

R: THE GOOD, THE BAD, AND THE UGLY

John D. Cook

M. D. Anderson Cancer Center



CLINT EASTWOOD

THE GOOD
AND THE BAD
AND THE UGLY

DIRECTED BY SERGIO LEONE

Personal background

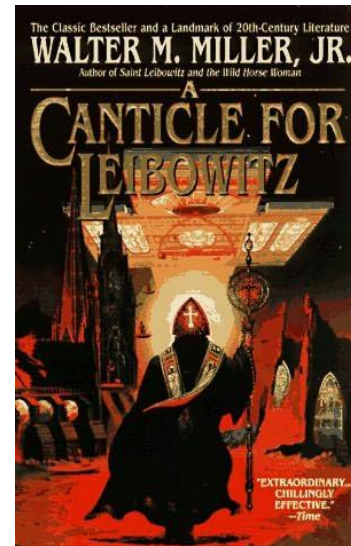
What is R?

- Open source statistical language
- De facto standard for statistical research
- Grew out of Bell Labs' S (1976, 1988)
- Influenced by Scheme, Fortran
- Quirky, flawed, and an enormous success

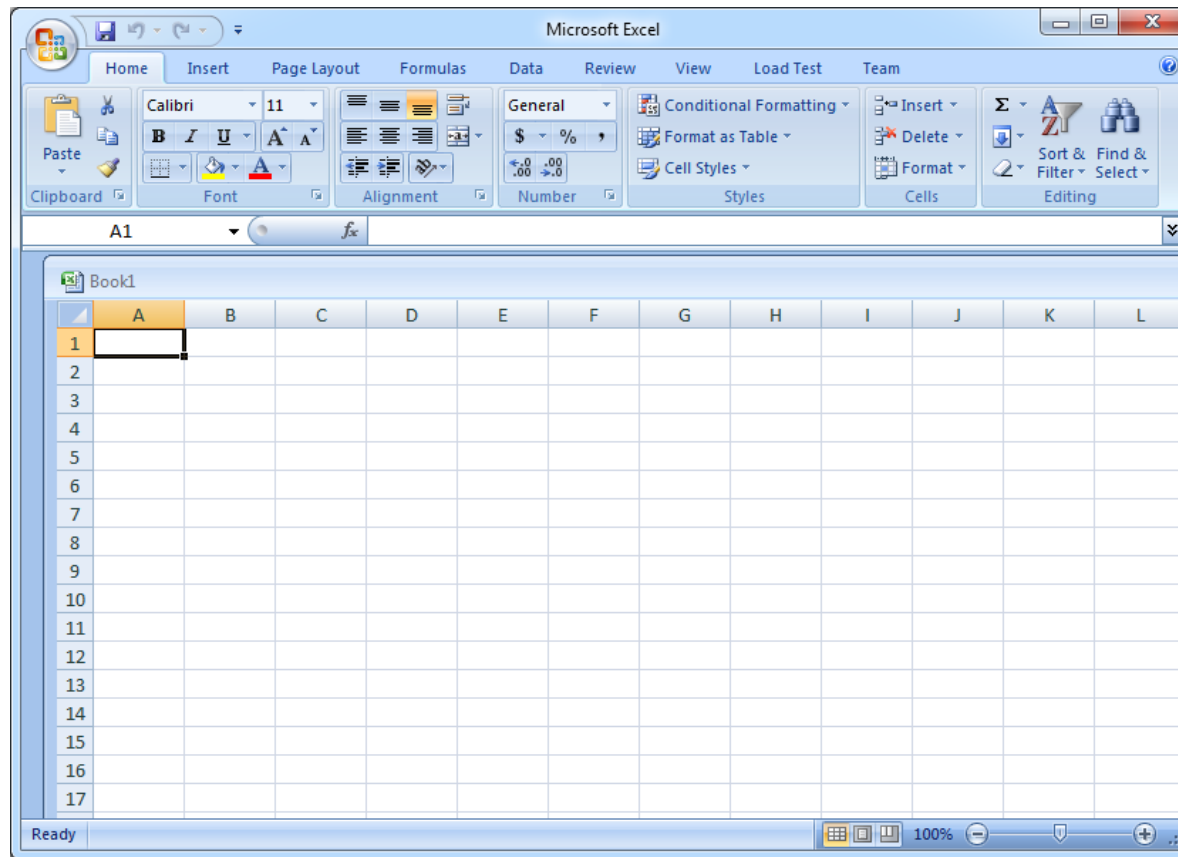


No really, what is R?

