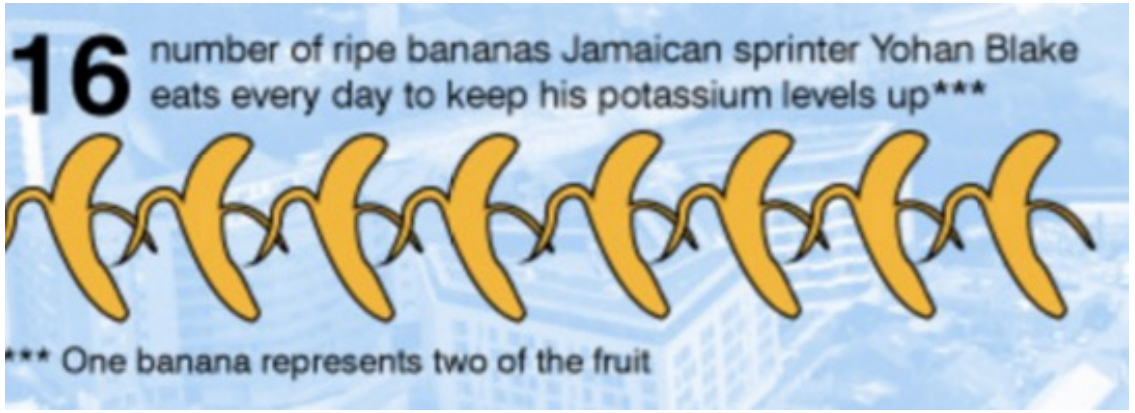
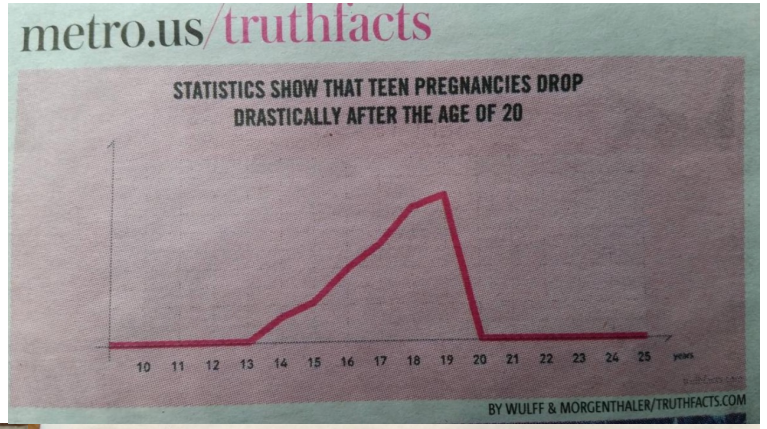
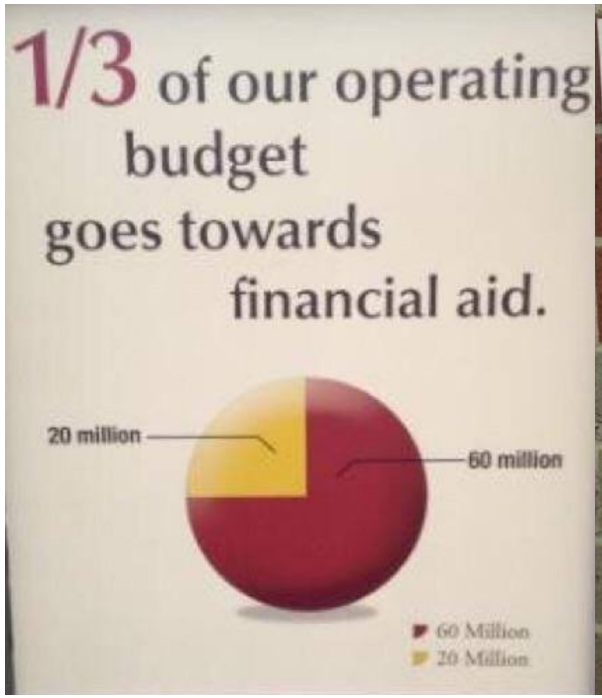
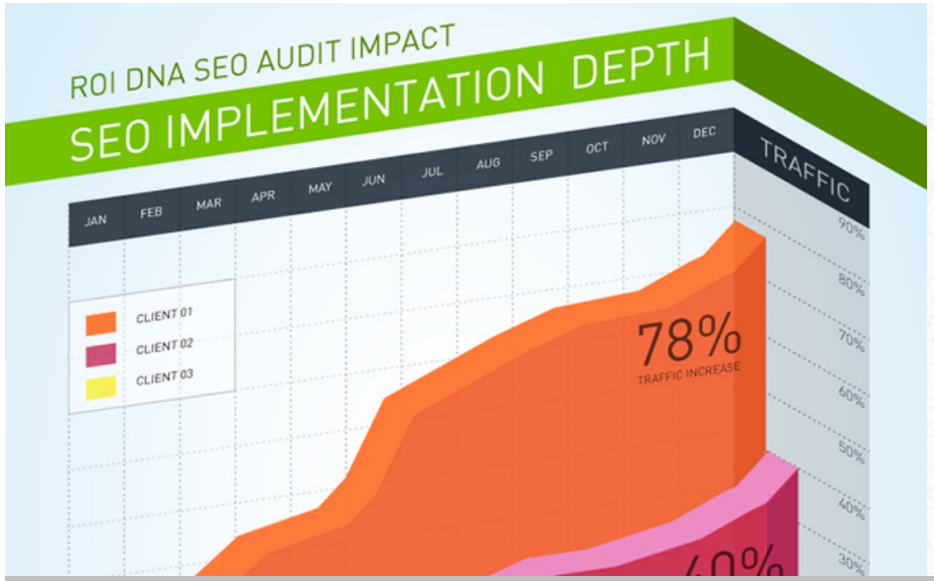
201ab Quantitative methods Visualization

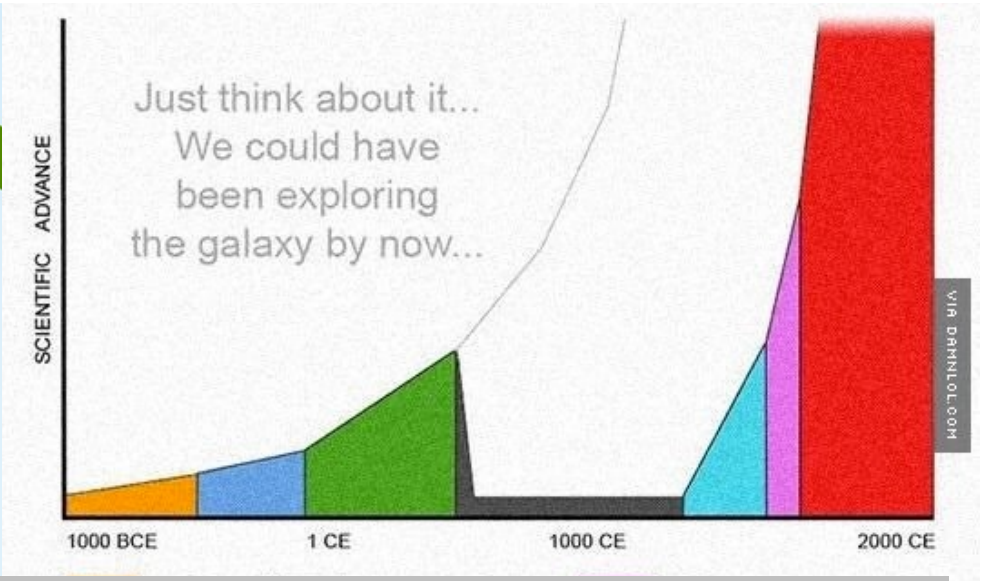




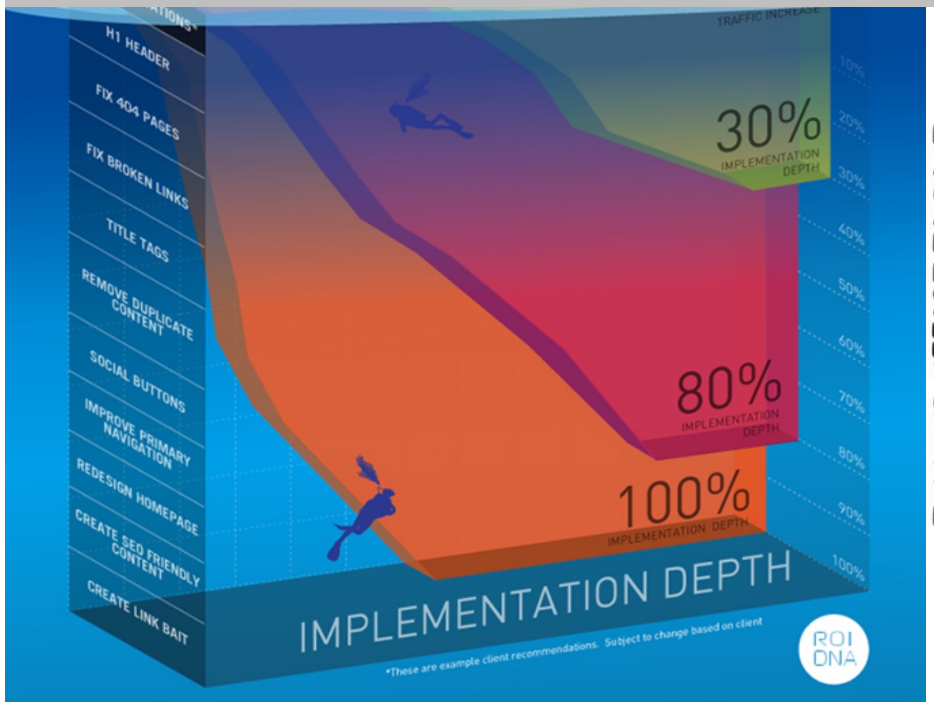


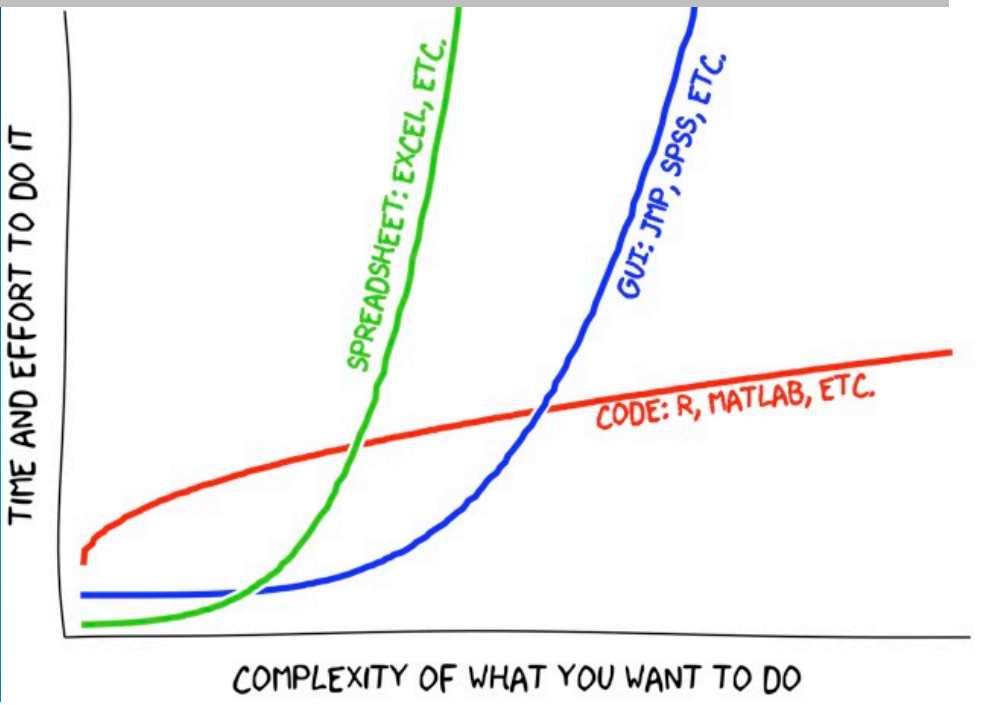
- Visualization failure modes
- Cool vs informative visualizations
- Ways graphs can mislead
- Making a graph pretty
- ggplot: grammar of graphics

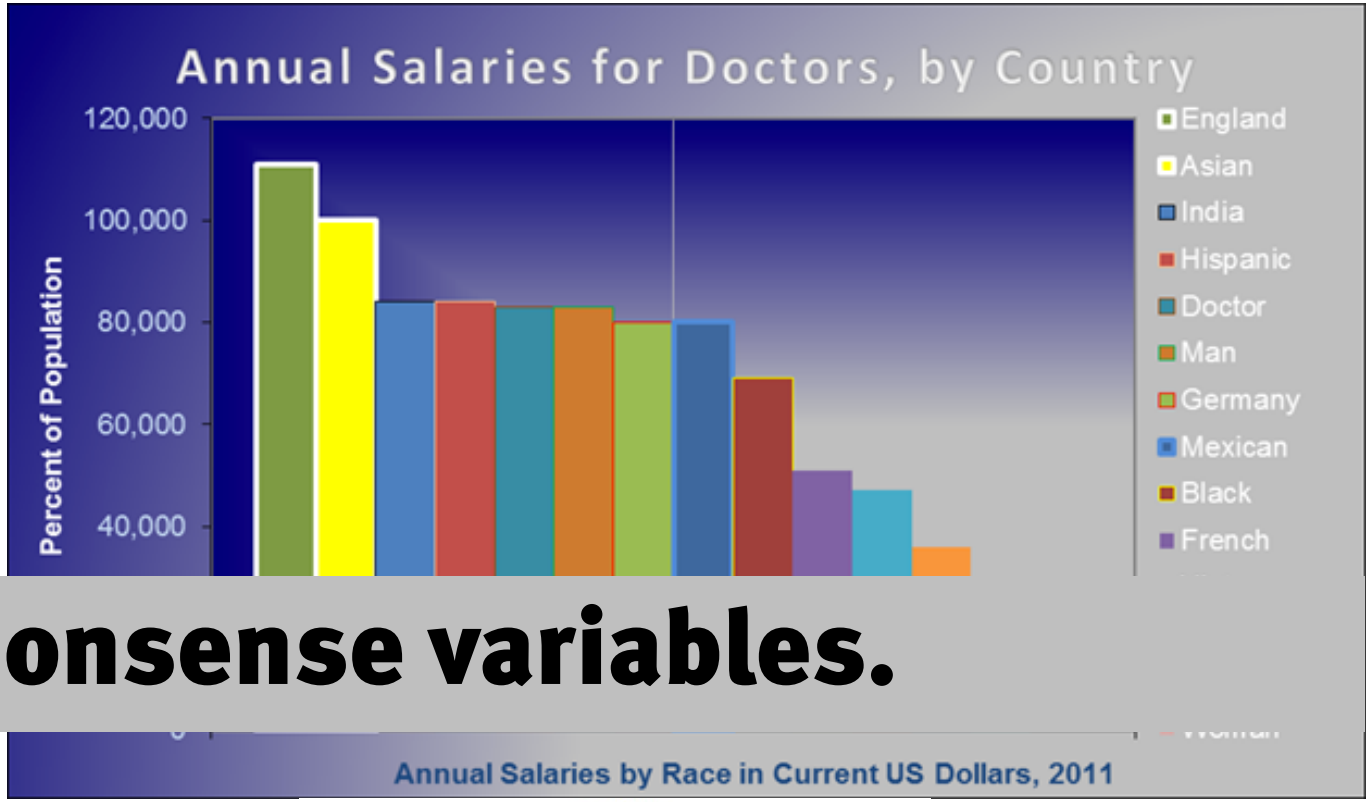




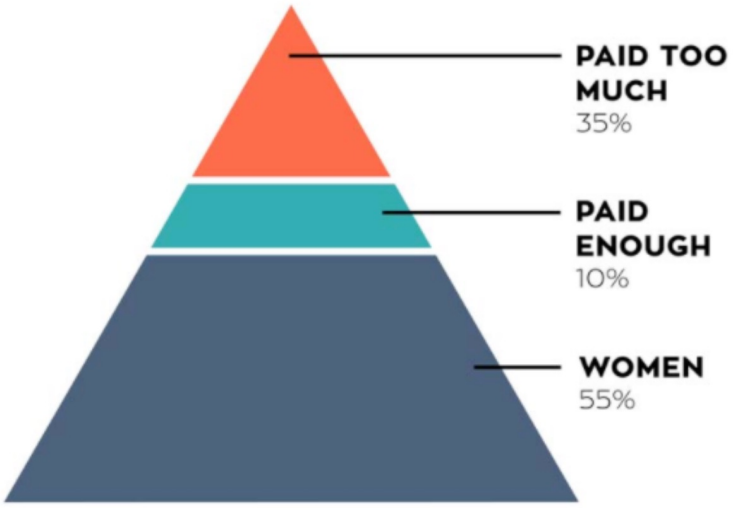
Entirely made up.

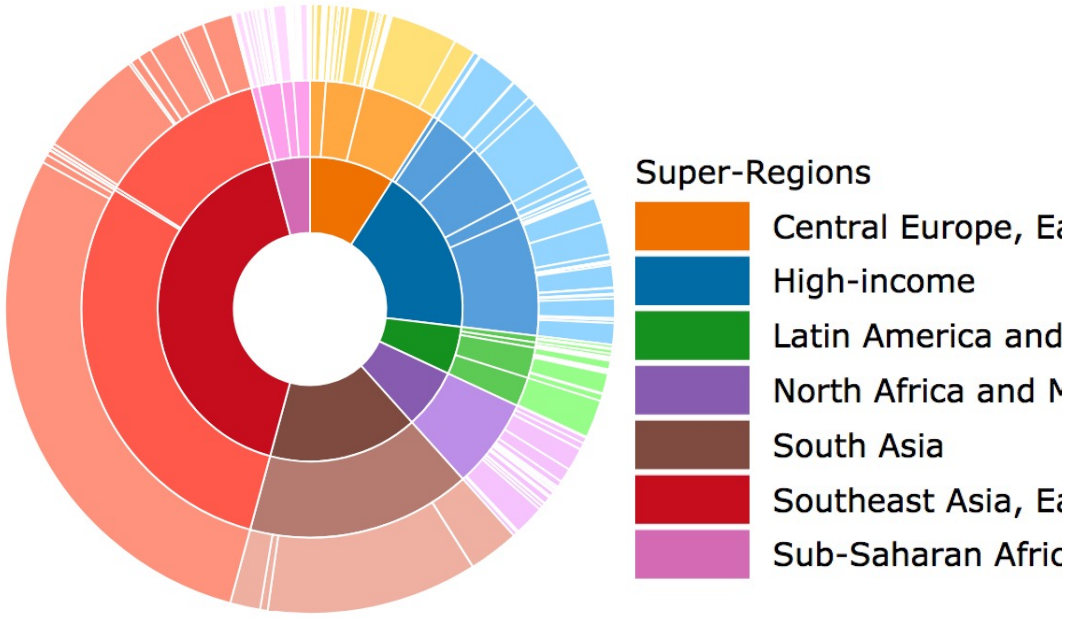


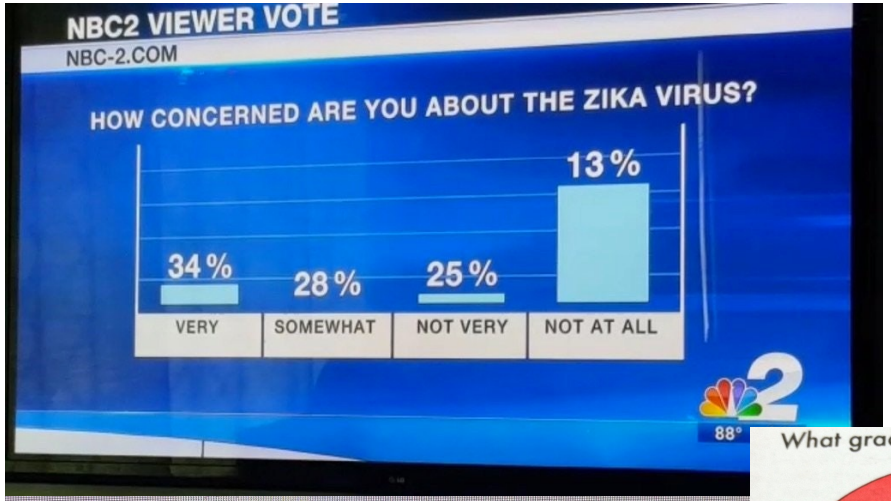


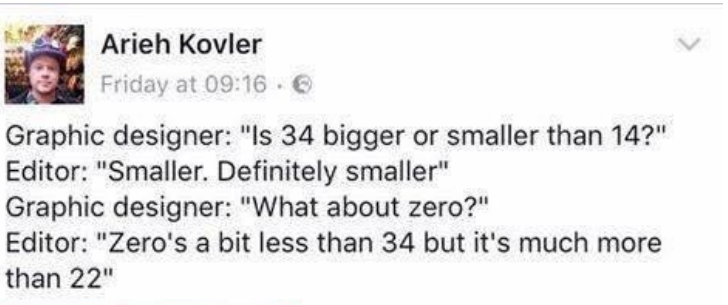


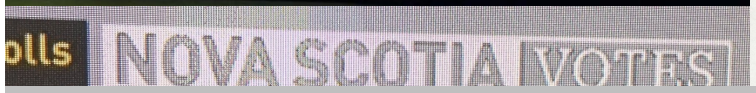
Nonsense variables.

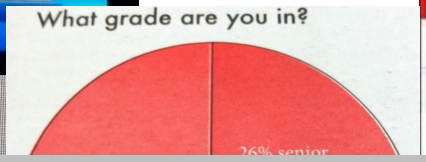




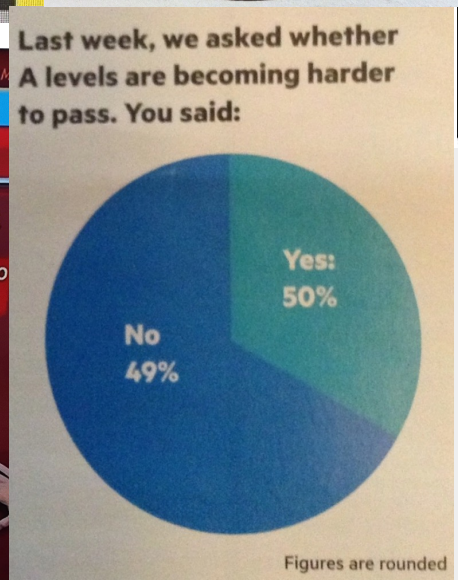


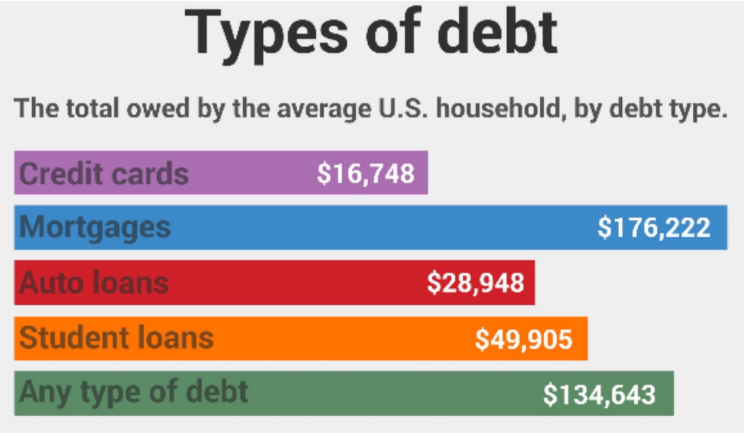


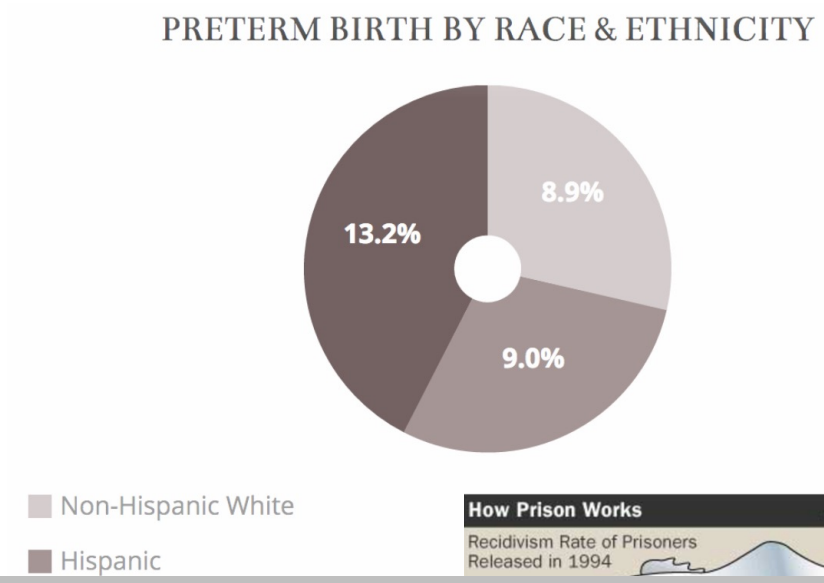


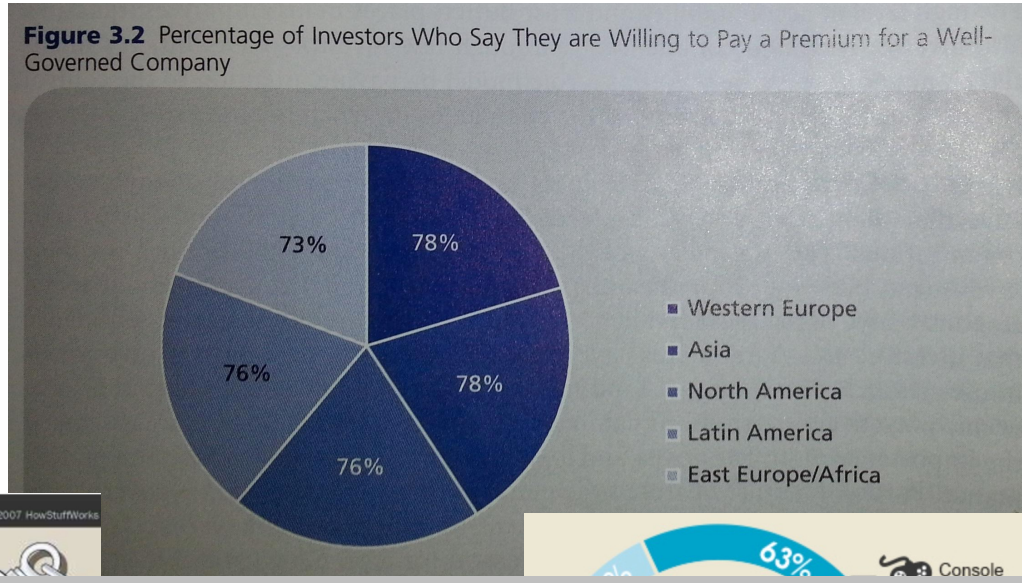


Graph independent of data.

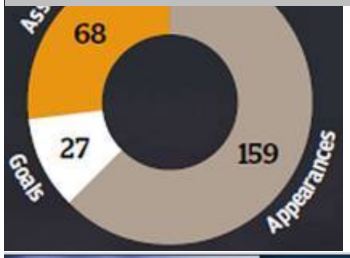


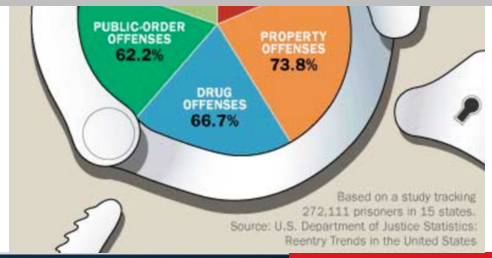


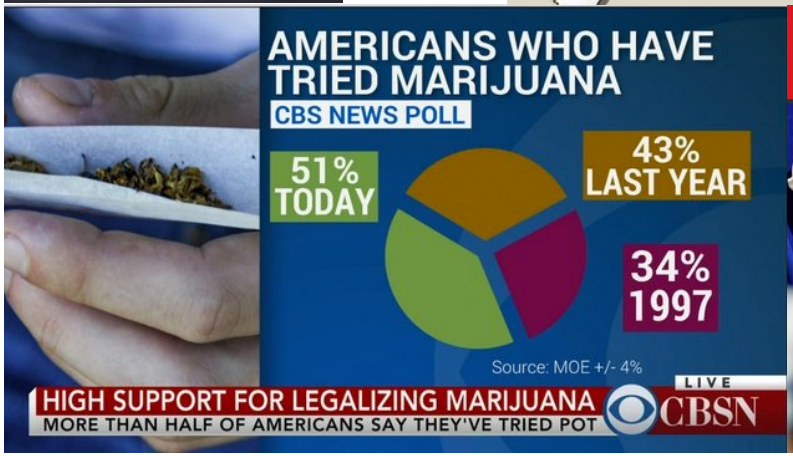


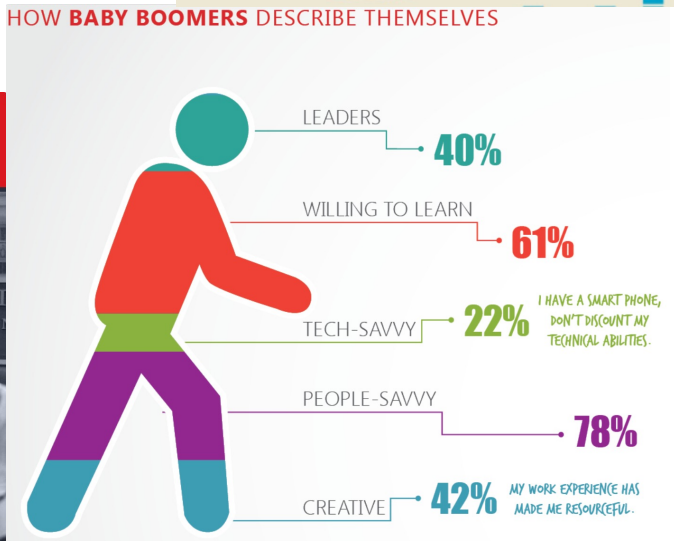


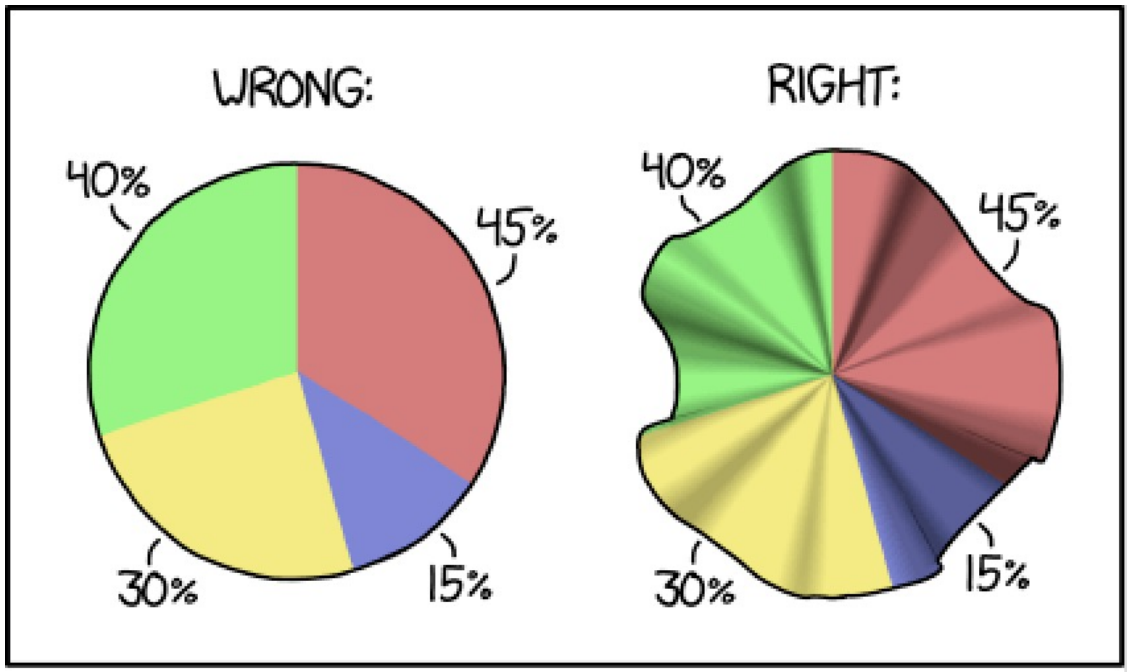
Multiple variables graphed as one.





