



How Software Affects Humans' Conceptions of Data: A Case Study in R Syntaxes

Amelia McNamara

@AmeliaMN

University of St Thomas

St Paul, MN USA

“Under promise & over deliver.”

–McGraw Hill

~~“Under promise & over deliver.”~~

~~–McGraw Hill~~

“Over promise & under deliver.”


–me (this time)

A blue metal truss structure is shown against a light-colored wall. The structure consists of several horizontal and diagonal beams connected by joints. The wall behind it is covered in a pattern of shadows cast by the structure, creating a complex geometric design. The text "My general research framework" is centered over the image in a large, black, sans-serif font.

My general research framework



Key Attributes of a Modern Statistical Computing Tool

Amelia McNamara 

Statistical and Data Sciences, Smith College, Northampton, MA

ABSTRACT

In the 1990s, statisticians began thinking in a principled way about how computation could better support the learning and doing of statistics. Since then, the pace of software development has accelerated, advancements in computing and data science have moved the goalposts, and it is time to reassess. Software continues to be developed to help do and learn statistics, but there is little critical evaluation of the resulting tools, and no accepted framework with which to critique them. This article presents a set of attributes necessary for a modern statistical computing tool. The framework was designed to be broadly applicable to both novice and expert users, with a particular focus on making more supportive statistical computing environments. A modern statistical computing tool should be accessible, provide easy entry, privilege data as a first-order object, support exploratory and confirmatory analysis, allow for flexible plot creation, support randomization, be interactive, include inherent documentation, support narrative, publishing, and reproducibility, and be flexible to extensions. Ideally, all these attributes could be incorporated into one tool, supporting users at all levels, but a more reasonable goal is for tools designed for novices and professionals to “reach across the gap,” taking inspiration from each others’ strengths.

ARTICLE HISTORY

Received September 2016
Revised May 2018

KEYWORDS

Bootstrap; Data visualization;
Exploratory data analysis;
Randomization;
Reproducibility; Software
design; Software evaluation

1. Introduction

Tools shape the way we see the world, and statistical computing tools are starting to blur, and we believe this lowers the barrier

Table 1. Summary of attributes.

1. Accessibility
 2. Easy entry for novice users
 3. Data as a first-order persistent object
 4. Support for a cycle of exploratory and confirmatory analysis
 5. Flexible plot creation
 6. Support for randomization throughout
 7. Interactivity at every level
 8. Inherent documentation
 9. Simple support for narrative, publishing, and reproducibility
 10. Flexibility to build extensions
-

	Accessibility	Easy entry	Data as first-order object	EDA/CDA	Flexible plotting	Randomization	Interactivity	Inherent documentation	Narrative, publishing, reproducibility	Flexibility for extensions
Graphing calculators	*	✓								
Excel	*	✓					✓			
applets	*	✓				✓	✓			
TinkerPlots	*	✓		✓	✓	✓	✓	✓		
Fathom	*	✓		✓	✓	✓	✓	✓		
R	✓		✓	✓	✓	*	*		✓	✓
Python	✓		✓		*	*	*		✓	✓
SAS software			✓	✓		*			*	✓
Stata software			✓	✓		*			*	✓

Table 1: A summary of many currently-available tools for learning and doing statistics, and how they satisfy the attributes outlined in this paper. Asterisks indicate partial satisfaction of the attribute. For example, most tools are not accessible, either because of prohibitive cost or because they do not support disabled users. R and Python are free and can be used with adaptive technology. R, Python, SAS software, and Stata software get an asterisk for randomization because it is possible within the system, but difficult for novices. Similarly, R and Python can be used to create interactive graphics, but it is difficult, and SAS software and Stata software can be used to create reproducible reports, although it is difficult.

On the State of Computing in Statistics Education: Tools for Learning and for Doing. pre-print <http://bit.ly/StateOfComputingPreprint>

tools designed for **learning** statistics are typically:

- graphical
- interactive
- intuitive
- supportive of EDA

but:

- don't support reproducibility
- can't handle real data

The logo for Fathom, featuring the word "Fathom" in a bold, blue, sans-serif font. The letter "o" is replaced by a 3D-rendered yellow sphere with a grid pattern.The logo for TinkerPlots, with "Tinker" in orange and "Plots" in blue. A dotted line arches over the text, and a 3D orange sphere with blue dots is positioned between the two words.

StatKey

to accompany [Statistics: Unlocking the Power of Data](#)
by Lock, Lock, Lock, Lock, and Lock

Rossman/Chance Applet Collection

tools designed for **doing** statistics are typically:

- powerful
- flexible
- reproducible
- supportive of extensions

but:

- hard to get started using
- not interactive

The logo for Stata, consisting of the word "Stata" in white, bold, sans-serif font on a dark teal rectangular background.The logo for R, featuring a stylized blue letter "R" inside a grey, metallic-looking ring.The logo for SPSS, with "SPSS" in white, bold, sans-serif font on a red square background. Below it, the text "AN IBM COMPANY" is written in smaller white font.The logo for SAS, featuring a blue stylized "S" followed by "sas" in black, lowercase, sans-serif font. Below it is the tagline "THE POWER TO KNOW" in grey.The logo for MATLAB, with "MATLAB" in black, bold, sans-serif font. Below it is the tagline "The Language of Technical Computing" in a smaller, italicized font.



We need a bridge between the two



We need a bridge between the two

Could be software, or curriculum. Today, I'm focused on "curriculum."

mobilize

Home

Curriculum

IDS Partnership

Professional Development

About Us

Contact Us



<https://www.mobilizingcs.org/>

mobilize





```
Terminal Shell Edit View Window Help
amelia — R — 80x24
Last login: Wed Apr 16 15:39:29 on ttys000
Amelias-MacBook-Air:~ amelia$ R

R version 3.0.2 (2013-09-25) -- "Frisbee Sailing"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin10.8.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> █
```



```

R version 3.0.2 (2013-09-25) -- "Frisbee Sailing"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin10.8.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.62 (6558) x86_64-apple-darwin10.8.0]

[History restored from /Users/amelia/.Rapp.history]

> |
```



~/Dropbox/Documents/Teaching/101c - RStudio

```
49 mydata=data.frame(Y, X)
50
51 require(leaps)
52 ms2 = regsubsets(Y~poly(X,10), data=mydata, nvmax=10)
53 coef(ms2,4)
54 ms3 = regsubsets(Y~poly(X,10, raw=TRUE), data=mydata, nvmax=10)
55 coef(ms3,4)
56 ...
57
58 **Back to polynomial regression**
59 -----
60 So, the "raw" parameter determines whether you use orthogonal polynomials or raw polynomial. They work
61 out about the same when you do predictions, so it doesn't really matter which one you use.
62
63 Lets plot the fits. First, we need to do some predictions.
64 ```{r}
65 ageLim = range(Wage$age)
66 ageGrid = seq(from=ageLim[1], to=ageLim[2])
67
68 m2 = lm(wage~poly(age, 3), data=Wage)
69 m3 = lm(wage~poly(age, 2), data=Wage)
70 m4 = lm(wage~age, data=Wage)
71
72 predictions1 = predict(m1, newdata=list(age=ageGrid))
73 predictions1 = c(predictions1, predict(m2, newdata=list(age=ageGrid)))
74 predictions1 = c(predictions1, predict(m3, newdata=list(age=ageGrid)))
75 predictions1 = c(predictions1, predict(m4, newdata=list(age=ageGrid)))
76
77 predData = data.frame(ageGrid = rep(ageGrid, 4), preds=predictions1, poly=c(rep(4, length(ageGrid)), rep
78 (3, length(ageGrid)), rep(2, length(ageGrid)), rep(1, length(ageGrid))))
79 predData$poly = factor(predData$poly)
80
```

Environment History

Global Environment

Environment is empty

Files Plots Packages Help Viewer

R: Fitting Linear Models

Fitting Linear Models

Description

lm is used to fit linear models. It can be used to carry out regression, single stratum analysis of variance and analysis of covariance (although [aov](#) may provide a more convenient interface for these).

Usage

```
lm(formula, data, subset, weights, na.action,
   method = "qr", model = TRUE, x = FALSE, y = FALSE, qr = TRUE,
   singular.ok = TRUE, contrasts = NULL, offset, ...)
```

Arguments

formula	an object of class " formula " (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
data	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which lm is called.
subset	an optional vector specifying a subset of observations to be used in the fitting process.

Console ~/Dropbox/Documents/Teaching/101c/ ↻

```
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```


Having a better IDE reduces friction, but students still get stuck on syntax

Syntax is the set of rules that govern what code works and doesn't work in a programming language. Most programming languages offer one standardized syntax, but R allows package developers to specify their own syntax. As a result, there are many (equally valid) R syntaxes.

R Syntax Comparison :: CHEAT SHEET <http://bit.ly/R-syntax-sheet>

Dollar sign syntax

```
goal(data$x, data$y)
```

SUMMARY STATISTICS:

one continuous variable:
`mean(mtcars$mpg)`

one categorical variable:
`table(mtcars$cyl)`

two categorical variables:
`table(mtcars$cyl, mtcars$am)`

one continuous, one categorical:
`mean(mtcars$mpg[mtcars$cyl==4])`
`mean(mtcars$mpg[mtcars$cyl==6])`
`mean(mtcars$mpg[mtcars$cyl==8])`

PLOTTING:

one continuous variable:
`hist(mtcars$disp)`

`boxplot(mtcars$disp)`

one categorical variable:
`barplot(table(mtcars$cyl))`

two continuous variables:
`plot(mtcars$disp, mtcars$mpg)`

two categorical variables:
`mosaicplot(table(mtcars$am, mtcars$cyl))`

one continuous, one categorical:
`histogram(mtcars$disp[mtcars$cyl==4])`
`histogram(mtcars$disp[mtcars$cyl==6])`
`histogram(mtcars$disp[mtcars$cyl==8])`

`boxplot(mtcars$disp[mtcars$cyl==4])`
`boxplot(mtcars$disp[mtcars$cyl==6])`
`boxplot(mtcars$disp[mtcars$cyl==8])`

WRANGLING:

subsetting:
`mtcars[mtcars$mpg>30,]`

making a new variable:
`mtcars$efficient[mtcars$mpg>30] <- TRUE`
`mtcars$efficient[mtcars$mpg<30] <- FALSE`

Formula syntax

