an introduction to R graphics for epidemiologists

Graphics

Charles DiMaggio, PhD, MPH, PA-C

Center for Injury Epidemiology and Prevention
Departments of Anesthesiology and Epidemiology
Columbia University
New York, NY 10032

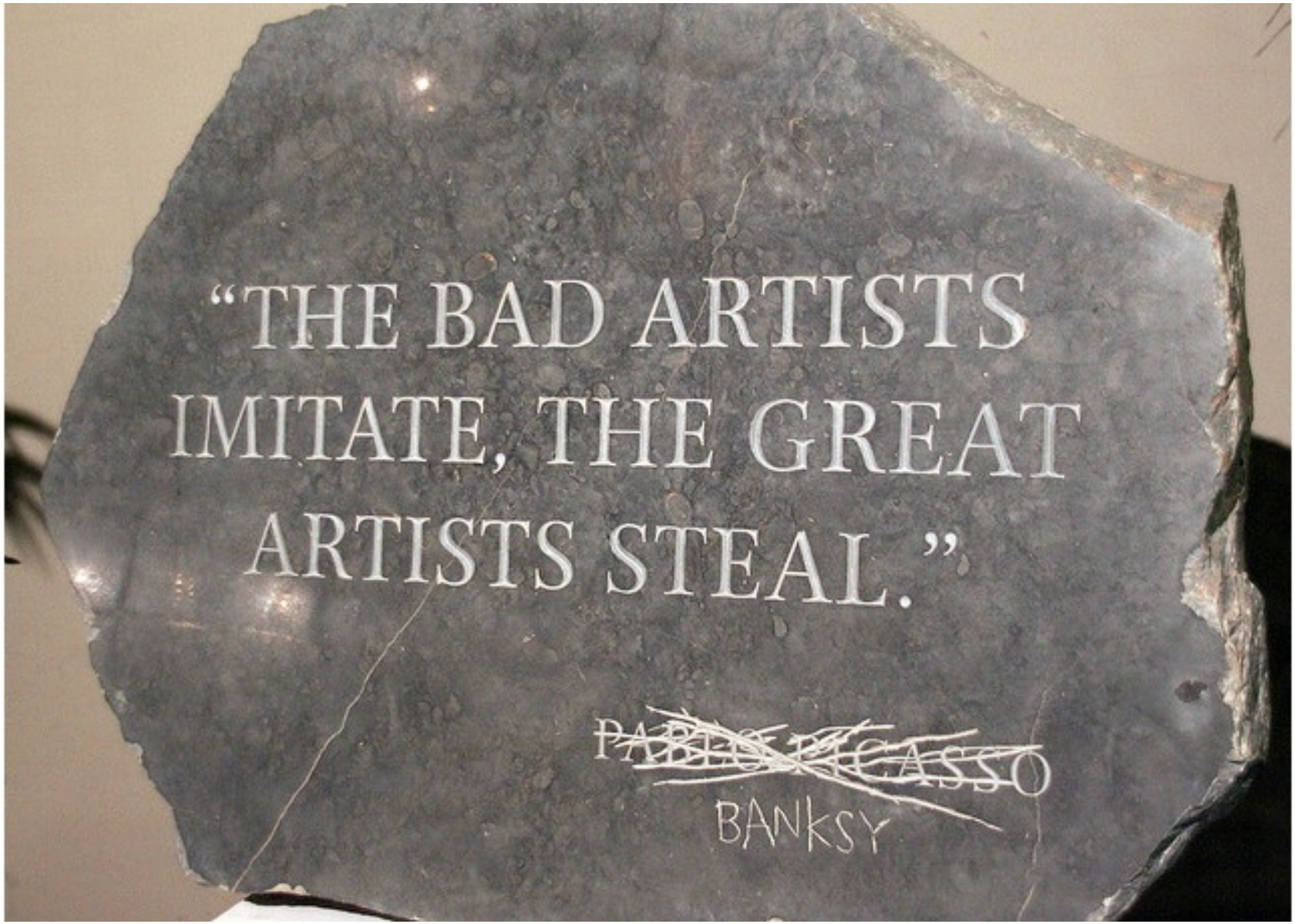
cjd11@columbia.edu

executive masters program
data visualization course
Fall 2013

Outline

1. some things people are doing with R
   - just because you can doesn't mean you should
2. installing and getting around in R
3. basic graphs in R
4. graphing examples
   - a line with confidence limits
   - epidemic curve
5. ggplot2

I am no artist, but I do steal...



Credit where credit is due...

- Tomas Aragon, MD, DrPH
  - Applied Epidemiology Using R
  - `http://www.epitools.net/`
- John Fox, PhD
  - An Introduction to Statistical Computing in R
  - `http://socserv.mcmaster.ca/jfox/Courses/UCLA/index.html`
- Bill Venebles, PhD
  - An Introduction to R
  - `cran.r-project.org/doc/manuals/R-intro.pdf`
- Hadley Wickham, PhD
  - ggplot2, plyr, stringr, lubridate, etc..., etc...
  - `http://had.co.nz`

what is R?