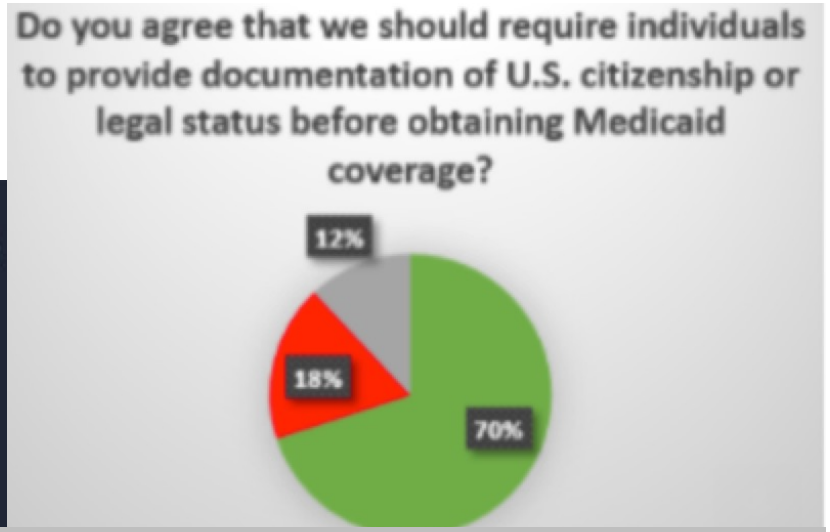




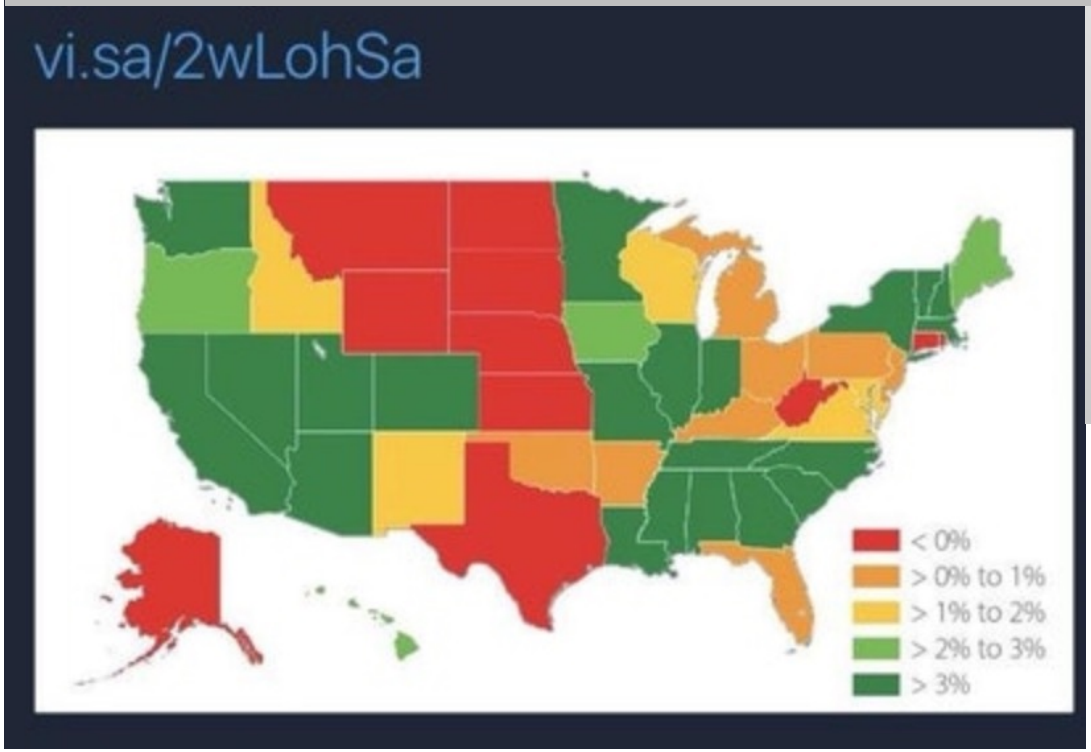


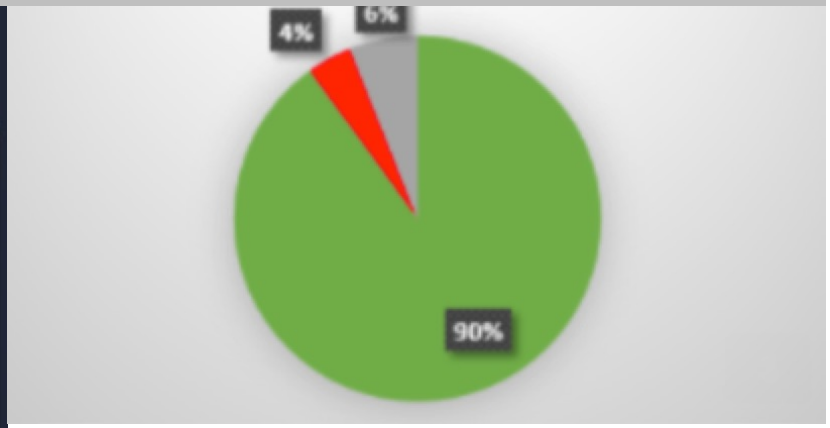
HOW TO MAKE A PIE CHART IF YOUR PERCENTAGES DON'T ADD UP TO 100

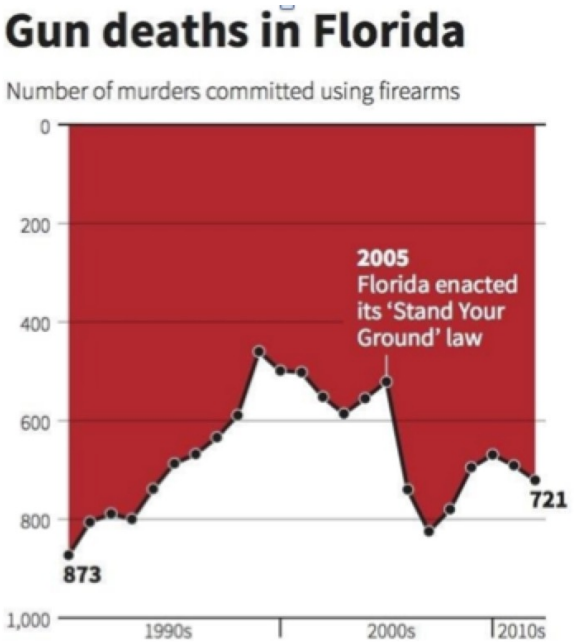
Credit: xkcd

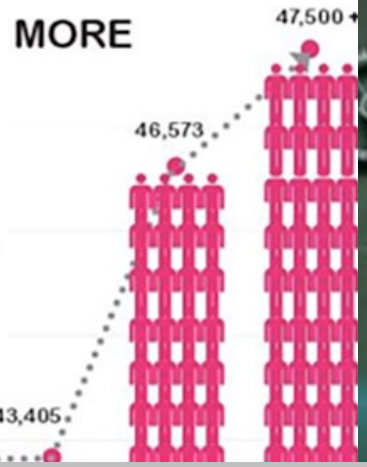


Not labeled (or mislabeled).

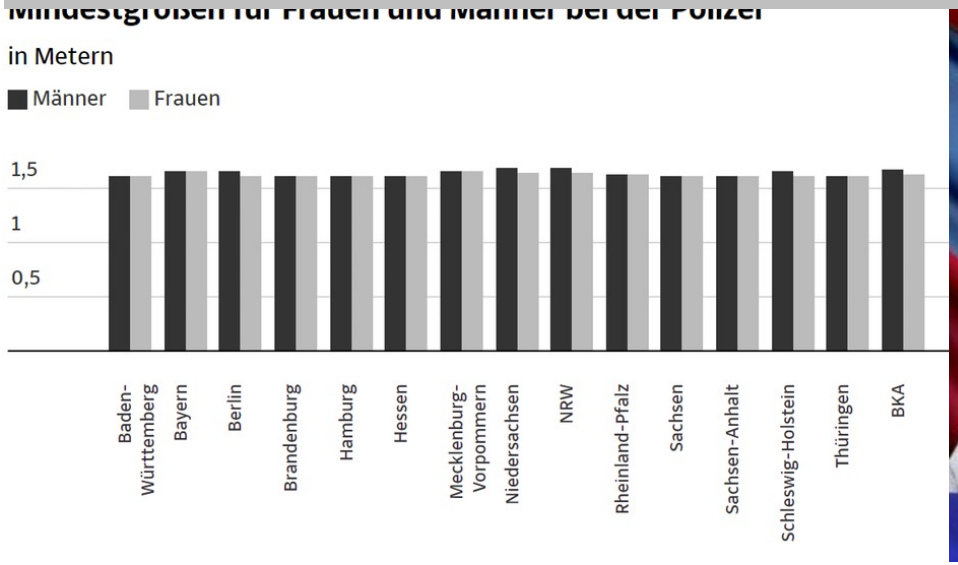


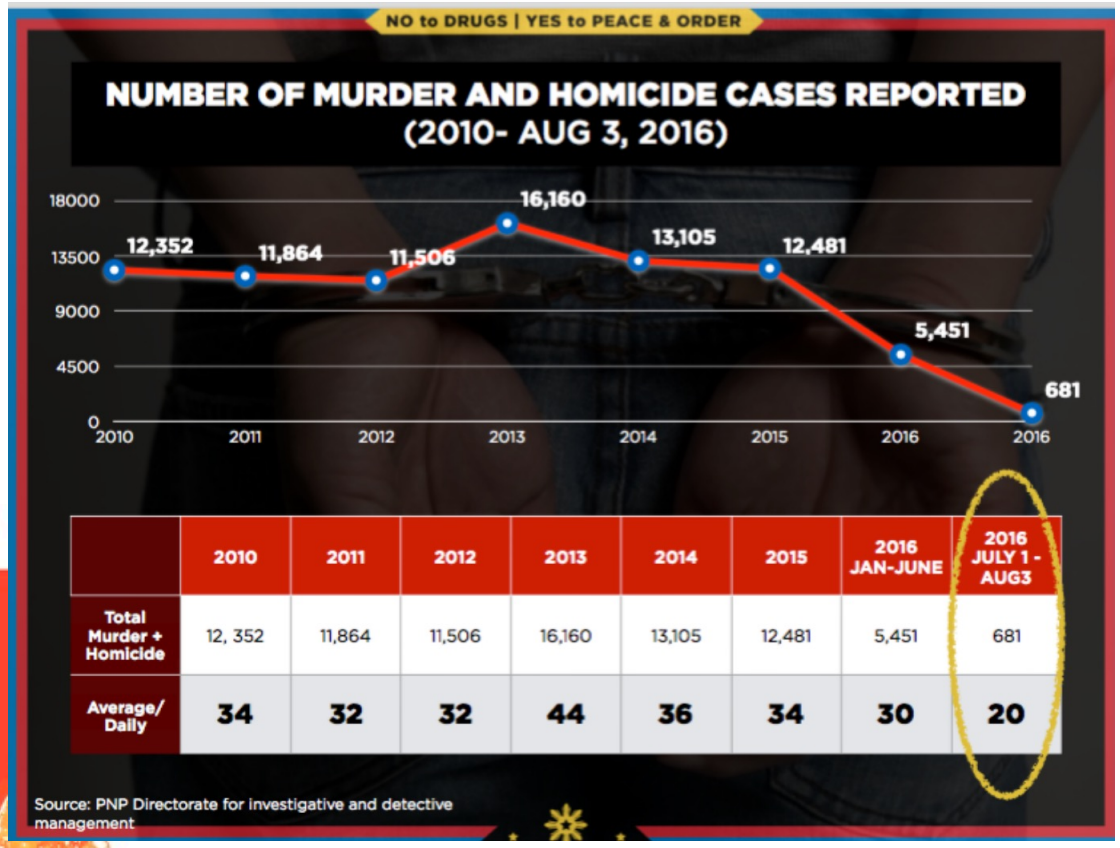




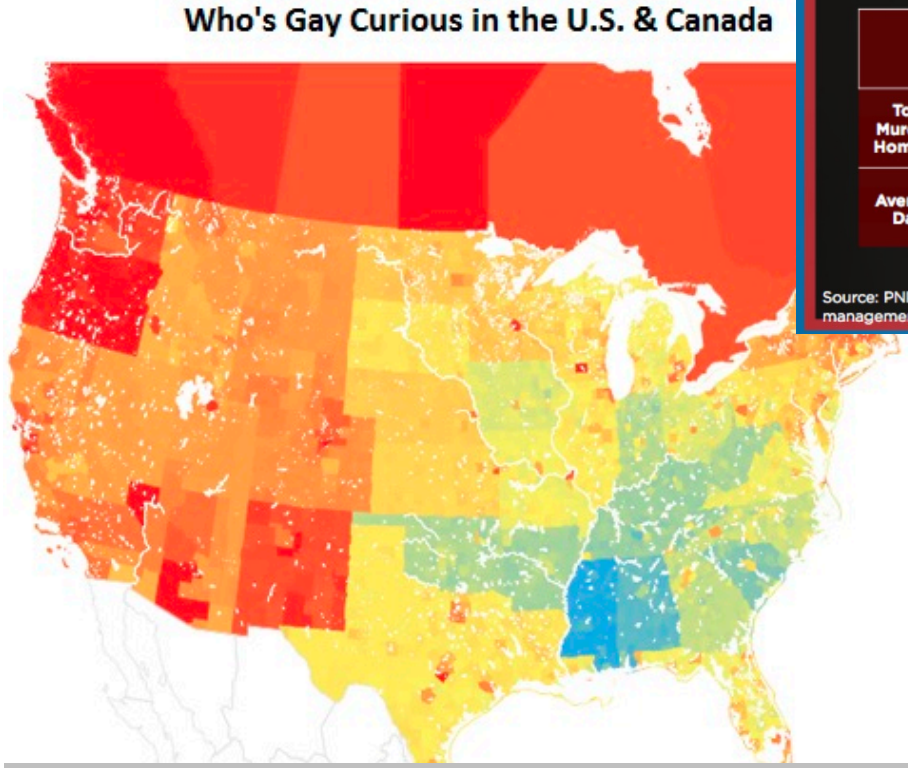


Misleading or useless axis scales.

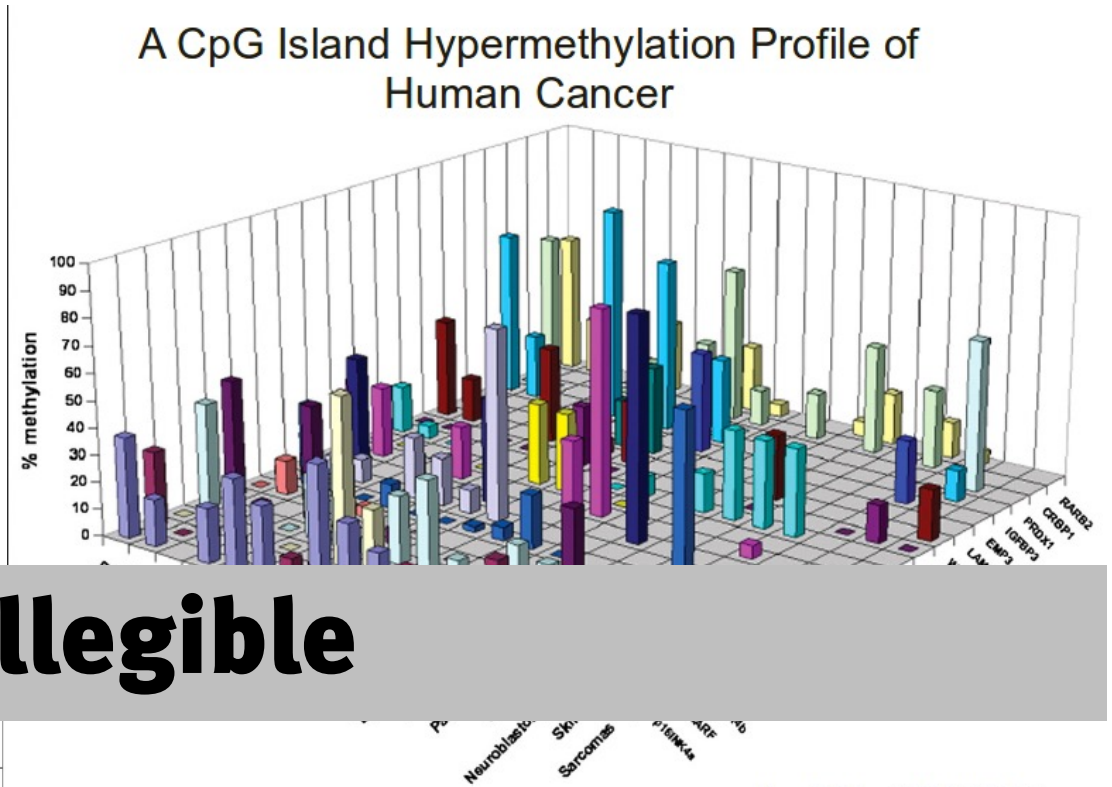




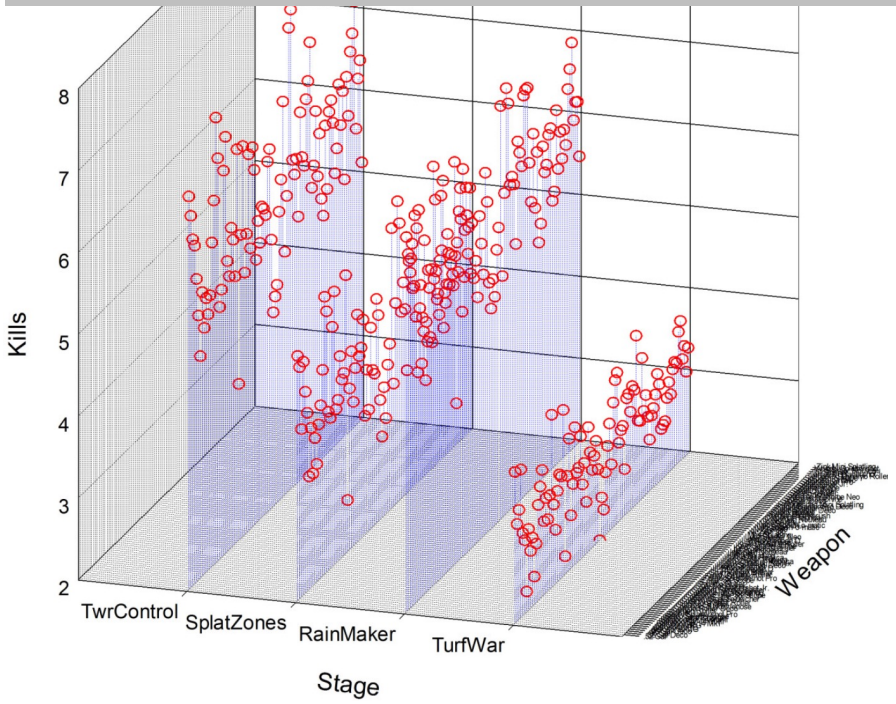
	2010	2011	2012	2013	2014	2015	2016 JAN-JUNE	2016 JULY 1 - AUG3
Total Murder + Homicide	12, 352	11,864	11,506	16,160	13,105	12,481	5,451	681
Average/ Daily	34	32	32	44	36	34	30	20

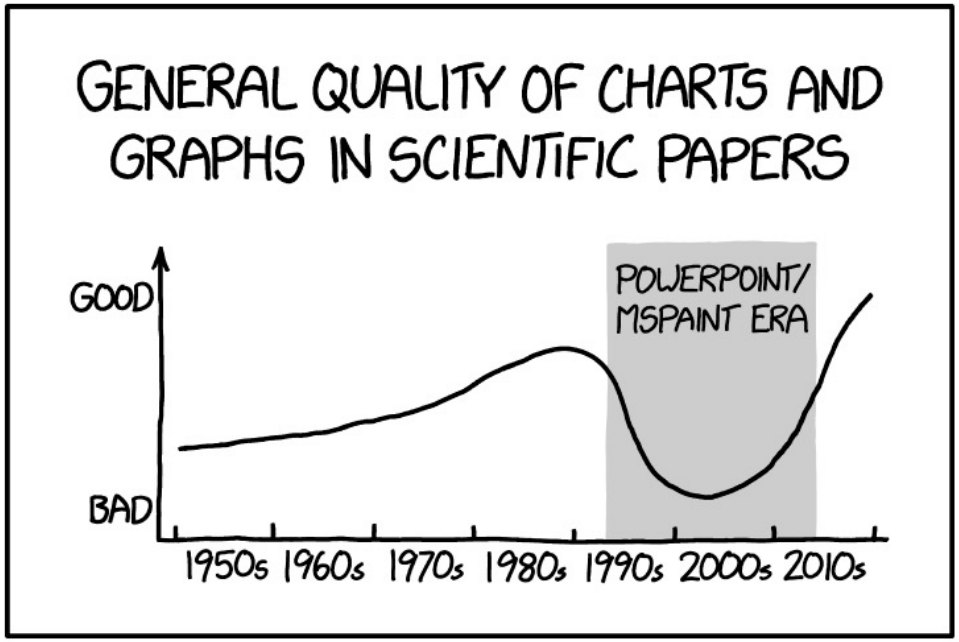


Misleading binning.


A CpG Island Hypermethylation Profile of Human Cancer
% methylation
100
90
80
70
60
50
40
30
20
10
0
RARB2
CRBP1
PDX1
IGFBP3
EMP3
Neuroblastoma
Sarcomas
Hum. Mol. Genet. (2007) 16:R50-59

Illegible


Kills
8
7
6
5
4
3
2
TwrControl
SplatZones
RainMaker
TurfWar
Stage
Weapon


GENERAL QUALITY OF CHARTS AND GRAPHS IN SCIENTIFIC PAPERS
GOOD
BAD
POWERPOINT/ MSPAINT ERA
1950s 1960s 1970s 1980s 1990s 2000s 2010s

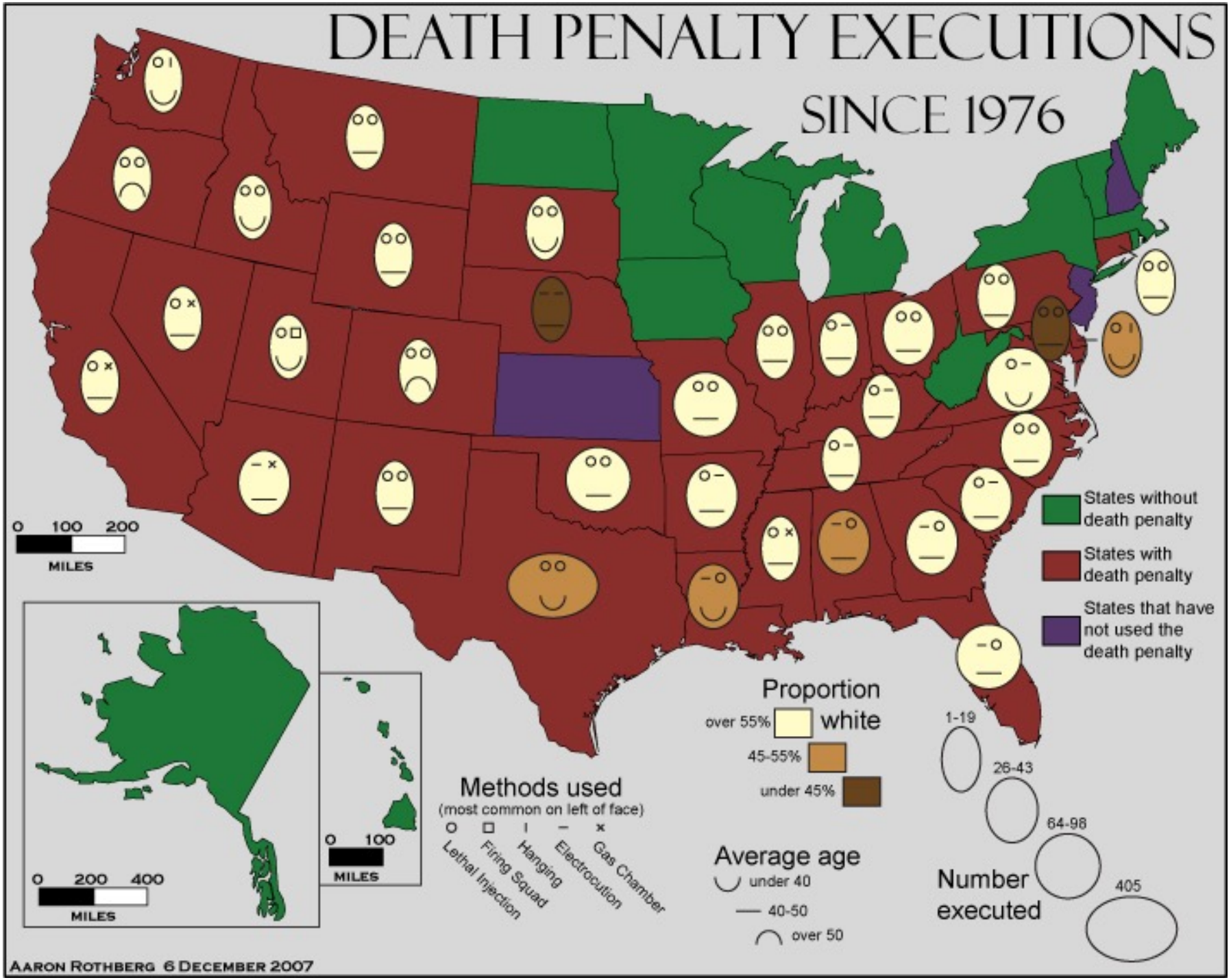
Credit: xkcd

Visualization failure modes

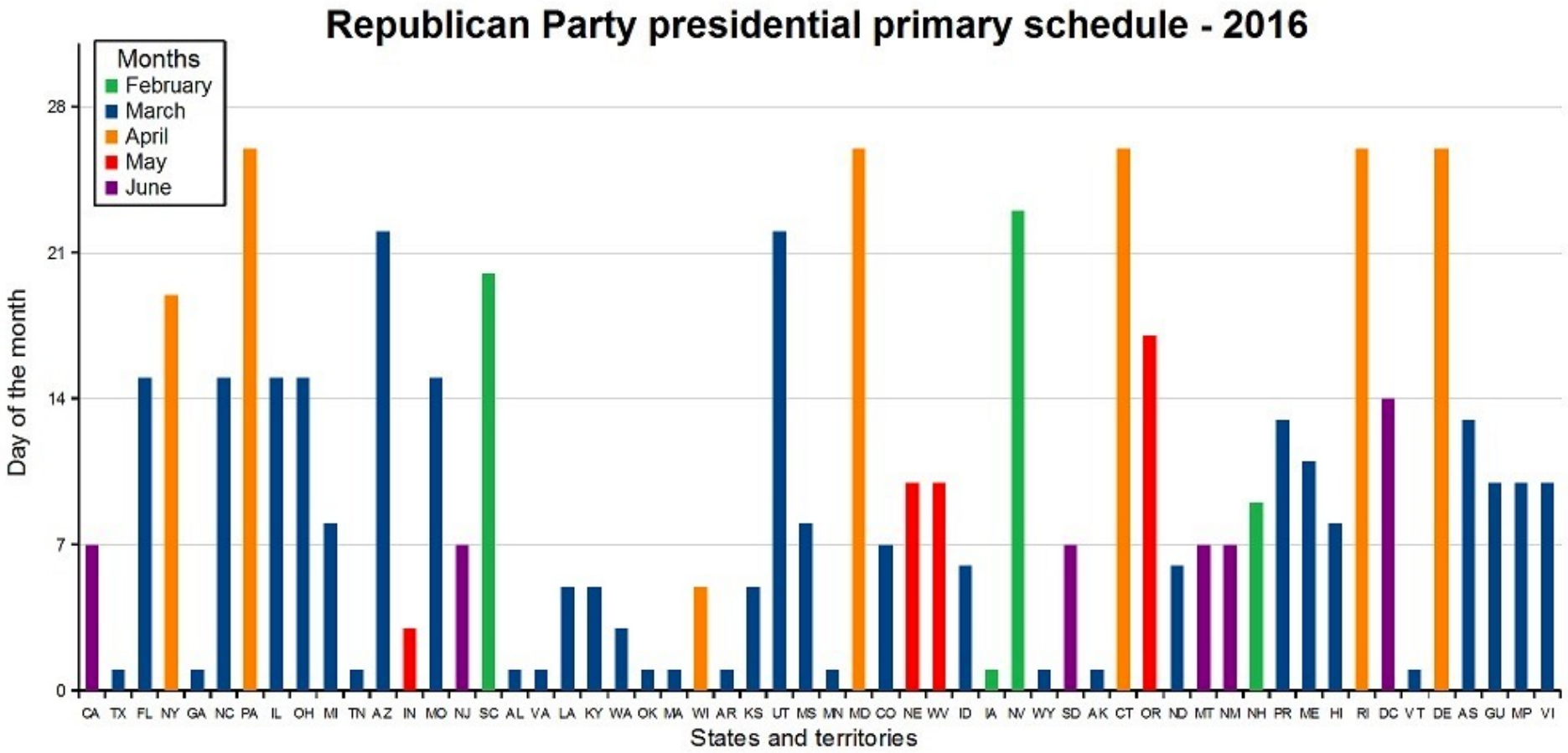
- Completely made up.
- Nonsense variables/relationships.
- Graph independent of data.
- Multiple variables treated as one.
- Not labeled, or mislabeled.
- Misleading / unusable scales.
- Misleading binning.
- Illegible.
- Crazy mapping from variables -› visual properties.