```
goal(y~x|z, data=data, group=w)
```

SUMMARY STATISTICS:

one continuous variable:
`mosaic::mean(~mpg, data=mtcars)`

one categorical variable:
`mosaic::tally(~cyl, data=mtcars)`

two categorical variables:
`mosaic::tally(cyl~am, data=mtcars)`

one continuous, one categorical:
`mosaic::mean(mpg~cyl, data=mtcars)`

tilde

PLOTTING:

one continuous variable:
`lattice::histogram(~disp, data=mtcars)`

`lattice::bwplot(~disp, data=mtcars)`

one categorical variable:
`mosaic::bargraph(~cyl, data=mtcars)`

two continuous variables:
`lattice::xyplot(mpg~disp, data=mtcars)`

two categorical variables:
`mosaic::bargraph(~am, data=mtcars, group=cyl)`

one continuous, one categorical:
`lattice::histogram(~disp|cyl, data=mtcars)`

`lattice::bwplot(cyl~disp, data=mtcars)`

The variety of R syntaxes give you many ways to “say” the same thing

read across the cheatsheet to see how different syntaxes approach the same problem

Tidyverse syntax

```
data %>% goal(x)
```

SUMMARY STATISTICS:

one continuous variable:
`mtcars %>% dplyr::summarize(mean(mpg))`

one categorical variable:
`mtcars %>% dplyr::group_by(cyl) %>% dplyr::summarize(n())`

two categorical variables:
`mtcars %>% dplyr::group_by(cyl, am) %>% dplyr::summarize(n())`

one continuous, one categorical:
`mtcars %>% dplyr::group_by(cyl) %>% dplyr::summarize(mean(mpg))`

PLOTTING:

one continuous variable:
`ggplot2::qplot(x=mpg, data=mtcars, geom = "histogram")`

`ggplot2::qplot(y=disp, x=1, data=mtcars, geom="boxplot")`

one categorical variable:
`ggplot2::qplot(x=cyl, data=mtcars, geom="bar")`

two continuous variables:
`ggplot2::qplot(x=disp, y=mpg, data=mtcars, geom="point")`

two categorical variables:
`ggplot2::qplot(x=factor(cyl), data=mtcars, geom="bar") + facet_grid(~am)`

one continuous, one categorical:
`ggplot2::qplot(x=disp, data=mtcars, geom = "histogram") + facet_grid(~cyl)`

`ggplot2::qplot(y=disp, x=factor(cyl), data=mtcars, geom="boxplot")`

WRANGLING:

subsetting:
`mtcars %>% dplyr::filter(mpg>30)`

making a new variable:
`mtcars <- mtcars %>% dplyr::mutate(efficient = if_else(mpg>30, TRUE, FALSE))`

the pipe

Summary statistics three ways



base

```
> mean(mtcars$mpg[mtcars$cyl==4])  
[1] 26.66364  
> mean(mtcars$mpg[mtcars$cyl==6])  
[1] 19.74286  
> mean(mtcars$mpg[mtcars$cyl==8])  
[1] 15.1
```

mosaic

```
> mean(mpg~cyl, data=mtcars)  
      4      6      8  
26.66364 19.74286 15.10000
```

dplyr

```
> mtcars %>%  
+   group_by(cyl) %>%  
+   summarize(mean(mpg))  
# A tibble: 3 x 2  
  cyl `mean(mpg)`  
  <dbl>      <dbl>  
1     4         26.7  
2     6         19.7  
3     8         15.1
```

Scatterplot three ways

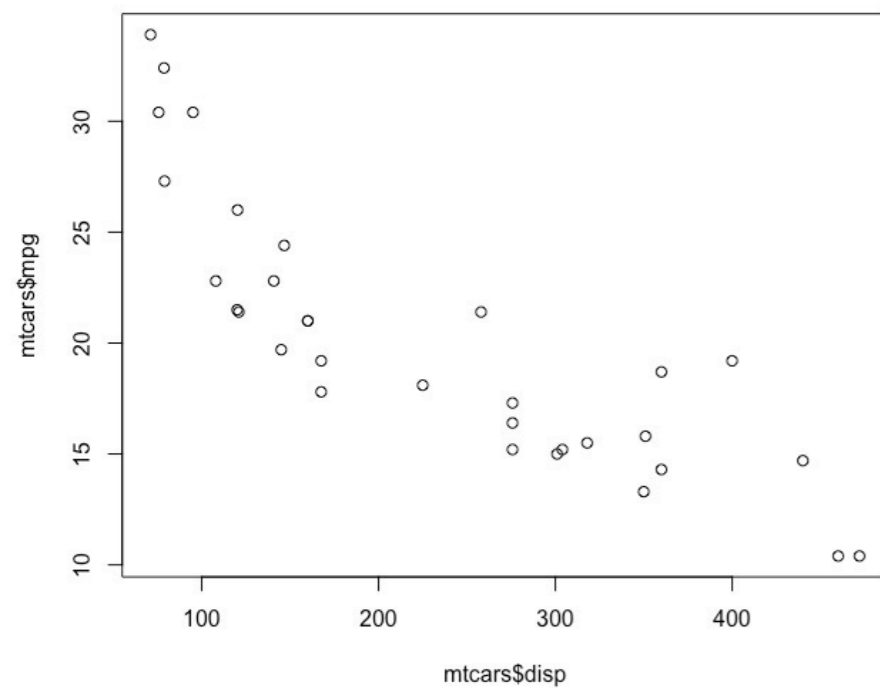


base

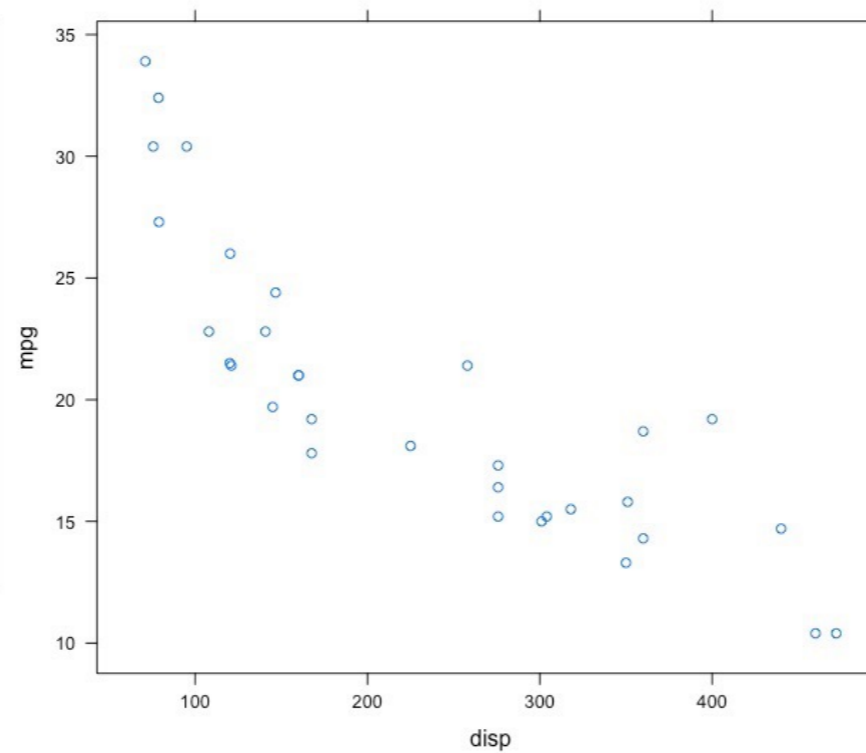
lattice

ggplot2

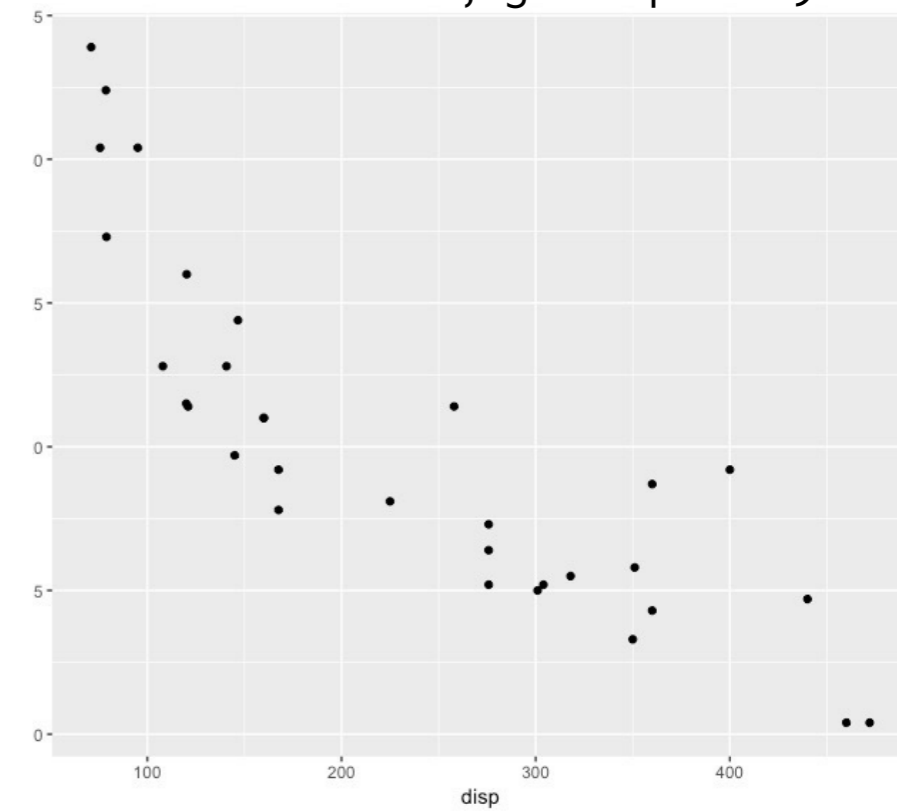
```
plot(mtcars$disp, mtcars$mpg)
```