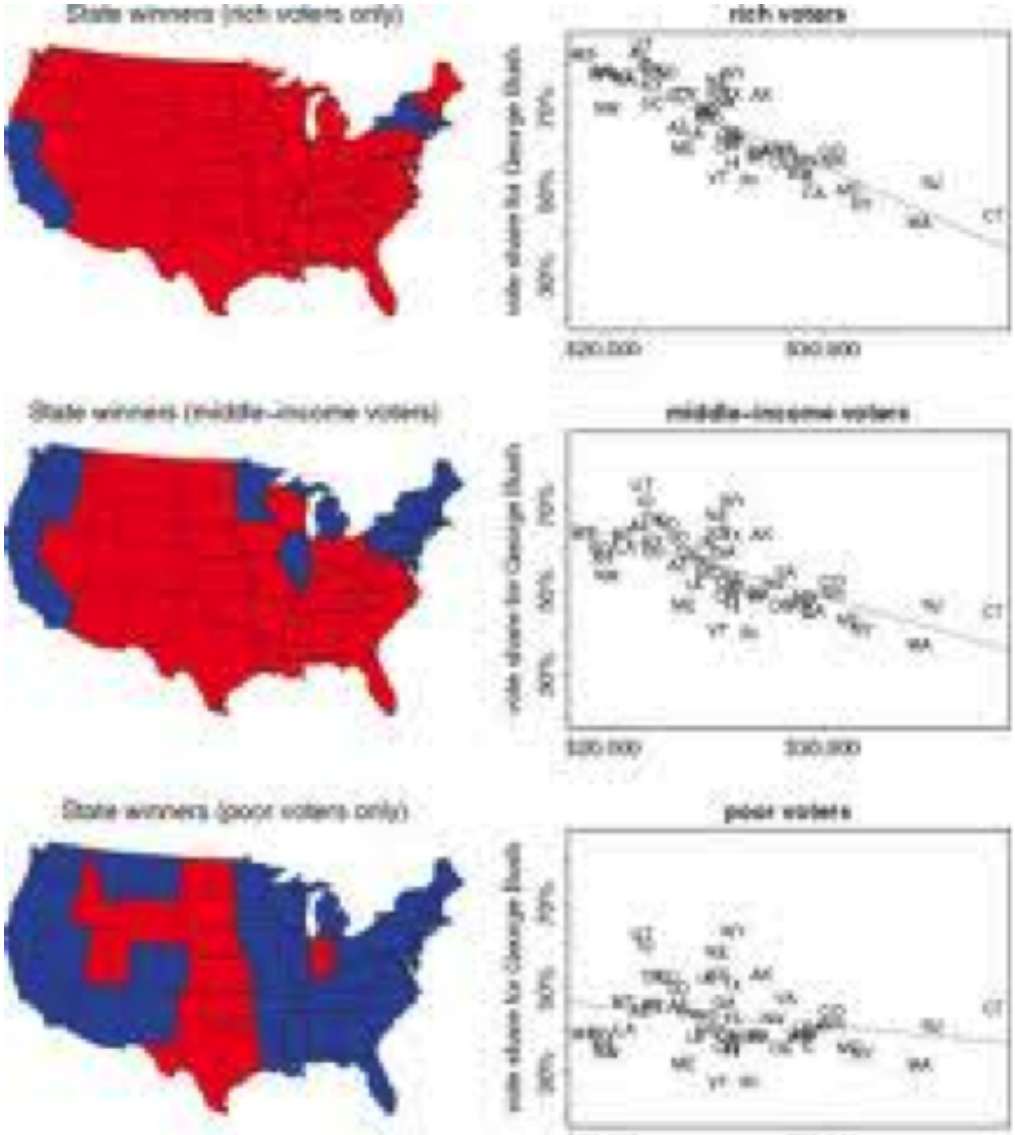
A flexible, scalable, **free** tool for the description, analysis, visual display, exploration and interpretation of data.

- a calculator
- a suite of statistical tools
- a graphics creator
- a programming language
- a simulation lab
- a means of scientific documentation and discourse

It is uniquely suited to epidemiological analysis.

Predicting Elections

Andrew Gelman, "Red State, Blue State, Rich State, Poor State"