“You don't *have* a soul, Doctor.
You *are* a soul.
You *have* a body, temporarily.”



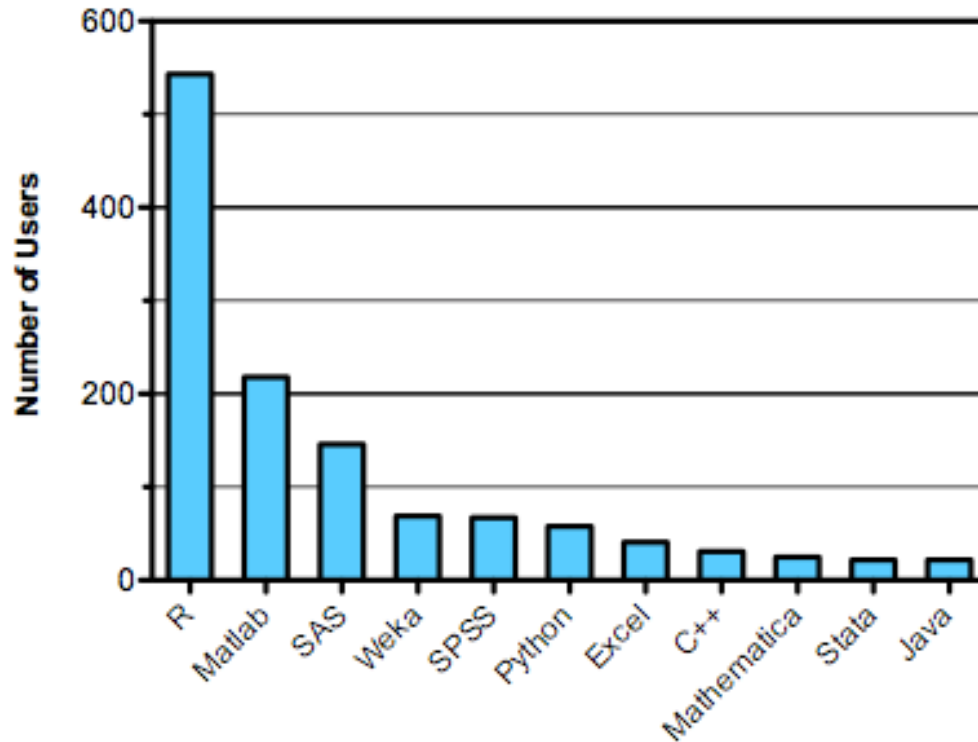
Comparison to Excel



Comparison to Emacs