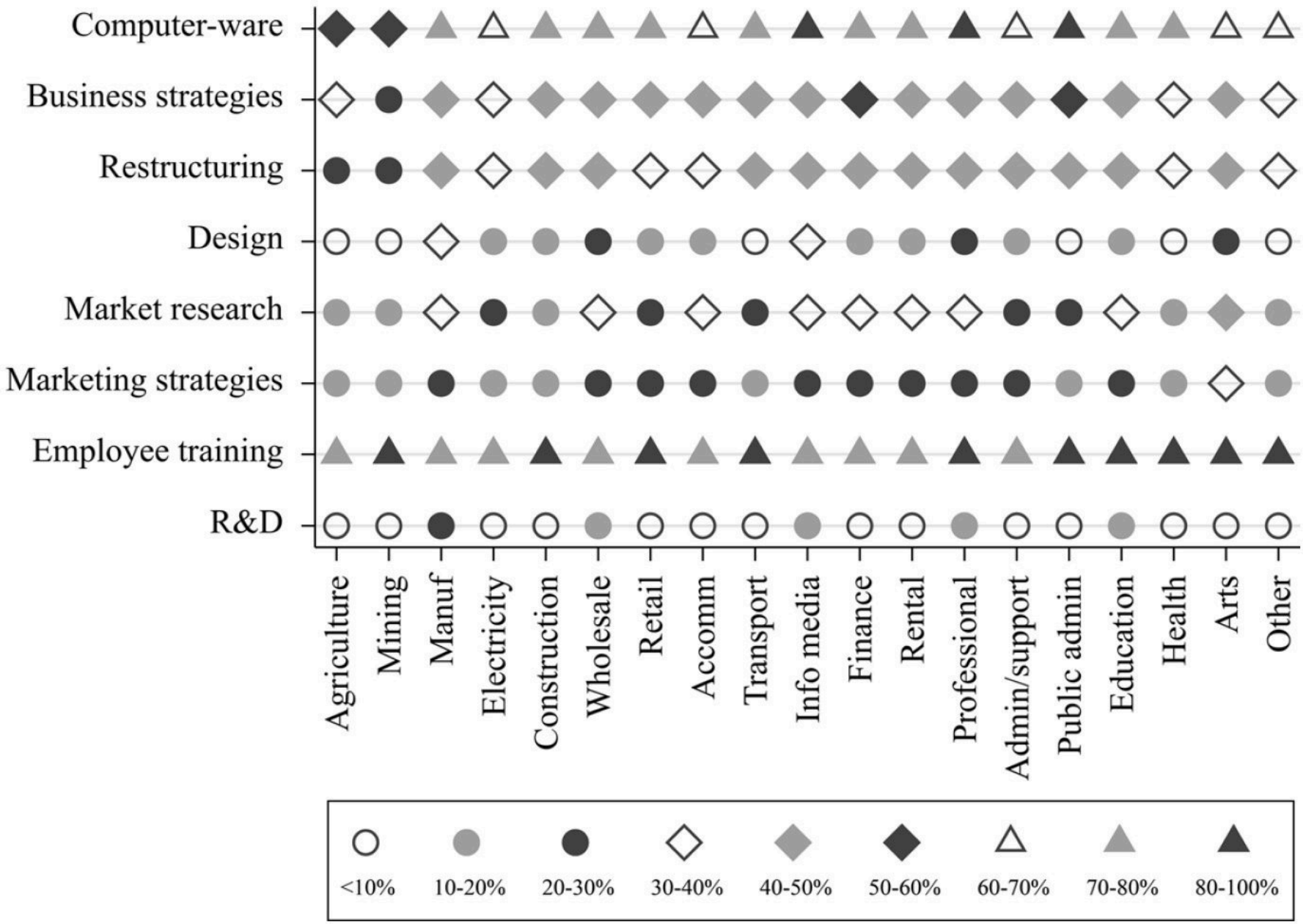
Peak time for sports and leisure

@hnrklndbrg | Source: American Time Use Survey

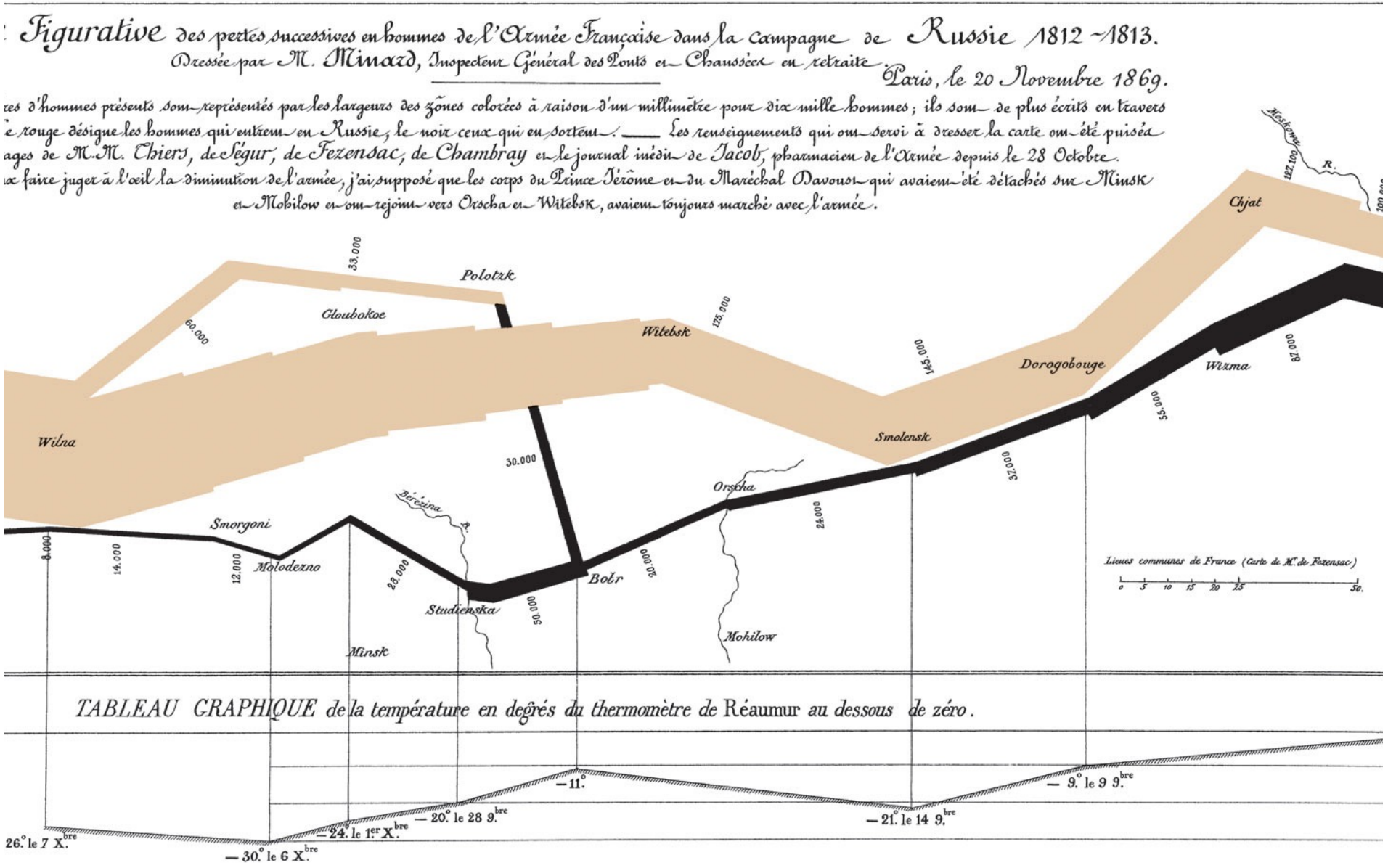

DEATH PENALTY EXECUTIONS
SINCE 1976
0 100 200
MILES
0 200 400
MILES
0 100
MILES
States without death penalty
States with death penalty
States that have not used the death penalty
Proportion white
over 55%
45-55%
under 45%
Methods used
(most common on left of face)
Lethal Injection
Firing Squad
Hanging
Electrocution
Gas Chamber
Average age
under 40
40-50
over 50
Number executed
1-19
26-43
64-98
405
AARON ROTHBERG 6 DECEMBER 2007

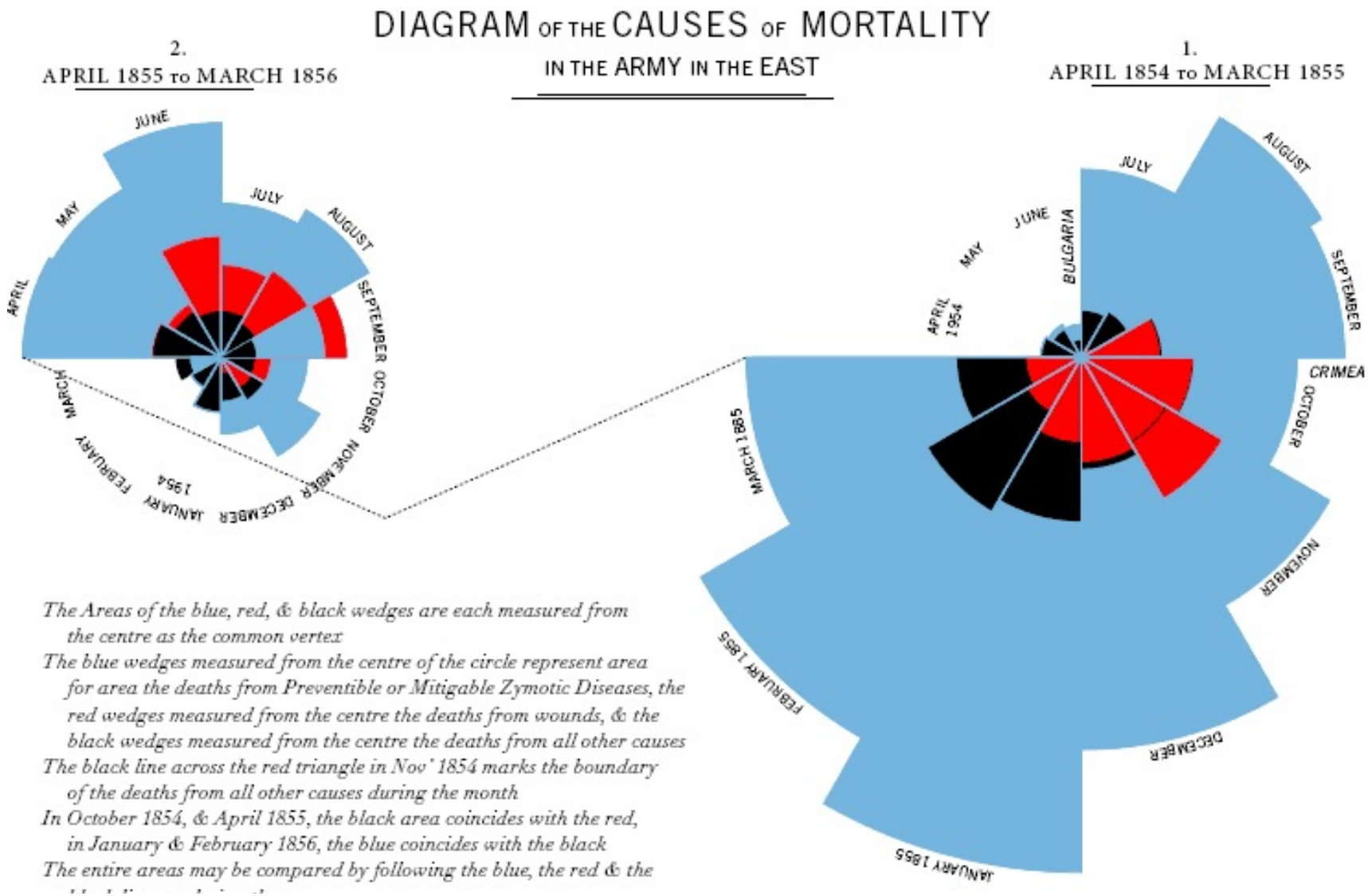
Ed


Republican Party presidential primary schedule - 2016
Months
February
March
April
May
June
Day of the month
0
7
14
21
28
CA TX FL NY GA NC PA IL OH MI TN AZ IN MO NJ SC AL VA LA KY WA OK MA WI AR KS UT MS MN MD CO NE WV ID IA NV WY SD AK CT OR ND MT NM NH PR ME HI RI DC VT DE AS GU MP VI
States and territories


Computer-ware
Business strategies
Restructuring
Design
Market research
Marketing strategies
Employee training
R&D
Agriculture
Mining
Manuf
Electricity
Construction
Wholesale
Retail
Accomm
Transport
Info media
Finance
Rental
Professional
Admin/support
Public admin
Education
Health
Arts
Other
<10%
10-20%
20-30%
30-40%
40-50%
50-60%
60-70%
70-80%
80-100%

- Visualization failure modes
- Cool vs scientific visualizations
- Making a graph pretty
- ggplot: grammar of graphics
- How to graph common data types.

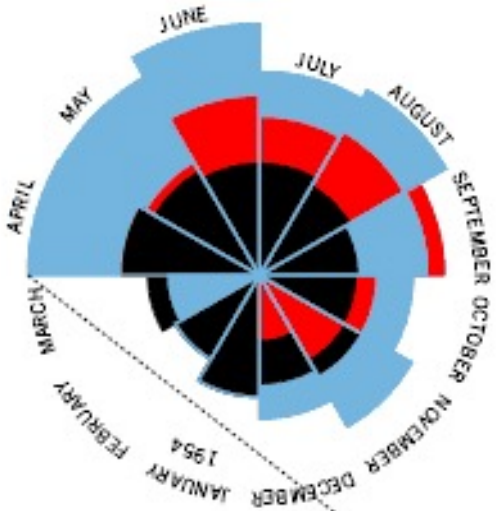

Figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812 - 1813.
Dressée par M. Minard, Inspecteur Général des Ponts et Chaussées en retraite.
Paris, le 20 Novembre 1869.
Wilna
Smorgoni
Molodexno
Minsk
Studienska
Bobr
Mohilow
Orscha
Smolensk
Dorogobouge
Wixma
Chjat
Polotzk
Gloubokoe
Witebsk
Berézina R.
Lieues communes de France (Carte de M.r de Fezensac)
TABLEAU GRAPHIQUE de la température en degrés du thermomètre de Réaumur au dessous de zéro.
— 30.° le 6 X.bre
26.° le 7 X.bre
— 24.° le 1er X.bre
— 20.° le 28 9.bre
— 11.°
— 21.° le 14 9.bre
— 9.° le 9 9.bre



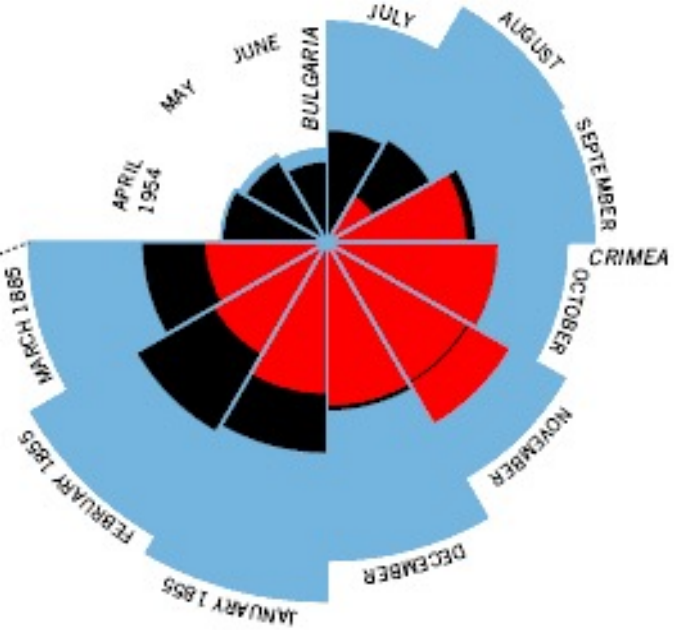
From dynamicdiagrams.com

DIAGRAM OF THE CAUSES OF MORTALITY
IN THE ARMY IN THE EAST

2.
APRIL 1855 TO MARCH 1856

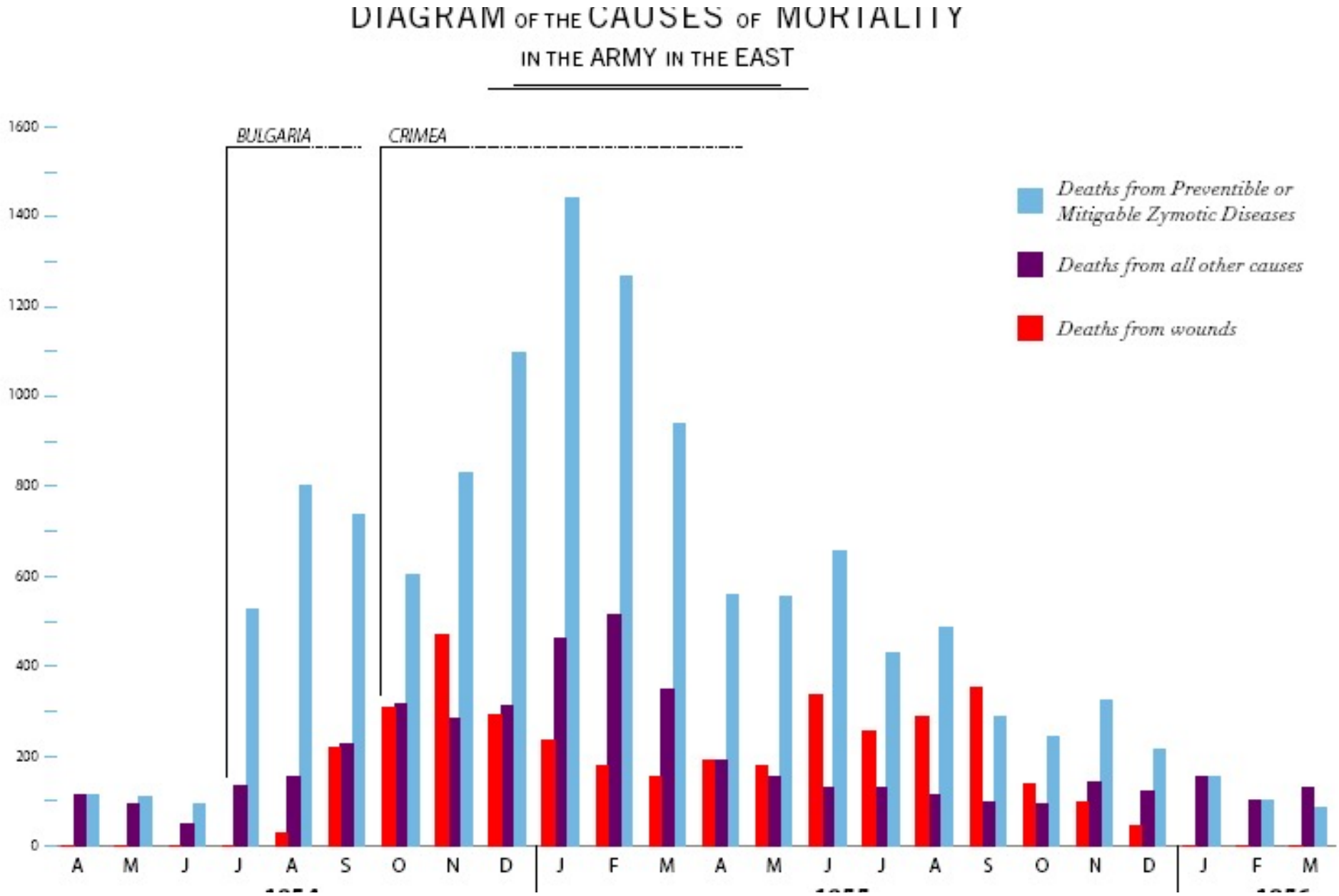


1.
APRIL 1854 TO MARCH 1855

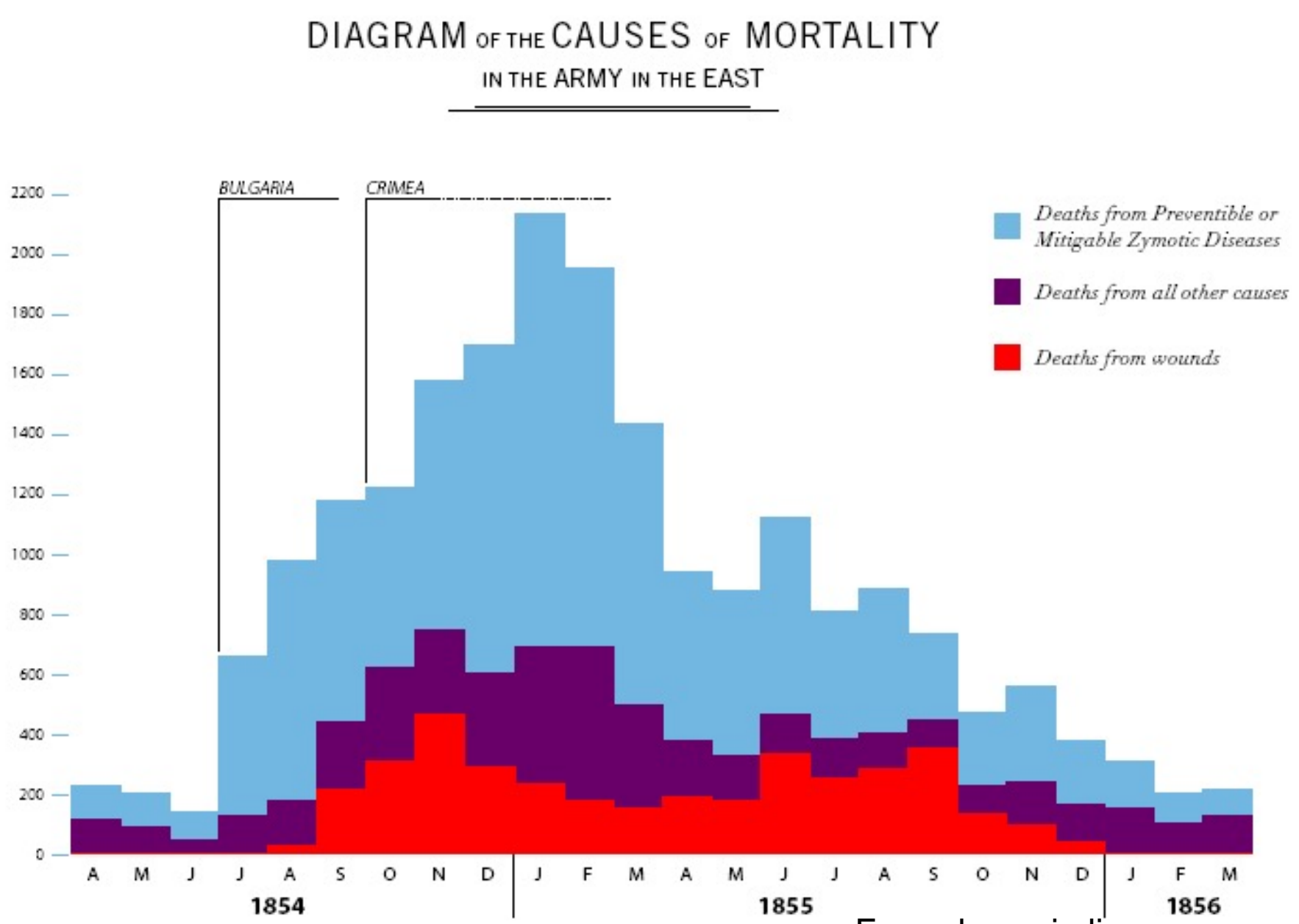


The Areas of the blue, red, & black wedges are each measured from the centre as the common vertex
The blue wedges measured from the centre of the circle represent area for area the deaths from Preventible or Mitigable Zymotic Diseases, the red wedges measured from the centre the deaths from wounds, & the black wedges measured from the centre the deaths from all other causes
The black line across the red triangle in Nov' 1854 marks the boundary of the deaths from all other causes during the month
In October 1854, & April 1855, the black area coincides with the red, in January & February 1856, the blue coincides with the black
The entire areas may be compared by following the blue, the red & the

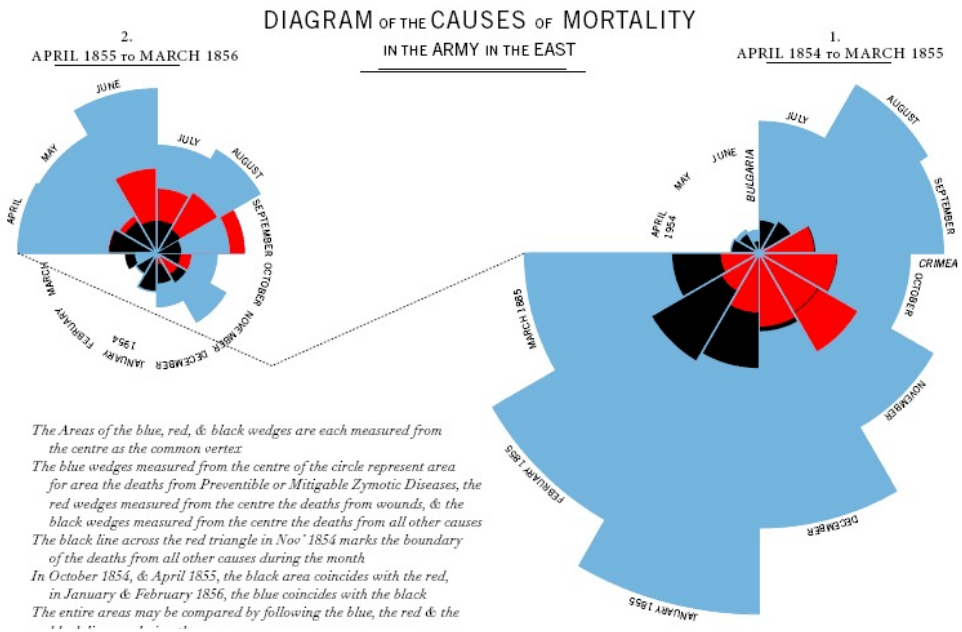
From dynamicdiagrams.com



From dynamicdiagrams.com



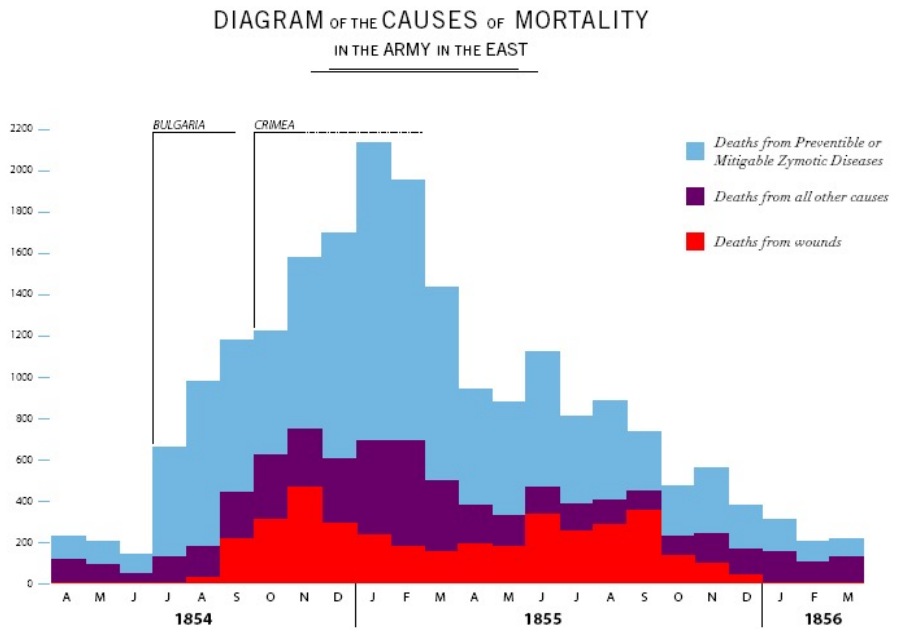
From dynamicdiagrams.com



This one.