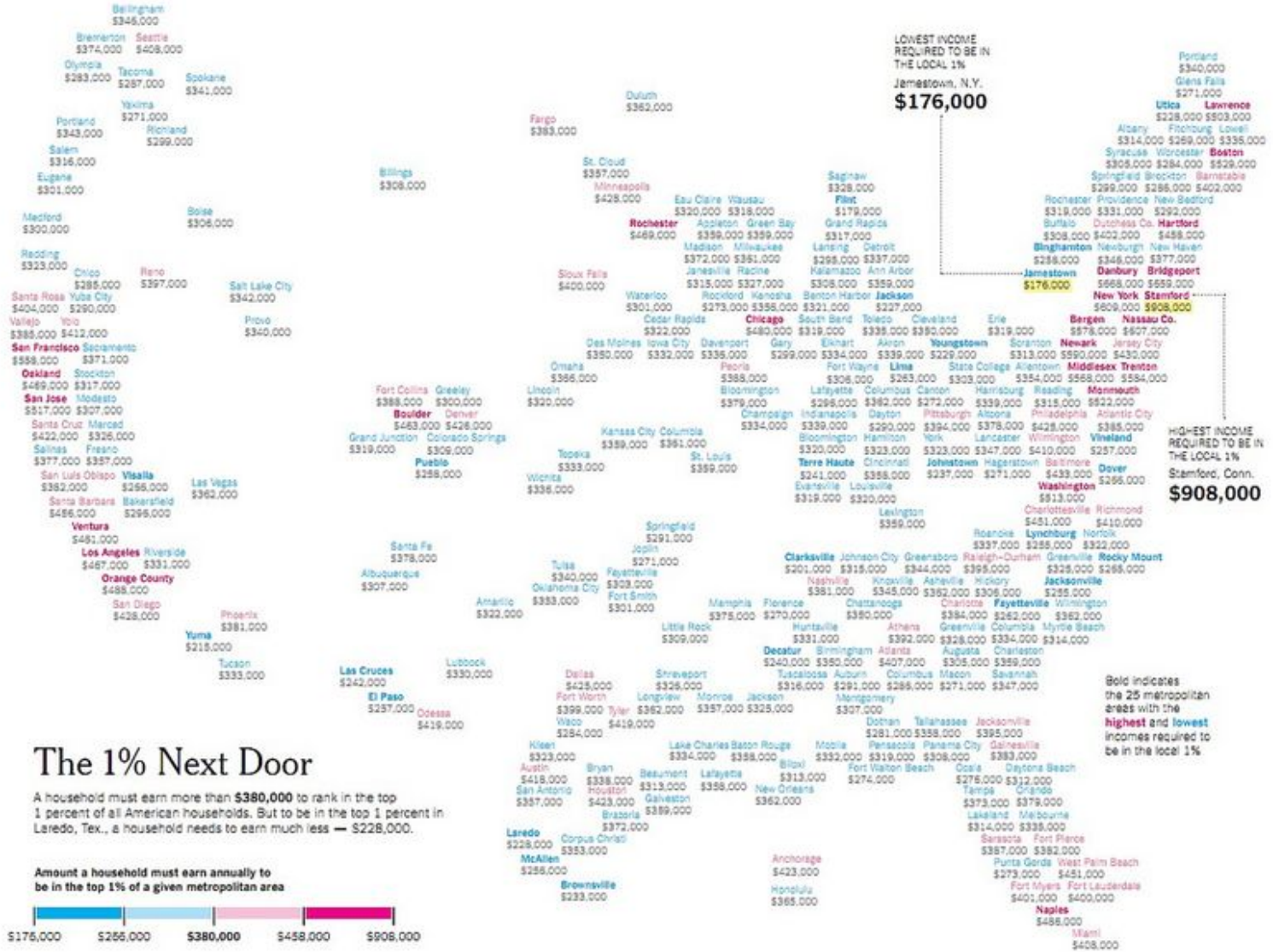
Crawling and Scraping the Web

John Muschelli, Andrew Jaffe, Jeffrey Leek. Simply Statistics Blog

Making Money

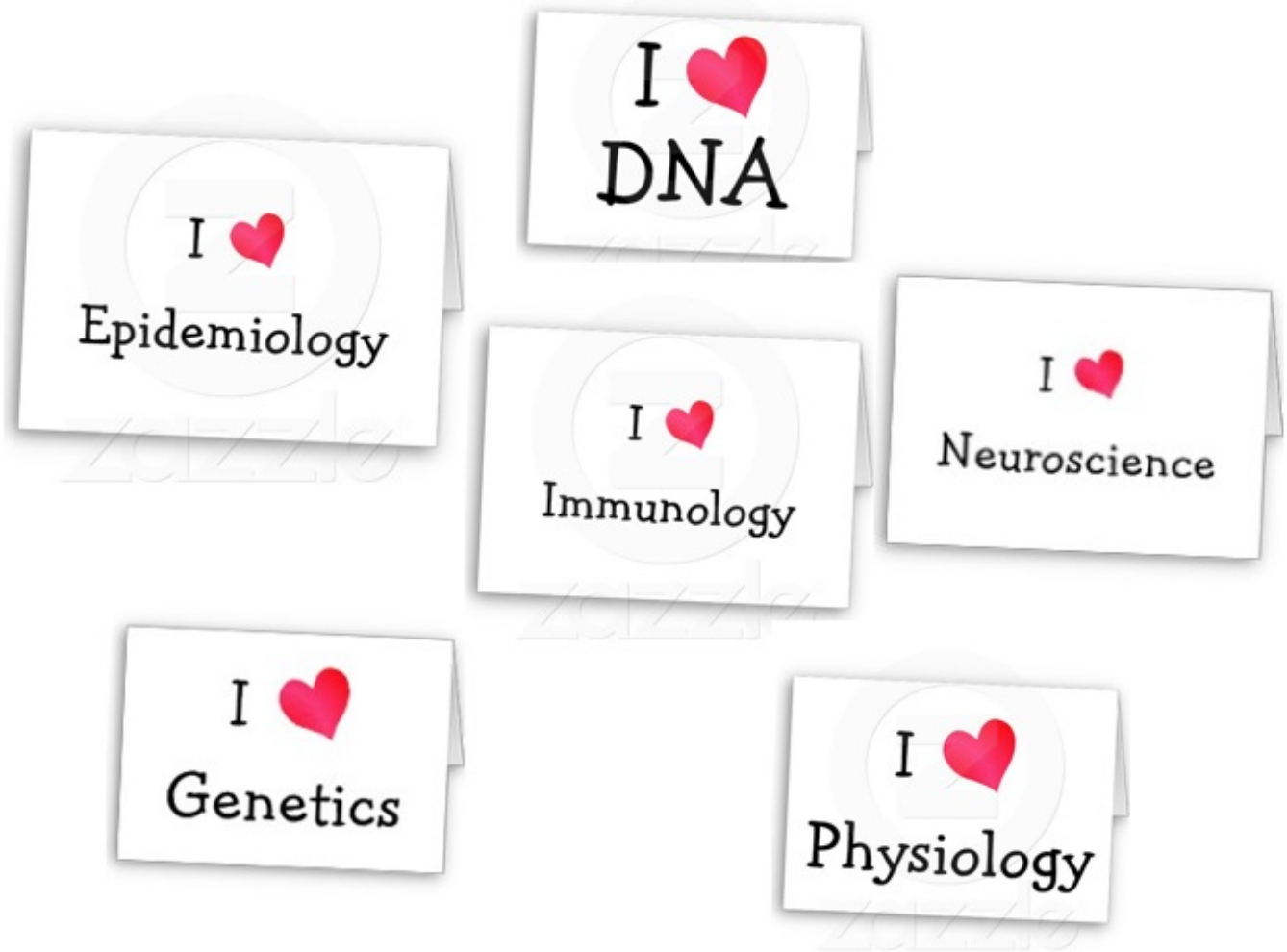
Publishing Newspapers

New York Times is an R Shop



Doing Science

Developed by Scientists for Scientists

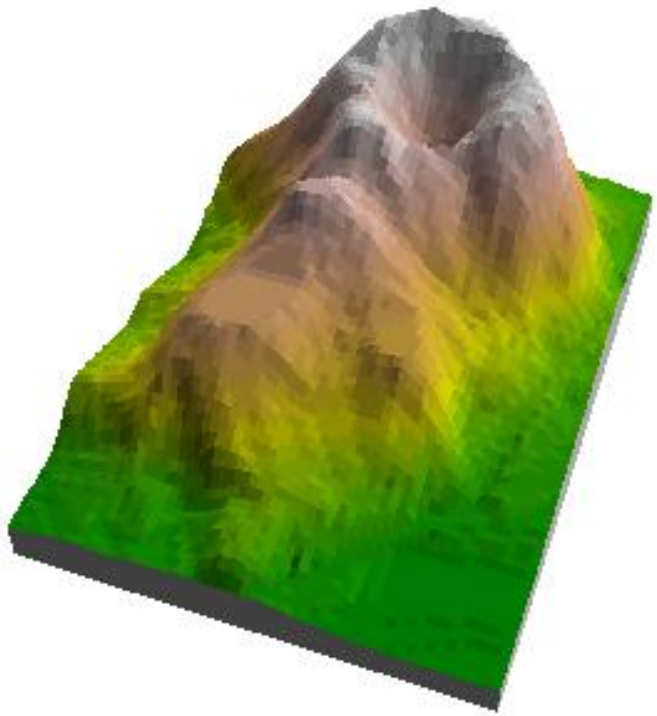


making some stunning graphics

Paul Butler Visualizing Facebook Friends

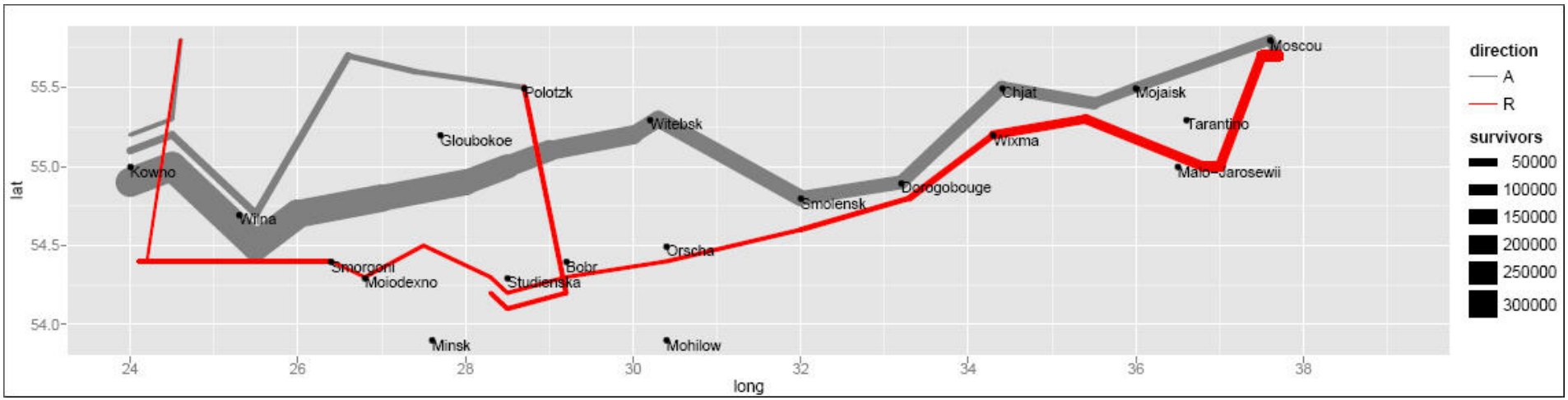
Maunga Whau

R Development Core Team



Napoleon's March to Moscow (Minard)