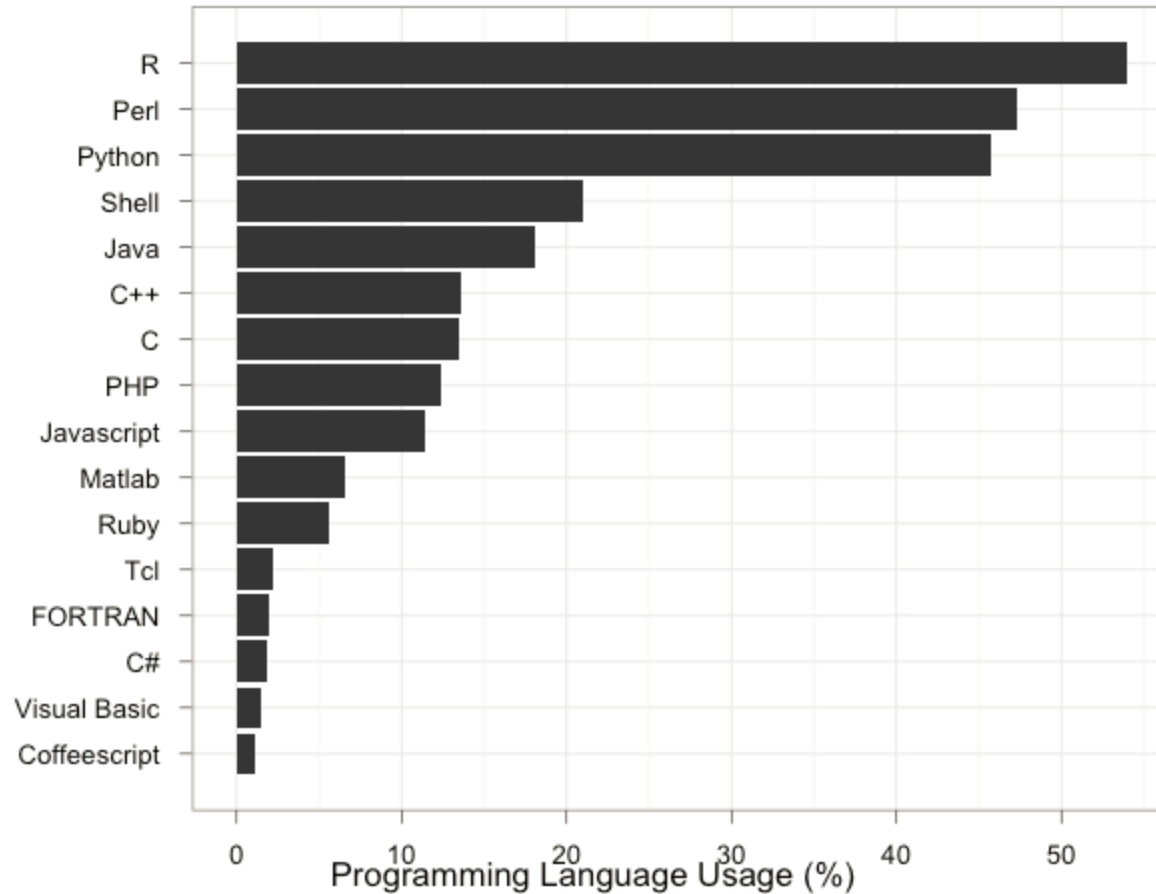


R in data analysis



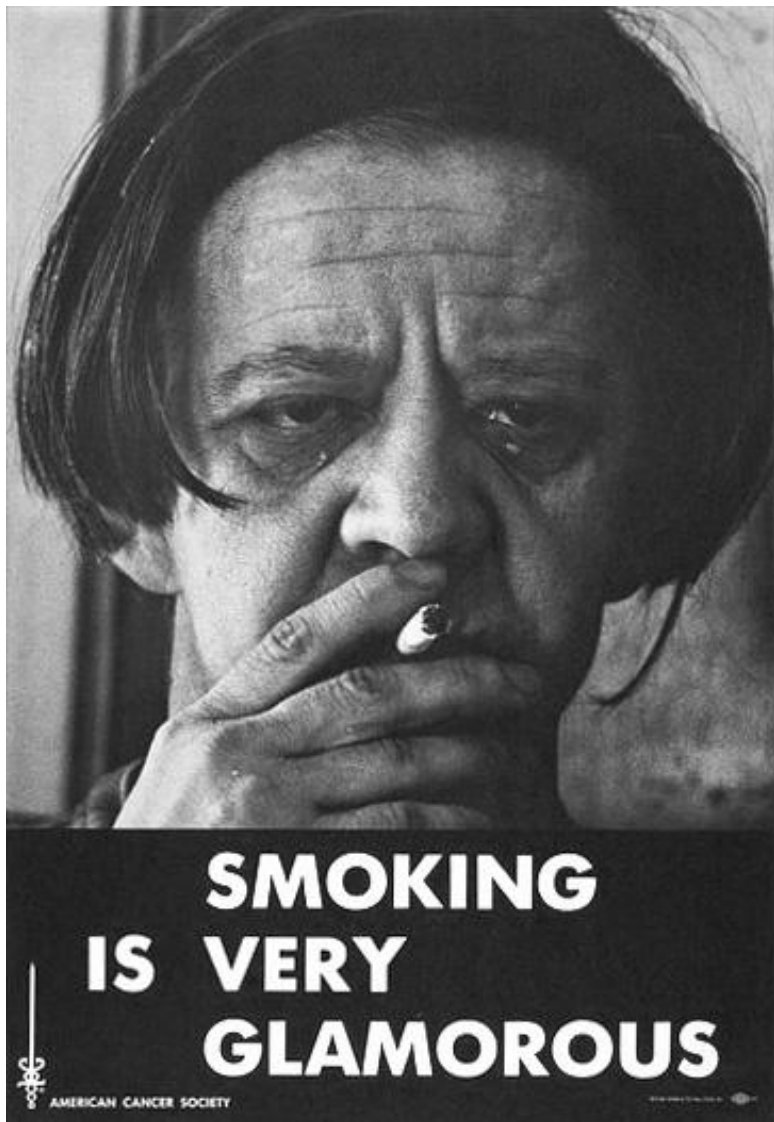
Languages used in Kaggle.com data analysis competition 2011
Source: <http://r4stats.com/popularity>

