Open-Source for Open Science
-A very brief introduction to R

Rebecca Clark
November 7, 2014
with material adapted from Melanie Frazier

A brief history

1976 - S language developed at AT&T Bell Labs

1993 - S is translated into R: Ross Ihaka and Robert Gentlemen; made decision to go open-source

1997 - International R Core team established as mailing lists, participants grew

2000 - R v.1.0.0 released

Early to mid-2000's - Thousands of programmers and statisticians contributing to the effort

Today: Us taking advantage of all this hard work!

But what does this mean for you?

It's all you will ever need

(at least in terms of data)

Shape data

Species	Size
grasshopper	1.2
grasshopper	1.5
grasshopper	2.1
grasshopper	1.6
grasshopper	1.7
fly	0.2
fly	0.4
fly	0.1
fly	0.2
beetle	5
beetle	4.7
beetle	3.2
beetle	3.1
beetle	2

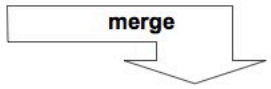
Species	AvgSize
grasshopper	1.2
fly	0.2
beetle	5

tapply(data$Size, data$Species, mean)
or:
It pays to learn Hadley Wickham's "reshape" package (not to be confused with the function "reshape").
http://had.co.nz/reshape/

Shape data

Species	Size
grasshopper	1.2
grasshopper	1.5
grasshopper	2.1
grasshopper	1.6
grasshopper	1.7
fly	0.2
fly	0.4
fly	0.1
fly	0.2
beetle	5
beetle	4.7
beetle	3.2
beetle	3.1
beetle	2

Species	Order
grasshopper	homoptera
grasshopper	homoptera
grasshopper	homoptera
grasshopper	homoptera
grasshopper	homoptera
fly	diptera
fly	diptera
fly	diptera
fly	diptera
beetle	coleoptera
beetle	coleoptera
beetle	coleoptera
beetle	coleoptera
beetle	coleoptera



Species	Size	Order
grasshopper	1.2	homoptera
grasshopper	1.5	homoptera
grasshopper	2.1	homoptera
grasshopper	1.6	homoptera
grasshopper	1.7	homoptera
fly	0.2	diptera
fly	0.4	diptera
fly	0.1	diptera
fly	0.2	diptera
beetle	5	coleoptera
beetle	4.7	coleoptera
beetle	3.2	coleoptera
beetle	3.1	coleoptera
beetle	2	coleoptera

merge(data1, data2, by="Species")

Shape data

Site	Species	Count
A	Grasshopper	2
A	Fly	3
A	Beetle	6
B	Grasshopper	1
B	Fly	1
C	Grasshopper	10
D	Grasshopper	100
D	Fly	11
D	Beetle	12
E	Grasshopper	3
E	Fly	5
E	Beetle	7
F	Fly	9
G	Beetle	10

	G	F	B
A	2	3	6
B	1	1	--
C	10	--	--
D	100	11	12
E	3	5	7
F	--	9	--
G	--	--	10

From Hadley Wickham's "reshape" package

Analyze

base package (includes the basics and loads automatically):

linear regression

ANOVA

chi-square

generalized linear models

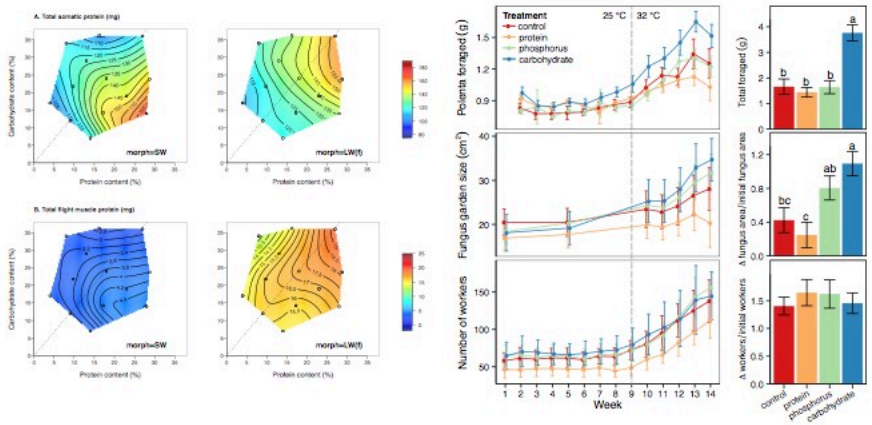
...