Hadley Wickham (ggplot2)



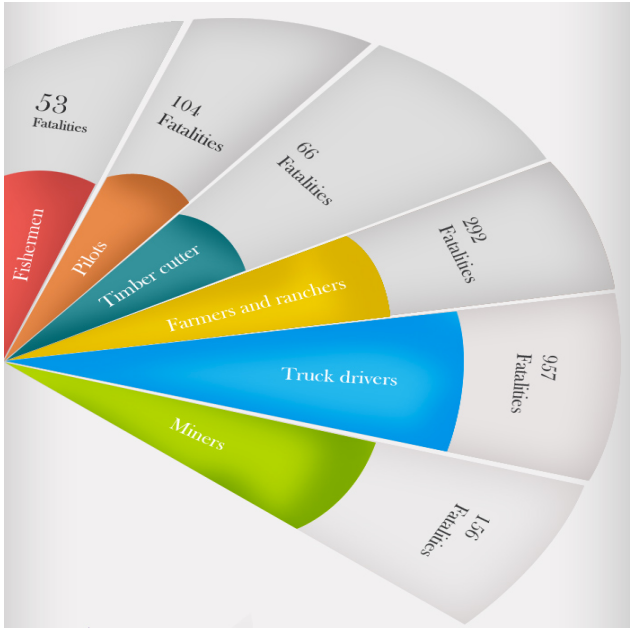
Animations

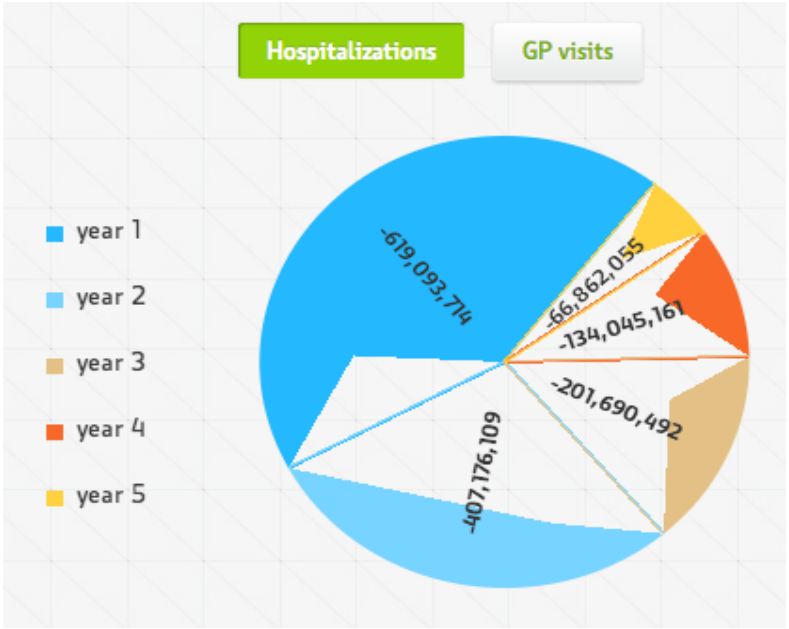
Click on the links

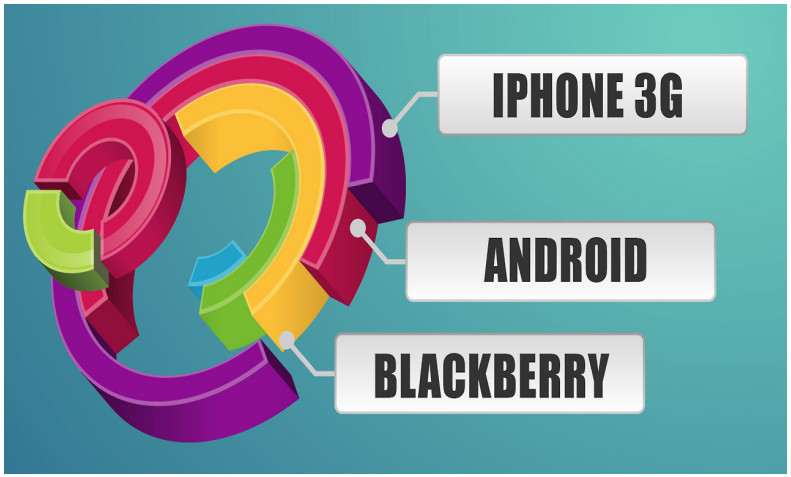
- The Age of Sail
- Wind Map
- Mandelbrot Set

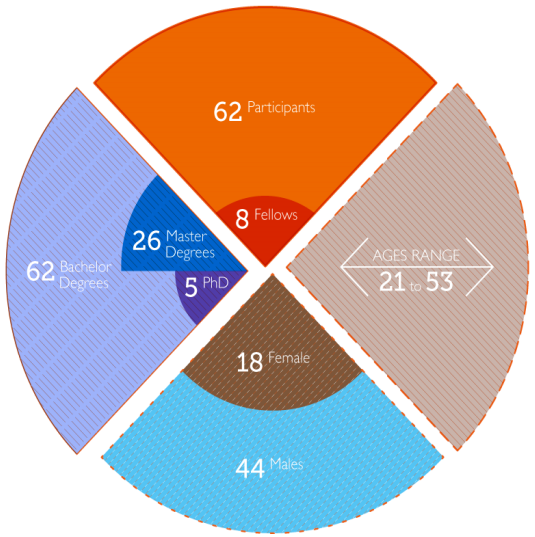
Outline

too much pie

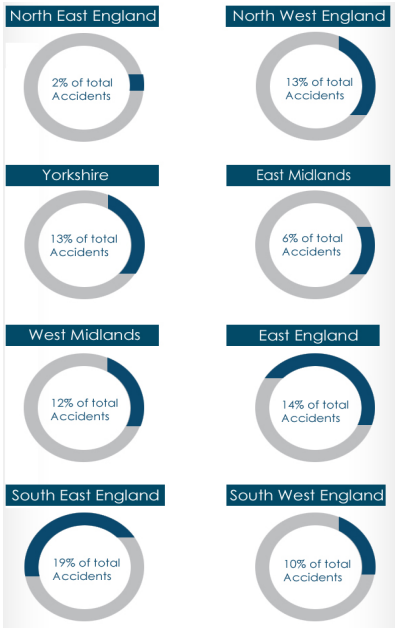

53 Fatalities
104 Fatalities
66 Fatalities
992 Fatalities
957 Fatalities
156 Fatalities
Fishermen
Pilots
Timber cutter
Farmers and ranchers
Truck drivers
Miners