R in bioinformatics (2012)



http://bioinfsurvey.org/analysis/programming_languages/

So what is using R like?



"Using R is a bit akin to smoking.

The beginning is difficult, one may get headaches and even gag the first few times.

But in the long run, it becomes pleasurable and even addictive.

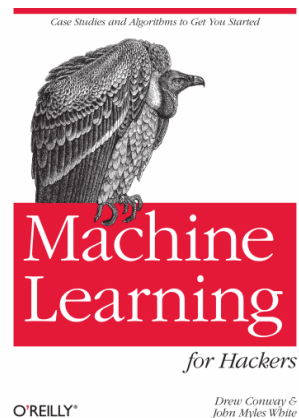
Yet, deep down, for those willing to be honest, there is something not fully healthy in it."

-- Francois Pinard



“... R has a unique and somewhat prickly syntax and tends to have a steeper learning curve than other languages.”

Drew Conway
John Myles White



So why do statisticians use R?

“The best thing about R is that it was written by statisticians.
The worst thing about R ...”

Bo Cowgill, Google

What are statisticians like?

- Different priorities than software developers
- Different priorities than mathematicians
- Learn bits of R in parallel with statistics

R is a DSL

- To understand a DSL, start with D, not L.
- The alternative to R isn't Python or C#, it's SAS.
- People love their DSL, and will use it outside of its domain.

Why a statistical DSL?

- Statistical functions easily accessible
- Convenient manipulation of tables
- Vector operations
- Smooth handling of missing data
- Patterns for common tasks

Some advantages of R

- Batteries included, one namespace
 - Contrast Python + matplotlib + SciPy + IPython
- Designed for **interactive** data analysis
- Easier to program than, e.g., SAS
- Open source, interpreted, portable
- Succinct notation for querying and filtering
- Succinct notation for linear regression

Examples

Set all NA elements of x to 0.

```
x[ is.na(x) ] <- 0
```

```
z <- log( x[y > 7] )
```

Examples

Fit a linear regression model to w as a function of x , y , and z , including a constant term and all first order interaction terms except xz .

```
model <- lm(w ~ (x + y + z)^2 - x:z)
```

Least squares fit to $w = a + b x + c y + d z + e xy + f yz$

Simple regression

growth	tannin
12	0
10	1
8	2
11	3
6	4
7	5
2	6
3	7
3	8

Regression example

```
> data <- read.table("example.txt", header=T)
> attach(data)
> names(data)
[1] "growth" "tannin"
> model <- lm( growth ~ tannin )
> summary(model)
...
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	11.7556	1.0408	11.295	9.54e-06	***
tannin	-1.2167	0.2186	-5.565	0.000846	***

...

Residual standard error: 1.693 on 7 degrees of freedom
Multiple R-squared: 0.8157, Adjusted R-squared:
0.7893
F-statistic: 30.97 on 1 and 7 DF, p-value: 0.0008461

Motor Trend metadata

mtcars {datasets}

R Documentation

Motor Trend Car Road Tests

Description

The data was extracted from the 1974 *Motor Trend* US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models).

Usage