Analyze

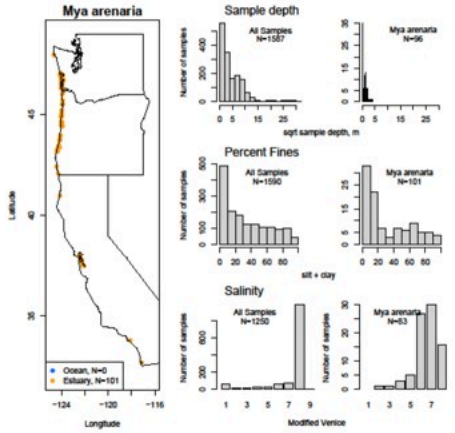
November 30, 2010: **2610** contributed **packages**
May 26, 2014: **5577** available packages

ape Analysis of phylogenetics and evolution
lme4 & **nlme** Hierarchical models
adehabitat Analysis of habitat selection by animals
vegan Community Ecology Package (PRIMER+)
RgoogleMaps Overlays Google map tiles in R
SwissAir Air quality data from Switzerland for one year in 30-minute resolution
sudoku Sudoku puzzle generator and solver

Visualize



base graphics, **ggplot2**, **lattice**



Melanie Frazier

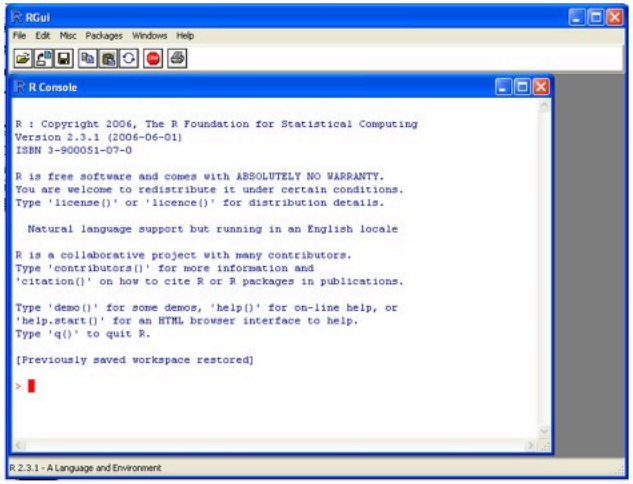
maps, **mapdata**, **maptools**

...and it's *open-source*! i.e. it's free, plus there are **other benefits:**

-Full access to algorithms and implementation
-Forums allow researchers to explore and expand methods used to analyze data
-Ensures that scientists around the world - not just ones in rich countries - are co-owners to software tools for research
-Promotes reproducible research by providing open and accessible tools

So, what's the catch?

The GUI: Where are the buttons??



- Not well-organized system
- No commercial support or all-in-one manual
- Figuring out an analysis or how to use a function on your own can be frustrating