```
xyplot(mpg~disp, data=mtcars)
```



```
qplot(x=disp, y=mpg,  
      data=mtcars, geom="point")
```

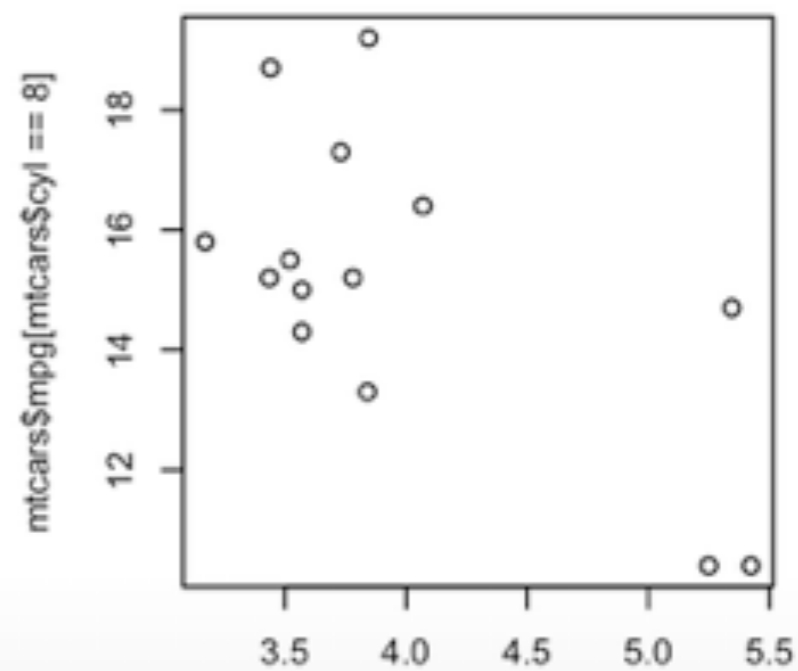
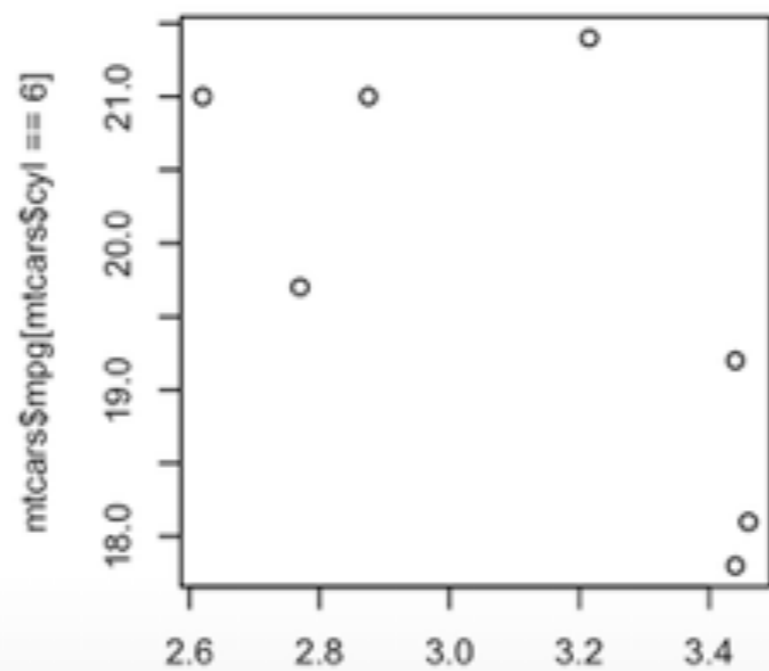
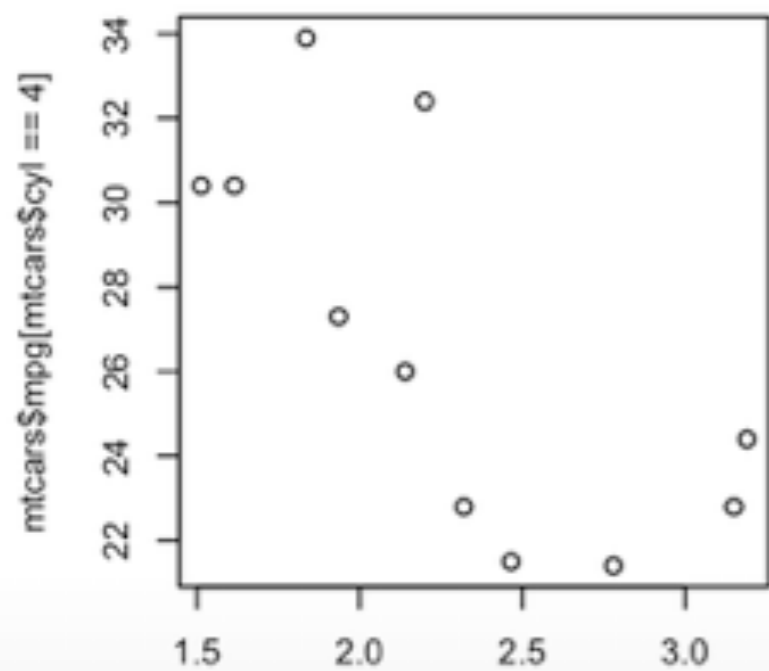
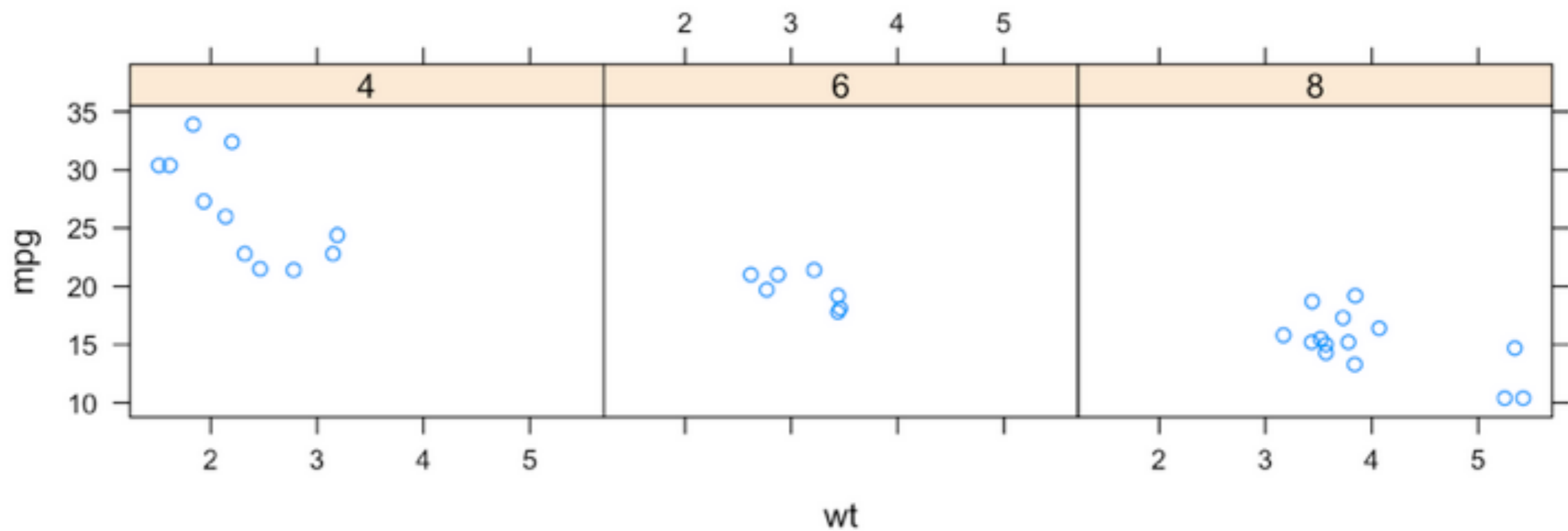




```
xyplot(mpg ~ wt | as.factor(cyl), data = mtcars)
```

vs.

```
par(mfrow = c(1,3))  
plot(mtcars$wt[mtcars$cyl == 4], mtcars$mpg[mtcars$cyl == 4])  
plot(mtcars$wt[mtcars$cyl == 6], mtcars$mpg[mtcars$cyl == 6])  
plot(mtcars$wt[mtcars$cyl == 8], mtcars$mpg[mtcars$cyl == 8])
```



`mtcars$wt[mtcars$cyl == 4]`

`mtcars$wt[mtcars$cyl == 6]`

`mtcars$wt[mtcars$cyl == 8]`

Formula syntax



- `lattice` graphics (or now, `ggformula`)
- `mosaic` statistics
- `mobilizr` additional functions

“Less volume, more creativity.”

–Mike McCarthy / mosaic package philosophy



A lot of times you end up putting in a lot more volume, because you are teaching fundamentals and you are teaching concepts that you need to put in, but you may not necessarily use because they are building blocks for other concepts and variations that will come off of that ... In the offseason you have a chance to take a step back and tailor it more specifically towards your team and towards your players."

Mike McCarthy, Head Coach, Green Bay Packers

"tidyverse" syntax



- Expects "tidy" data
- Uses the pipe (`%>%`, often pronounced "then") to string together verbs
- Verbs include `select()`, `filter()`, `mutate()`, `group_by()`

Next: [Vector arithmetic](#), Previous: [Simple manipulations numbers and vectors](#), Up: [Simple manipulations numbers and vectors](#) [[Contents](#)][[Index](#)]

2.1 Vectors and assignment

R operates on named *data structures*. The simplest such structure is the numeric *vector*, which is a single entity consisting of an ordered collection of numbers. To set up a vector named `x`, say, consisting of five numbers, namely 10.4, 5.6, 3.1, 6.4 and 21.7, use the R command

```
> x <- c(10.4, 5.6, 3.1, 6.4, 21.7)
```

This is an *assignment* statement using the *function* `c()` which in this context can take an arbitrary number of vector *arguments* and whose value is a vector got by concatenating its arguments end to end.⁷

A number occurring by itself in an expression is taken as a vector of length one.

Notice that the assignment operator (`<-`), which consists of the two characters `<` (“less than”) and `-` (“minus”) occurring strictly side-by-side and it ‘points’ to the object receiving the value of the expression. In most contexts the `=` operator can be used as an alternative.

Assignment can also be made using the function `assign()`. An equivalent way of making the same assignment as above is with:

