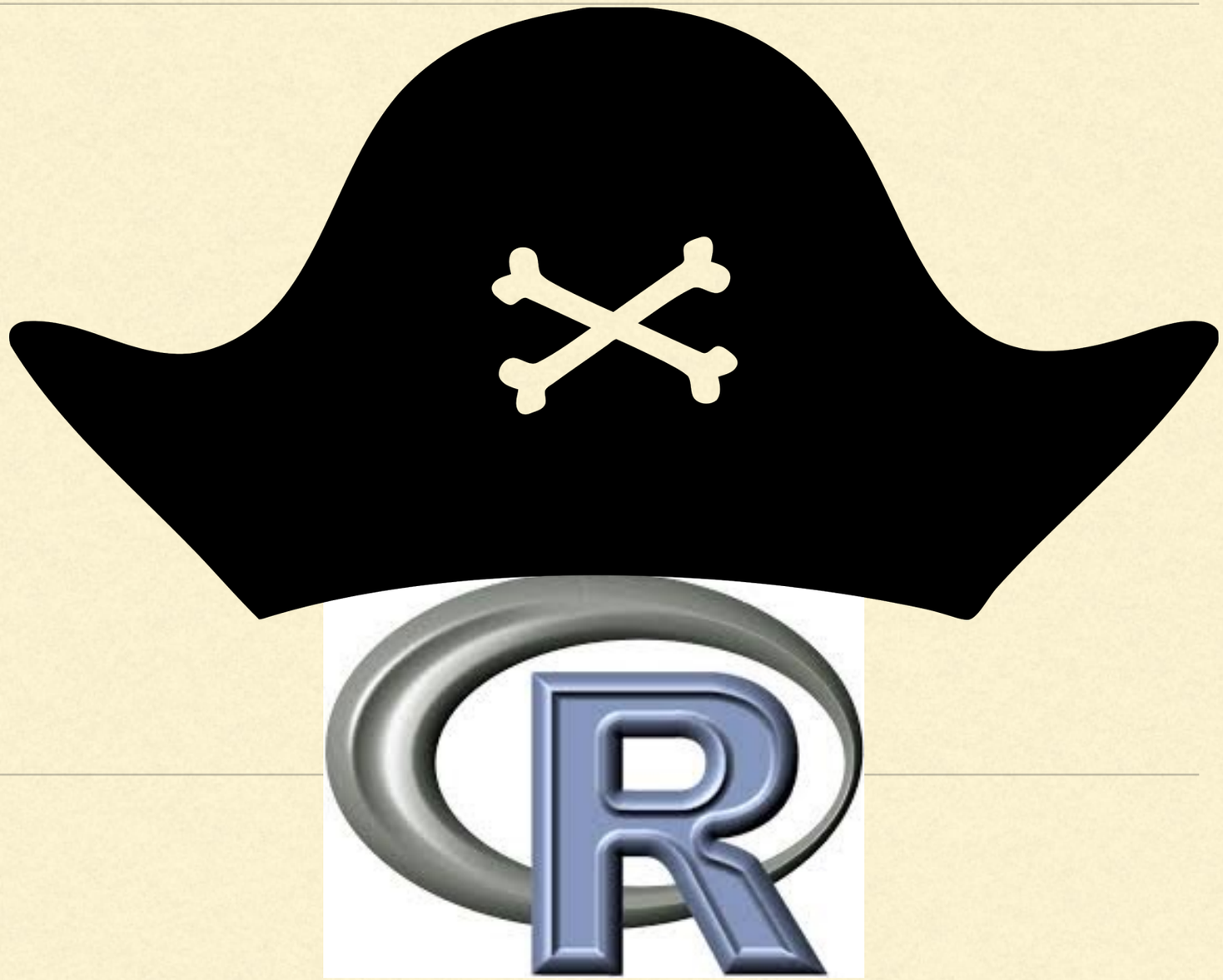


ARGH

Amy Willis
@AmyDWillis
Assistant Professor
Department of Biostatistics
University of Washington