You have to learn a new language...

```
ylimit <- c(-.5,2) # y should be three for unrestricted
plot(1,1,
     ylim=ylimit,
     xlim=c(start.year,2010),
     pch=" ",
     ann=F,
     axes=F)
# temperature change equals zero line
lines(c(start.year-1,2010),c(0,0), col="black", lty="dotted", lwd=lwd.axes)
for(i in 1:length(sem.n.temp)){
  lines(rep(x.points[i]+pol.jit, 2), c(mean.polar[i]+sem.polar[i],mean.polar[i]-sem.polar[i]), col=col.pol, lwd=lwd.err)
  lines(rep(x.points[i], 2), c(mean.n.temp[i]+sem.n.temp[i],mean.n.temp[i]-sem.n.temp[i]),col=col.temp, lwd=lwd.err)
  lines(rep(x.points[i], 2), c(mean.s.temp[i]+sem.s.temp[i],mean.s.temp[i]-sem.s.temp[i]),col=col.temp, lwd=lwd.err)
  lines(rep(x.points[i]+trop.jit, 2), c(mean.tropical[i]+sem.tropical[i],mean.tropical[i]-sem.tropical[i]),col=col.trop, lwd=lwd.err)
}
points(x.points+pol.jit, mean.polar, type="b", col=col.pol, lty="solid", lwd=lwd.data, pch=pch.all, cex=cex.pt)
points(x.points, mean.n.temp, type="b", col=col.temp, lty="solid", lwd=lwd.data, pch=pch.all, cex=cex.pt)
points(x.points, mean.s.temp, type="b", col=col.temp, lty="dashed", lwd=lwd.data, pch=pch.all, cex=cex.pt)
points(x.points+trop.jit, mean.tropical, type="b", col=col.trop, lty="solid", lwd=lwd.data, pch=pch.all, cex=cex.pt)
#
ylabel=c(-.5,0,.5,1,1.5,2)
axis(2, at=ylabel, labels=NA, las=1, lwd=lwd.axes, tcl=tcl.axes)
par(ps=ps.axes)
text(rep(1972.5, length(ylabel)), ylabel, c("-0.5","    0","  0.5","  1.0","  1.5","  2.0"), adj=c(-1,0.5))
xlabel <- seq(start.year, 2010, 5)
axis(1, at=xlabel, labels=NA, lwd=lwd.axes, tcl=tcl.axes)
text(xlabel[c(1, 3, 5, 7)], rep(-.7, length(xlabel[c(1, 3, 5, 7)])), xlabel[c(1, 3, 5, 7)], adj=c(0.5,0.5))
#text(xlabel[c(1, 3, 5, 7)], rep(-.7, length(xlabel[c(1, 3, 5, 7)])), xlabel[c(1, 3, 5, 7)], srt=-50, adj=c(0,0.5))
# add plot annotation
par(xpd=NA, ps=ps.subfig)
text(1981, max(ylabel), adj=c(0,0.5), "a", ps=ps.subfig, font=2, cex=1)
par(xpd=NA, ps=ps.axes.label)
text(1974, mean(ylabel), expression(paste("Change in temperature (",degree,"C)")), srt=90)
par(ps=ps.ann)
text(2004, 0.5, "tropical", col=col.trop, ps=ps.ann) # 2003,.48 for unrestricted,
text(1990, 0.75, "north temperate", col=col.temp, ps=ps.ann) # 1986,.7 for unrestricted
text(2002,-0.35, "south temperate", col=col.temp, ps=ps.ann) # 2002,-.4 for unrestricted
text(2000, 1.5, "arctic", col=col.pol, ps=ps.ann) # 1992,2.0 for unrestricted
#
```

Ultimately, this is a good thing.

- **Written record** of data manipulation and analyses
- It's a programming language, so actions can be repeated thousands of times
- Get the *exact* analysis/figure that you want

Horrifying Scenario 1:

Years after doing a project, another researcher requests your data.

You realize you have 4 versions of the dataset with slight variations in sample size and treatment groups.

R solution:

After the dataset is compiled, it is **not** altered!
Subsetting occurs in the R script and not the data.

Result: Data files remain pristine and untouched.

Horrifying Scenario 2:

You finally return to a dataset that you started and stopped analyzing a year (or so) ago.

You realize you can't remember how you did your original analysis.

R solution:

All analyses are from a script, which means you will have complete documentation of what you did.

Result: You can have a bad memory but still resume projects close to where you left off.

Enough of this. Time to take a look!

Wrap-Up: Where to go for help

Books

- List of recommended books at acromyrmex.net

Internets

- Stack Overflow
- r-help e-mail list
- ...many more

People