- Looks cooler!
- Provides a visual puzzle.
- Misrepresents magnitudes.
- Does not adhere to (modern!) convention.
- Makes it difficult to make quantitative comparisons, or extract numbers

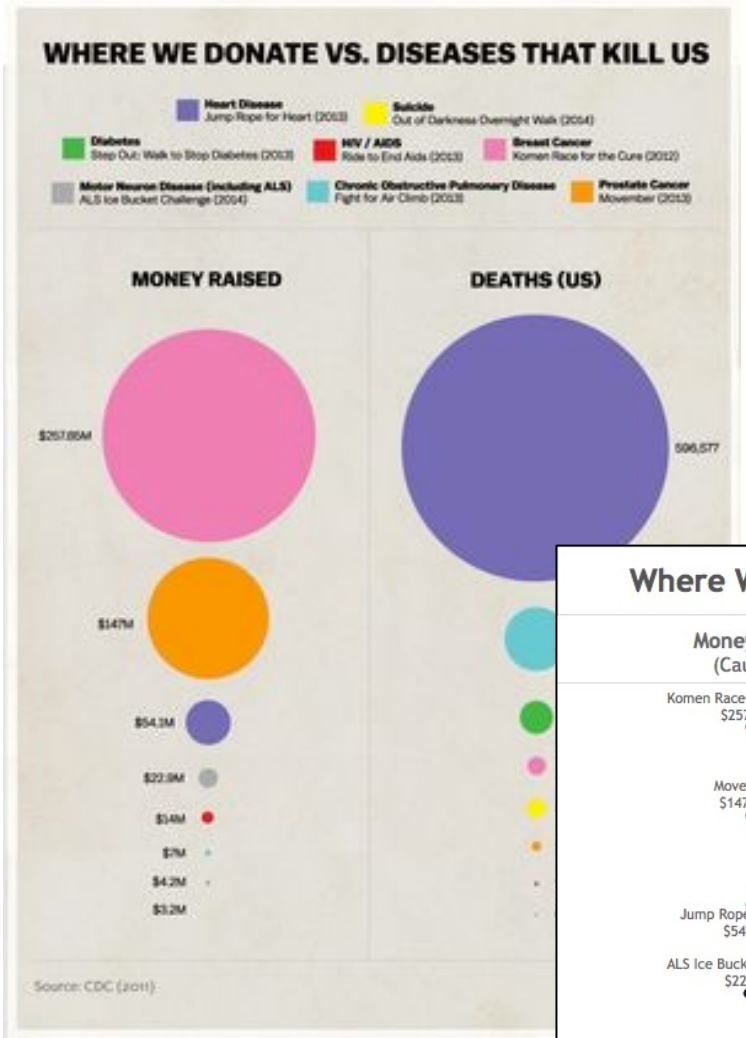
This is a **bad scientific data display**
But it is a cool visualization

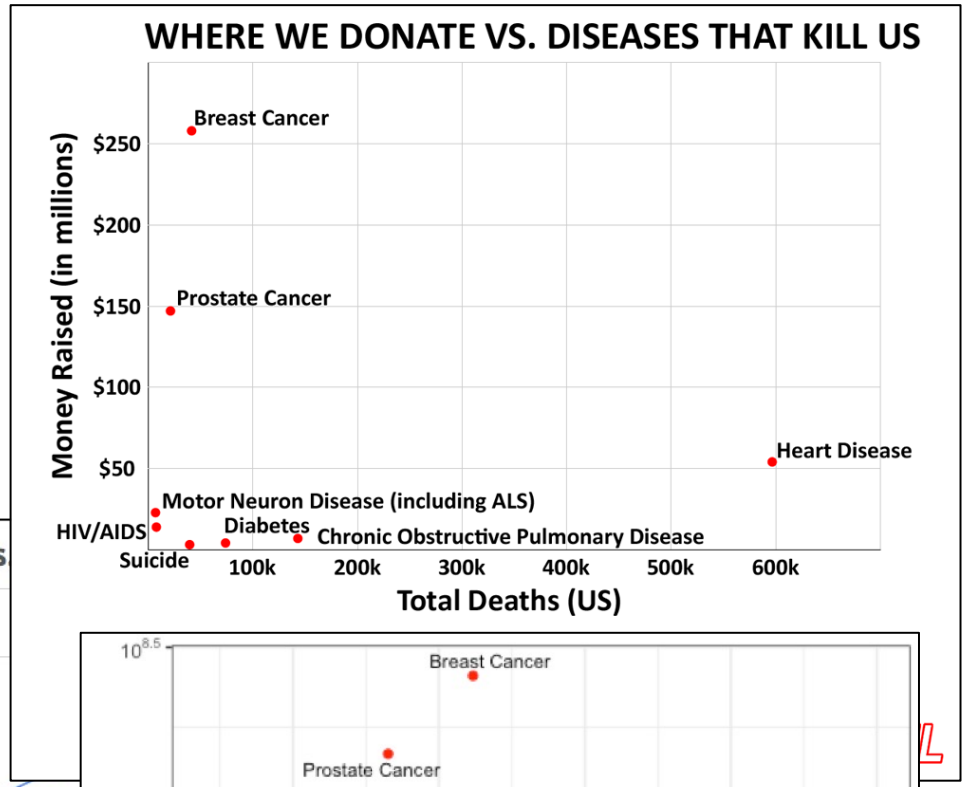


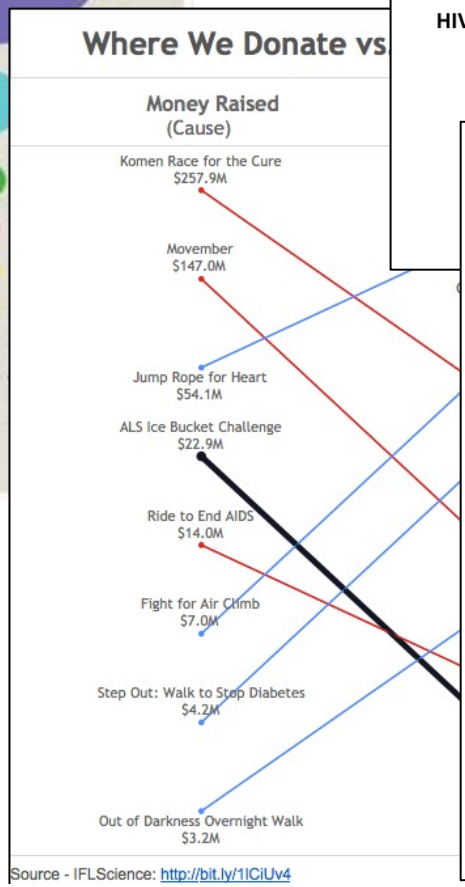
This one.

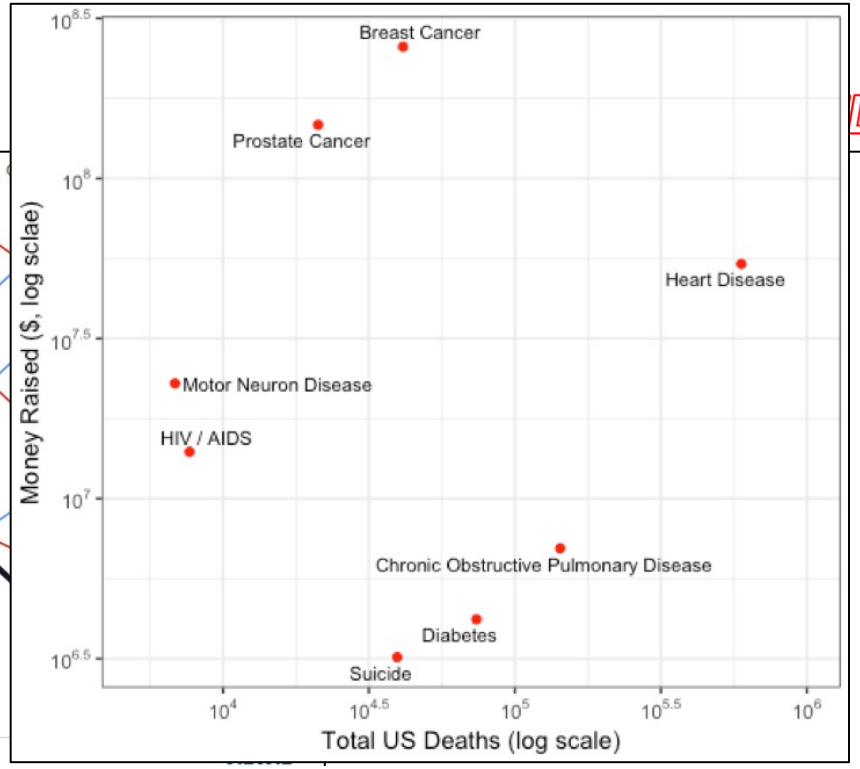
- Looks a bit more boring
- Is much easier to parse and understand
- Accurately, quantitatively represents magnitudes.
- Adheres to modern convention
- Makes it easy to make quantitative comparisons, and extract numbers

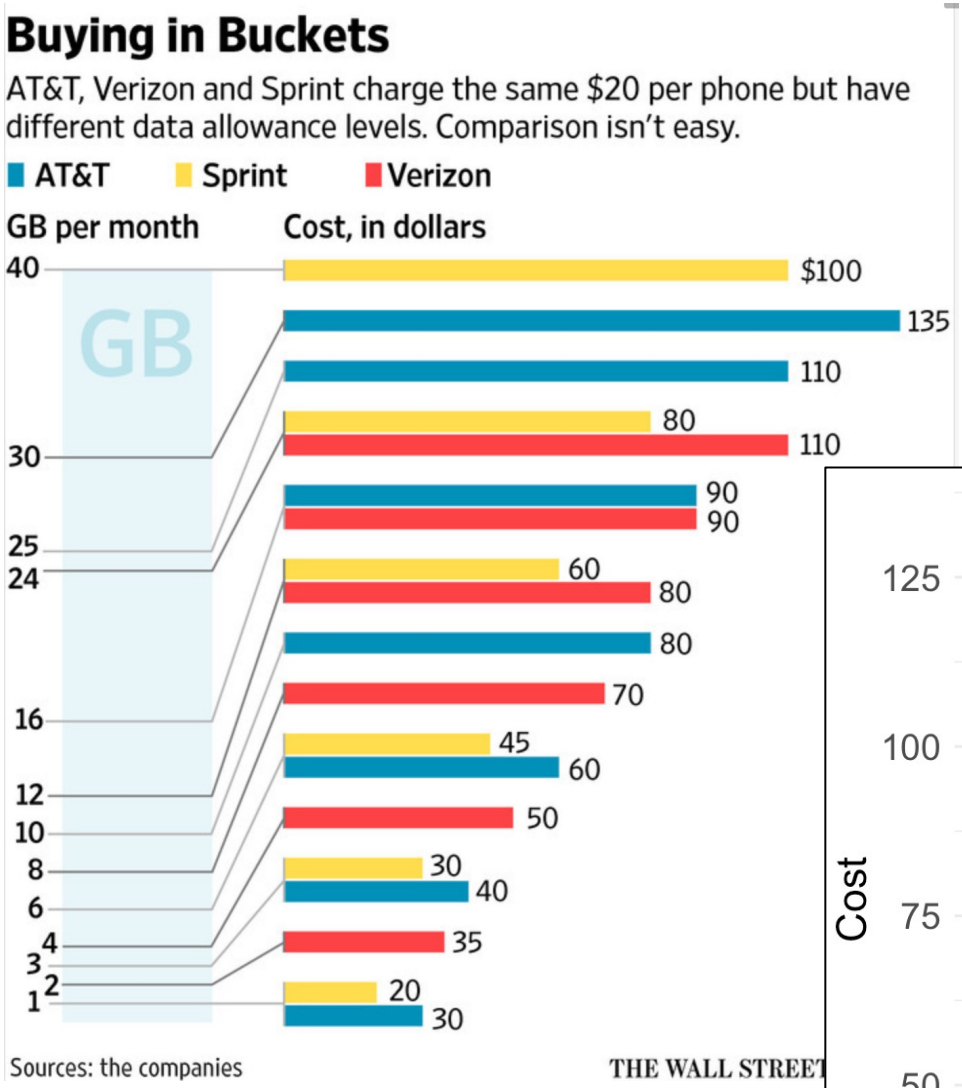
This is a **good scientific data display**
But might not be as interesting a visualization

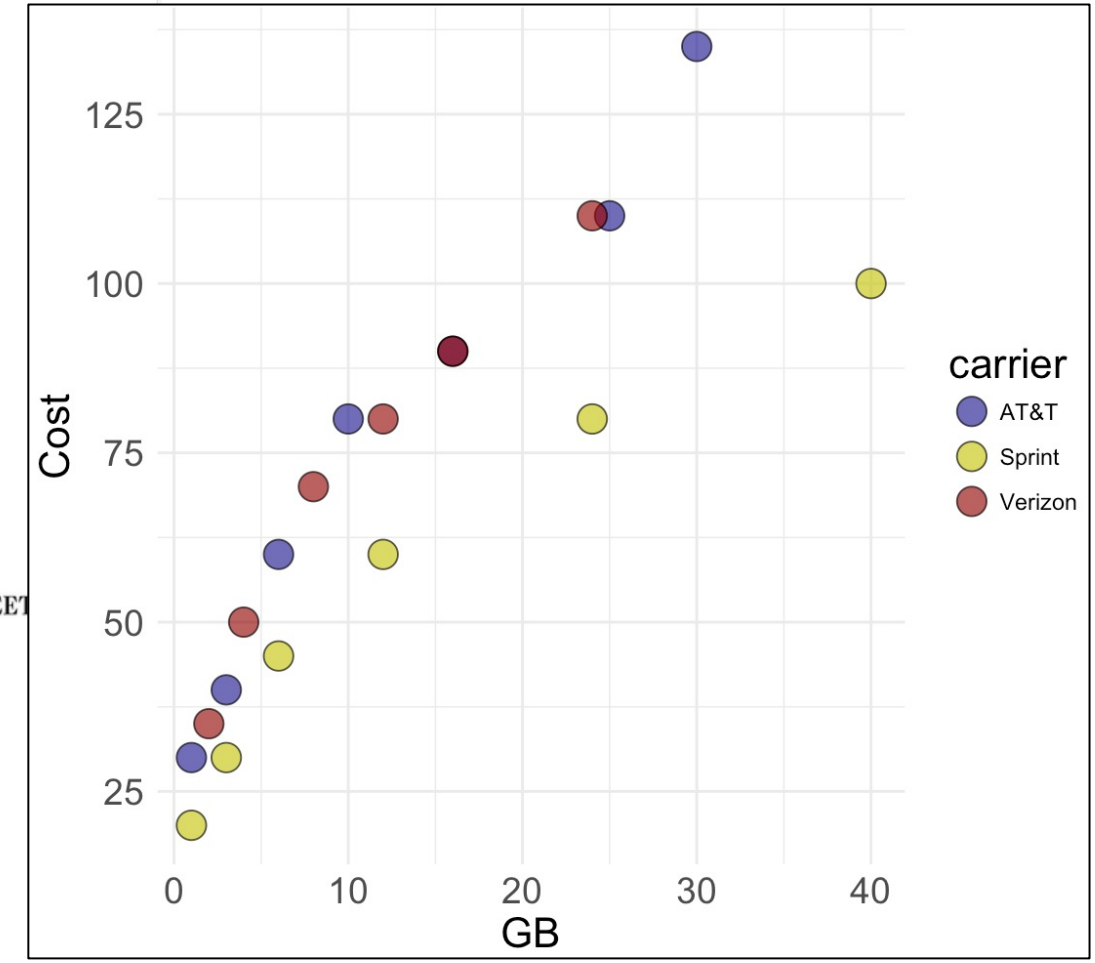

WHERE WE DONATE VS. DISEASES THAT KILL US
Heart Disease
Jump Rope for Heart (2013)
Suicide
Out of Darkness Overnight Walk (2014)
Diabetes
Step Out: Walk to Stop Diabetes (2013)
HIV / AIDS
Ride to End Aids (2013)
Breast Cancer
Komen Race for the Cure (2012)
Motor Neuron Disease (including ALS)
ALS Ice Bucket Challenge (2014)
Chronic Obstructive Pulmonary Disease
Fight for Air Climb (2013)
Prostate Cancer
Movember (2013)
MONEY RAISED
DEATHS (US)
$257.9M
$147M
$54.1M
$22.9M
$14M
$7M
$4.2M
$3.2M
596,577
Source: CDC (2011)


WHERE WE DONATE VS. DISEASES THAT KILL US
Money Raised (in millions)
$250
$200
$150
$100
$50
Breast Cancer
Prostate Cancer
Heart Disease
Motor Neuron Disease (including ALS)
HIV/AIDS
Diabetes
Chronic Obstructive Pulmonary Disease
Suicide
100k
200k
300k
400k
500k
600k
Total Deaths (US)


Where We Donate vs.
Money Raised
(Cause)
Komen Race for the Cure
$257.9M
Movember
$147.0M
Jump Rope for Heart
$54.1M
ALS Ice Bucket Challenge
$22.9M
Ride to End AIDS
$14.0M
Fight for Air Climb
$7.0M
Step Out: Walk to Stop Diabetes
$4.2M
Out of Darkness Overnight Walk
$3.2M
Source - IFLScience: http://bit.ly/1ICiUv4


Breast Cancer
Prostate Cancer
Heart Disease
Motor Neuron Disease
HIV / AIDS
Chronic Obstructive Pulmonary Disease
Diabetes
Suicide
Money Raised ($, log sclae)
Total US Deaths (log scale)


Buying in Buckets
AT&T, Verizon and Sprint charge the same $20 per phone but have different data allowance levels. Comparison isn't easy.
AT&T
Sprint
Verizon
GB per month
Cost, in dollars
GB
40
30
25
24
16
12
10
8
6
4
3
2
1
$100
135
110
80
110
90
90
60
80
80
70
45
60
50
30
40
35
20
30
Sources: the companies
THE WALL STREET


Cost
125
100
75
50
25
0
10
20
30
40
GB
carrier
AT&T
Sprint
Verizon

- Visualization failure modes
- Cool vs scientific visualizations
- Making a graph pretty
- ggplot: grammar of graphics
- How to graph common data types.

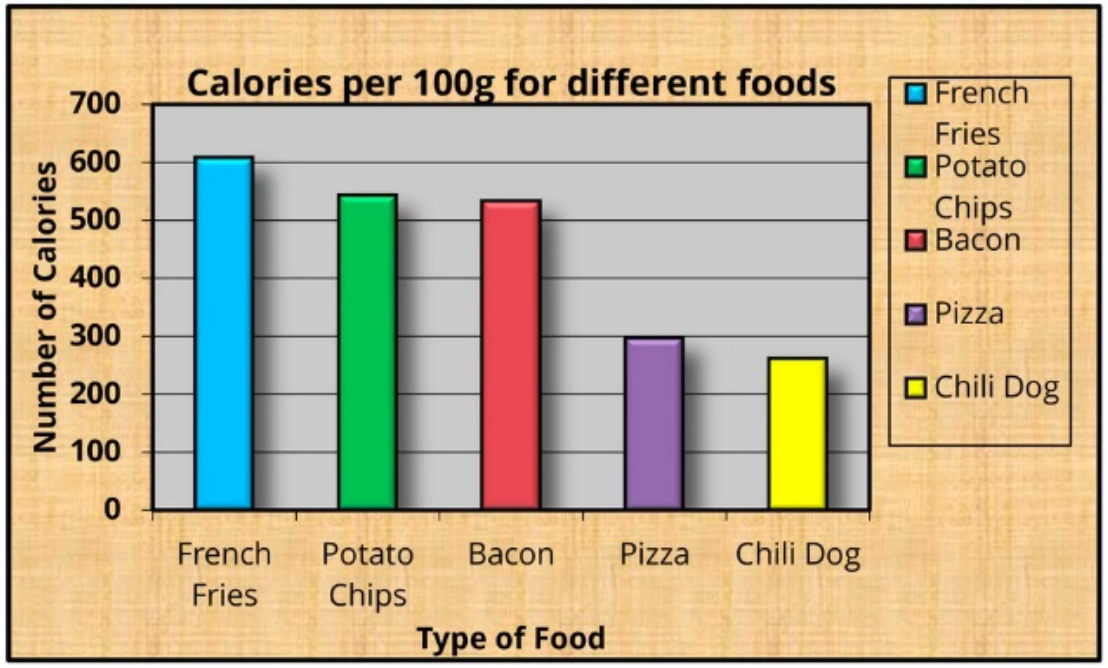
Remove

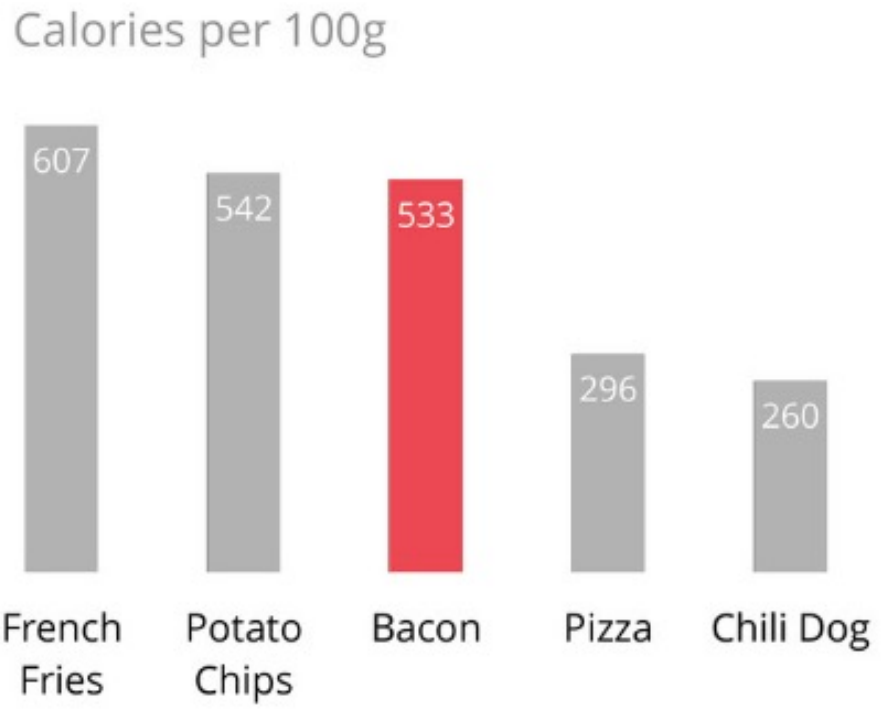
to improve

(the **data-ink** ratio)

Created by Darkhorse Analytics

www.darkhorseanalytics.com





May have gone a bit overboard into "visualization" territory – looks good, but starts violating some conventions:

- No Y axis
- Y axis label used as title

- Visualization failure modes
- Cool vs informative visualizations
- Making a graph pretty
- ggplot: grammar of graphics
- Graphs for common types of data.

```
library(ggplot2)

Fig <- ggplot(data=...,
              mapping=aes(...)) +
       facet_*() +
       geom_*() +
       stat_*() +
       scale_*() +
       theme*()
```

Basic operation:

Take a tidy data frame

map variables onto different aesthetic variables (e.g., x, y, color, fill, size, shape, alpha, group).

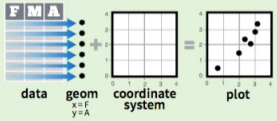
Draw some geom(etric entity) according to that mapping (e.g., point, line, tile, area, ribbon, etc.)

Data Visualization with ggplot2 Cheat Sheet

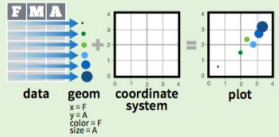
RStudio

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 **qplot()** or **ggplot()**

aesthetic mappings · data · geom

qplot(x = cty, y = hwy, color = cyl, data = mpg, geom = "point"**)**

Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

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

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

data · add layers, elements with + · layer = geom + default stat + layer specific mappings · additional elements

```
ggplot(mpg, aes(hwy, cty)) +
 geom_point(aes(color = cyl)) +
 geom_smooth(method ="lm") +
 coord_cartesian() +
 scale_color_gradient() +
 theme_bw()
```

Add a new layer to a plot with a **geom_*()** or **stat_*()** 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' x 5' 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))

a + geom_area(stat = "bin")
x, y, alpha, color, fill, linetype, size
b + geom_area(aes(y = ..density..), stat = "bin")

a + geom_density(kernel = "gaussian"**)**
x, y, alpha, color, fill, linetype, size, weight
b + geom_density(aes(y = ..county..))

a + geom_dotplot()
x, y, alpha, color, fill

a + geom_freqpoly()
x, y, alpha, color, linetype, size
b + geom_freqpoly(aes(y = ..density..))

a + geom_histogram(binwidth = 5**)**
x, y, alpha, color, fill, linetype, size, weight
b + geom_histogram(aes(y = ..density..))

Discrete

b <- ggplot(mpg, aes(fl))

b + geom_bar()
x, alpha, color, fill, linetype, size, weight

Graphical Primitives

c <- ggplot(map, aes(long, lat))

c + geom_polygon(aes(group = group)**)**
x, y, alpha, color, fill, linetype, size

d <- ggplot(economics, aes(date, unemploy))

d + geom_path(lineend="butt", linejoin="round', linemitre=1**)**
x, y, alpha, color, linetype, size

d + geom_ribbon(aes(ymin=unemploy - 900, ymax=unemploy + 900)**)**
x, ymax, ymin, alpha, color, fill, linetype, size

e <- ggplot(seals, aes(x = long, y = lat))

e + geom_segment(aes(
xend = long + delta_long,
yend = lat + delta_lat)**)**
x, xend, y, yend, alpha, color, linetype, size

e + geom_rect(aes(xmin = long, ymin = lat, xmax= long + delta_long, ymax = lat + delta_lat)**)**
xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size

Two Variables

Continuous X, Continuous Y

f <- ggplot(mpg, aes(cty, hwy))

f + geom_blank()

f + geom_jitter()
x, y, alpha, color, fill, shape, size

f + geom_point()
x, y, alpha, color, fill, shape, size

f + geom_quantile()
x, y, alpha, color, linetype, size, weight

f + geom_rug(sides = "bl"**)**
alpha, color, linetype, size

f + geom_smooth(model = lm**)**
x, y, alpha, color, fill, linetype, size, weight

f + geom_text(aes(label = cty)**)**
x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

Discrete X, Continuous Y

g <- ggplot(mpg, aes(class, hwy))

g + geom_bar(stat = "identity")
x, y, alpha, color, fill, linetype, size, weight

g + geom_boxplot()
lower, middle, upper, x, ymax, ymin, alpha, color, fill, linetype, shape, size, weight

g + geom_dotplot(binaxis = "y", stackdir = "center"**)**
x, y, alpha, color, fill

g + geom_violin(scale = "area"**)**
x, y, alpha, color, fill, linetype, size, weight

Discrete X, Discrete Y

h <- ggplot(diamonds, aes(cut, color))

h + geom_jitter()
x, y, alpha, color, fill, shape, size

Continuous Bivariate Distribution

i <- ggplot(movies, aes(year, rating))

i + geom_bin2d(binwidth = c(5, 0.5)**)**
xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size, weight

i + geom_density2d()
x, y, alpha, colour, linetype, size

i + geom_hex()
x, y, alpha, colour, fill size

Continuous Function

j <- ggplot(economics, aes(date, unemploy))

j + geom_area()
x, y, alpha, color, fill, linetype, size

j + geom_line()
x, y, alpha, color, linetype, size

j + geom_step(direction = "hv"**)**
x, y, alpha, color, linetype, size

Visualizing error

df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
k <- ggplot(df, aes(grp, fit, ymin = fit-se, ymax = fit+se))

k + geom_crossbar(fatten = 2**)**
x, y, ymax, ymin, alpha, color, fill, linetype, size

k + geom_errorbar()
x, ymax, ymin, alpha, color, linetype, size, width (also **geom_errorbarh()**)

k + geom_linerange()
x, ymin, ymax, alpha, color, linetype, size

k + geom_pointrange()
x, y, ymin, ymax, alpha, color, fill, linetype, shape, size

Maps

data <- data.frame(murder = USArrests$Murder, state = tolower(rownames(USArrests)))
map <- map_data("state")
l <- ggplot(data, aes(fill = murder))

l + geom_map(aes(map_id = state), map = map**) + expand_limits(**x = map$long, y = map$lat**)**
map_id, alpha, color, fill, linetype, size

Three Variables

seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2))
m <- ggplot(seals, aes(long, lat))

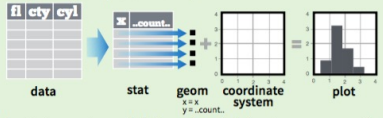
m + geom_contour(aes(z = z)**)**
x, y, z, alpha, colour, linetype, size, weight

m + geom_raster(aes(fill = z), hjust=0.5, vjust=0.5, interpolate=FALSE**)**
x, y, alpha, fill

m + geom_tile(aes(fill = z)**)**
x, y, alpha, color, fill, linetype, size

Stats - An alternative way to build a layer

Some plots visualize a **transformation** of the original data set. Use a **stat** to choose a common transformation to visualize, e.g. **a + geom_bar(stat = "bin")**



Each stat creates additional variables to map aesthetics to. These variables use a common **..name..** syntax.

stat functions and geom functions both combine a stat with a geom to make a layer, i.e. **stat_bin(geom="bar")** does the same as **geom_bar(stat="bin")**

stat function | layer specific mappings | variable created by transformation

i + stat_density2d(aes(fill = ..level..), geom = "polygon", n = 100)

geom for layer | parameters for stat

1D distributions

a + stat_bin(binwidth = 1, origin = 10**)**
x, y | ..count.., ..ncount.., ..density.., ..ndensity..
a + stat_bindot(binwidth = 1, binaxis = "x"**)**
x, y, | ..count.., ..ncount..
a + stat_density(adjust = 1, kernel = "gaussian"**)**
x, y, | ..count.., ..density.., ..scaled..

2D distributions

f + stat_bin2d(bins = 30, drop = TRUE**)**
x, y, fill | ..count.., ..density..
f + stat_binhex(bins = 30**)**
x, y, fill | ..count.., ..density..
f + stat_density2d(contour = TRUE, n = 100**)**
x, y, color, size | ..level..

3 Variables

m + stat_contour(aes(z = z)**)**
x, y, z, order | ..level..
m+ stat_spoke(aes(radius= z, angle = z)**)**
angle, radius, x, xend, y, yend | ..x.., ..xend.., ..y.., ..yend..
m + stat_summary_hex(aes(z = z), bins = 30, fun = mean**)**
x, y, z, fill | ..value..
m + stat_summary2d(aes(z = z), bins = 30, fun = mean**)**
x, y, z, fill | ..value..

Comparisons

g + stat_boxplot(coef = 1.5**)**
x, y | ..lower.., ..middle.., ..upper.., ..outliers..
g + stat_ydensity(adjust = 1, kernel = "gaussian", scale = "area"**)**
x, y | ..density.., ..scaled.., ..count.., ..n.., ..violinwidth.., ..width..

Functions

f + stat_ecdf(n = 40**)**
x, y | ..x.., ..y..
f + stat_quantile(quantiles = c(0.25, 0.5, 0.75), formula = y ~ log(x), method = "rq"**)**
x, y | ..quantile.., ..x.., ..y..
f + stat_smooth(method = "auto", formula = y ~ x, se = TRUE, n = 80, fullrange = FALSE, level = 0.95**)**
x, y | ..se.., ..x.., ..y.., ..ymin.., ..ymax..