```
> assign("x", c(10.4, 5.6, 3.1, 6.4, 21.7))
```

The usual operator, `<-`, can be thought of as a syntactic short-cut to this.

Assignments can also be made in the other direction, using the obvious change in the assignment operator. So the same assignment could be made using

```
> c(10.4, 5.6, 3.1, 6.4, 21.7) -> x
```



If an expression is used as a complete command, the value is printed *and lost*⁸. So now if we were to use the command

```
> 1/x
```



David Robinson

Chief Data Scientist at DataCamp, works in R and Python.

-  Email
-  Twitter
-  Github
-  Stack Overflow

Subscribe

Teach the tidyverse to beginners

A few years ago, I wrote a post [Don't teach built-in plotting to beginners \(teach ggplot2\)](#). I argued that ggplot2 was not an advanced approach meant for experts, but rather a suitable introduction to data visualization.

Many teachers suggest I'm overestimating their students: "No, see, my students are beginners...". If I push the point, they might insist I'm not understanding just how much of a beginner these students are, and emphasize they're looking to keep it simple and teach the basics, and that that students can get to the advanced methods later....

My claim is that this is precisely backwards. ggplot2 is easier to teach beginners, not harder, and makes constructing plots simpler, not more complicated.

I've [continued making this argument](#) in the years since, and I like to think our side is "winning." Even people that defend teaching base R plotting often treat it as an "underdog" opinion, which you never would have seen just a few years ago.

There's another debate that has popped up recently on Twitter and in conversations (many this week at the useR conference), about how to teach general R programming and data manipulation, and about the role of the ["tidyverse"](#) in such education. Just like ggplot2, this is a subject close to my heart.

Why I don't use ggplot2

Jeff Leek  2016/02/11

Some of my colleagues think of me as super data-sciencey compared to other academic statisticians. But one place I lose tons of street cred in the data science community is when I talk about ggplot2. For the 3 data type people on the planet who still don't know what that is, [ggplot2](#) is an R package/phenomenon for data visualization. It was created by Hadley Wickham, who is (in my opinion) perhaps the most important statistician/data scientist on the planet. It is one of the best maintained, most important, and really well done R packages. Hadley also supports R software like few other people on the planet.



But I don't use ggplot2 and I get nervous when other people do.

I get no end of grief for this from [Hilary and Roger](#) and especially from [drob](#), among many others. So I thought I would explain why and defend myself from the internet hordes. To understand why I don't use it, you have to understand the three cases where I use data visualization.

1. When creating exploratory graphics - graphs that are fast, not to be shown to anyone else and help me to explore a data set
2. When creating expository graphs - graphs that i want to put into a publication that have to be very carefully made.
3. When grading student data analyses.

Let's consider each case.

Teaching R to New Users - From tapply to the Tidyverse

 Roger Peng  2018/07/12

Abstract

The intentional ambiguity of the R language, inherited from the S language, is one of its defining features. Is it an interactive system for data analysis or is it a sophisticated programming language for software developers? The ability of R to cater to users who do not see themselves as programmers, but then allow them to slide gradually into programming, is an enduring quality of the language and is what has allowed it to gain significance over time. As the R community has grown in size and diversity, R's ability to match the needs of the community has similarly grown. However, this growth has raised interesting questions about R's value proposition today and how new users to R should be introduced to the system.

NOTE: A [video of this keynote address](#) is now available on YouTube if you would prefer to watch it instead.

Introduction

If we go to the [R web site](#) in order to discover what R is all about, the first sentence we see is

R is a free software environment for statistical computing and graphics.

I haven't been to the R web site in quite some time, but it struck me that the word "data" does not appear in that first sentence.

Lots of opinions!

Lots of opinions!

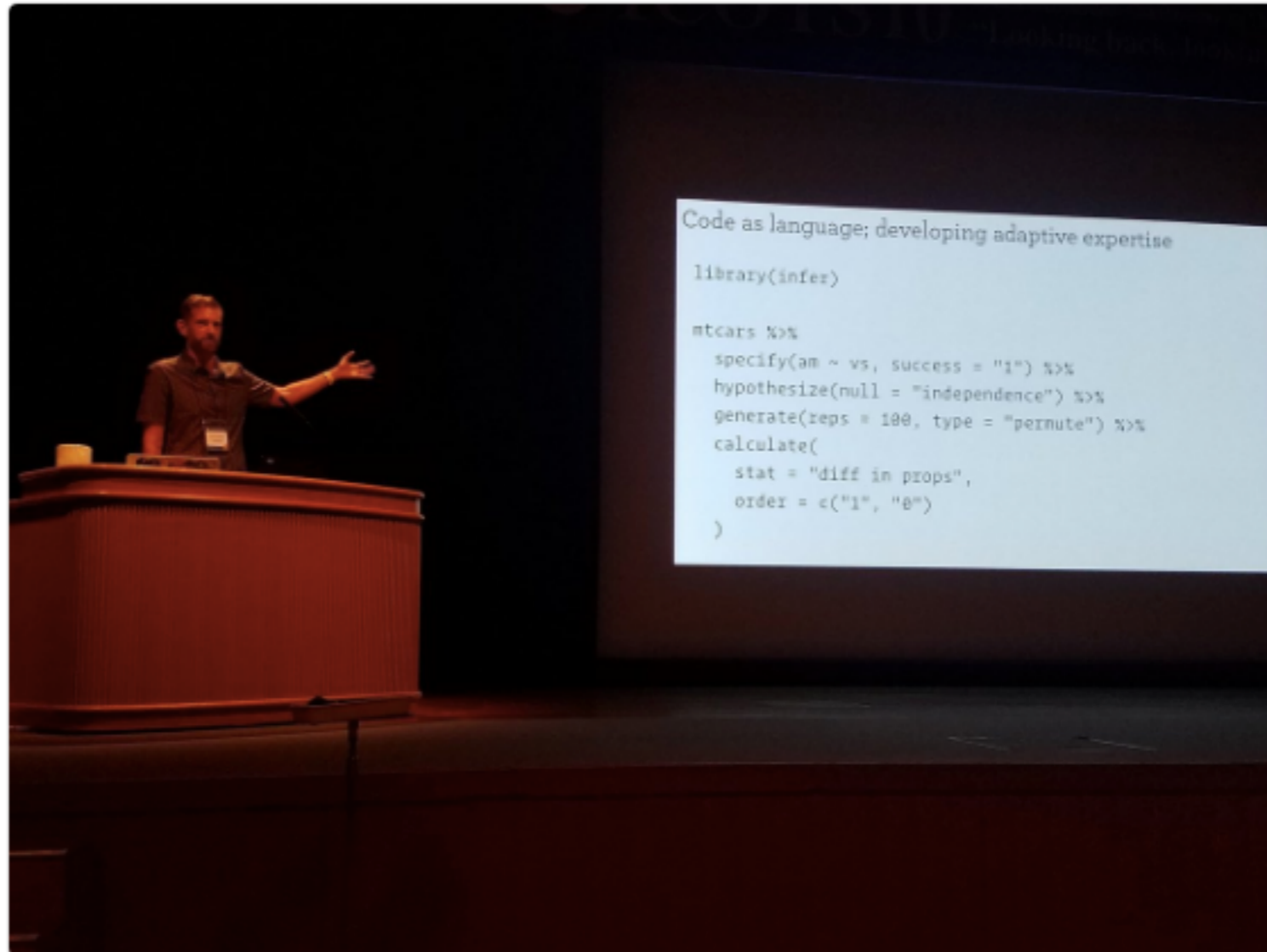
But, not much data...



Amelia McNamara
@AmeliaMN

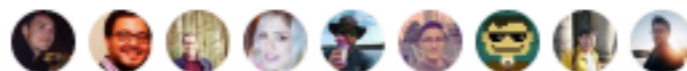


"Code as language" -@hadleywickham
#icots10 🥰🥰



12:26 AM - 10 Jul 2018

30 Retweets 129 Likes









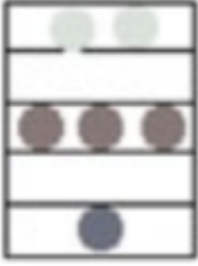


Lera Boroditsky | TEDWomen 2017

How language shapes the way we think

<http://bit.ly/LanguageShapesThinking>

Statistical research

View of Data	Perceptual Unit	Data Structure	Student Observation
Pointer	?		We said our favorite colors
Case Value			Juan likes red
Classifier			Three like red
Aggregate			Half like red

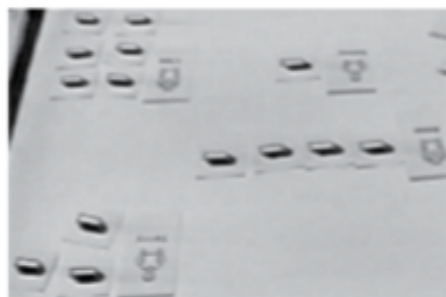
Konold, C. et al. "Data seen through different lenses."
Educational Studies in Mathematics, 2014.

The Development of Graph Understanding in the Mathematics Curriculum

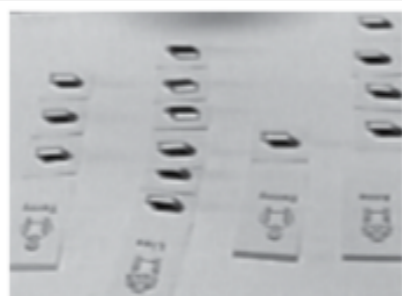
Jane Watson and Noleine Fitzallen



Level 1 (Prestructural/Idiosyncratic)



Level 2 (Unistructural)



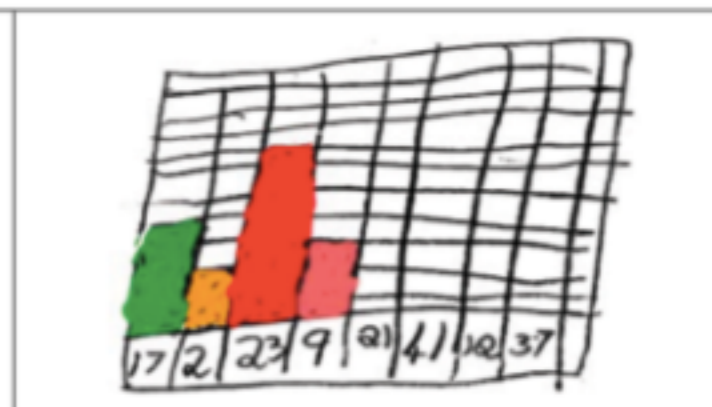
Level 3 (Multistructural)



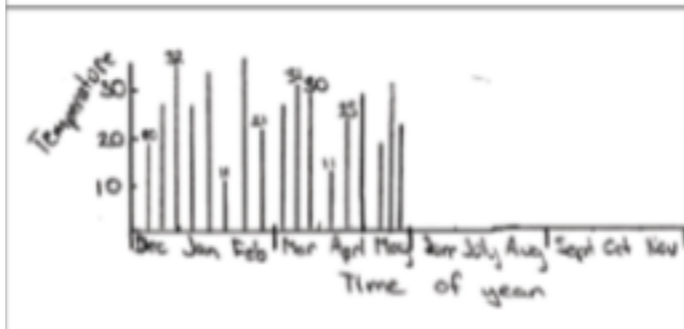
Level 4 (Relational)



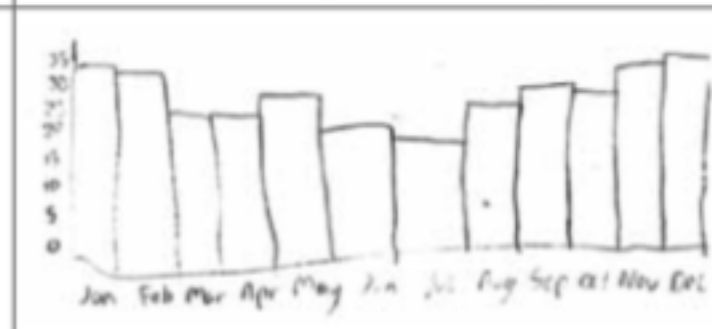
Level 1 (Prestructural)



Level 2 (Unistructural)



Level 3 (Multistructural)



Level 4 (Relational)

Contrasting emerging conceptions of distribution in contexts of error and natural variation

Richard Lehrer and Leona Schauble

K <small>Jordan Erik Tyler</small>	1st.	Arms	Nose	Eyes
Mostly no clothes	details on shirts	K stick	dot	dot
stick arms	most have hair	1 sideways □ □	slash (/)	round
naked	no ears	3 down	2-D (>)	round
eyes	long hair	5 down	3-D	football shape
looks like skeleton	Good -shaped heads			
no hair or dinosaur hair	bows or decorations			
triangle head	most have shoelaces			

Figure 6.2 Data structures for modeling classification of the age of the artist.

Exploring informal inference with interactive visualization software

Andee Rubin, James Hammerman and Cliff Konold

- types of variability:
 - variability due to errors of measurement
 - variability due to multiple causes
 - sample to sample variability

Reasoning about Data Analysis. Dani Ben-Zvi

- Identifying focus, beginning from irrelevant and local information
- Describing variability informally in raw data
- Formulating a statistical hypothesis that accounts for variability
- Accounting for variability when comparing groups using frequency tables
- Using center and spread measures to compare groups
- Modeling variability informally through handling outlying values
- Noticing and distinguishing the variability within and between the distributions in a graph.

Comparing Box Plot Distributions: A Teacher's Reasoning

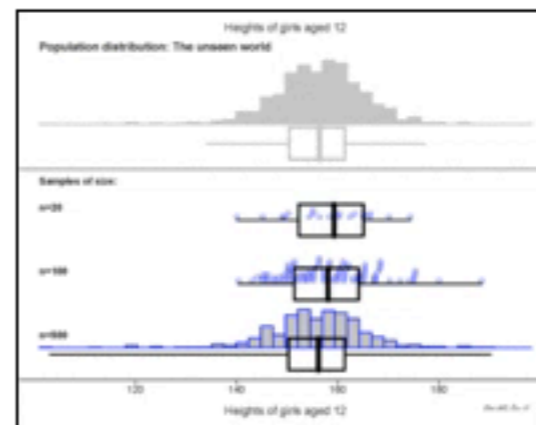
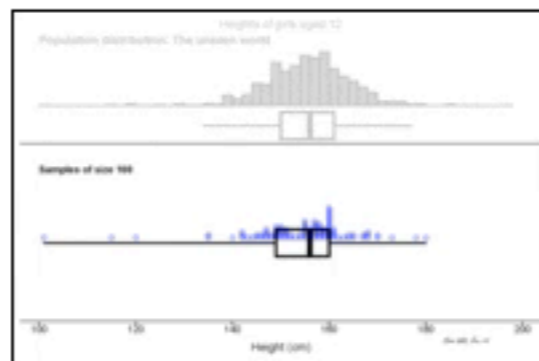
Maxine Pfannkuch