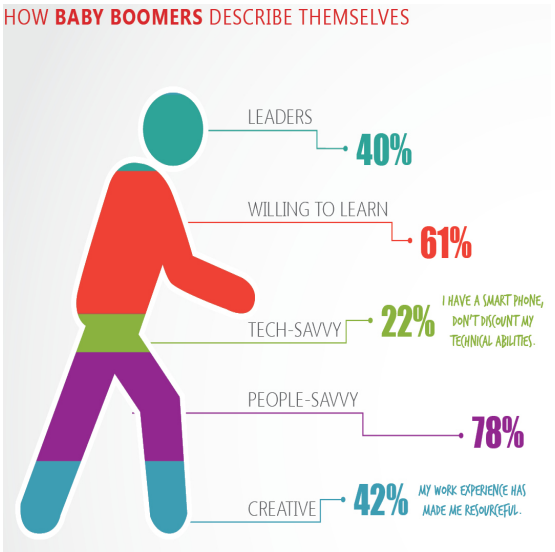

Hospitalizations
GP visits
year 1
year 2
year 3
year 4
year 5
-619,093,714
-66,862,055
-134,045,161
-201,690,492
-407,176,109


IPHONE 3G
ANDROID
BLACKBERRY


62 Participants
8 Fellows
26 Master Degrees
5 PhD
62 Bachelor Degrees
AGES RANGE
21 to 53
18 Female
44 Males

more than the sum of its parts





what R is not

- a GUI experience
- initially easy and intuitive
- warrantied in any way (if it runs, it can be on CRAN)
- a DBMS
- well suited to enormous data sets (*but that is changing...*)
  - historical 32-bit limit ($2^{31} - 1$) on size of a vector
  - in R, objects like matrices are actually vectors
  - R stores everything in RAM
  - rules of thumb: 100,000 rows, 20 variables (*very* conservative), single object 10% of RAM, etc...
  - change is in the air
    - 64 bit computing, increased RAM
    - interest in 'big data' in R
      `http://www.cybaea.net/Blogs/Data/Big-data-for-R.html`
    - packages like *ff, bigmemory* take advantage of "paralellism" (Hadoop, MapReduce)

so, why learn R?

Many options for epidemiological computing: SAS, STATA, SPSS, Mathematica, Excel....

But, if you want to ...

- accomplish many tasks with a single tool
- better understand the methods you use
- use methods not available in any other program
- develop and share your own methods
- collaborate with wide community of scientific colleagues

...R might be for you.

and did I mention it's free?

Outline

how to install R

1. go to `http://www.r-project.org/`
2. select CRAN (Comprehensive R Archive Network) from left menu
3. link to nearby geographic site (e.g., http://software.rc.fas.harvard.edu/mirrors/R/)
4. select your operating system
5. chose "Base" installation
6. save R-X.X.X-win32.exe (windows) or R-X.X.X-mini.dmg (Mac OS X)
7. run the installation program accepting defaults

R is a calculator

math operators and functions

```
arithmetic   + , - , * , /
power    ^
```

convert 68 degrees Fahrenheit to Celsius ($C^0 = \frac{5}{9}(F^0 - 32)$)

```
5/9*(68-32)
```

assignment operator

'memory' key

<-

```
y <- 5^3 #assignment (no display )
y
(y <- 5^3) #assignment (display)
```

functions

R "apps"

fx()

math operators and functions

```
mathematical functions -  sqrt, log, exp, sin, cos, tan
simple functions -   max, min, length, sum, mean, var, sort
```

```
abs(-23) #absolute value
exp(8) # exponentiation
log(exp(8)) # natural logarithm
sqrt(64) # square root
```

concatenation function

combine or "vectorize"

c()

,

```
x <- c(1,2,3,4,5)
x
y<-c("a", "b", "c", "d")
y
```

getting started in R graphics

Learn by Example

```
demo(graphics)
example(plot)
example(persp)
```

for many user-written examples

Outline

graphing the Duncan data set

John Fox car (companion to applied regression) package

Load the data

```
install.packages("car")
library(car)
data(Duncan)
?Duncan
```

univariate

```
plot(Duncan$income)
plot(Duncan$income, type="l")
plot(Duncan$income, type="h")
plot(Duncan$income, type="h", col="red", lty=3, lwd=5)
```

bivariate

```
plot(Duncan$income, Duncan$education)
abline(lm(Duncan$income~ Duncan$education))
```

add titles and labels

```
plot(Duncan$income, Duncan$education, main="Relationship of
Income and Education", ylab="Education Index",
xlab="Yearly Income (Thousands)")
abline(lm(Duncan$income~Duncan$education), lty=2)
```

R graphic flavors

- base - high-level functions create plots, titles, etc, low-level functions add to existing plots
  - become comfortable with basic plotting before moving on to things like mapping or ggplot2
  - Venables Chapter 12 is your friend
  - Quick-R has a very nice overview
- lattice - (Deepayan Sarkar) based on Trellis from S, allows multi-panels, work on grid
- ggplot2 - (Hadley Wickham) based on "The Grammer of Graphics" (Leland Wilkinson) layer elements to build a graphic
- sp - (Roger Bivand) great package for mapping and spatial analysis (depends on other equally nice packages)

plot()