```
mtcars
```

Format

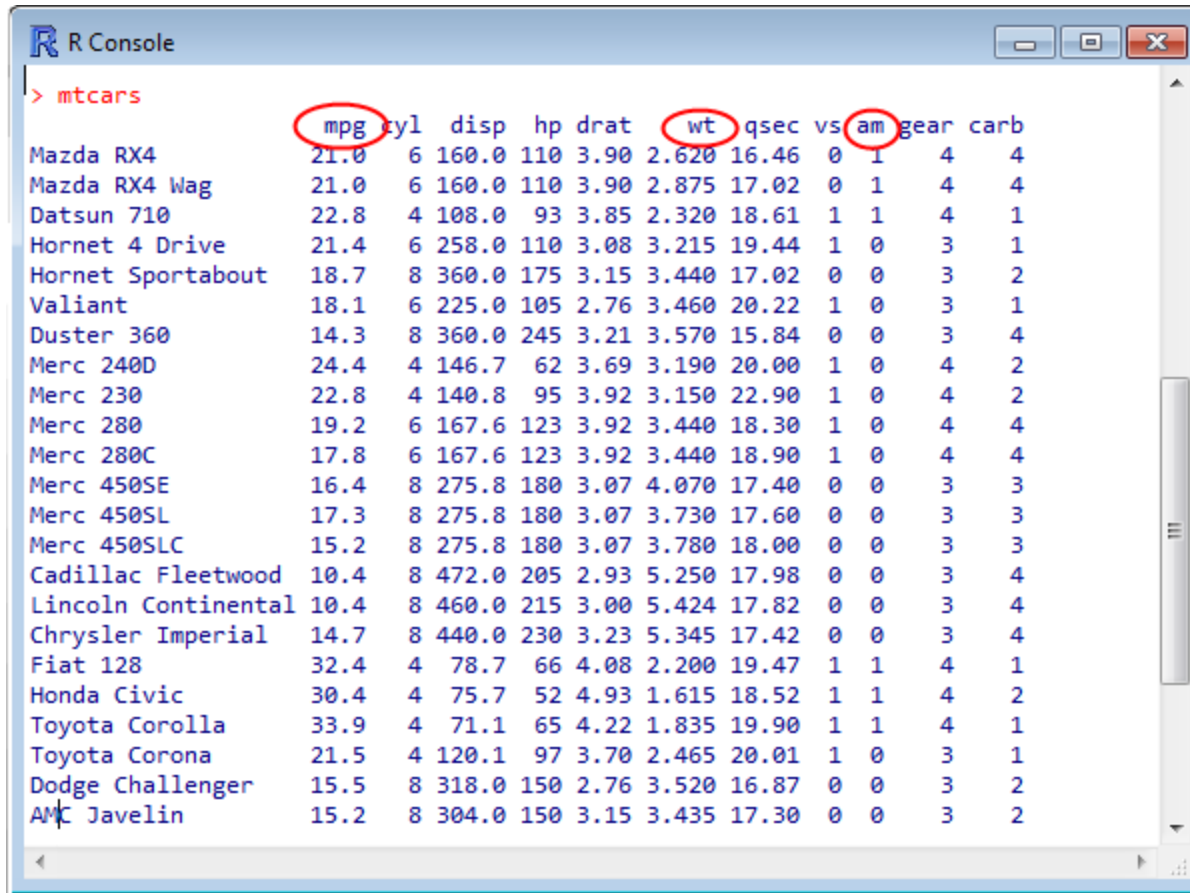
A data frame with 32 observations on 11 variables.

- [, 1] mpg Miles/(US) gallon
- [, 2] cyl Number of cylinders
- [, 3] disp Displacement (cu.in.)
- [, 4] hp Gross horsepower
- [, 5] drat Rear axle ratio
- [, 6] wt Weight (lb/1000)
- [, 7] qsec 1/4 mile time