Teacher: Now I know the numbers are different, the males are *bigger* than the females, but *it's not that different*, it's not like one's 100 and the other's – you know? So, it's another contributing thing, men's stuff is *more spread out*. But *it's not massively different*, especially when you see it on the graph, you know, *it's not that different*, can you accept that? Okay, so at the moment, I've *got some conflicting kind of information*, right median – females are *more* clever, but when I look at the whole graph, the whole graph's *a bit more higher* for males. They're a little bit different in their spreads but you know, so *I'm still not ready to say yes* males have got a higher IQ than females.

Making the Call

Chris Wild, Nick Horton, Maxine Pfannkuch, Matt Regan

One Population



Animated Gifs

Click for: [n=30](#) [n=100](#) [n=300](#)

[View Full Size GIF](#)

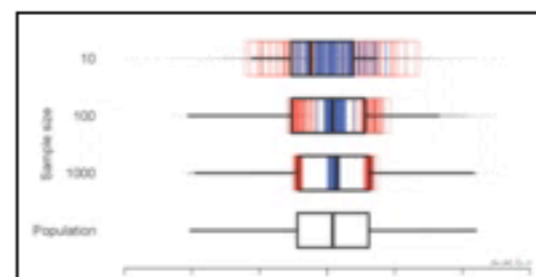
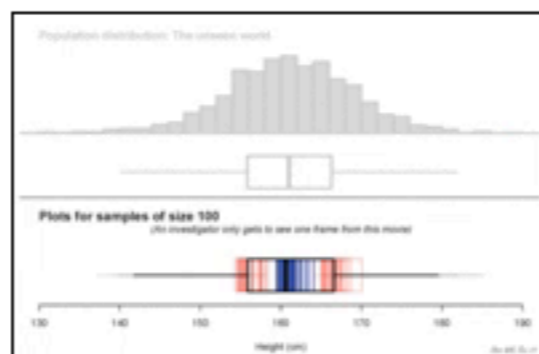
PDFs

Click for: [n=30](#) [n=100](#) [n=300](#)

[View Full size PDF](#)

1(a)

1(b)



Animated Gifs

Click for: [n=30](#) [n=100](#) [n=300](#)

[View full size GIF](#)

PDFs

Click for: [n=30](#) [n=100](#) [n=300](#)

[View full size PDF](#)

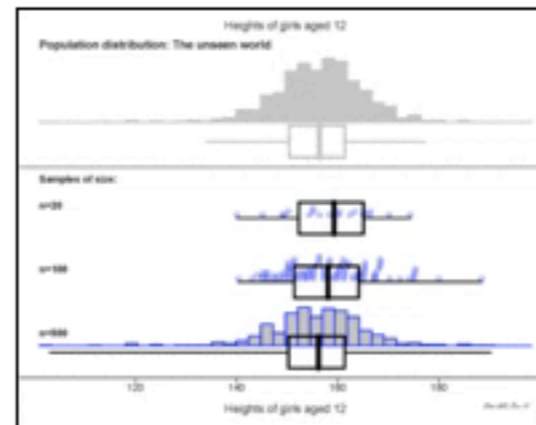
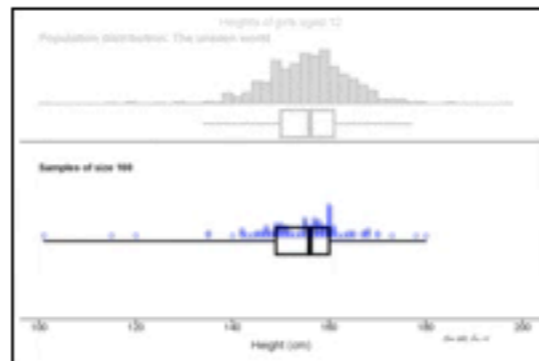
2(a)

2(b)

Making the Call

Chris Wild, Nick Horton, Maxine Pfannkuch, Matt Regan

One Population



Animated Gifs

Click for: [n=30](#) [n=100](#) [n=300](#)

[View Full Size GIF](#)

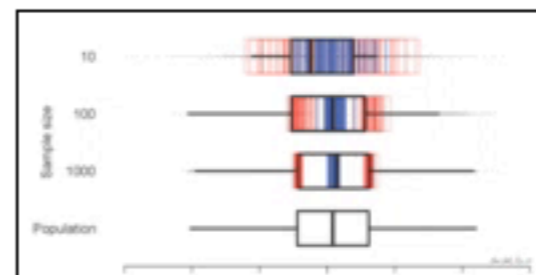
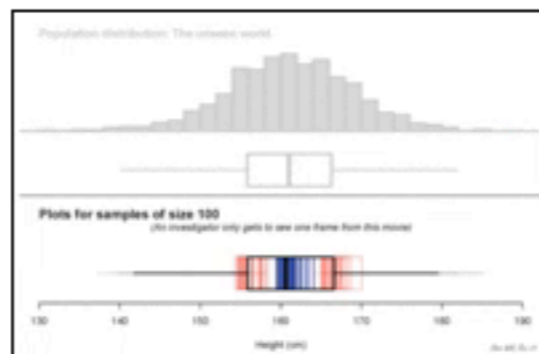
PDFs

Click for: [n=30](#) [n=100](#) [n=300](#)

[View Full size PDF](#)

1(a)

1(b)



Animated Gifs

Click for: [n=30](#) [n=100](#) [n=300](#)

[View full size GIF](#)

PDFs

Click for: [n=30](#) [n=100](#) [n=300](#)

[View full size PDF](#)

2(a)

2(b)

The Research Frontier: Where Technology Interacts with the Teaching and Learning of Data Analysis and Statistics

Susan Friel

- Technology is an amplifier of statistical power
- Technology as a reorganizer of physical and mental work
 - Shifting activity to higher cognitive levels
 - Changing the objects on which an activity may focus
 - Focusing on transforming and analyzing representations
 - Supporting a situated cognition perspective
 - Understanding statistical concepts dynamically through the use of graphics
 - Confronting "representative ambiguity"

Computer science
(education) research



<http://bit.ly/EvidenceAboutProgrammers>

SCHLOSS DAGSTUHL
Leibniz-Zentrum für Informatik



[About Dagstuhl](#)

Program

[Publications](#)

[Library](#)

[dblp](#)

You are here: [Program](#) » [Seminar Calendar](#) » [Seminar Homepage](#)

<https://www.dagstuhl.de/18061>

February 4 – 9 , 2018, Dagstuhl Seminar 18061

Evidence About Programmers for Programming Language Design

Organizers

Stefan Hanenberg (Universität Duisburg-Essen, DE)

Brad A. Myers (Carnegie Mellon University – Pittsburgh, US)

Bonita Sharif (Youngstown State University, US)

Andreas Stefik (Univ. of Nevada – Las Vegas, US)



[Dagstuhl Seminars](#)
[Dagstuhl Perspectives](#)
[GI-Dagstuhl Seminars](#)
[Events](#)
[Research Guests](#)
[Seminar Calendar](#)
[All Events](#)

Book exhibition

🔗 **Books from the participants of the current Seminar**

Book exhibition in the library, ground floor, during the seminar week.

Documentation

In the series **Dagstuhl Reports** each Dagstuhl Seminar and Dagstuhl Perspectives Workshop is documented. The **seminar organizers**, in cooperation with the **collector**, prepare a report that includes contributions from the **participants' talks** together with a summary of the seminar.

Download 📄 **overview leaflet (PDF)**.

Publications

Furthermore, a comprehensive peer-reviewed collection of research papers can be published in the series **Dagstuhl Follow-Ups**.

Dagstuhl's Impact



EDUCATOR OPINION VS BLACKBOX

Do educators predict Blackbox mistake frequency?
Is this affected by experience?

An Empirical Investigation into Programming Language Syntax Andreas Stefik and Susanna Siebert

For this article, **we conducted four empirical studies on programming language syntax** as part of a larger analysis into the, so called, programming language wars. We first present **two surveys** conducted with students on the intuitiveness of syntax, which we used to garner formative clues on what words and symbols might be easy for novices to understand. We followed up with **two studies on the accuracy rates of novices** using a total of six programming languages: Ruby, Java, Perl, Python, Randomo, and Quorum. Randomo was designed by randomly choosing some keywords from the ASCII table (a metaphorical placebo). To our surprise, **we found that languages using a more traditional C-style syntax** (both Perl and Java) **did not afford accuracy rates significantly higher than a language with randomly generated keywords, but** that **languages which deviate** (Quorum, Python, and Ruby) **did**.

<http://bit.ly/EmpiricalProgrammingSyntax>

This idea—that programming will provide exercise for the highest mental faculties, and that the cognitive development thus assured for programming will generalize or transfer to other content areas in the child’s life—is a great hope. Many elegant analyses offer reasons for this hope, although there is an important sense in which the arguments ring like the overzealous prescriptions for studying Latin in Victorian times.

Roy Pea, 1983

Psychology, linguistics,
...? research

Many analysts, one dataset: Making transparent how variations in analytical choices affect results

Silberzahn, R. and Uhlmann, E.L. and Martin, D.P. and Anselmi, P. and Aust, F. and Awtrey, E. and Bahník, Š. and Bai, F. and Bannard, C. and Bonnier, E. and Carlsson, R. and Cheung, F. and Christensen, G. and Clay, R. and Craig, M.A. and Dalla Rosa, A. and Dam, L. and Evans, M.H. and Flores Cervantes, I. and Fong, N. and Gamez-Djokic, M. and Glenz, A. and Gordon-McKeon, S. and Heaton, T.J. and Hederos, K. and Heene, M. and Hofelich Mohr, A.J. and Högden, F. and Hui, K. and Johannesson, M. and Kalodimos, J. and Kaszubowski, E. and Kennedy, D.M. and Lei, R. and Lindsay, T.A. and Liverani, S. and Madan, C.R. and Molden, D. and Molleman, E. and Morey, R.D. and Mulder, L.B. and Nijstad, B.R. and Pope, N.G. and Pope, B. and Prenoveau, J.M. and Rink, F. and Robusto, E. and Roderique, H. and Sandberg, A. and Schlüter, E. and Schönbrodt, F.D. and Sherman, M.F. and Sommer, S.A. and Sotak, K. and Spain, S. and Spörlein, C. and Stafford, T. and Stefanutti, L. and Tauber, S. and Ullrich, J. and Vianello, M. and Wagenmakers, E.-J. and Witkowiak, M. and Yoon, S. and Nosek, B.A. (2017) *Many analysts, one dataset: making transparent how variations in analytical choices affect results*. *Advances in Methods and Practices in Psychological Science* . ISSN 2515-2459 (In Press)

<https://psyarxiv.com/qkwst/>

Please give a name to the block:

Please shortly explain what you did in this block:

What were the other (if any) alternatives you considered in order to achieve the results of this block?

Please describe each alternative and explain its advantages and disadvantages. By clicking on "Add another alternative", you can add additional alternatives.

Alternative

Advantages of this alternative

Disadvantages of this alternative

Alternative

Advantages of this alternative

Disadvantages of this alternative

Why did you choose your option?

What preconditions should be fulfilled to successfully execute this block?