General Purpose

ggplot() + stat_function(aes(x = -3:3), fun = dnorm, n = 101, args = list(sd=0.5)**)**
x | ..y..
f + stat_identity()
ggplot() + stat_qq(aes(sample=1:100), distribution = qt, dparams = list(df=5)**)**
sample, x, y | ..x.., ..y..
f + stat_sum()
x, y, size | ..size..
f + stat_summary(fun.data = "mean_cl_boot"**)**
f + stat_unique()

Scales

Scales control how a plot maps data values to the visual values of an aesthetic. To change the mapping, add a custom scale.

n <- b + geom_bar(aes(fill = fl)**)**
n

scale_ | aesthetic to adjust | prepackaged scale to use | scale specific arguments

n + scale_fill_manual(
values = c("skyblue", "royalblue", "blue", "navy"),
limits = c("d", "e", "p", "r"), breaks =c("d", "e", "p", "r"),
name = "fuel", labels = c("D", "E", "P", "R")**)**

range of values to include in mapping | title to use in legend/axis | labels to use in legend/axis | breaks to use in legend/axis

General Purpose scales

Use with any aesthetic:
alpha, color, fill, linetype, shape, size

scale_*_continuous() - map cont' values to visual values
scale_*_discrete() - map discrete values to visual values
scale_*_identity() - use data values **as** visual values
scale_*_manual(values = c()**)** - map discrete values to manually chosen visual values

X and Y location scales

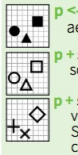
Use with x or y aesthetics (x shown here)

scale_x_date(labels = date_format("%m/%d"), breaks = date_breaks("2 weeks")**)** - treat x values as dates. See ?strptime for label formats.
scale_x_datetime() - treat x values as date times. Use same arguments as scale_x_date().
scale_x_log10() - Plot x on log10 scale
scale_x_reverse() - Reverse direction of x axis
scale_x_sqrt() - Plot x on square root scale

Color and fill scales

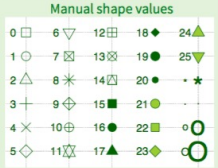
Discrete	Continuous
n <- b + geom_bar(aes(fill = fl)**)**	**o <- a + geom_dotplot(**aes(fill = ..x..)**)**
n + scale_fill_brewer(palette = "Blues"**)** For palette choices: library(RColorBrewer) display.brewer.all()	**o + scale_fill_gradient(**low = "red", high = "yellow"**)**
	o + scale_fill_gradient2(low = "red", hight = "blue", mid = "white", midpoint = 25**)**
n + scale_fill_grey(start = 0.2, end = 0.8, na.value = "red"**)**	**o + scale_fill_gradientn(**colours = terrain.colors(6)**)** Also: rainbow(), heat.colors(), topo.colors(), cm.colors(), RColorBrewer::brewer.pal()

Shape scales



p <- f + geom_point(aes(shape = fl)**)**
p + scale_shape(solid = FALSE**)**
p + scale_shape_manual(values = c(3:7)**)** Shape values shown in chart on right

Manual shape values



Size scales

q <- f + geom_point(aes(size = cyl)**)**

q + scale_size_area(max = 6**)** Value mapped to area of circle (not radius)

Coordinate Systems

r <- b + geom_bar()

r + coord_cartesian(xlim = c(0, 5)**)**
xlim, ylim
The default cartesian coordinate system

r + coord_fixed(ratio = 1/2**)**
ratio, xlim, ylim
Cartesian coordinates with fixed aspect ratio between x and y units

r + coord_flip()
xlim, ylim
Flipped Cartesian coordinates

r + coord_polar(theta = "x", direction=1 **)**
theta, start, direction
Polar coordinates

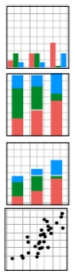
r + coord_trans(xtrans = "sqrt"**)**
xtrans, ytrans, limx, limy
Transformed cartesian coordinates. Set extras and strains to the name of a window function.

z + coord_map(projection = "ortho", orientation=c(41, -74, 0)**)**
projection, orientation, xlim, ylim
Map projections from the mapproj package (mercator (default), azequalarea, lagrange, etc.)

Position Adjustments

Position adjustments determine how to arrange geoms that would otherwise occupy the same space.

s <- ggplot(mpg, aes(fl, fill = drv))



s + geom_bar(position = "dodge")
Arrange elements side by side

s + geom_bar(position = "fill")
Stack elements on top of one another, normalize height

s + geom_bar(position = "stack")
Stack elements on top of one another

f + geom_point(position = "jitter")
Add random noise to X and Y position of each element to avoid overplotting

Each position adjustment can be recast as a function with manual **width** and **height** arguments

s + geom_bar(position = position_dodge(width = 1))

Themes

r + theme_bw()
White background with grid lines

r + theme_grey()
Grey background (default theme)

r + theme_classic()
White background no gridlines

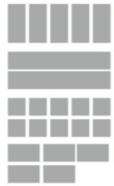
r + theme_minimal()
Minimal theme

ggthemes - Package with additional ggplot2 themes

Faceting

Facets divide a plot into subplots based on the values of one or more discrete variables.

t <- ggplot(mpg, aes(cty, hwy)) + geom_point()



t + facet_grid(. ~ fl)
facet into columns based on fl
t + facet_grid(year ~ .)
facet into rows based on year
t + facet_grid(year ~ fl)
facet into both rows and columns
t + facet_wrap(~ fl)
wrap facets into a rectangular layout

Set **scales** to let axis limits vary across facets

t + facet_grid(y ~ x, scales = "free")
x and y axis limits adjust to individual facets
- **"free_x"** - x axis limits adjust
- **"free_y"** - y axis limits adjust

Set **labeller** to adjust facet labels

t + facet_grid(. ~ fl, labeller = label_both)

fl: c	fl: d	fl: e	fl: p	fl: r

t + facet_grid(. ~ fl, labeller = label_bquote(alpha ^ .(x)**))**

α^c	α^d	α^e	α^p	α^r

t + facet_grid(. ~ fl, labeller = label_parsed)

c	d	e	p	r

Labels

t + ggtitle("New Plot Title"**)**
Add a main title above the plot
t + xlab("New X label"**)**
Change the label on the X axis
t + ylab("New Y label"**)**
Change the label on the Y axis
t + labs(title =" New title", x = "New x", y = "New y"**)**
All of the above

Use scale functions to update legend labels

Legends

t + theme(legend.position = "bottom"**)**
Place legend at "bottom", "top", "left", or "right"
t + guides(color = "none"**)**
Set legend type for each aesthetic: colorbar, legend, or none (no legend)
t + scale_fill_discrete(name = "Title", labels = c("A", "B", "C")**)**
Set legend title and labels with a scale function.

Zooming

Without clipping (preferred)
t + coord_cartesian(xlim = c(0, 100), ylim = c(10, 20)**)**

With clipping (removes unseen data points)

t + xlim(0, 100**) + ylim(**10, 20**)**
t + scale_x_continuous(limits = c(0, 100)**) + scale_y_continuous(**limits = c(0, 100)**)**

- Visualization failure modes
- Cool vs informative visualizations
- Making a graph pretty
- ggplot: grammar of graphics
- Graphs for common types of data.

- Practice in R.

- More exotic graph types / considerations

Goal: show how response/dependent variable(s) change with explanatory/independent variable(s).

What kind of variables? Categorical? Numerical?

Helps to think of it as an abstract formula of sorts, e.g.,:

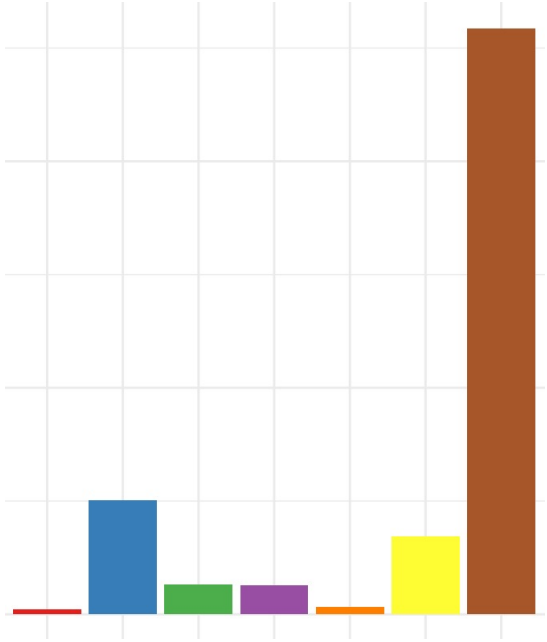
How does height (numerical response) vary across sex (categorical), nationality (categorical), and parents' income (numerical):

numerical ~ 2*categorical + numerical

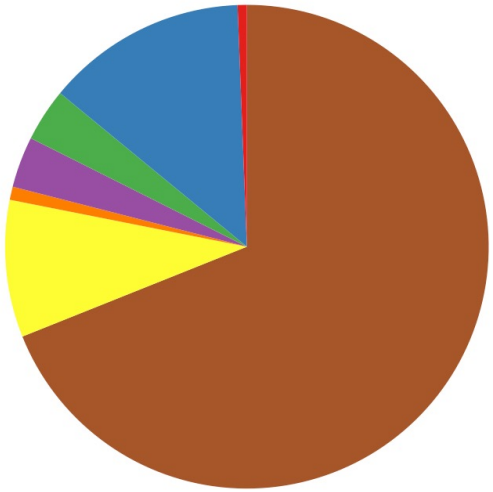
This abstraction helps you pick starting points for graphs.

categorical ~ 0

(1 categorical response variable, with 0 explanatory variables)

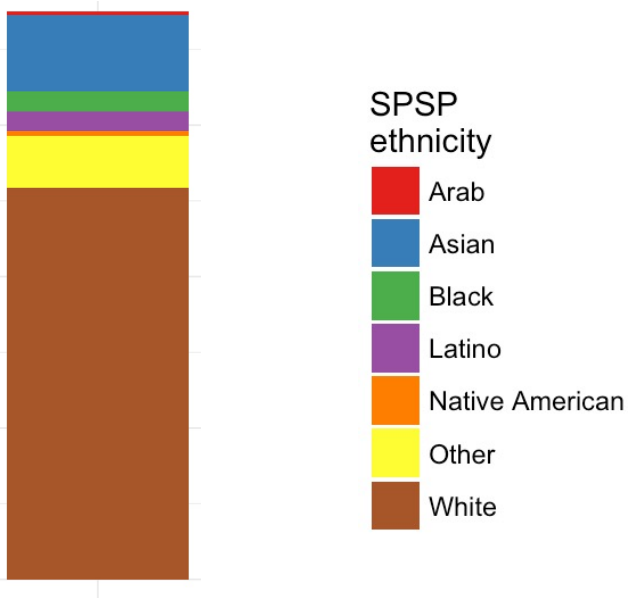


Histogram
barplot of counts
++ Easiest comparisons
- Hardest proportion



Pie chart
- Hardest comparisons
++ easiest proportion

- Waste of ink
- Considered tacky.



Stacked bar plot
+ easy-ish comparisons
+ easy-ish proportion

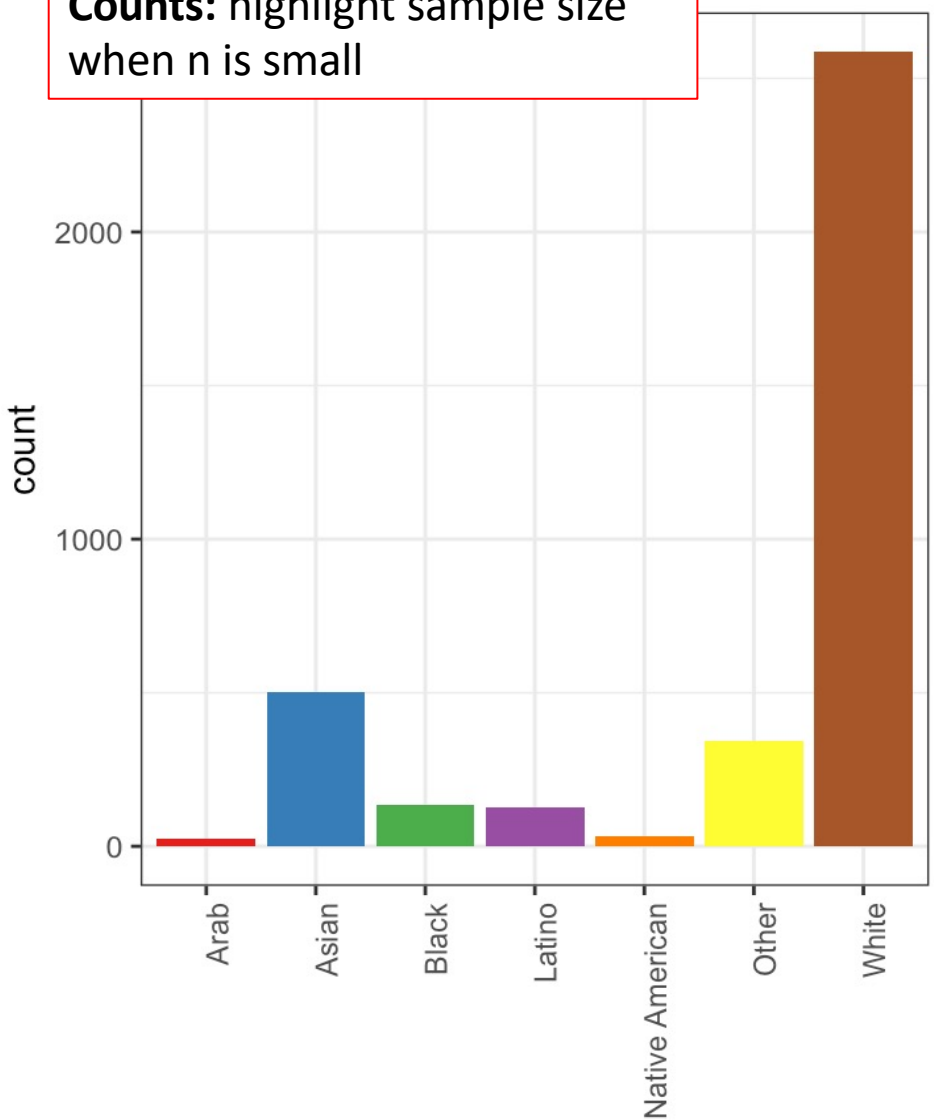
+ socially acceptable pie chart

Data: `http://vulstats.ucsd.edu/data/spsp.demographics.cleaned.csv`

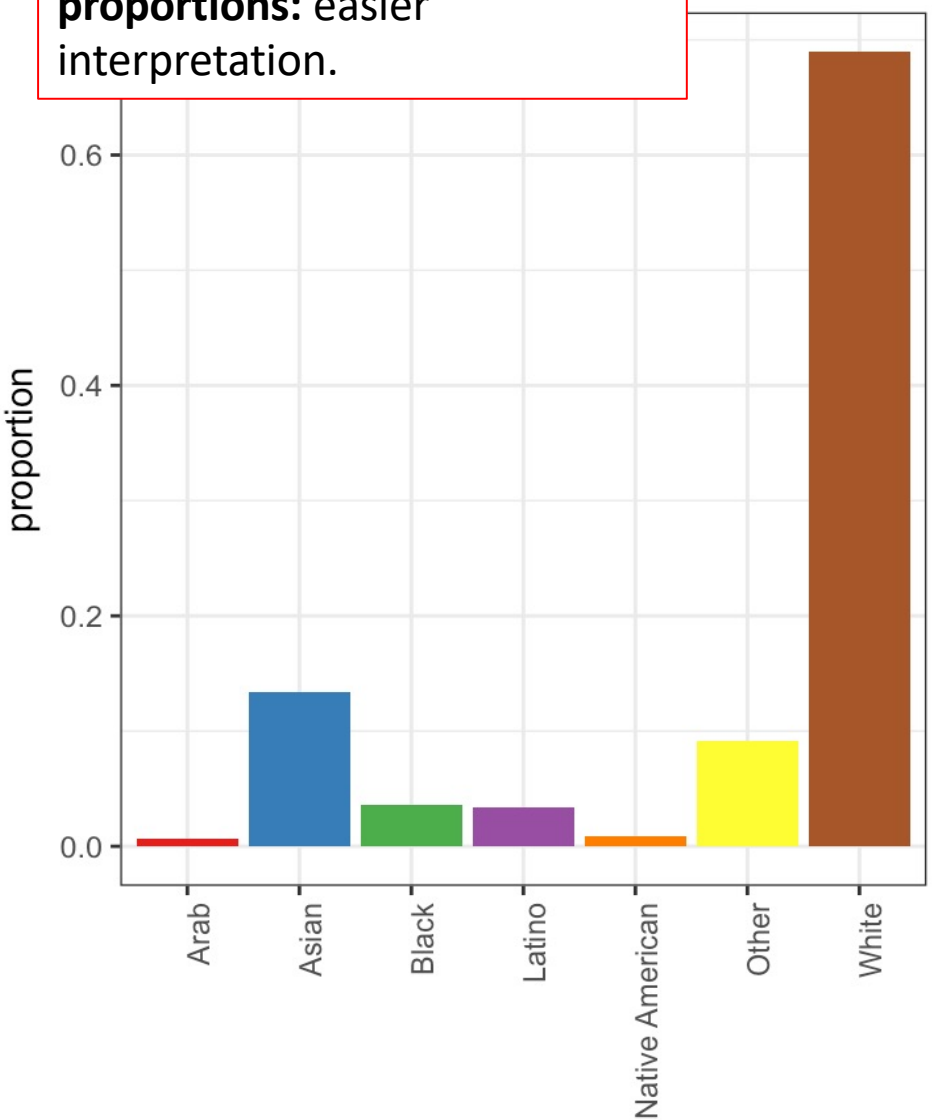
categorical ~ 0

(1 categorical response variable, with 0 explanatory variables)

Counts: highlight sample size when n is small

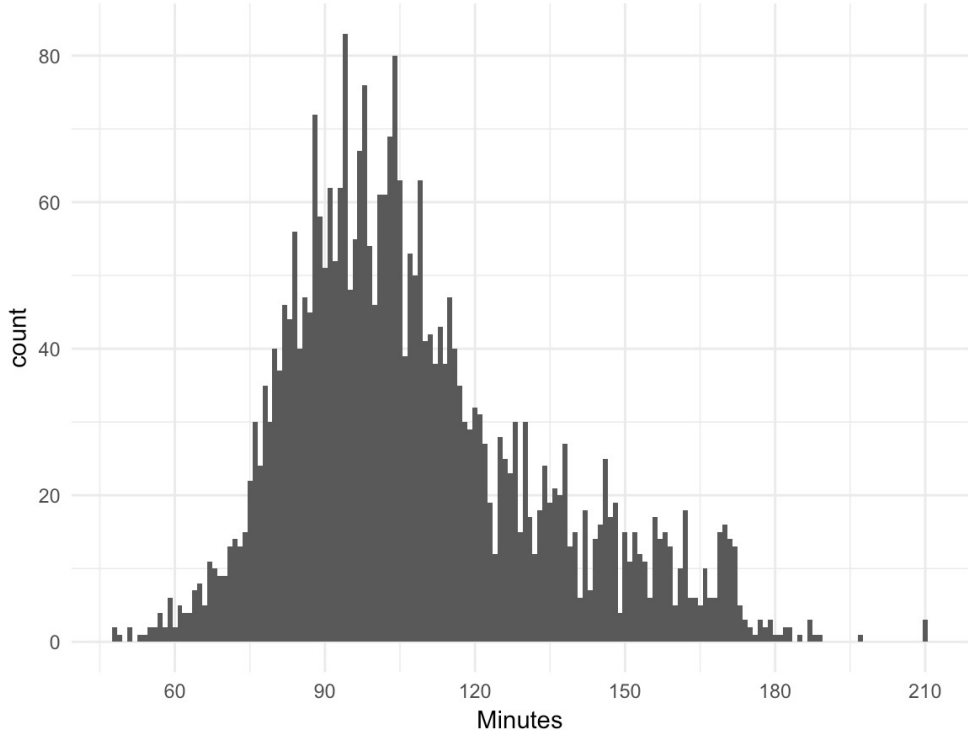


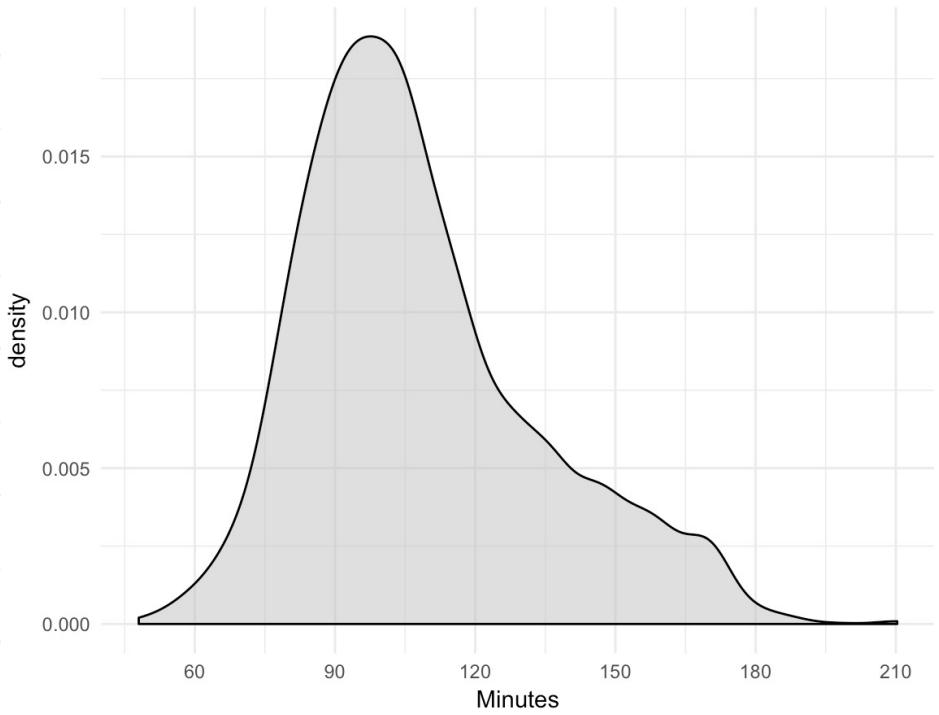
proportions: easier interpretation.



numerical ~ o

(1 numerical response variable, with o explanatory variables)





Histogram

+ Portrays noisiness.
- Impression sensitive to bins

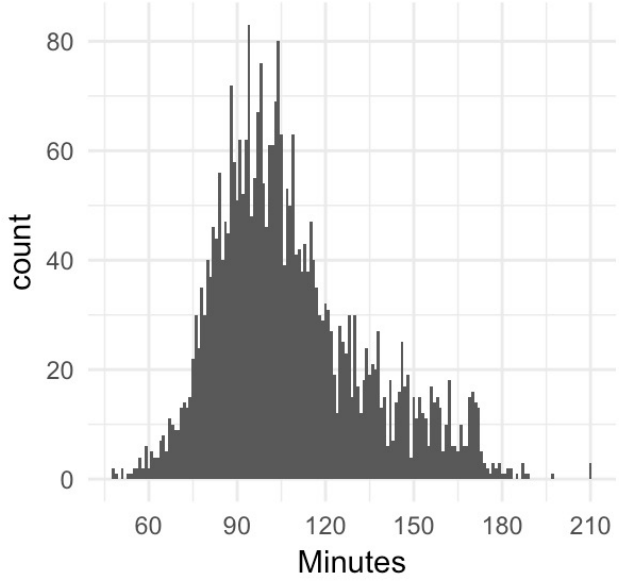
Smoothed density

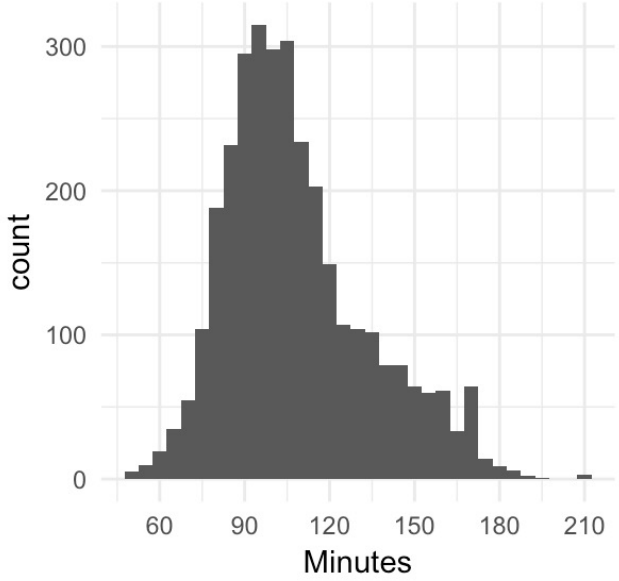
- Obscures noisiness
+ not too sensitive to reasonable kernel width.

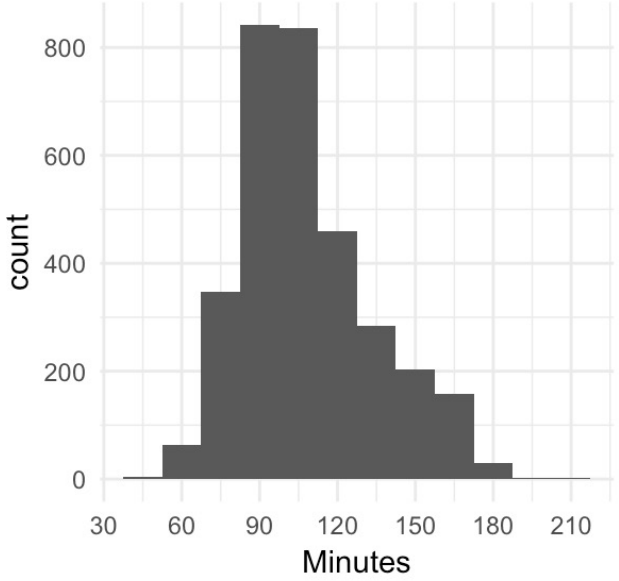
Data: `http://vulstats.ucsd.edu/data/cal1020.cleaned.Rdata`

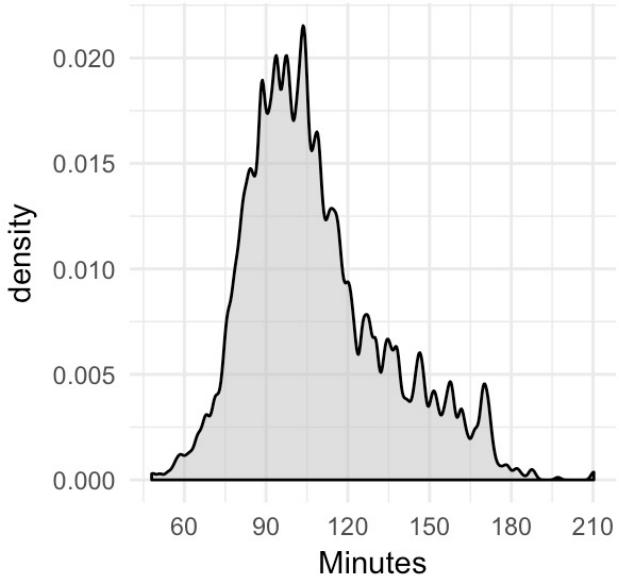
numerical ~ 0

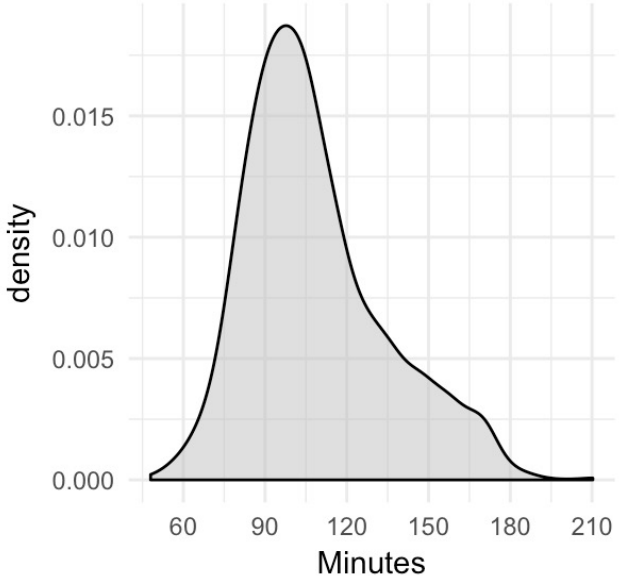
(1 numerical response variable, with 0 explanatory variables)

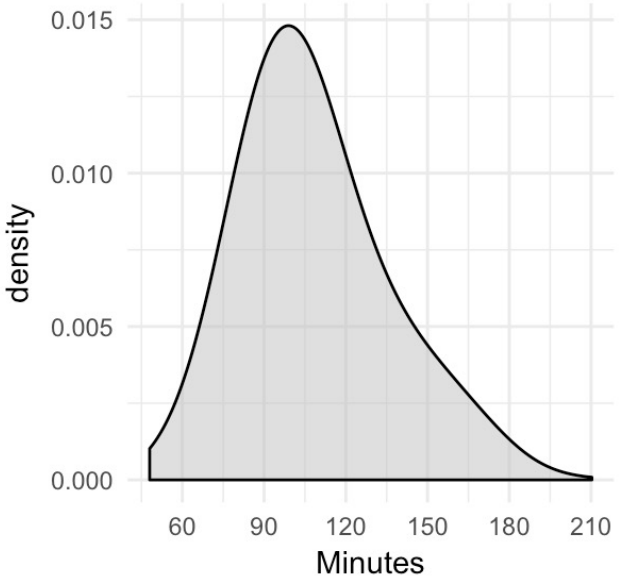






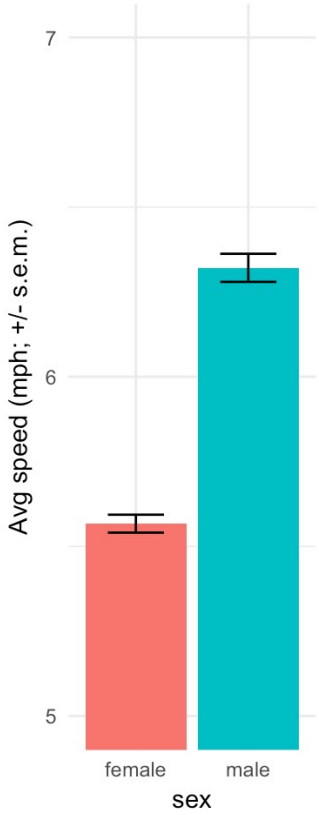




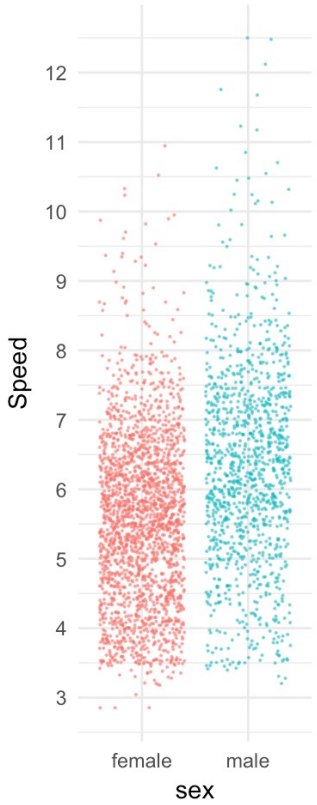


numerical ~ categorical

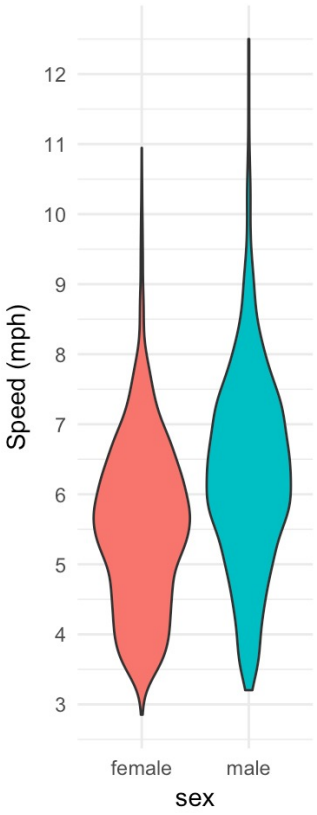
(1 numerical response variable, with 1 categorical explanatory variable)



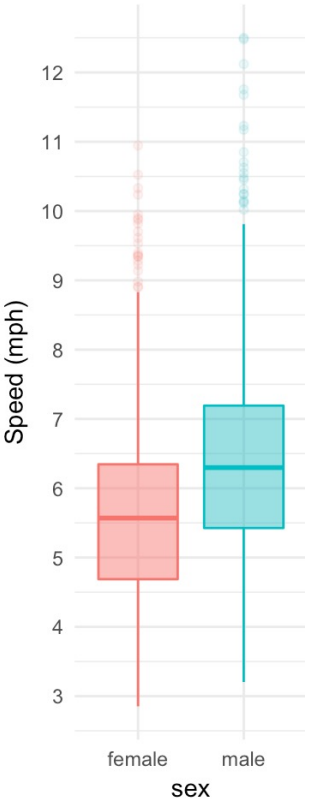
Mean+error



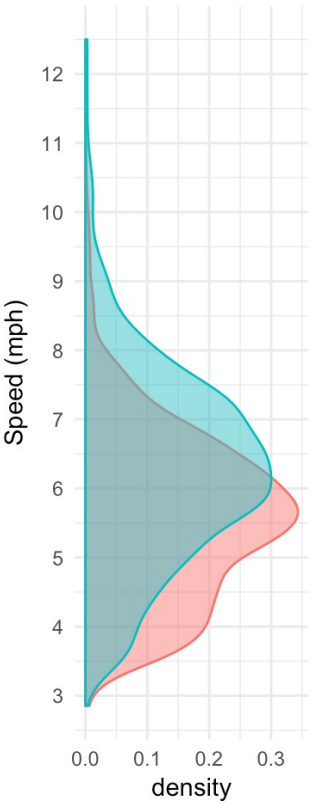
Jitter



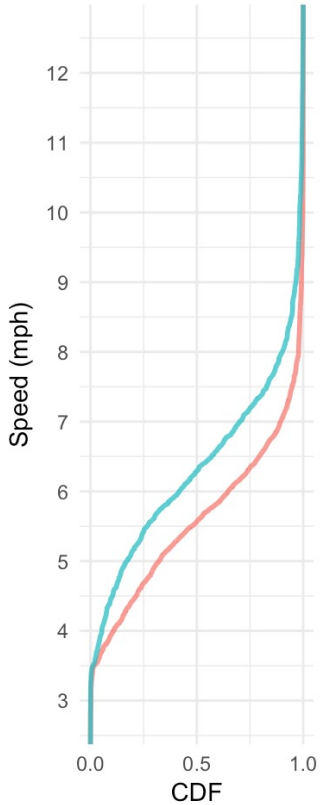
violin



boxplot



densities
(coords flipped)



Emp CDF
(coords flipped)

Easy stat. comparison

Useful when n is small

Useful when n is large

Best when coords not flipped, Best for few categories (<4?).