- generic, high-level, type of plot depends on *class* of arguments
- plot(x,y) - scatterplot
- plot (x) - time series if x is vector, barplot if x is a factor
- plot(f,y) - boxplots if f is factor and y is vector

high-level graphing functions

- hist(x) histograms, lets R choose the breaks
- hist(x, nclass=n) - you choose the number of breaks, (probability=TRUE) bars represent relative frequencies instead of counts
- qqnorm(x) - plots x against normal equivalent
- image(x,y,z) - 3-variable plots, returns a grid
- contour(x,y,z) - returns contour lines,
- persp(x,y,z) - returns 3D image

basic graphics parameters

- lty - line type
- col - color
- lwd - line width
- cex - character expansion (scales characters smaller or larger)
- font, font.axis, font.label, font.main, font.sub
- mfrow=() mfcol=() - for multiple figures, e.g. mfrow=c(3,2) sets up a 3 by 2 figure

plotting a line with upper and lower confidence intervals

- residential proximity to wtc and anxiety
- number of dxs by increasing number of miles from WTC holding the other variables constant at their median values
- 4 sets of numbers: lower limit, point estimate, upper limit and variable against which to plot (in this case miles)

read in data

```
#miles from WTC
miles<-c(2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24,
     26, 28, 30)
# point estimate
est<-c(570.7778, 527.9492, 488.3342, 452.1437, 418.2168,
386.8357, 357.8092, 330.9608, 306.4333, 283.4399, 262.1718,
242.4996, 224.3035, 207.6803, 192.0969)
#lower limit
low<-c(450.7893, 445.4121, 437.4664, 420.7337, 382.9866,
337.6467, 294.4178, 255.6988, 221.6279, 192.2891, 166.6674,
144.4596, 125.3362, 108.6357, 94.16043)
#upper limit
upper<-c(711.2329, 619.5541, 542.398, 483.9589, 456.2313,
445.4121, 437.9041, 431.3846, 425.8129, 421.1546, 416.5473,
412.4026, 408.2991, 404.6409, 400.6147)
```

the plot

```
#main plot
# note zero the axes, add axes titles etc
plot(miles, est, xlab="Distance from WTC in Miles",
 ylab="Number of Anxiety-Related Diagnoses",
  ylim=c(0,700), pch=15)

#add confidence limit lines
points(miles, low)
points(miles, upper)
lines(miles, est)
lines(miles, low)
lines(miles, upper)
```

plotting an epidemic curve

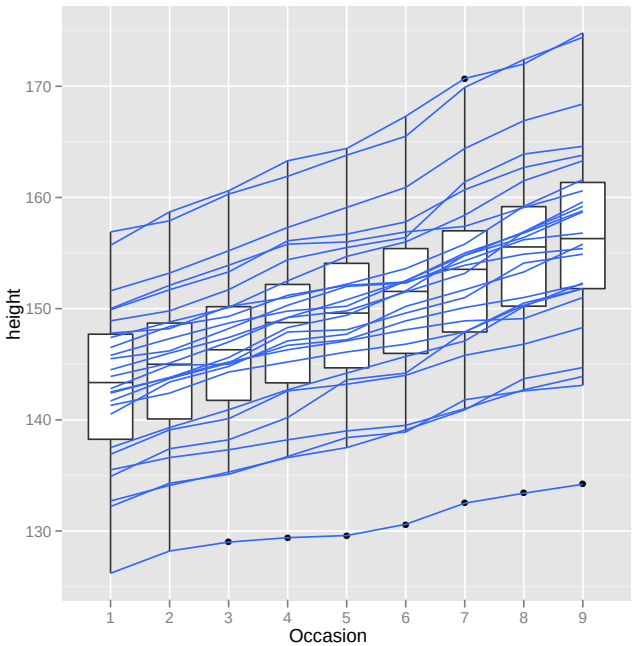
epitools

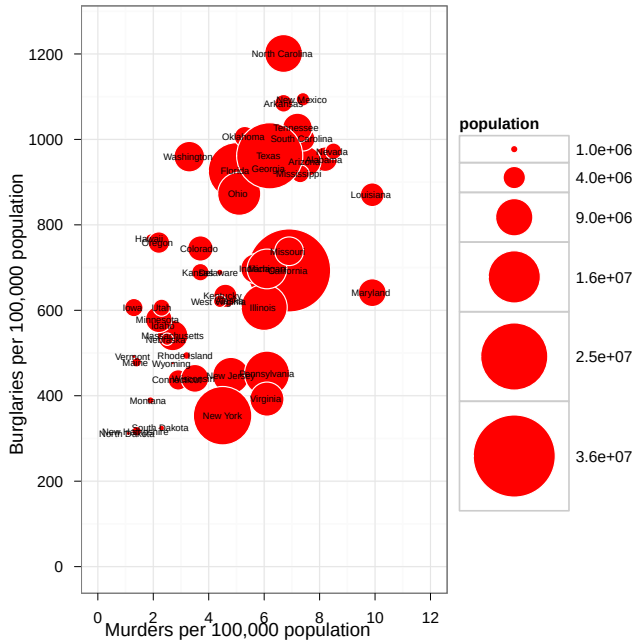
```
library(epitools)
sampdates <- seq(as.Date("2004-07-15"),
      as.Date("2004-09-15"), 1)
x <- sample(sampdates, 100, rep=TRUE)
xs <- sample(c("Male","Female"), 100, rep=TRUE)
epicurve.weeks(x)
epicurve.weeks(x, strata = xs)
rr <- epicurve.weeks(x, strata = xs, segments = TRUE)
```

Outline

the ggplot2 package

- Developed by Hadley Wickham
- Based on the Grammar of Graphics (Wilkinson, 2005)
- Plots built up by adding layers
- Uses (somewhat) idiosyncratic vocabulary
  - data are *mapped* to attributes or *aesthetics* using *geometries* and *scales* and can be displayed as multiple plots or *facets*