```
set.seed(170513)
n <- 200
d <- data.frame(a = rnorm(n))
d$b <- .4 * (d$a + rnorm(n))
head(d)
library(ggplot2)
ggplot(d, aes(a, b)) +
  geom_point() +
  theme_minimal()
library(ggplot2)
library(ggplot2)
ggplot2(d, aes(a, b)) +
  geom_point() +
  theme_minimal()
install.packages('ggplot')
library(ggplot2)
ggplot(d, aes(a, b)) +
  geom_point() +
  theme_minimal()
ggplot(d, aes(a, b)) +
  geom_point(shape = 16, size = 5) +
  theme_minimal()
ggplot(d, aes(a, b, color = a)) +
  geom_point(shape = 16, size = 5, show.legend = FALSE)
+
  theme_minimal()
d$pc <- predict(prcomp(~a+b, d))[, 1]
ggplot(d, aes(a, b, color = pc)) +
  geom_point(shape = 16, size = 5, show.legend = FALSE)
+
  theme_minimal()
ggplot(d, aes(a, b, color = pc)) +
  geom_point(shape = 16, size = 5, show.legend = FALSE)
+
  theme_minimal() +
  scale_color_gradient(low = "#0091ff", high = "#f0650e")
```

Crowdsourcing Data Analysis, Martin Schweinsberg et al

Crowdsourcing Data Analysis, Martin Schweinsberg et al

Edit block

Please give a name to the block:

Create different scatter plots

Please shortly explain *what* you did in this block:

I created a scatter plot to check the correlation between variable X and Y. In addition, I changed the color to improve the design of visualisation.

What were the other (if any) alternatives you considered in order to achieve the results of this block?

Please describe each alternative and explain its advantages and disadvantages. By clicking on "Add another alternative", you can add additional alternatives.

Alternative

Just calculating correlation coefficient Rho

```
set.seed(170513)
n <- 200
d <- data.frame(a = rnorm(n))
d$b <- .4 * (d$a + rnorm(n))
head(d)
library(ggplot2)
ggplot(d, aes(a, b)) +
  geom_point() +
  theme_minimal()
library(ggplot2)
library(ggplot2)
ggplot2(d, aes(a, b)) +
  geom_point() +
  theme_minimal()
install.packages('ggplot')
library(ggplot2)
ggplot(d, aes(a, b)) +
  geom_point() +
```

Features (A,B) and (C,D) are aggregated using Average and weighted sum respectively

Model 2 proposed

Parameters fitted using Logic 1

Visualization of results

Update model

Redo analysis

Analysis of performance on test set

Rerun Analysis on test set

Analysis of performance on training set

of this alternative

ADD ANOTHER ALTERNATIVE

REMOVE LAST ALTERNATIVE

Why did you choose your option?

I suspected that variable X and Y correlate because ...

What preconditions should be fulfilled to successfully execute this block?

Both, X and Y variables should be calculated based on the raw data using metric A

SHOW DIFF

DELETE BLOCK

LOAD FILES

SAVE

CANCEL

```
theme_minimal() +
scale_color_gradient(low = "#0091ff", high = "#f0650e")
```

R Syntax Comparison :: CHEAT SHEET <http://bit.ly/R-syntax-sheet>

Dollar sign syntax

```
goal(data$x, data$y)
```

SUMMARY STATISTICS:

one continuous variable:
`mean(mtcars$mpg)`

one categorical variable:
`table(mtcars$cyl)`

two categorical variables:
`table(mtcars$cyl, mtcars$am)`

one continuous, one categorical:
`mean(mtcars$mpg[mtcars$cyl==4])`
`mean(mtcars$mpg[mtcars$cyl==6])`
`mean(mtcars$mpg[mtcars$cyl==8])`

PLOTTING:

one continuous variable:
`hist(mtcars$disp)`

`boxplot(mtcars$disp)`

one categorical variable:
`barplot(table(mtcars$cyl))`

two continuous variables:
`plot(mtcars$disp, mtcars$mpg)`

two categorical variables:
`mosaicplot(table(mtcars$am, mtcars$cyl))`

one continuous, one categorical:
`histogram(mtcars$disp[mtcars$cyl==4])`
`histogram(mtcars$disp[mtcars$cyl==6])`
`histogram(mtcars$disp[mtcars$cyl==8])`

`boxplot(mtcars$disp[mtcars$cyl==4])`
`boxplot(mtcars$disp[mtcars$cyl==6])`
`boxplot(mtcars$disp[mtcars$cyl==8])`

WRANGLING:

subsetting:
`mtcars[mtcars$mpg>30,]`

making a new variable:
`mtcars$efficient[mtcars$mpg>30] <- TRUE`
`mtcars$efficient[mtcars$mpg<30] <- FALSE`

Formula syntax