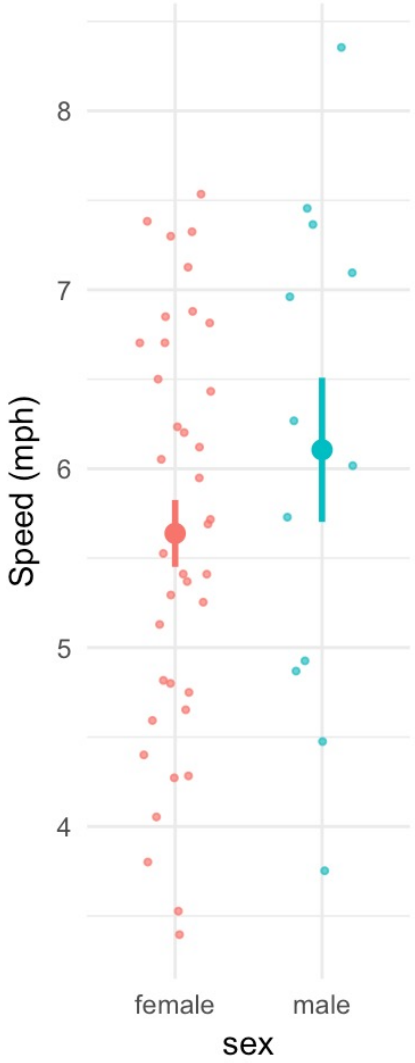
numerical ~ categorical

(1 numerical response variable, with 1 categorical explanatory variable)

- Always put error bars on bar charts (std. error or CI are fine)
- Look at rawer data (e.g,. strip charts) before going to more compressed plots.
- By removing the solid bar from a bar chart, you can add a good visualization of data distribution. This is better.

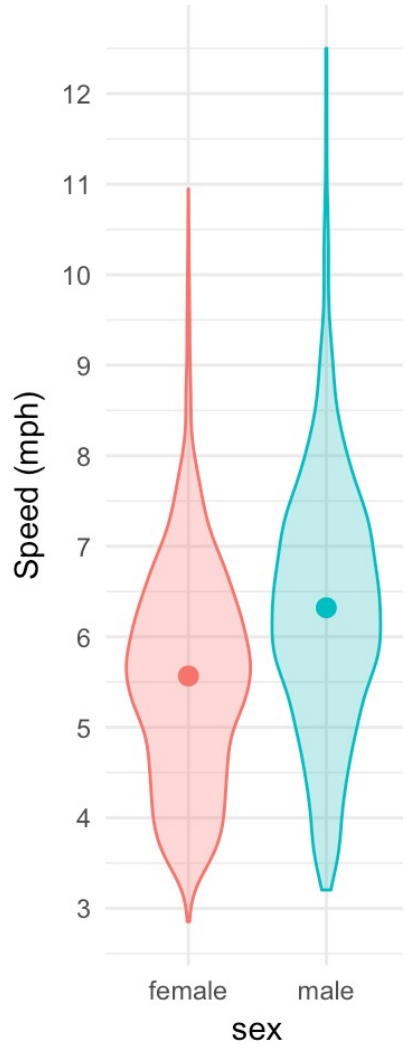
numerical ~ categorical

(my suggestions)



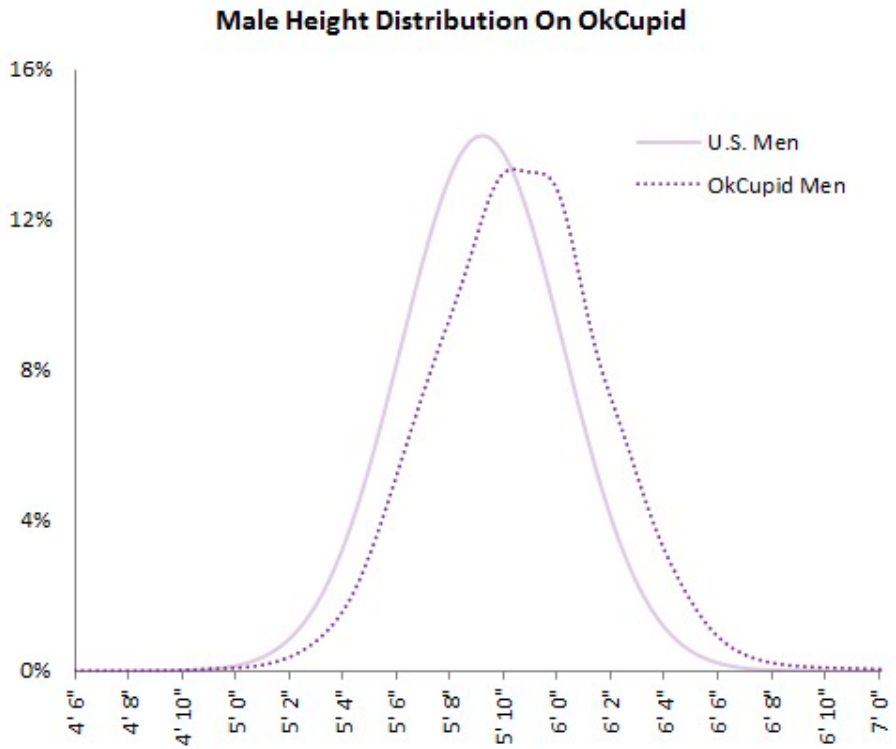
With small n:
Show all the data points with jitter
(here, data are sub-sampled to generate a low n scenario)

With large n:
Show distribution with violin or density.

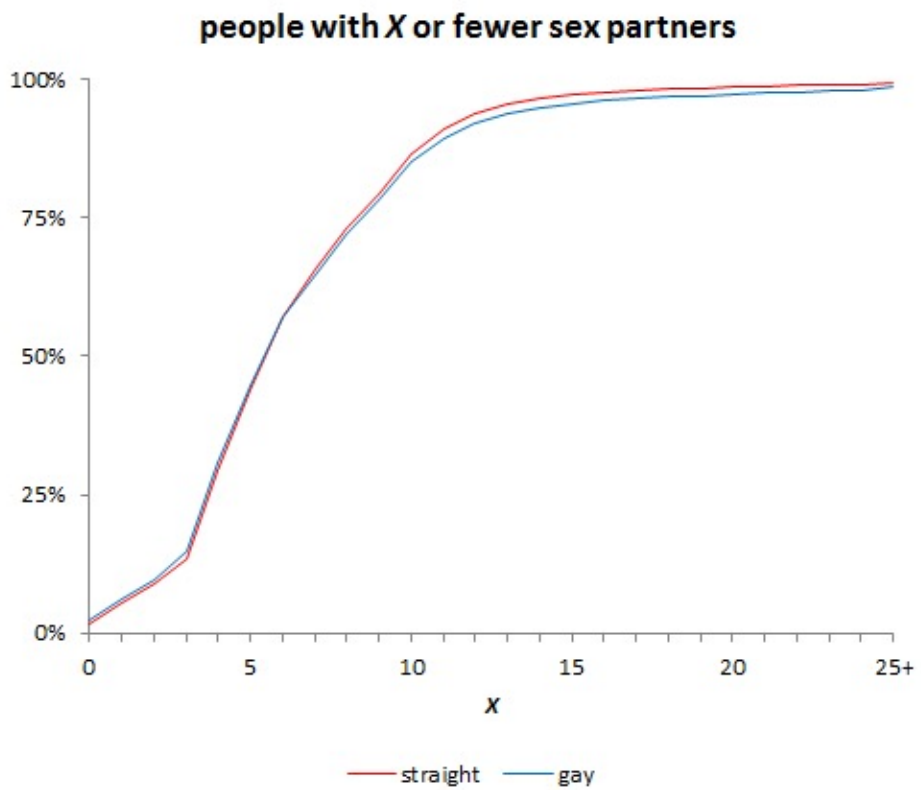


numerical ~ categorical

(eclectic plots, useful with large n, weird distributional differences)



Overlayed density/histograms
With large n can show weird differences.

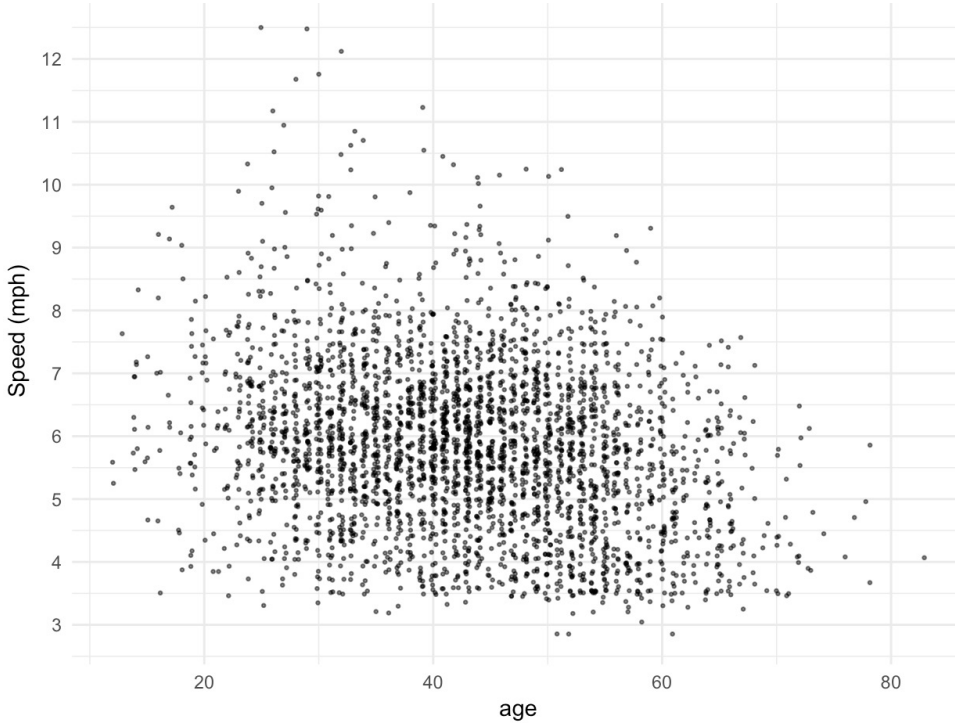


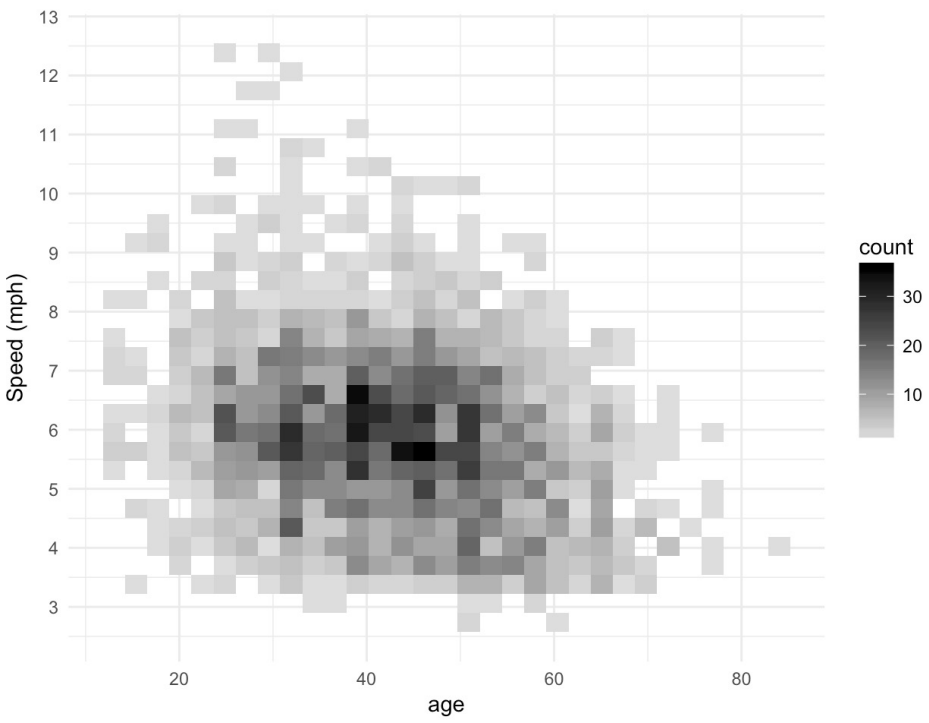
Cumulative distribution functions
Highlights differences in the tails.
Only useful with really large n
(so tails aren't just noise).

numerical ~ numerical

(1 numerical response variable, with 1 numerical explanatory variable)

2 x numerical ~ 0



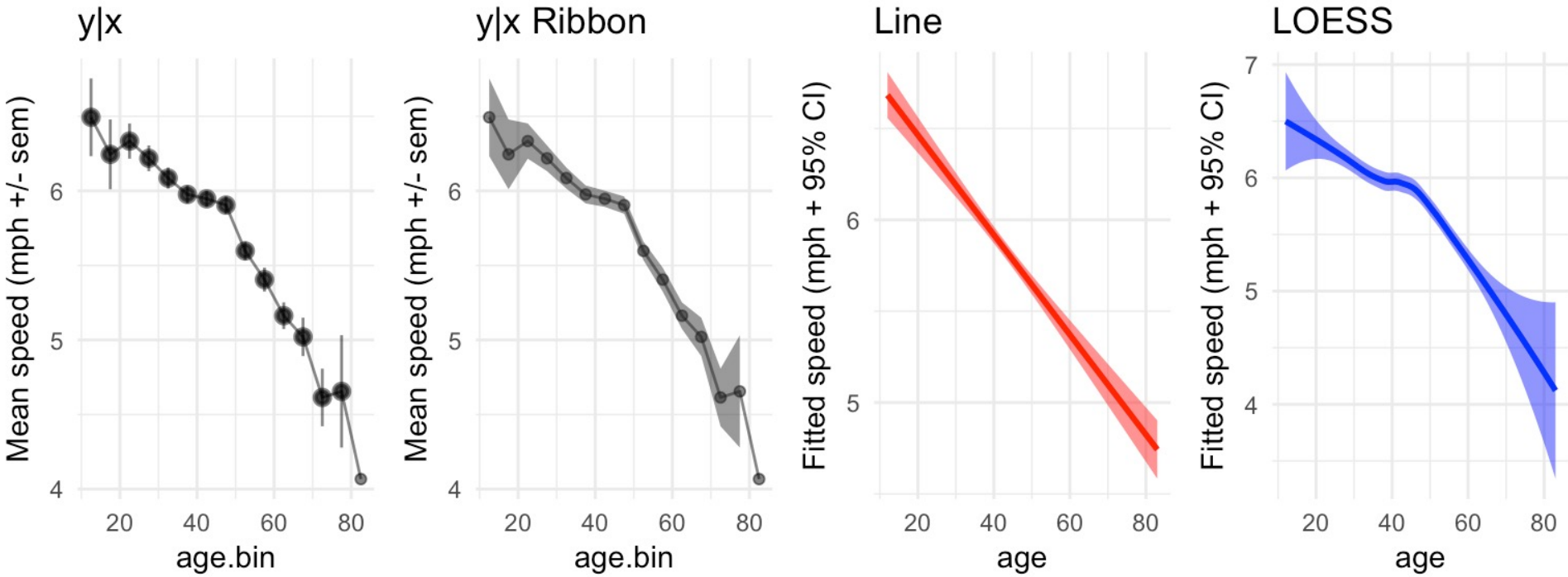


Scatterplot:
Best option with small n.
Hard to make legible with large n.

2D histogram heatmap:
Useless for small n.
Best option with large n.

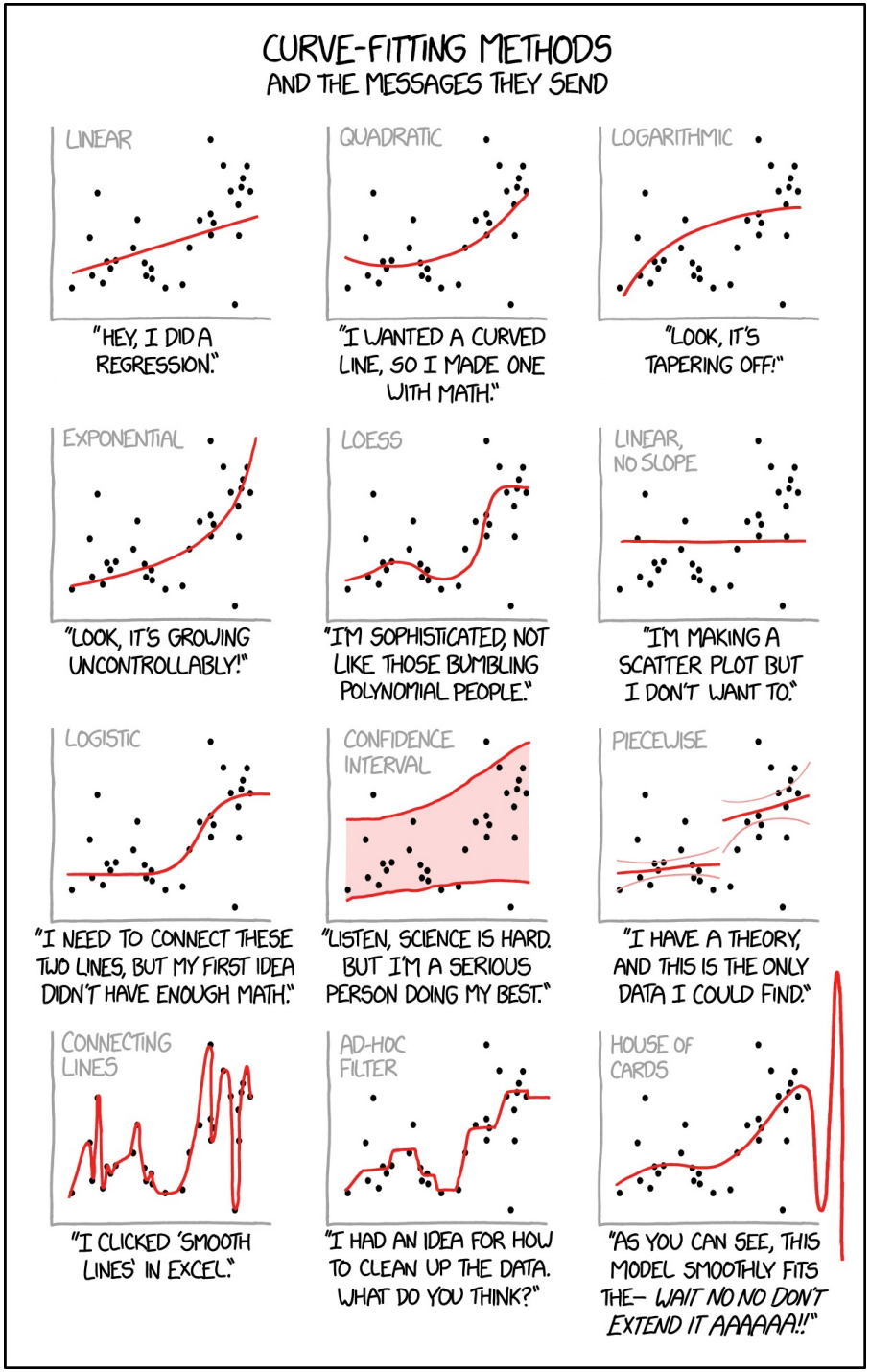
numerical ~ numerical

(1 numerical response variable, with 1 numerical explanatory variable)



Conditional means
This will require binning by x.

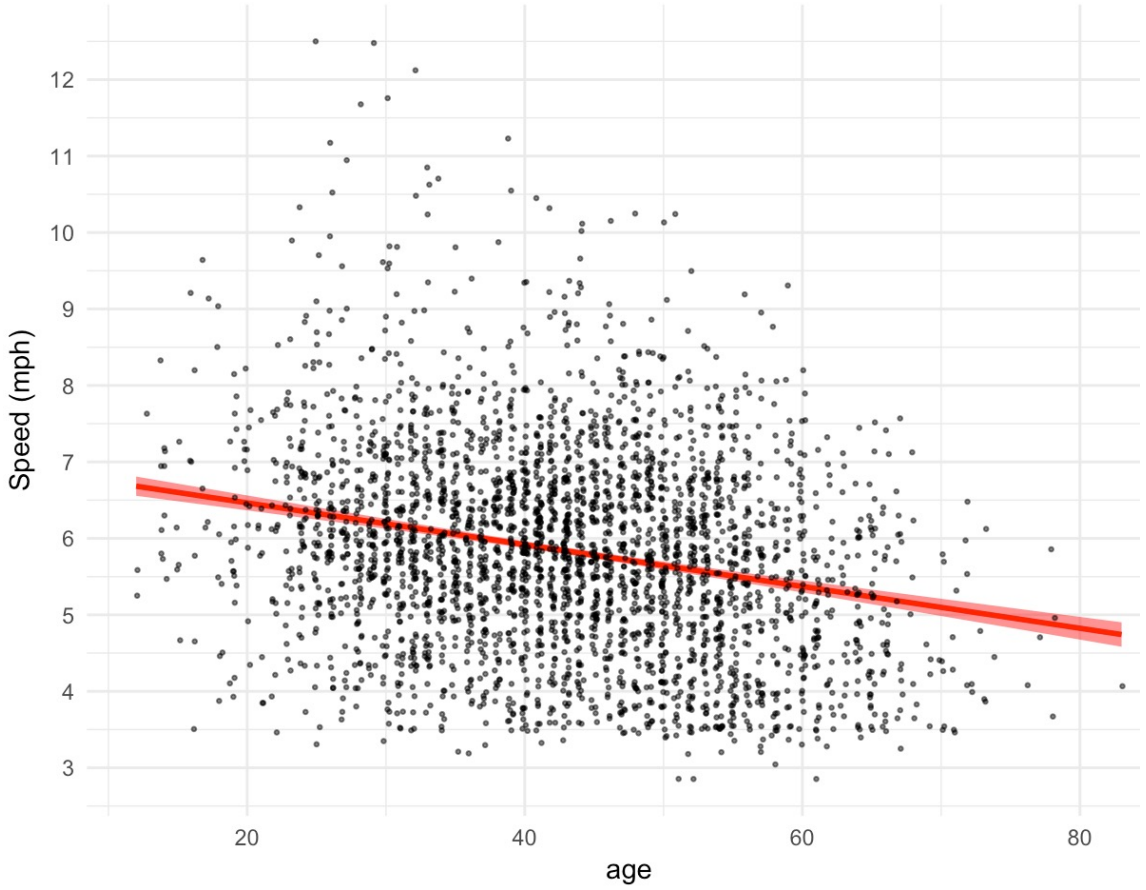
Fitted conditional means
Very rarely should you show these on their own, without the raw data.
Generally: use method=lm, rather than loess.


CURVE-FITTING METHODS
AND THE MESSAGES THEY SEND
LINEAR
"HEY, I DID A REGRESSION."
QUADRATIC
"I WANTED A CURVED LINE, SO I MADE ONE WITH MATH."
LOGARITHMIC
"LOOK, IT'S TAPERING OFF!"
EXPONENTIAL
"LOOK, IT'S GROWING UNCONTROLLABLY!"
LOESS
"I'M SOPHISTICATED, NOT LIKE THOSE BUMBLING POLYNOMIAL PEOPLE."
LINEAR, NO SLOPE
"I'M MAKING A SCATTER PLOT BUT I DON'T WANT TO."
LOGISTIC
"I NEED TO CONNECT THESE TWO LINES, BUT MY FIRST IDEA DIDN'T HAVE ENOUGH MATH."
CONFIDENCE INTERVAL
"LISTEN, SCIENCE IS HARD. BUT I'M A SERIOUS PERSON DOING MY BEST."
PIECEWISE
"I HAVE A THEORY, AND THIS IS THE ONLY DATA I COULD FIND."
CONNECTING LINES
"I CLICKED 'SMOOTH LINES' IN EXCEL."
AD-HOC FILTER
"I HAD AN IDEA FOR HOW TO CLEAN UP THE DATA. WHAT DO YOU THINK?"
HOUSE OF CARDS
"AS YOU CAN SEE, THIS MODEL SMOOTHLY FITS THE– WAIT NO NO DON'T EXTEND IT AAAAAA!!"