- [, 8] vs V/S
- [, 9] am Transmission (0 = automatic, 1 = manual)
- [,10] gear Number of forward gears
- [,11] carb Number of carburetors

Source

Henderson and Velleman (1981), Building multiple regression models interactively. *Biometrics*, 37, 391–411.

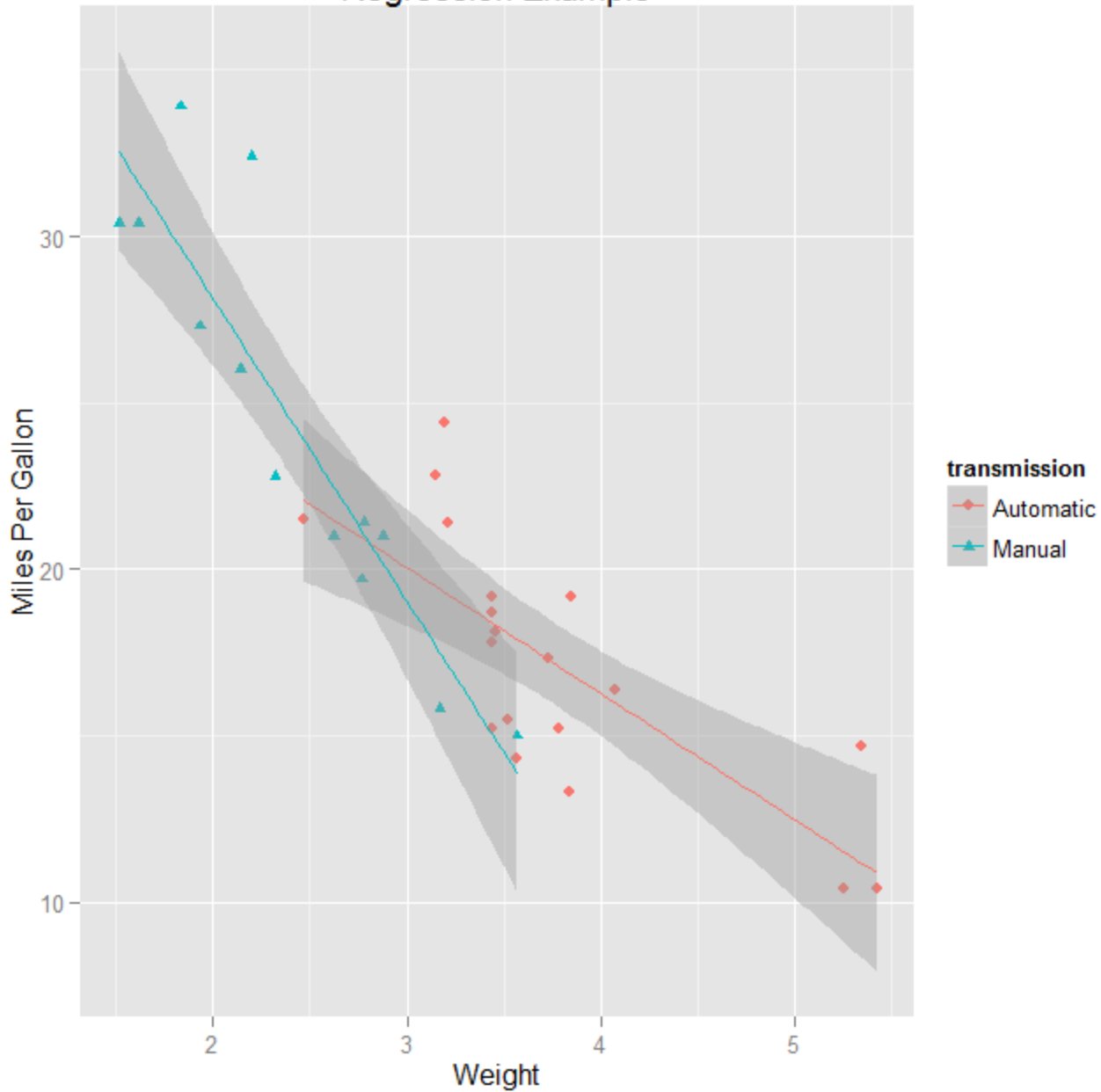
Motor Trend data