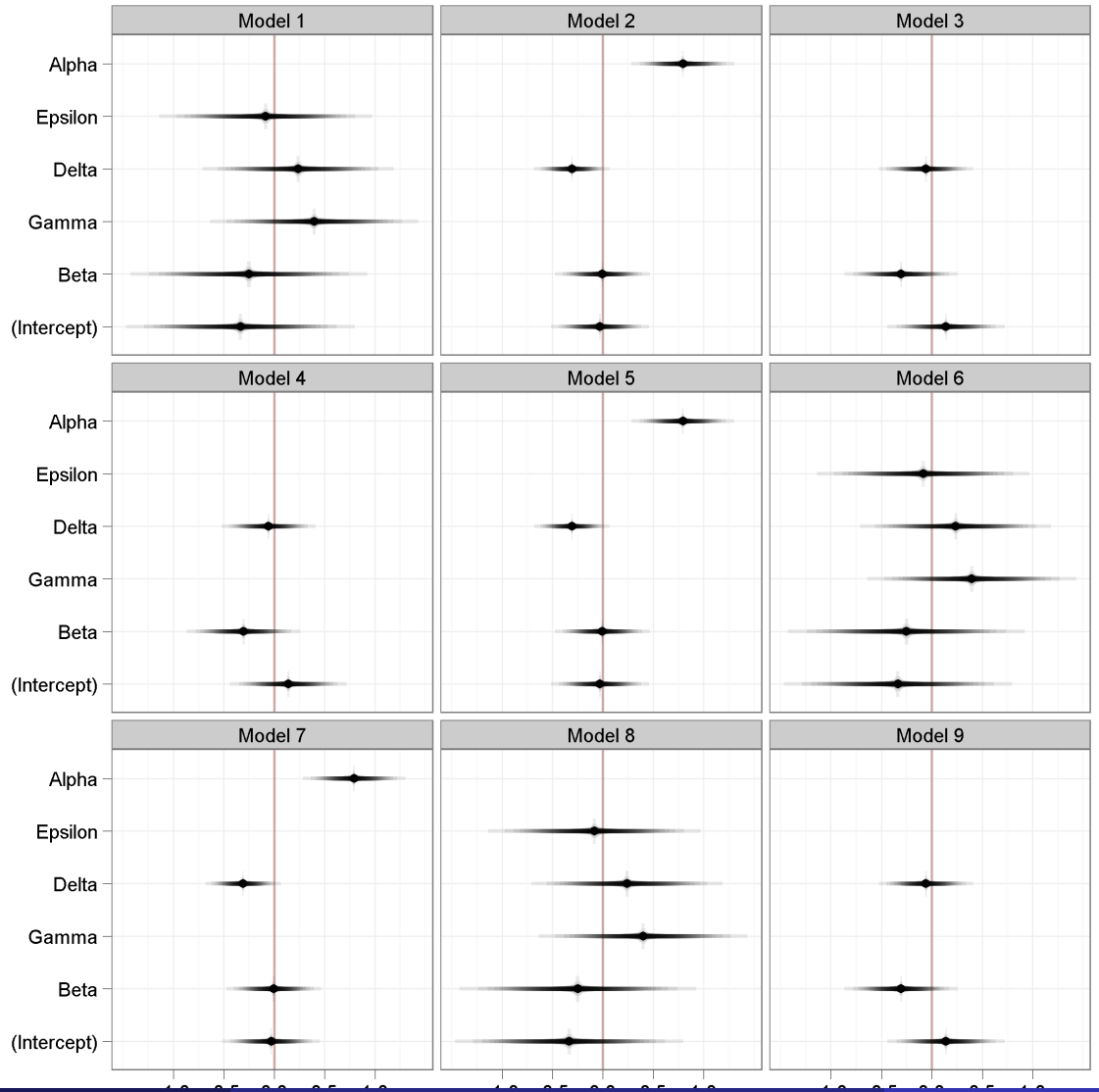
the grammar of graphics

plots built in layers that consist of elements

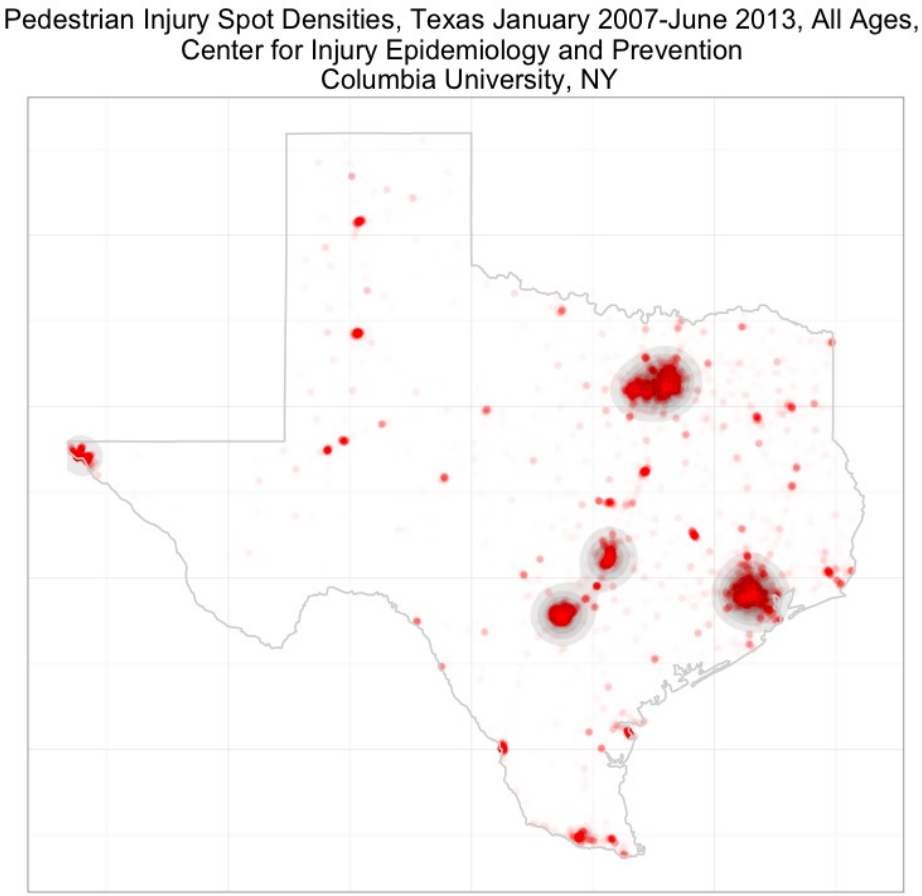
- geometries - represent data (points, bars, lines)
  - geom_xxx
- aesthetics - colors, shapes, sizes
  - aes()
  - consider whether makes sense for the data (discrete vs. continuous, ordered vs. unordered)
- scales - map geometries to space (linear, logarithmic)
- statistics - identity, mean
- coordinates - how elements represented on page ("canvas")

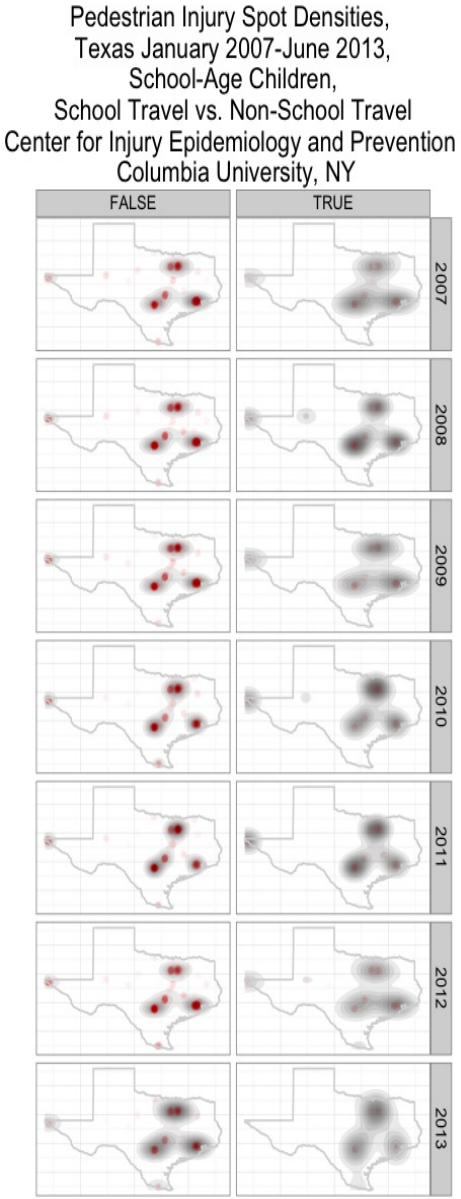
Regression Coefficients (Faceting)