Credit: xkcd

numerical ~ numerical

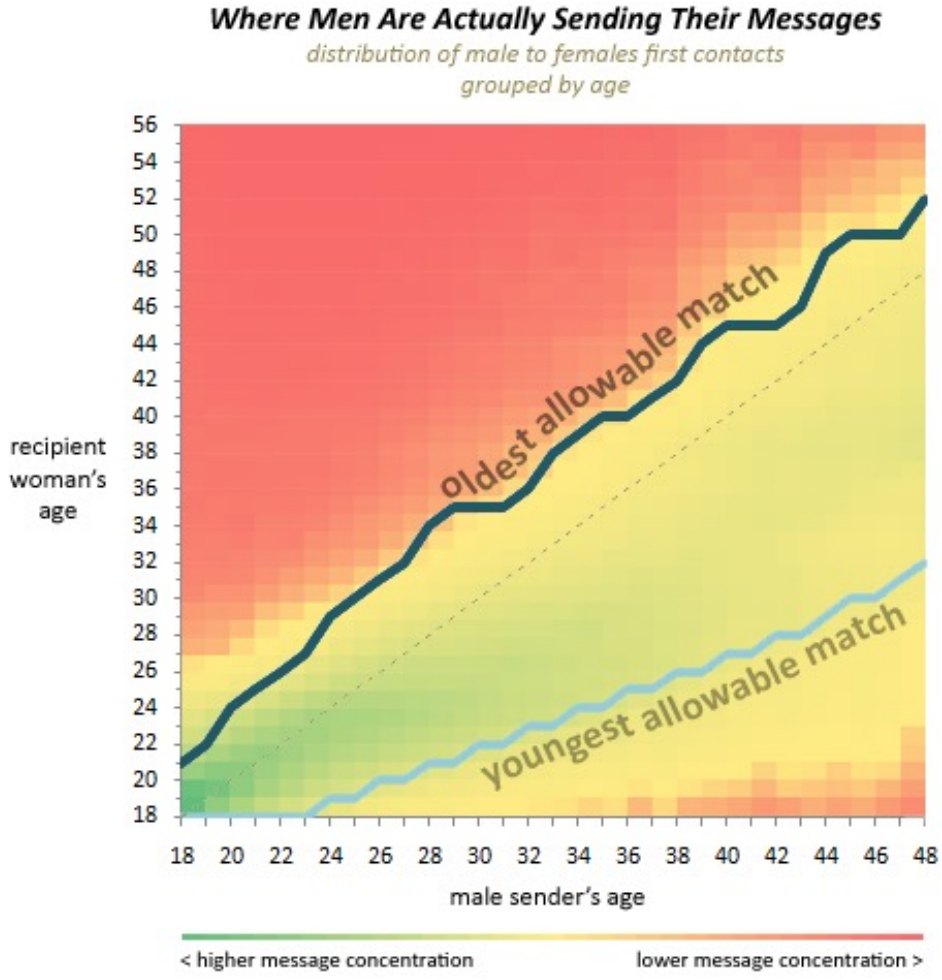
(my recommendation)

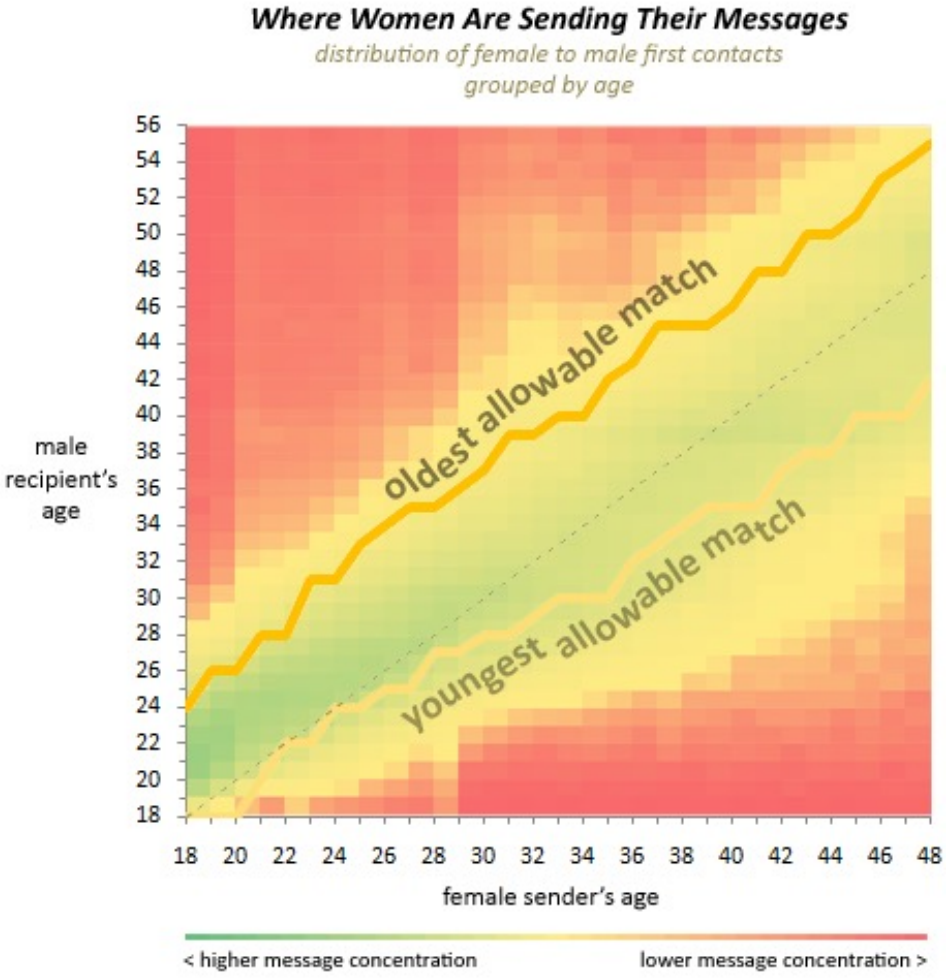


My recommendation:
Show data, show fit.

numerical ~ numerical

(1 numerical response variable, with 1 numerical explanatory variable)



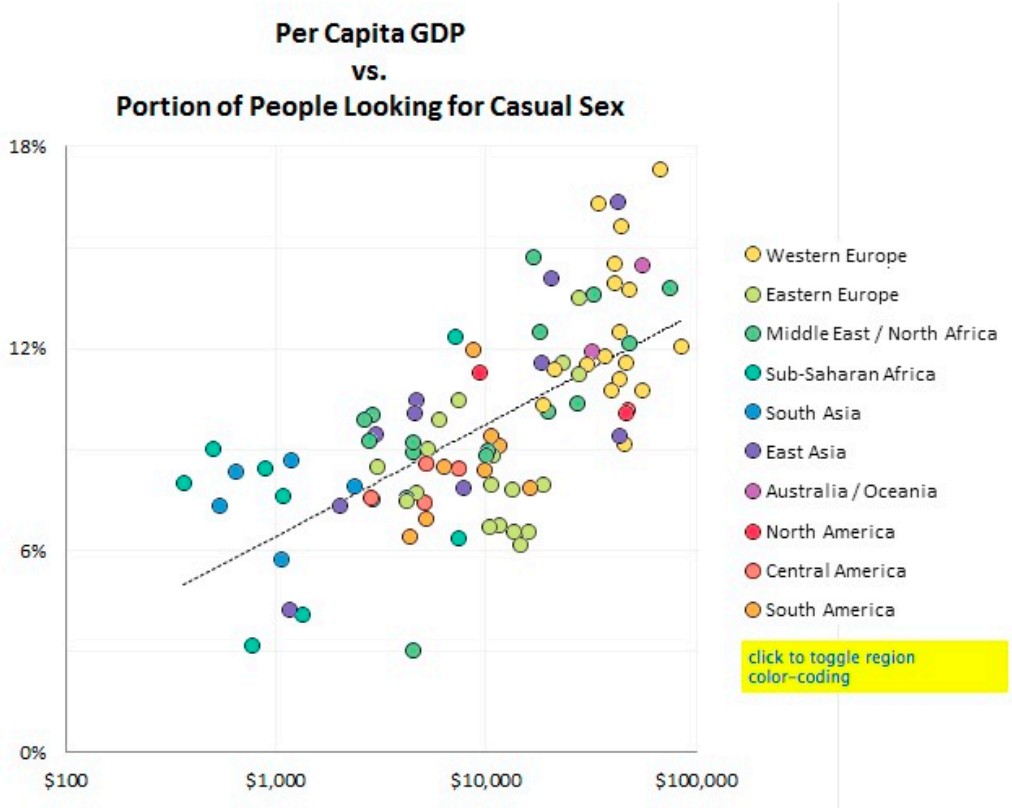


Normalization by x useful when you don't care about distribution over x.

Note: you are unlikely to luxuriate in this much data.

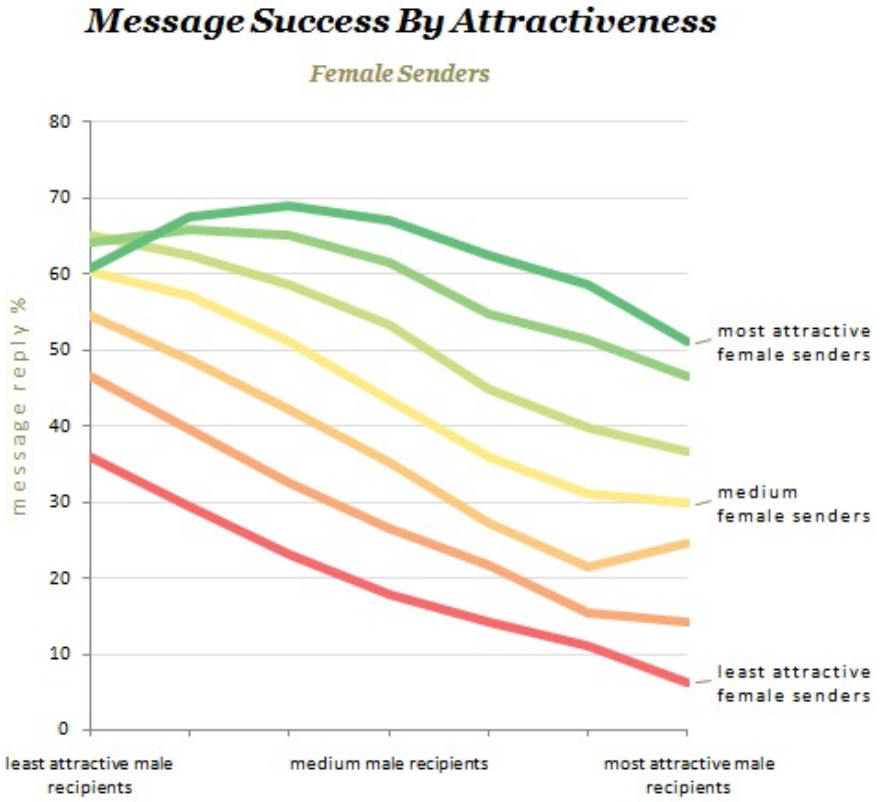
numerical ~ numerical + categorical

(1 numerical response, with numerical & categorical explanatory variable)



Color-coded scatterplot
Hard to parse with lots of data.

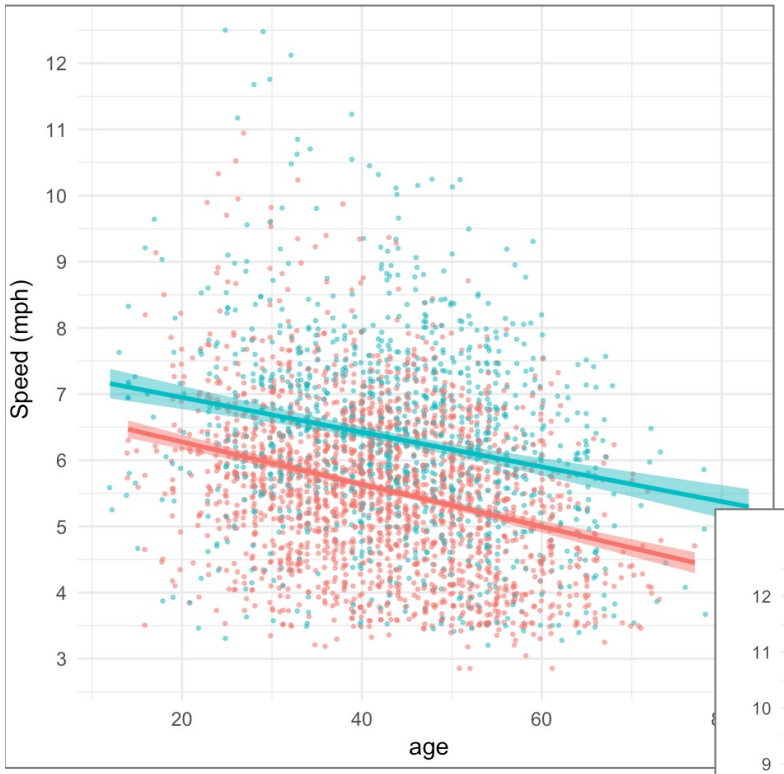
Note importance of explanatory variable on the x axis!



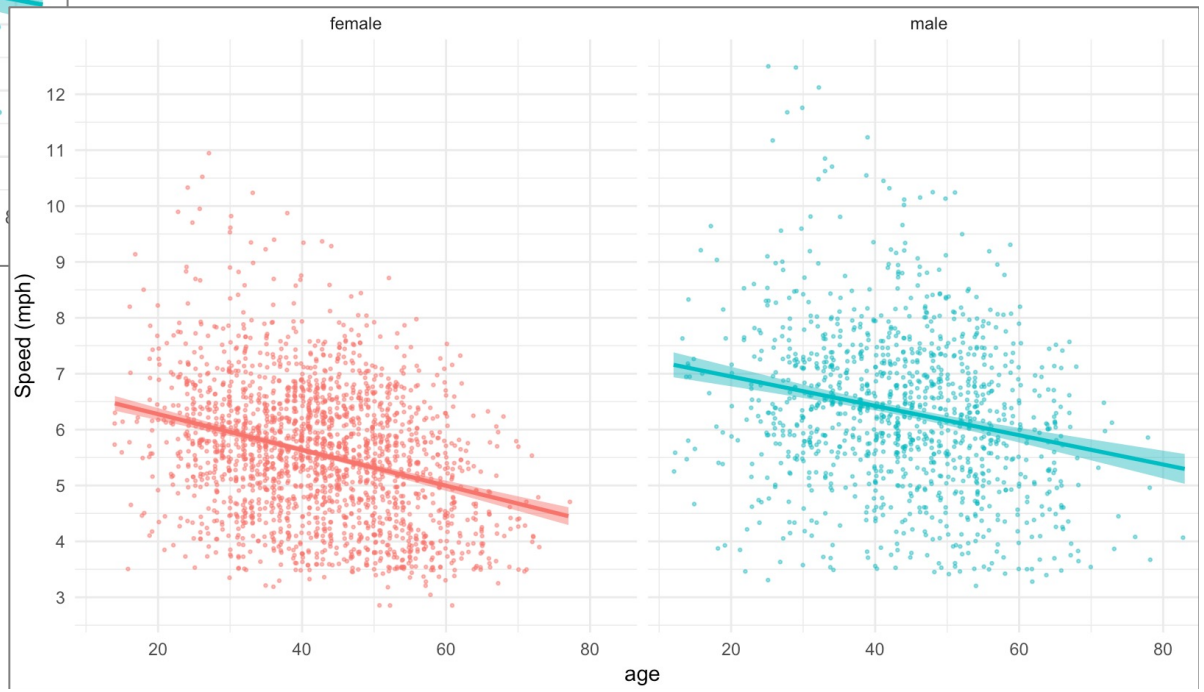
Fitted lines / conditional means.
Show error bars.
If y is smooth in x, show conditional means (as in here).
Bin width matters.

numerical ~ numerical + categorical

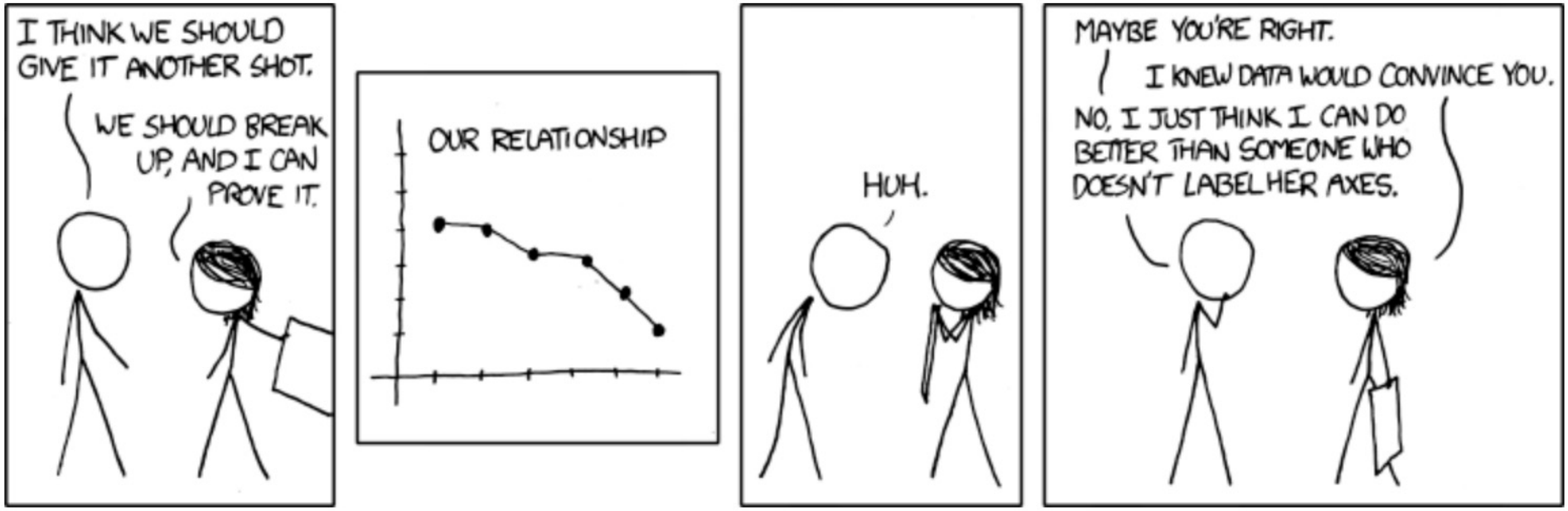
(1 numerical response, with numerical & categorical explanatory variable)



If scatterplots are important, split into facets with large n.
If line comparison is important, keep in same panel.



General pointers

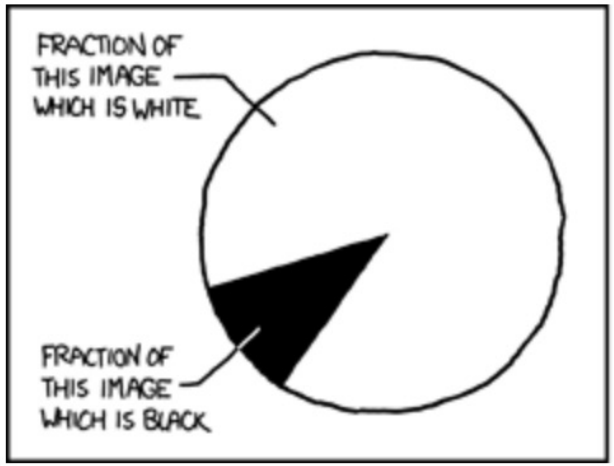


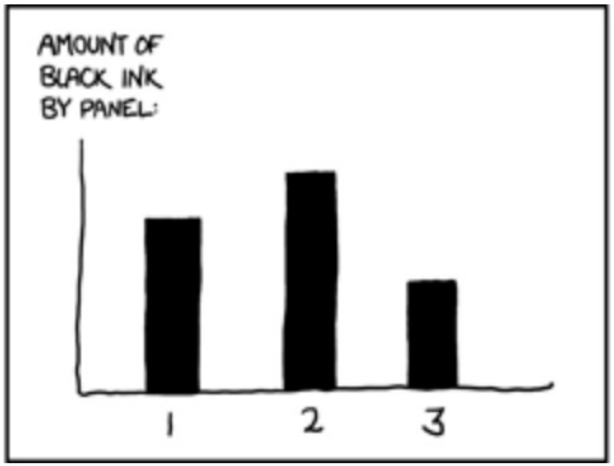
General pointers

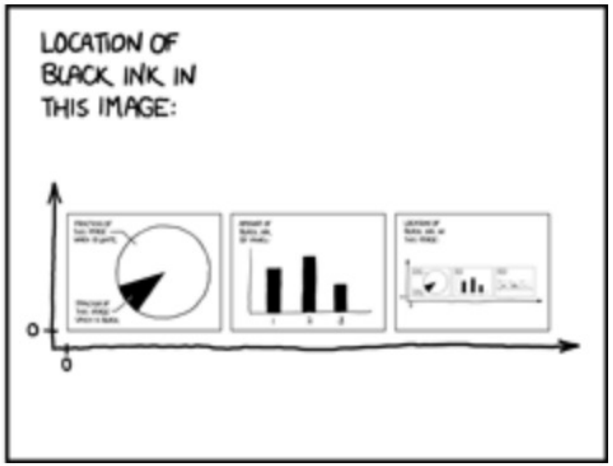
- Label your axes.
- Follow conventions
  - Explanatory variable on x axis.
  - Don’t get creative – respect variable types.
  - Don’t make visualization puzzles
- Convey information clearly, numerically
- Represent uncertainty! (distribution, error, confidence)
- Be wary of binning artifacts / thresholding
- Cool visualizations are not good science graphs

Graph priorities

- Interpretable without requiring caption or puzzle
  - Label all axes, legends, etc. intuitively.
  - No spiffy visualization puzzles.
- Facilitate quantitative interpretation and comparison
  - Easy to estimate numbers from graph
  - Be wary of binning/thresholding
- Permit inferential statistics by eye
  - Represent distribution/variability, uncertainty/error
- Follow conventions for the relationship/data presented
- Graphs should not waste ink and should look pretty


FRACTION OF
THIS IMAGE
WHICH IS WHITE
FRACTION OF
THIS IMAGE
WHICH IS BLACK


AMOUNT OF
BLACK INK
BY PANEL:
1
2
3


LOCATION OF
BLACK INK IN
THIS IMAGE:
0
0

- Visualization failure modes
- Cool vs informative visualizations
- Making a graph pretty
- ggplot: grammar of graphics
- Graphs for common types of data.

- Practice in R.

- More esoteric graph types / considerations

http://vulstats.ucsd.edu/data/duckworth-grit-scale-data/data-coded.csv

```
Observations: 4,270
Variables: 27
$ country           <chr> "RO", "US", "US", "K
$ surveyelapse      <int> 174, 120, 99, 5098,
$ education         <int> 4, 2, 1, 3, 4, 3, 3,
$ urban             <int> 3, 3, 2, 2, 2, 3, 2,
$ gender            <chr> "female", "female",
$ engnat            <int> 2, 1, 2, 1, 2, 2, 1,
$ age               <int> 28, 19, 16, 30, 38,
$ hand              <chr> "right", "right", "r
$ religion          <int> 1, 6, 0, 6, 2, 12, 3
$ orientation       <int> 1, 1, 1, 1, 1, 1, 1,
$ race              <chr> "white or indigenous
$ voted             <chr> "yes", "no", "no", "
$ married           <chr> "never", "never", "n
$ familysize        <int> 2, 3, 3, 6, 3, 1, 1,
$ operatingsystem   <chr> "Windows", "Macintos
$ browser           <chr> "Chrome", "Chrome",
$ screenw           <int> 1366, 1280, 1920, 16
$ screenh           <int> 768, 800, 1080, 900,
$ introelapse       <int> 69590, 33657, 95550,
$ testelapse        <int> 307, 134, 138, 4440,
$ extroversion      <int> 1, 10, -12, -11, -18
$ neuroticism       <int> 18, 30, 23, 6, 23, 2
$ agreeableness     <int> 19, 15, 9, 20, 9, 18
$ conscientiousness <int> 4, 11, 10, 20, 14, 1
$ openness          <int> 26, 24, 23, 22, 12,
$ grit              <int> 0, -5, -3, -16, -1,
$ vocabulary        <int> 10, 6, 11, 8, 4, 6,
```

Make plots to…

1. **Compare males and females on the big 5 personality traits:**
   - **extroversion**
   - **neuroticism**
   - **agreeableness**
   - **conscientiousness**
   - **openness**

2. **Evaluate the relationship between conscientiousness and grit?**
   - **does this relationship vary with sex?**

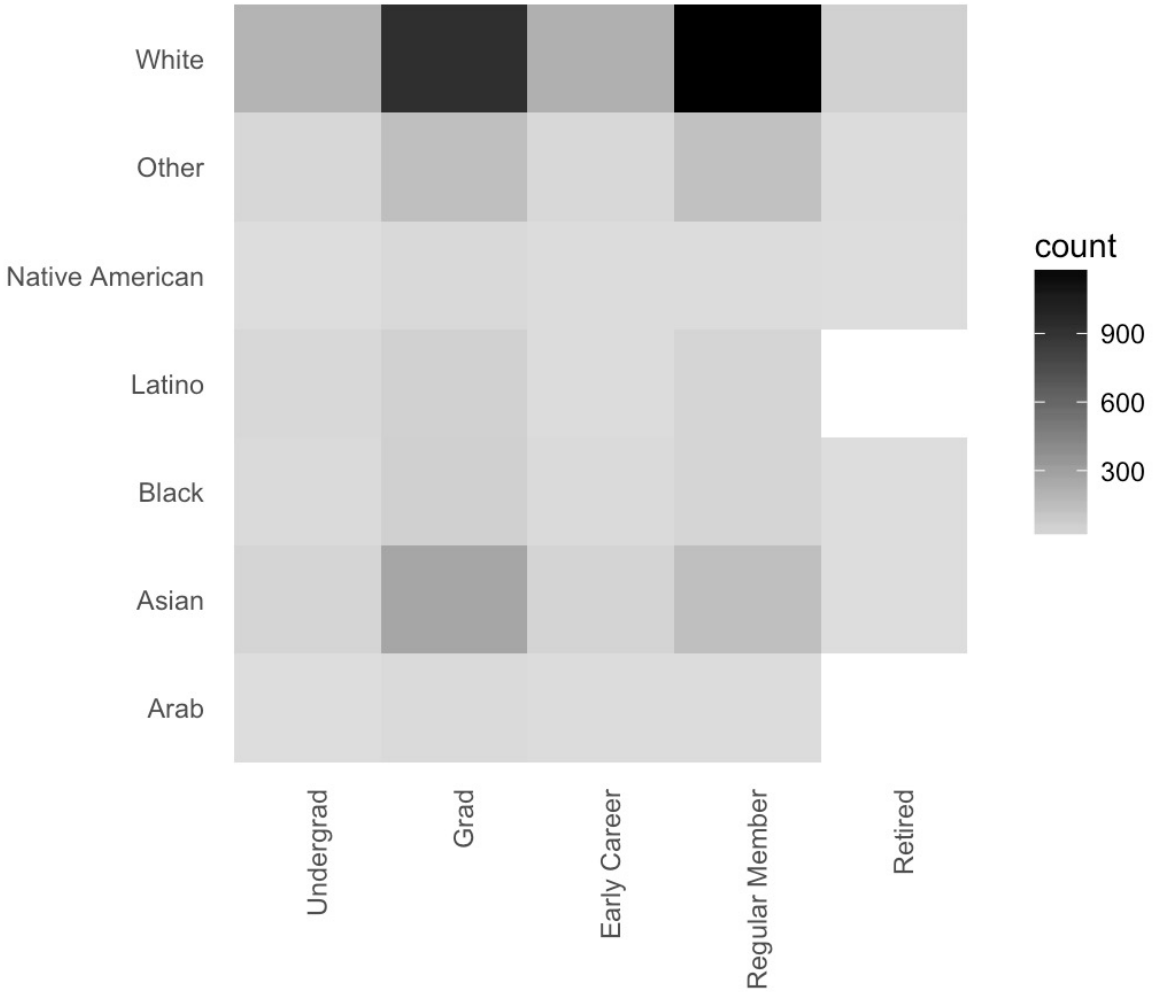
- Visualization failure modes
- Cool vs informative visualizations
- Making a graph pretty
- ggplot: grammar of graphics
- Graphs for common types of data.

- Practice in R.

- More esoteric graph types / considerations

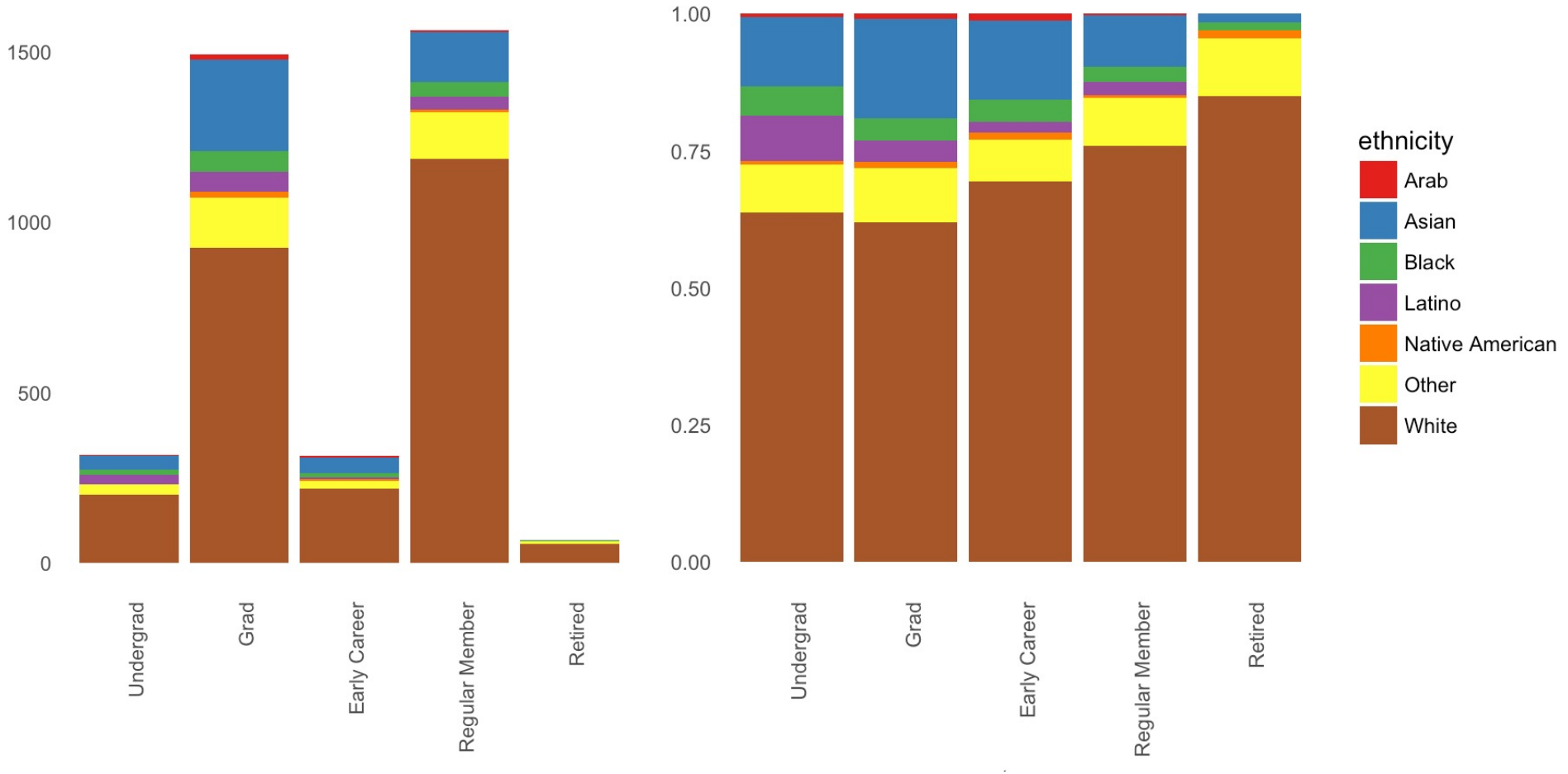
2 x categorical ~ 0

(2 categorical response variable, with 0 explanatory variables)