```
goal(y~x|z, data=data, group=w)
```

SUMMARY STATISTICS:

one continuous variable:
`mosaic::mean(~mpg, data=mtcars)`

one categorical variable:
`mosaic::tally(~cyl, data=mtcars)`

two categorical variables:
`mosaic::tally(cyl~am, data=mtcars)`

one continuous, one categorical:
`mosaic::mean(mpg~cyl, data=mtcars)`

tilde

PLOTTING:

one continuous variable:
`lattice::histogram(~disp, data=mtcars)`

`lattice::bwplot(~disp, data=mtcars)`

one categorical variable:
`mosaic::bargraph(~cyl, data=mtcars)`

two continuous variables:
`lattice::xyplot(mpg~disp, data=mtcars)`

two categorical variables:
`mosaic::bargraph(~am, data=mtcars, group=cyl)`

one continuous, one categorical:
`lattice::histogram(~disp|cyl, data=mtcars)`

`lattice::bwplot(cyl~disp, data=mtcars)`

The variety of R syntaxes give you many ways to “say” the same thing

read across the cheatsheet to see how different syntaxes approach the same problem

Tidyverse syntax

```
data %>% goal(x)
```

SUMMARY STATISTICS:

one continuous variable:
`mtcars %>% dplyr::summarize(mean(mpg))`

one categorical variable:
`mtcars %>% dplyr::group_by(cyl) %>% dplyr::summarize(n())`

two categorical variables:
`mtcars %>% dplyr::group_by(cyl, am) %>% dplyr::summarize(n())`

one continuous, one categorical:
`mtcars %>% dplyr::group_by(cyl) %>% dplyr::summarize(mean(mpg))`

PLOTTING:

one continuous variable:
`ggplot2::qplot(x=mpg, data=mtcars, geom = "histogram")`

`ggplot2::qplot(y=disp, x=1, data=mtcars, geom="boxplot")`

one categorical variable:
`ggplot2::qplot(x=cyl, data=mtcars, geom="bar")`

two continuous variables:
`ggplot2::qplot(x=disp, y=mpg, data=mtcars, geom="point")`

two categorical variables:
`ggplot2::qplot(x=factor(cyl), data=mtcars, geom="bar") + facet_grid(~am)`

one continuous, one categorical:
`ggplot2::qplot(x=disp, data=mtcars, geom = "histogram") + facet_grid(~cyl)`

`ggplot2::qplot(y=disp, x=factor(cyl), data=mtcars, geom="boxplot")`

WRANGLING:

subsetting:
`mtcars %>% dplyr::filter(mpg>30)`

making a new variable:
`mtcars <- mtcars %>% dplyr::mutate(efficient = if_else(mpg>30, TRUE, FALSE))`

the pipe

Open questions

- Is it easier to pick up one syntax than another?
- How do different R syntaxes impact users' conceptions of data?
- Does a particular syntax have better transference?

Even more stuff to
study

Our path to better science in less time using open data science tools

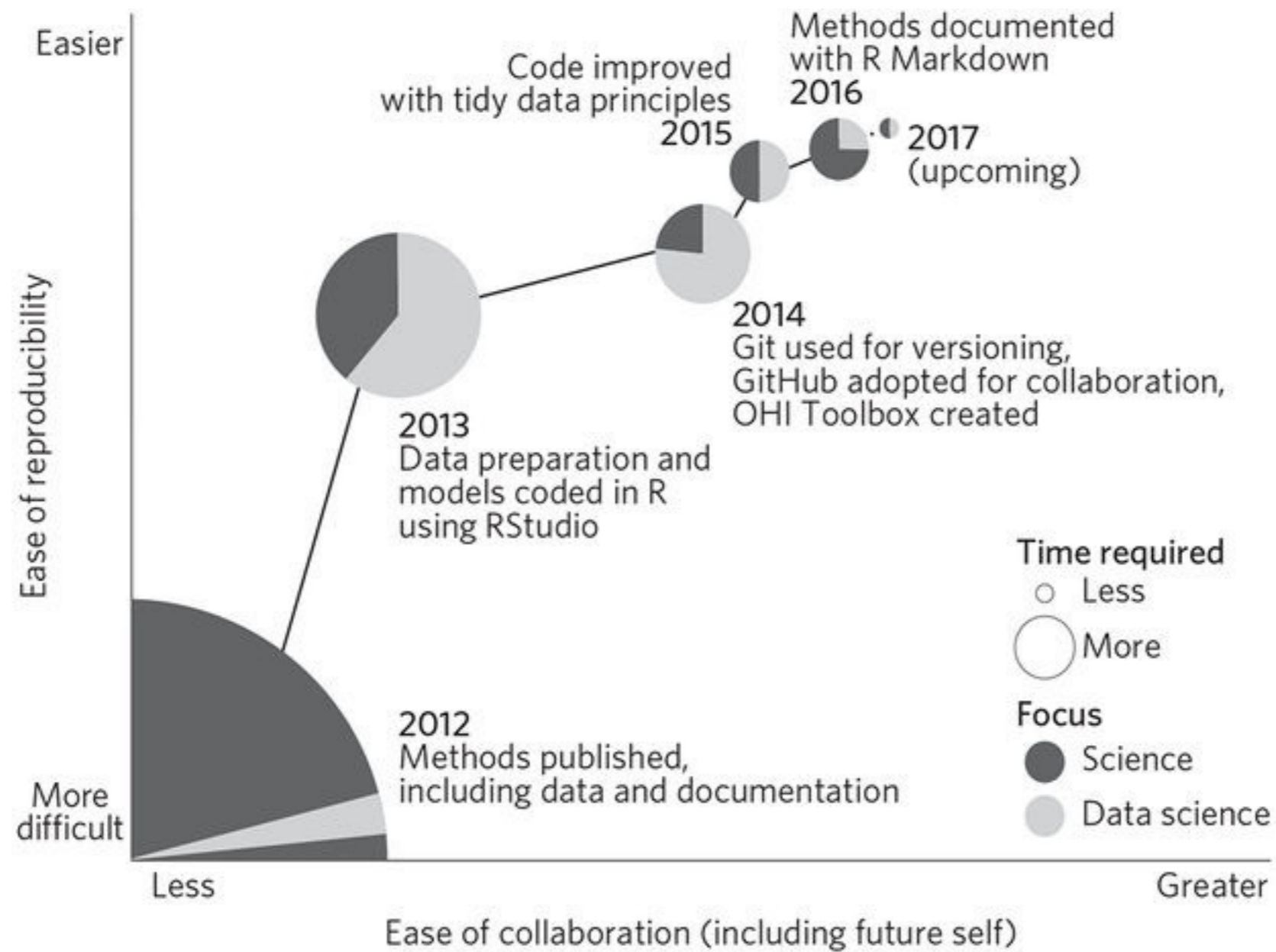
Julia Stewart Lowndes, et al. Nature Ecology & Evolution v1.
<https://www.nature.com/articles/s41559-017-0160>

We thought we were doing reproducible science. For the first global OHI assessment in 2012 we employed an approach to reproducibility that is standard to our field, which focused on scientific methods, not data science methods. Data from nearly one hundred sources were prepared manually—that is, without coding, typically in Microsoft Excel—which included organizing, transforming, rescaling, gap-filling and formatting data. Processing decisions were documented primarily within the Excel files themselves, e-mails, and Microsoft Word documents. We programmatically coded models and meticulously documented their development, (resulting in the 130-page supplemental materials), and upon publication we also made the model inputs (that is, prepared data and metadata) freely available to download. This level of documentation and transparency is beyond the norm for environmental science.

We decided to base our work in R and RStudio for coding and visualization, Git for version control, GitHub for collaboration, and a combination of GitHub and RStudio for organization, documentation, project management, online publishing, distribution and communication.

Data preparation: coding and documenting. Our first priority was to code all data preparation, create a standard format for final data layers, and do so using a single programmatic language, R. Code enables us to reproduce the full process of data preparation, from data download to final model inputs, and a single language makes it more practical for our team to learn and contribute collaboratively. We code in R and use RStudio to power our workflow because it has a user-friendly interface and built-in tools useful for coders of all skill levels, and, importantly, it can be configured with Git to directly sync with GitHub online (See ‘Collaboration’).

Sharing methods and instruction. We use R Markdown not only for data preparation but also for broader communication. R Markdown files can be generated into a wide variety of formatted outputs, including PDFs, slides, Microsoft Word documents, HTML files, books or full websites.



Our path to better science in less time using open data science tools. Julia Stewart Lowndes, et al. Nature Ecology & Evolution v1. <https://www.nature.com/articles/s41559-017-0160>

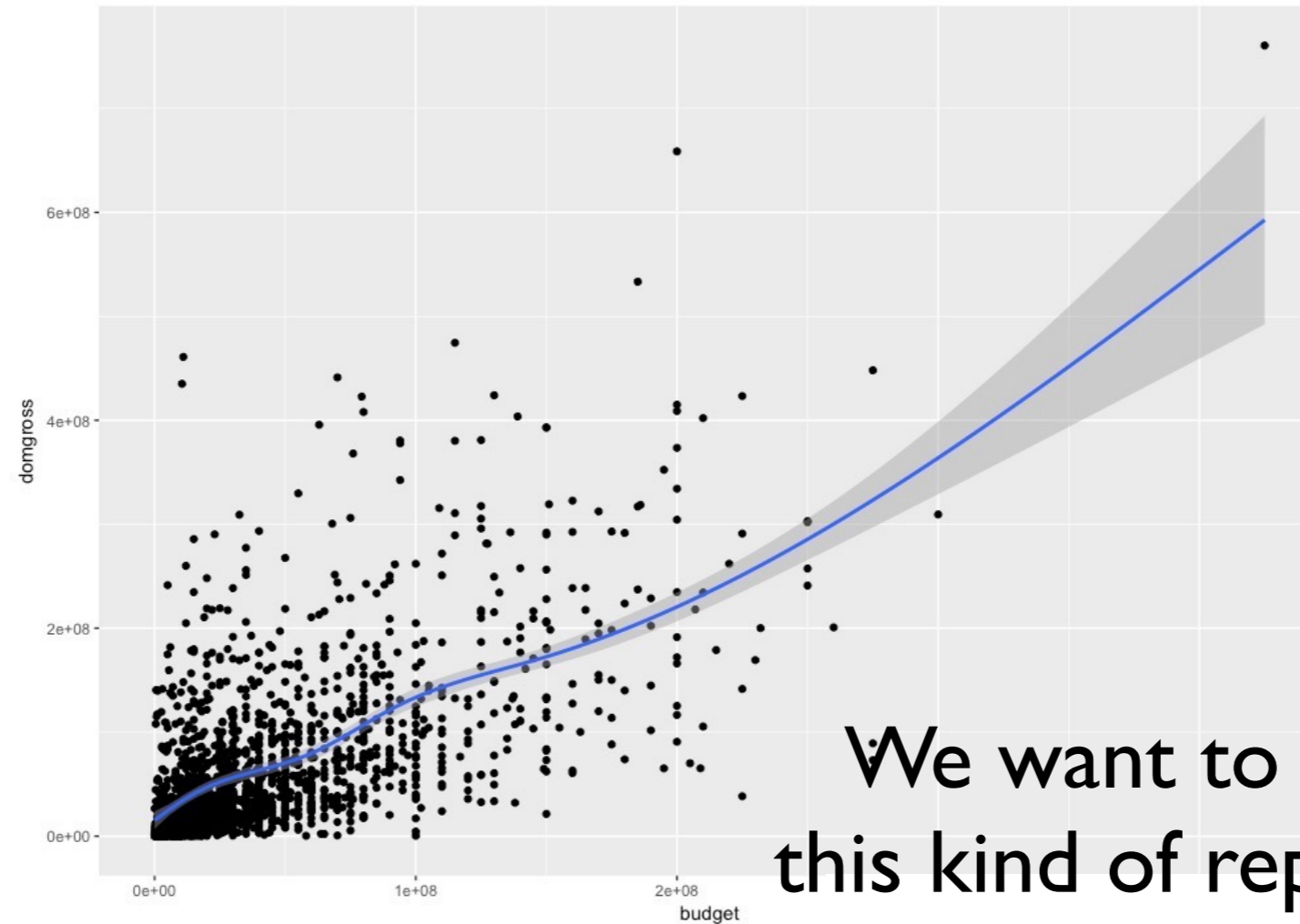