```
> mtcars
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2

Regression Example



Example from
"R in Action"
by Robert Kabacoff

Code for plot

```
library(ggplot2)

transmission <- factor(mtcars$am,
  levels = c(0, 1),
  labels = c("Automatic", "Manual"))

qplot(wt, mpg,
  data      = mtcars,
  color     = transmission,
  shape     = transmission,
  geom      = c("point", "smooth"),
  method    = "lm",
  formula   = y ~ x,
  xlab      = "Weight",
  ylab      = "Miles Per Gallon",
  main      = "Regression Example")
```

Language features

- Dynamically typed
- First-class functions, closures
- Objects (two ways!)
- Vector-oriented
- Pass by value
- Everything is nullable (two ways!)

Vectorization example

```
# generate and store one million random values  
x <- rnorm(1e6)  
y <- sum(x)
```

Good R style, bad C style



```
# save memory by generating one random value at a time  
s <- 0  
for ( i in 1:1e6 ) s <- s + rnorm(1)
```

Good C style, bad R style



Some Bad and some Ugly

Speed

Maybe 100x slower than C++,
though it varies greatly.



Tool support

Limited compared to, e.g.,
Visual Studio from 1995.



Safety

Designed for interactive use,
not production.



Hussaini Hanging Bridge (Pakistan)

Misuse

R users often only know R
and use it when inappropriate.



Guide to the Bad and the Ugly

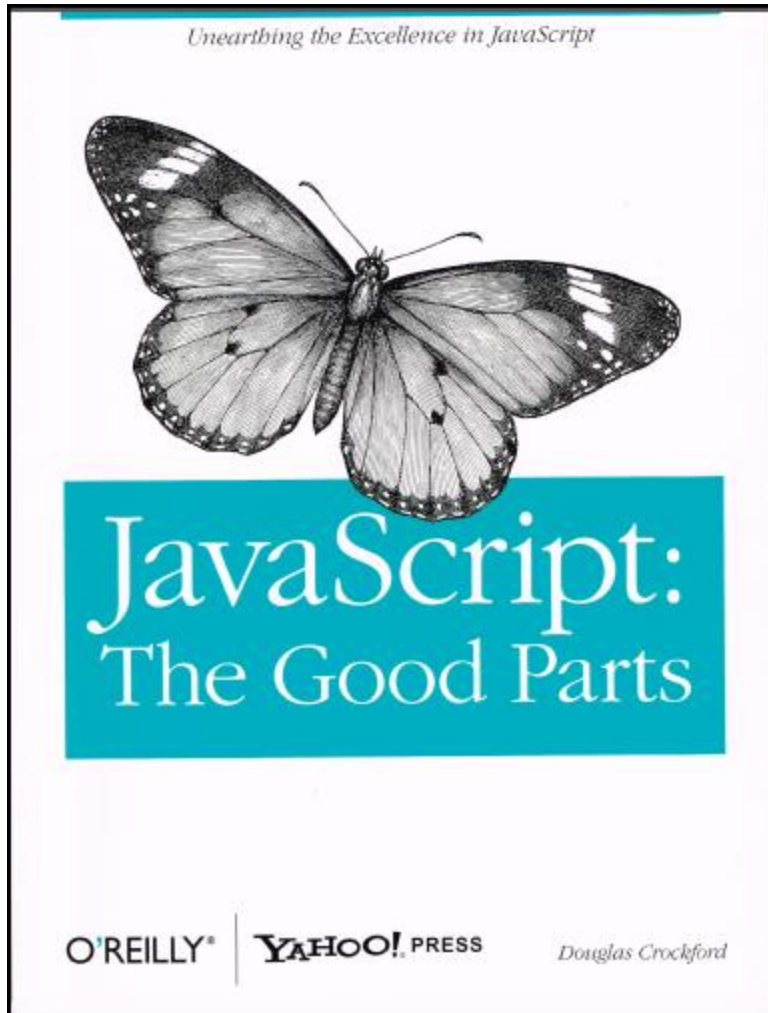


The R Inferno
by Patrick Burns

126 pages

[http://www.burns-stat.com/
pages/Tutor/R_inferno.pdf](http://www.burns-stat.com/pages/Tutor/R_inferno.pdf)

The book I wish someone would write



s/JavaScript/R/



Photo by David Walsh, <http://davidwalsh.name>

Lessons from R

- Data analysis is very different from system programming.
- People will put up with a lot to get their work done.
- People will use a familiar tool over a better tool if at all feasible.

Resources

- <http://www.r-project.org/>
- [http://www.johndcook.com/
R_language_for_programmers.html](http://www.johndcook.com/R_language_for_programmers.html)
- “The Art of R Programming”
by Normal Matloff
- @RLangTip