David Sparks



mapping pedestrian injuries





qplot()

qplot()

1. defaults for quick plotting
2. accepts transformed variables
3. define "aesthetics" by multiple variables categorized by things like color or size

graphically exploring motor trend car tests

```
library(ggplot2)
data(mtcars)
head(mtcars)
?mtcars
qplot(wt, mpg, data=mtcars)
qplot(log(wt), mpg-10, data=mtcars)
qplot(log(wt), mpg-10, data=mtcars, color=qsec)
qplot(log(wt), mpg-10, data=mtcars, color=qsec, size=cyl)
```

save plot and add to it

```
plot1<-qplot(wt, mpg, data=mtcars, geom=c("point", "smooth"))
plot1
plot2<-plot1+facet_wrap(~cyl)
plot2
```

from qplot() to ggplot()

(see Christophe Ladroue)

- more control
- basic form: ggplot() + geom_xxx()
- steps in creating plot
  1. define the data - has to be data frame
  2. add first layer - geometry (plot type) and its aesthetics (variables, categorizing factors)
  3. more layers - scales, facets, titles

```
data(diamonds)
set.seed(53)
small<-diamonds[sample(nrow(diamonds),1000),]

p1<-ggplot(small)
p2<-p1+geom_point(aes(x=carat,y=price,colour=cut))
p2
p2+scale_y_log10()+facet_wrap(~cut)+ggtitle("Cut by Price")
```

geoms, aesthetics, facets

- geoms need **"aesthetics"**
  - shapes defined by name of the geom, e.g. "geom_point"
  - x (and y) variables required
  - color, size, fill, alpha (transparency)
- varied **ggplot syntax**

```
ggplot(small,aes(x=carat,y=price,colour=cut))+geom_point()
ggplot(small,aes(x=carat,y=price))+geom_point(aes(colour=cut))
```

- **other geoms**: geom_smooth for trends

```
p<-ggplot(small,aes(x=carat,y=price))
p+geom_point()+geom_smooth()+facet_wrap(~cut)
p+geom_point()+geom_smooth(method="lm")+facet_wrap(~cut)
```

- **faceting** - to categorize variables
  - facet_wrap() - by single variable

```
p2+facet_wrap(~cut)
p2+facet_wrap(~cut,nrow=1)
p2+facet_wrap(~cut,ncol=1)
```

  - facet_grid - more than one variable

```
p2+facet_grid(cut~color)
```

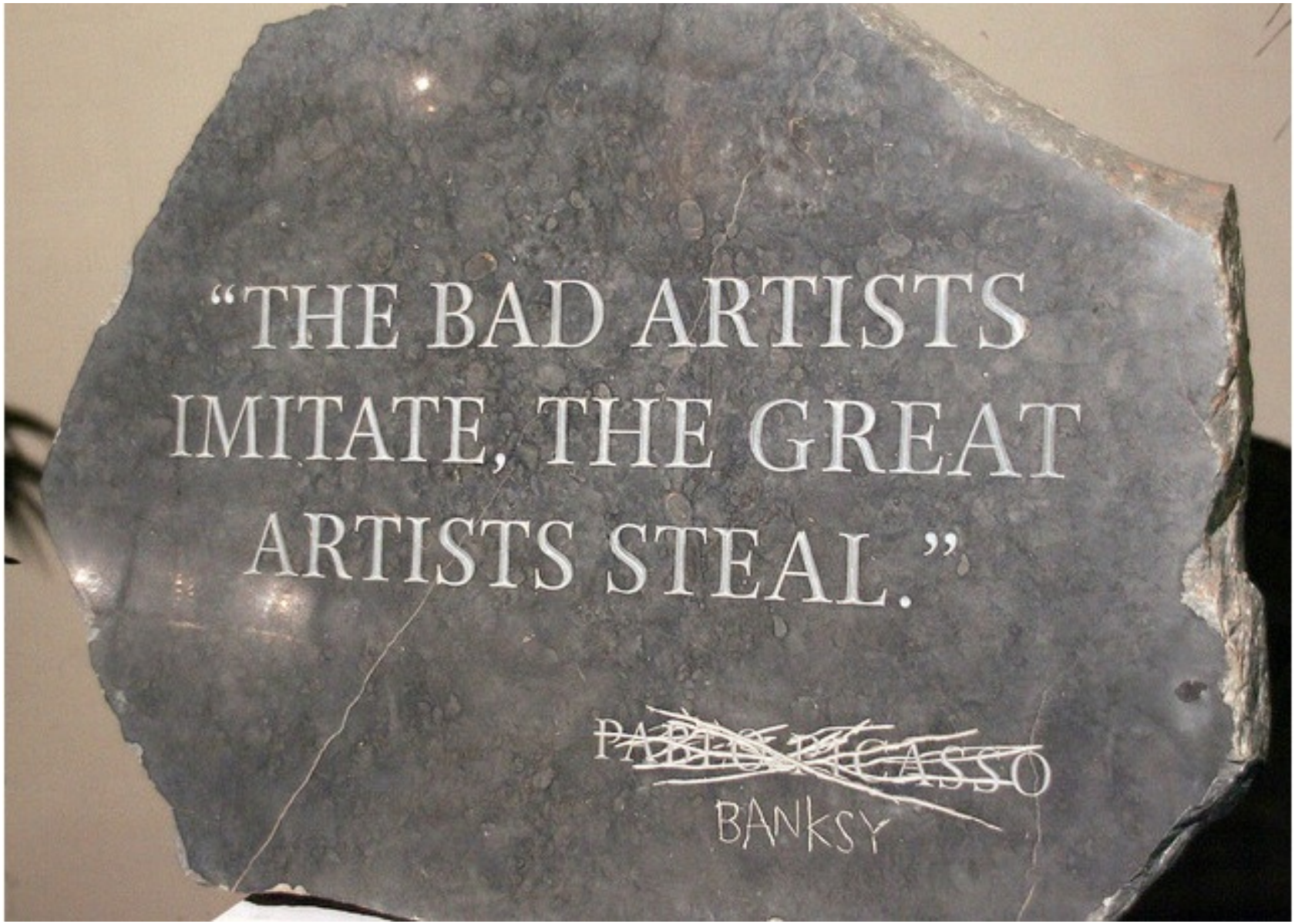


Figure: your turn