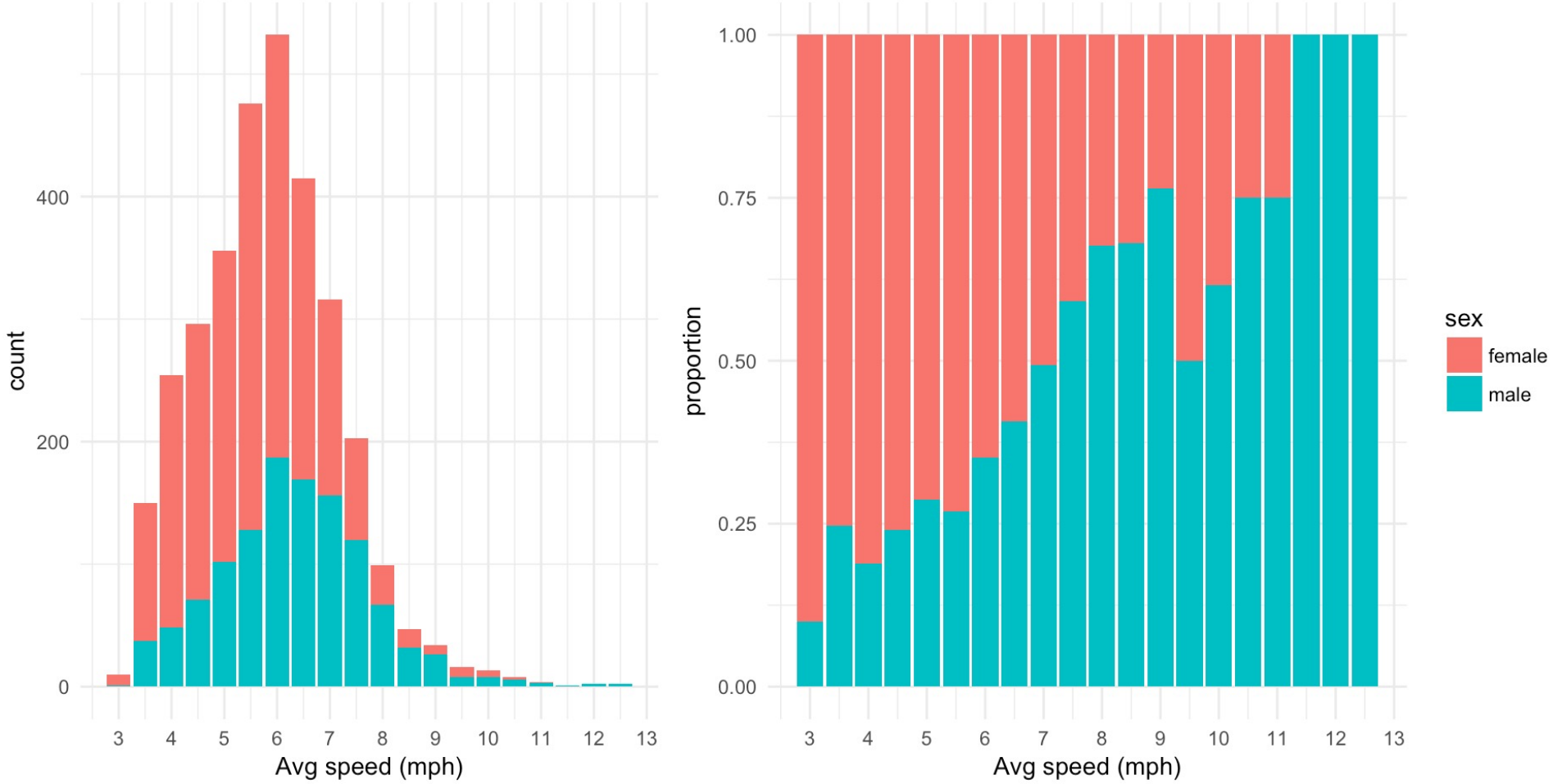
categorical ~ categorical

(1 categorical response variable, with 1 categorical explanatory variable)



categorical ~ numerical

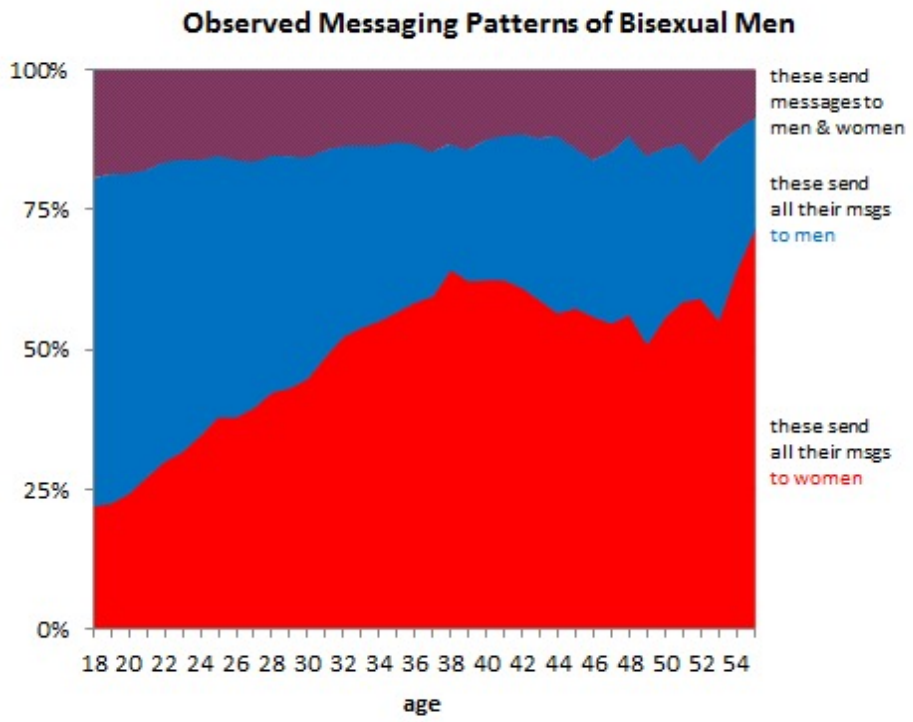
(1 categorical response variable, with 1 numerical explanatory variable)

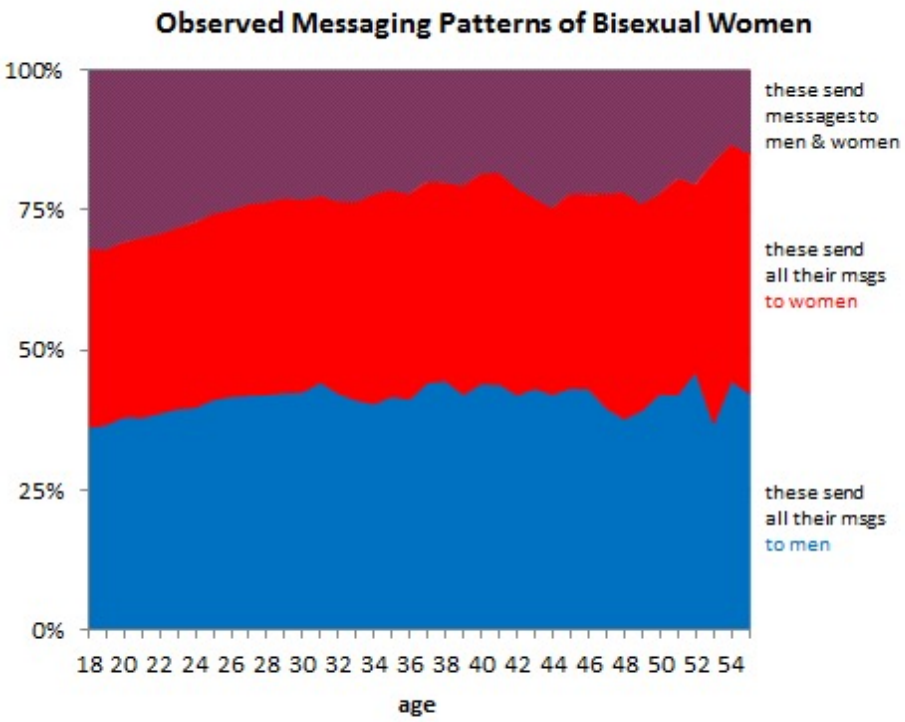


Stacked area charts. Generally, must round/bin numerical variable.
Stacked counts show the distribution of numerical variable.
Proportions show how categorical variable changes.

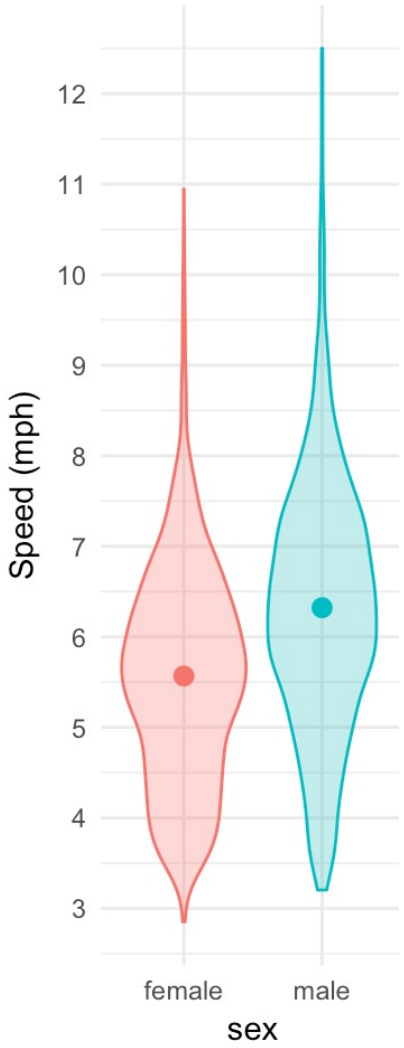
categorical ~ numerical

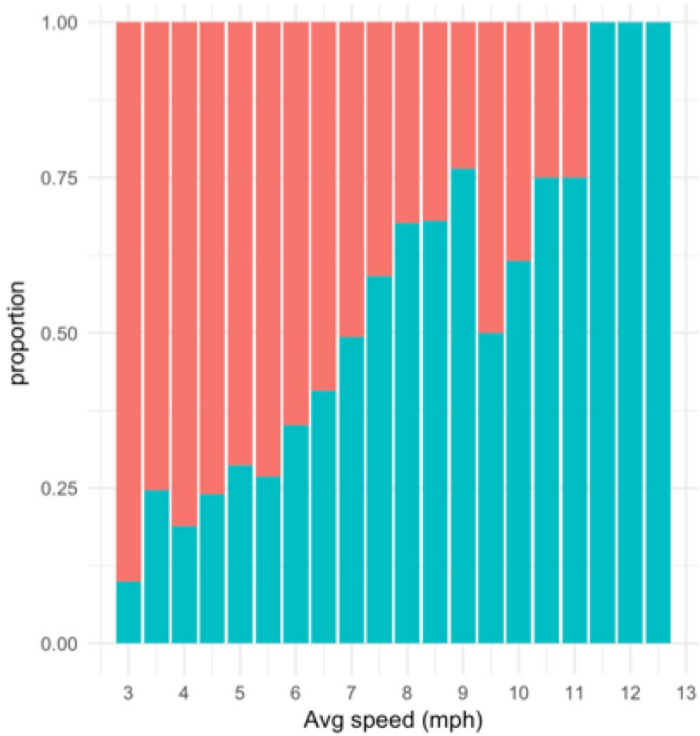
(with small n, binning must be very coarse; most useful with large n)





num. ~ cat. vs cat. ~ num.





Same data, but they invite different comparisons and interpretations.

numerical ~ 2 x categorical

(1 numerical response variable, with 1 categorical explanatory variable)

numerical ~ 2 x categorical

(1 numerical response variable, with 2 categorical explanatory variable)

Reply Rate By Race
male sender

	Asian - Female	Black - Female	Hispanic/Latin - Female	Indian - Female	Middle Eastern - Female	Native American - Female	Other - Female	Pacific Islander - Female	White - Female	
Asian - Male	22	34	22	18	27	31	30	23	21	22.2
Black - Male	17	28	19	21	21	27	24	25	21	21.7
Hispanic/Latin - Male	20	31	24	25	25	29	27	24	22	23.1
Indian - Male	20	34	19	18	23	27	28	24	20	20.8
Middle Eastern - Male	26	37	23	24	23	36	29	28	25	25.7
Native American - Male	26	34	26	27	32	35	32	29	27	27.8
Other - Male	25	32	26	21	23	34	32	22	26	26.8
Pacific Islander - Male	21	39	21	33	32	32	35	30	23	24.6
White - Male	29	38	30	30	28	34	33	29	29	29.2
	25.7	34.1	27.1	25.1	27.0	33.0	31.4	27.0	27.3	**27.6**

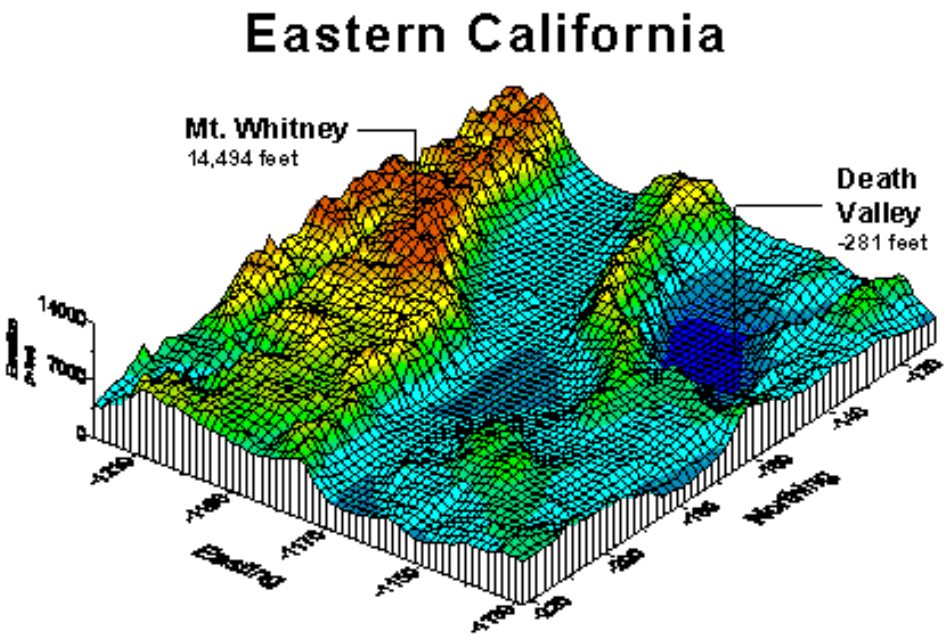
Reply Rates By Race
female sender

	Asian - Male	Black - Male	Hispanic/Latin - Male	Indian - Male	Middle Eastern - Male	Native American - Male	Other - Male	Pacific Islander - Male	White - Male	
Asian - Female	48	55	49	50	53	49	50	46	41	43.7
Black - Female	31	37	36	37	40	41	41	32	32	34.3
Hispanic/Latin - Female	51	46	48	45	50	45	48	48	40	42.5
Indian - Female	51	51	43	55	51	45	36	44	40	42.7
Middle Eastern - Female	51	55	54	63	56	63	52	48	47	49.5
Native American - Female	45	50	47	47	47	44	47	52	40	42.3
Other - Female	52	52	43	54	52	51	47	50	42	44.4
Pacific Islander - Female	51	57	49	35	60	53	50	46	44	46.0
White - Female	48	51	47	48	49	48	48	47	41	42.1
	47.3	46.9	46.4	48.2	49.7	47.3	47.5	46.2	40.5	**42.0**

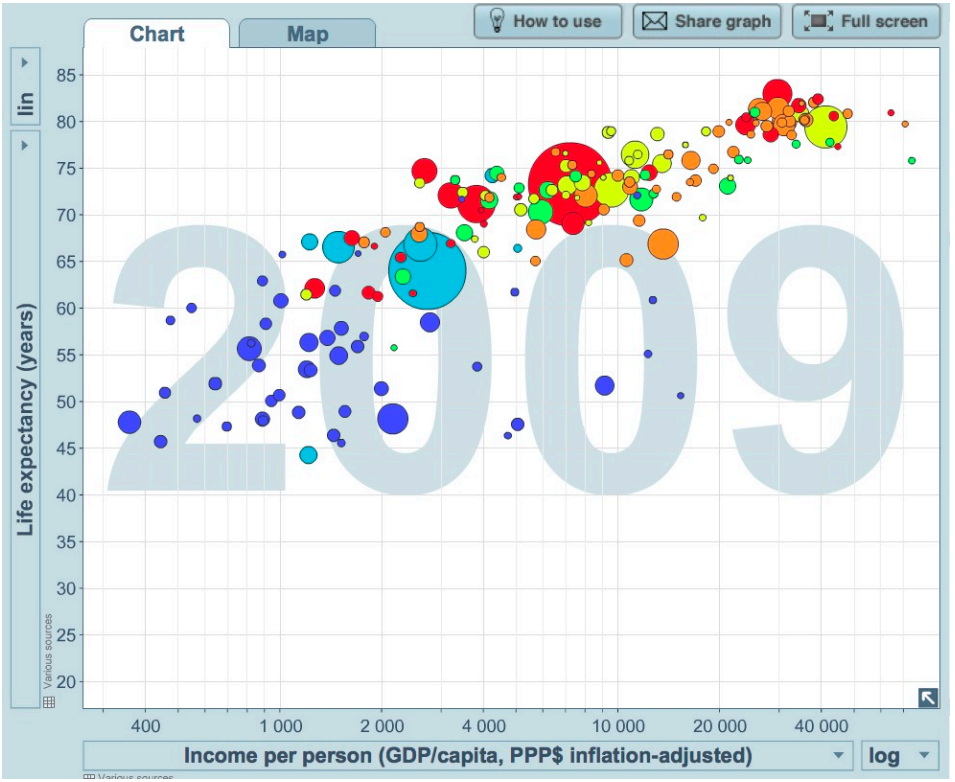
Notes: can't show error, so it better be tiny (as in here, with enormous n).
Which comparisons jump out is determined by number -> color mapping, so be careful.

numerical ~ 2 x numerical

(1 numerical response variable, with 2 numerical explanatory variable)



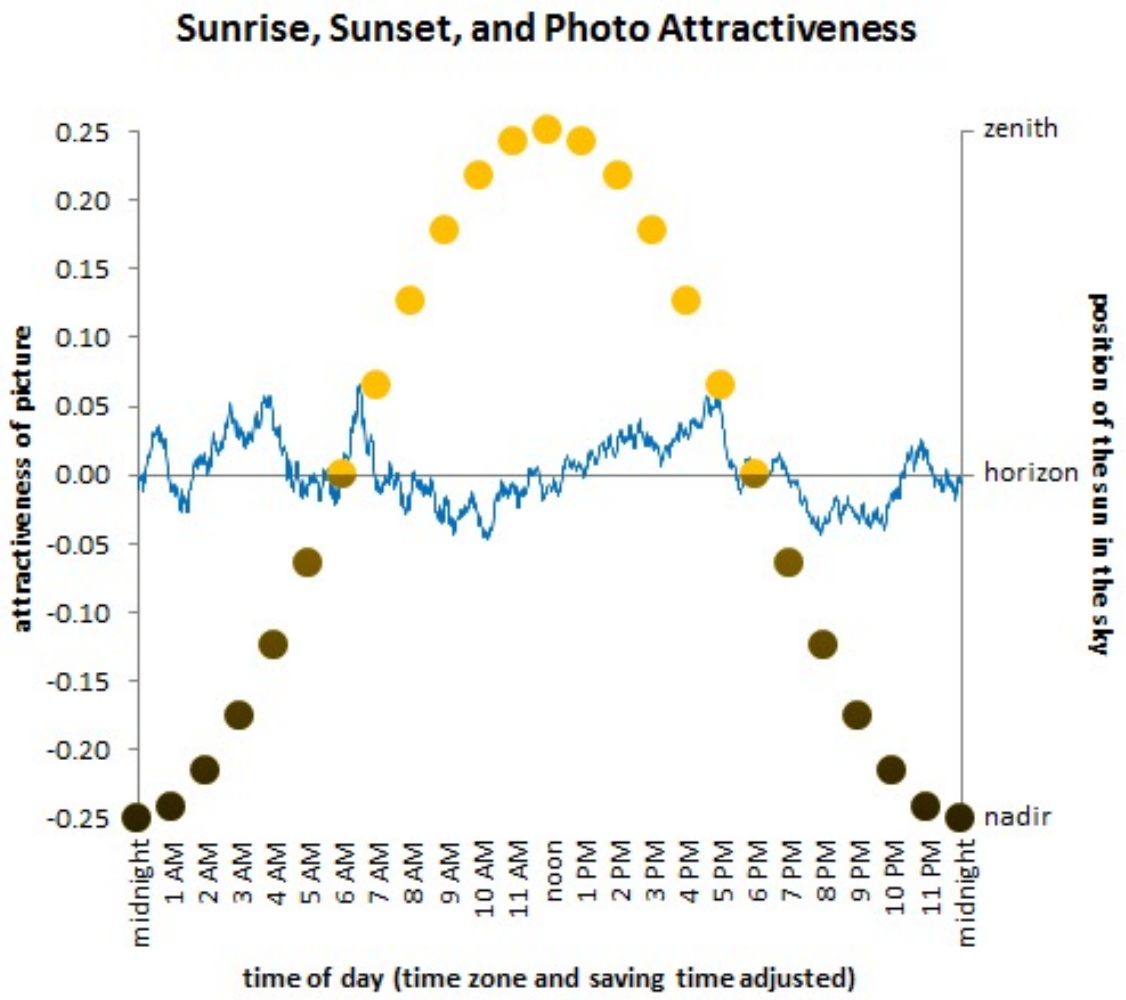
Heat map or surface plot
Generally your data need to be:
complete, smooth, abundant



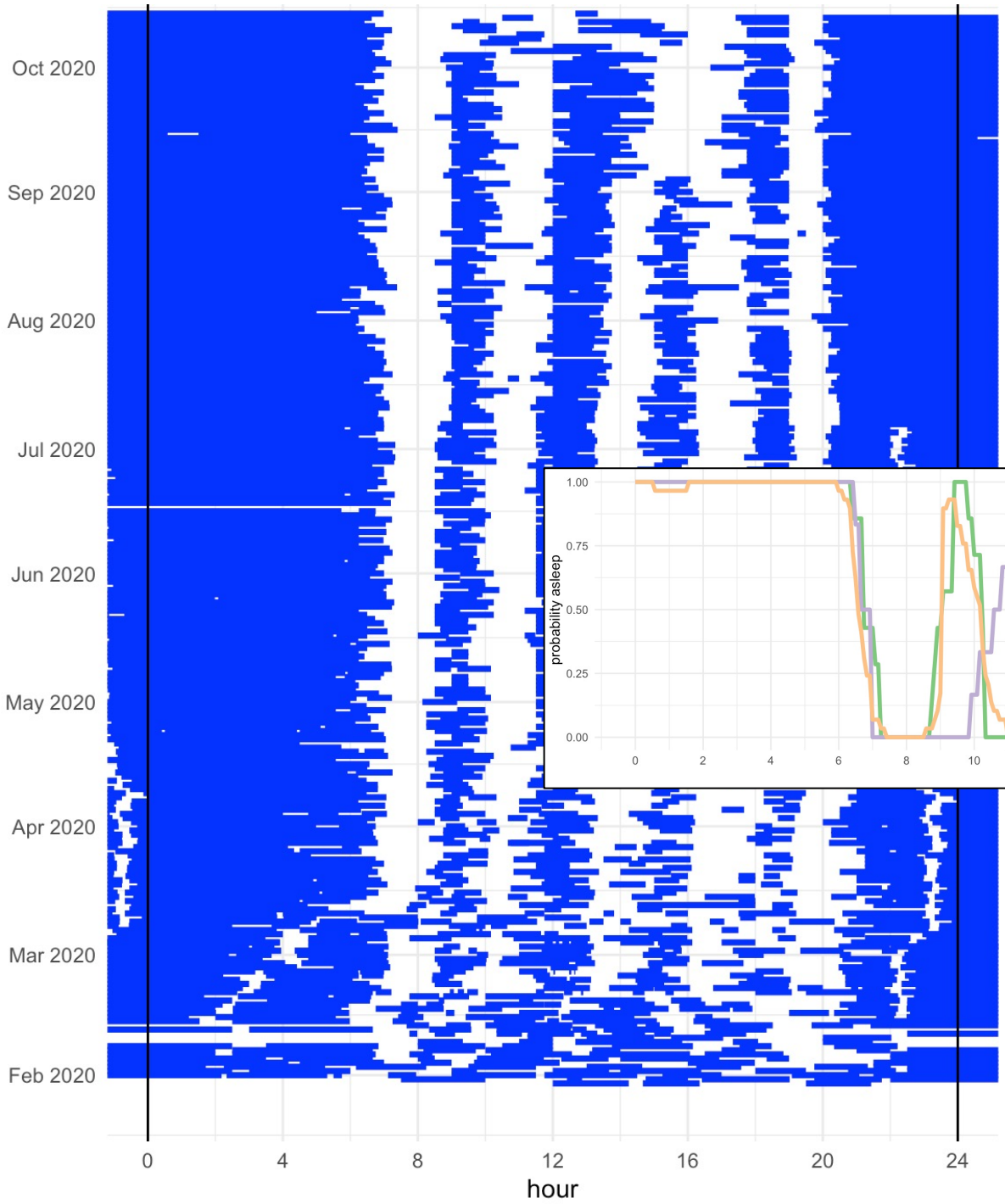
Bubble chart:
Comparisons across dot size are not easy, so that shouldn't be a very important variable.

2 x numerical ~ numerical

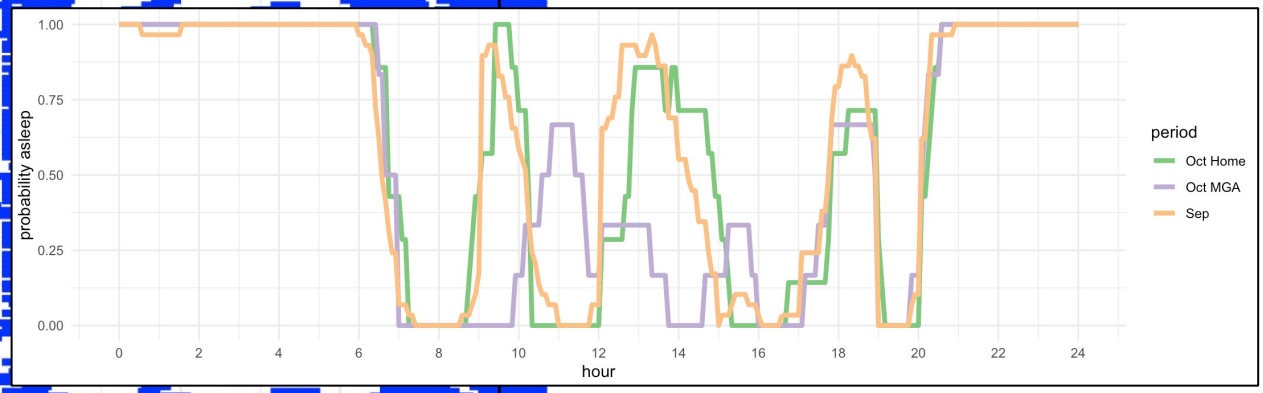
(2 numerical response variable, with 1 numerical explanatory variable)

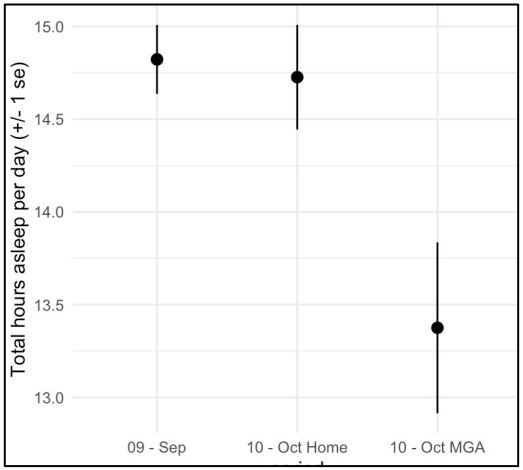


Double-axis plot.
Usually a terrible idea.


Oct 2020
Sep 2020
Aug 2020
Jul 2020
Jun 2020
May 2020
Apr 2020
Mar 2020
Feb 2020
0
4
8
12
16
20
24
hour


8
months


probability asleep
1.00
0.75
0.50
0.25
0.00
0
2
4
6
8
10
12
14
16
18
20
22
24
hour
period
Oct Home
Oct MGA
Sep


Total hours asleep per day (+/- 1 se)
15.0
14.5
14.0
13.5
13.0
09 - Sep
10 - Oct Home
10 - Oct MGA

- Visualization failure modes
- Cool vs informative visualizations
- Making a graph pretty
- ggplot: grammar of graphics
- Graphs for common types of data.

- Practice in R.

- More esoteric graph types / considerations

201A Schedule

vulstats.ucsd.edu/201a-schedule.html

code: summary stats, tidyverse

Week 2: Visualization

Readings

[notes]

R4DS: 2, 3

socviz: make a plot (the rest of this book may also be useful, bu

Tuesday

slides: visualization

code: ggplot

Wednesday

ggplot practice [code] [answers]

Thursday

https://vulstats.ucsd.edu/notes

vulstats.ucsd.edu/notes/visualizations.html

```
plot3 <- ggplot(cal1020, aes(x=sex, fill=sex, y=speed.mph))+
  geom_violin()+
  scale_y_continuous('Speed (mph)', breaks = seq(0, 15, by=1))+
  ggtitle('Violin')+
  theme_minimal()+
  theme(legend.position = 'none')
```

Box and whiskers plot

This is a visualization of a bunch of summary statistics of the distribution. By default, these summary statistics are: the median (middle line), the 25th and 75th percentile (edges of the box), 25th percentile - 1.5(IQR), and 75th percentile + 1.5(IQR) (the whiskers); and it shows the "outliers" (data points that are beyond those IQR intervals.

```
plot4 <- ggplot(cal1020, aes(x=sex, fill=sex, color=sex, y=speed.mph))+
  geom_boxplot(alpha=0.5, outlier.alpha = 0.1)+
  scale_y_continuous('Speed (mph)', breaks = seq(0, 15, by=1))+
  ggtitle('Boxplot')+
  theme_minimal()+
  theme(legend.position = 'none')
```

Overlayed densities

(here we flip the coordinates so we can picture them along side the other graphs)

```
plot5 <- ggplot(cal1020, aes(x=speed.mph, fill=sex, color=sex))+
  geom_density(alpha=0.5)+
  coord_flip()+
  scale_x_continuous('Speed (mph)', breaks = seq(0, 15, by=1))+
  ggtitle('Densities')+
  theme_minimal()+
  theme(legend.position = 'none')
```

Empirical cumulative distribution

(here we flip the coordinates so we can picture them along side the other graphs)

```
plot6 <- ggplot(cal1020, aes(x=speed.mph, fill=sex, color=sex))+
  stat_ecdf(geom='line', size=1, alpha=0.75)+
  coord_flip()+
  scale_x_continuous('Speed (mph)', breaks = seq(0, 15, by=1))+
  scale_y_continuous('CDF', breaks=c(0, 0.5, 1.0))+
  ggtitle('ECDF')+
  theme_minimal()+
  theme(legend.position = 'none')
```

Comparisons

```
gridExtra::grid.arrange(plot1, plot2, plot3, plot4, plot5, plot6, nrow=1)
```

The mean + standard error bar plot makes the implicit statistical comparison (t-test) easy to do by eye, but it obscures what the actual data look like (both the underlying variability, as well as it's messiness). The jittered plot is very faithful to the underlying data, but it's tricky to figure out how

Interactive plotting / manipulating

- Option 0: make particular kinds of graphs on request.
- Option 1: Molly's sleep data
- Option 2: babynames
- Option 3: personality and grit