```
temp_long = coastal_fish_scores %>%
  select(monitoring_area,core_indicator,taxa, score1,score2,score3) %>%
  group_by(monitoring_area, core_indicator,taxa) %>%
  gather(score_type,score,score1,score2,score3)%>%
  ungroup()

slope2 = slope2 %>%
  group_by(basin_name, core_indicator)%>%
  mutate(slope_mean_basin_indicator = mean(slope_mean_basin_taxa))%>%
  ungroup()

basin_n_obs = coastal_fish_scores_long %>%
  filter(score_type=="score1") %>% ## only select 1 score t
  select(Basin_HOLAS)%>%
  count(Basin_HOLAS)
```

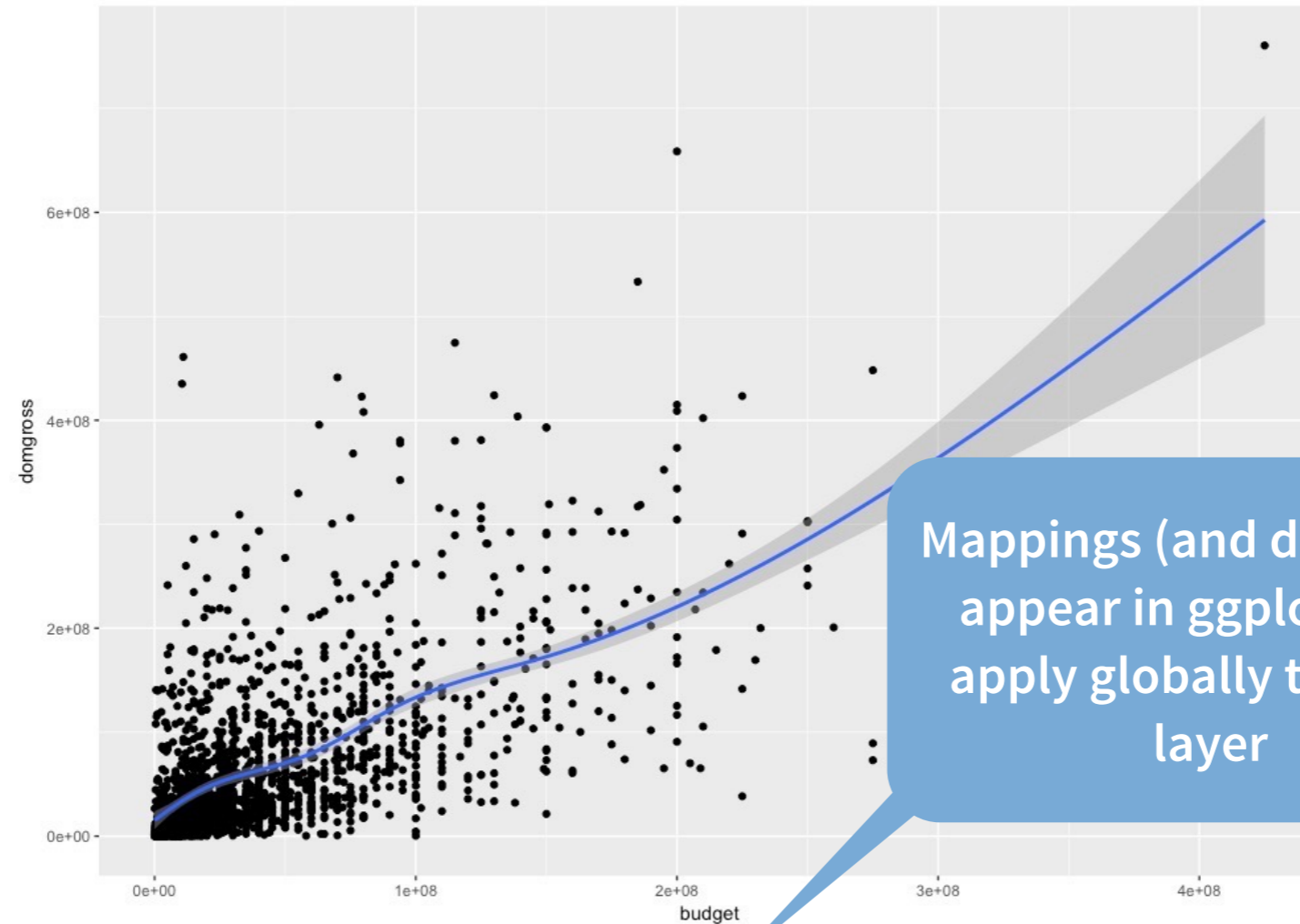
syntax within ggplot2



We want to avoid
this kind of repetition
in programming

```
ggplot(data = bechdel) +  
  geom_point(mapping = aes(x = budget, y = domgross)) +  
  geom_smooth(mapping = aes(x = budget, y = domgross))
```

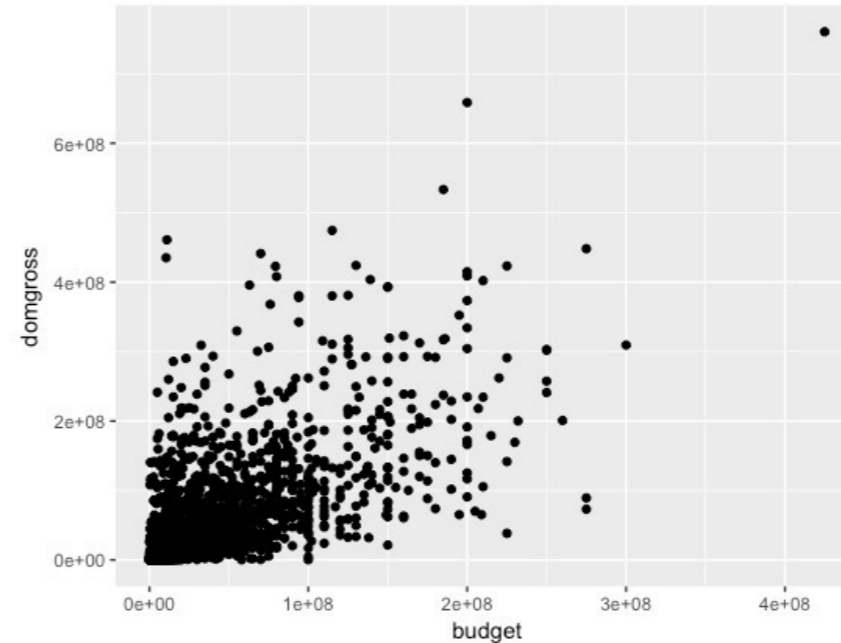
syntax within ggplot2



```
ggplot(data = bechdel, aes(x= budget, y = domgross)) +  
  geom_point() +  
  geom_smooth()
```

syntax within ggplot2

MANY ways to say the same thing



```
ggplot(bechdel) +  
  geom_point(aes(x = budget, y = domgross))  
ggplot(bechdel, aes(x = budget, y = domgross)) +  
  geom_point()  
ggplot(bechdel, aes(x= budget)) +  
  geom_point(aes(y = domgross))  
ggplot() +  
  geom_point(bechdel, aes(x = budget, y = domgross))
```


How can we study this?

- Observational studies (qualitative 🤖)
- Logs of code (🍜)
- DataCamp (📞 call me)
- Mechanical Turk? (🤖)



Thank you!

mcna6887@stthomas.edu

amcnamara@smith.edu

<http://bit.ly/studyingRsyntax>