variable by some unit like a state or country. For a toy example, we’ll map 1973 murder rates by state using the USArrests data.

Step 1: Format data

```r
suppressMessages(library(dplyr))
us = map_data("state")
head(us)
```

```r
##    long     lat group order region subregion
## 1 -87.46201 30.38968   1     1   alabama <NA>
## 2 -87.48493 30.37249   1     2   alabama <NA>
## 3 -87.52503 30.37249   1     3   alabama <NA>
## 4 -87.53076 30.33239   1     4   alabama <NA>
## 5 -87.57087 30.32665   1     5   alabama <NA>
## 6 -87.58806 30.32665   1     6   alabama <NA>
```
Chloropleth maps

```r
head(USArrests)
```

<table>
<thead>
<tr>
<th></th>
<th>Murder</th>
<th>Assault</th>
<th>UrbanPop</th>
<th>Rape</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alabama</td>
<td>13.2</td>
<td>236</td>
<td>58</td>
<td>21.2</td>
</tr>
<tr>
<td>Alaska</td>
<td>10.0</td>
<td>263</td>
<td>48</td>
<td>44.5</td>
</tr>
<tr>
<td>Arizona</td>
<td>8.1</td>
<td>294</td>
<td>80</td>
<td>31.0</td>
</tr>
<tr>
<td>Arkansas</td>
<td>8.8</td>
<td>190</td>
<td>50</td>
<td>19.5</td>
</tr>
<tr>
<td>California</td>
<td>9.0</td>
<td>276</td>
<td>91</td>
<td>40.6</td>
</tr>
<tr>
<td>Colorado</td>
<td>7.9</td>
<td>204</td>
<td>78</td>
<td>38.7</td>
</tr>
</tbody>
</table>

#mismatch between region and state
# need to add var to arrest data to match 1:1
arrest = USArrests %>%
  add_rownames("region") %>%
  # use mutate function from plyr
  mutate(region = tolower(region)) # make it all lowercase

# format to work with map
head(arrest)

## Source: local data frame [6 x 5]
##
## region Murder Assault UrbanPop Rape
## (chr) (dbl) (int) (int) (dbl)
## 1 alabama 13.2 236 58 21.2
## 2 alaska 10.0 263 48 44.5
## 3 arizona 8.1 294 80 31.0
## 4 arkansas 8.8 190 50 19.5
## 5 california 9.0 276 91 40.6
## 6 colorado 7.9 204 78 38.7
Chloropleth maps

Step 2: Plot the base map layer

```r
# must define map first

g = ggplot()
g + geom_map(data=us, map=us,
             aes(x=long, y=lat, map_id=region),
             fill="#ffffff", color="#ffffff", size=0.15)
```

```
Iris Malone
Advanced Data Visualization in R
November 6, 2015 61 / 68
```
Chloropleth maps

Step 3: Add our arrest data

g = g + geom_map(data=arrest, map=us,
    aes(fill=Murder, map_id=region),
    color="#ffffff", size=0.15)
g
Chloropleth maps

Step 4: Make it look pretty.

```r
library(tmap)
library(tmapextras)

# Load murder data
murder_data <- read.csv("murder_data.csv")

# Create a tmap object
murder_tmap <- tmap_init("state")

# Plot the data
murder_tmap <- tmap_plot(murder_tmap, murder_data, variable = "Murder")

# Add color scale
murder_tmap <- murder_tmap + scale_fill_continuous(low = "thistle2", high = "darkblue", guide = "colorbar")

# Add axis labels
murder_tmap <- murder_tmap + xlab("") + ylab("")

# Add map projection
murder_tmap <- murder_tmap + coord_map(albers, lat0 = 39, lat1 = 45)

# Customize theme
murder_tmap <- murder_tmap + theme(panel.border = element_blank()) + theme(panel.background = element_blank()) + theme(axis.ticks = element_blank()) + theme(axis.text = element_blank())

# Final result
print(murder_tmap)
```

```
g = g + scale_fill_continuous(low='thistle2', high='darkblue', guide='colorbar') + xlab("")
g = g + theme(panel.border = element_blank()) + theme(panel.background = element_blank()) + theme(axis.ticks = element_blank()) + theme(axis.text = element_blank())
g
```
```r
map1 = suppressMessages(get_map(
    location = 'Stanford University', zoom = 14, #zoom-in level
    maptype="satellite")) #map type
ggmap(map1)
```
Step 1: Pull a location you want to plot from Google maps.

```r
map = suppressMessages(get_map(location = 'Europe', zoom = 4))
ggmap(map)
```
Step 2: Get geocoordinates for points or locations you’re interested in.

europegps = suppressMessages(geocode(c("Lisbon, Portugal", "Eiffel Tower", "Berlin, Germany", "Crimea, Ukraine"), source="google"))

europegps

## lon     lat
## 1 -9.139337 38.72225
## 2  2.294481 48.85837
## 3 13.404954 52.52001
## 4 34.102417 44.95212
Step 3: Add geocoordinates to ggmap! It’s just like working with another ggplot object.

```r
ggmap(map) + geom_point(aes(x=europegps$lon, y = europegps$lat, lwd = 4, colour = "red") + ggttitle("Place I would like to Visit!")
```

![Place I would like to Visit!](image-url)
Summary

- ggplot is super powerful
Summary

- ggplot is super powerful but kind of annoying to learn
- ggplot is super powerful but kind of annoying to learn
- Ability to make complicated graphs awesome
Summary

- ggplot is super powerful but kind of annoying to learn
- Ability to make complicated graphs awesome
- Benefits outweigh the start-up costs
Summary

- ggplot is super powerful but kind of annoying to learn
- Ability to make complicated graphs awesome
- Benefits outweigh the start-up costs

**GGPLOT ALL THE THINGS**