R IS…

- the sound you make when you get one of the following errors

```
> taxaRel[,13]  <- rowSums(taxaRel)
Error in `[<-`(`*tmp*`, , 13, value = c(0.00010789038
3370496, 0.000139324277255312,  :
  subscript out of bounds
```

```
> heatmap(topTen, main="Common Taxa", labRow=topGener
a, labCol=Seasons)
Error in heatmap(topTen, main = "Common Taxa", labRow
= topGenera, labCol = Seasons) :
  object 'Seasons' not found
```

```
f(10)
Error in "a" + d : non-numeric argument to binary operator
```

```
Error in eval(expr, envir, enclos) : object 'JPA_Jan'
not found
```

```
> A%*%B
Error in A %*% B : non-conformable arguments
>
```

R IS...

- Open source software for statistical analysis
- Platform for developing and sharing statistical tools
- A big, fancy calculator

R IS…

- At best
  - a great option for analysing your data
  - a great tool for reproducible research
  - easy to install, easy to collaborate with
- At worst
  - argh…

R VS RSTUDIO

- R is the programming language
- R is the calculator
- RStudio is the interface
- RStudio is your friend

R VS RSTUDIO

RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help Thu 2:24 PM adw96

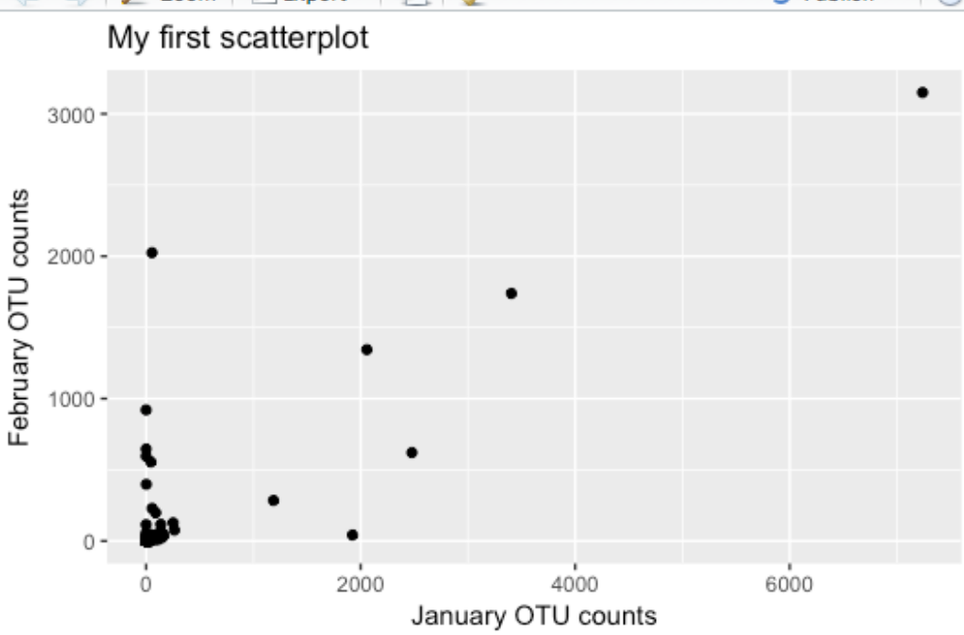
RStudio

Console ~/Documents/teaching/17-stamps/

```
Loading required package: Rcpp
Warning message:
package 'dada2' was built under R version 3.3.3
>
>
>
>
>
> library(dada2)
> ggplot(abundances, aes(JPA_Jan, JPA_Feb)) +
+   geom_point() +
+   labs(title="My first scatterplot") + # adds a tit
le
+   xlab("January OTU counts") + # adds labels
+   ylab("February OTU counts")
>
```

Files Plots Packages Help Viewer

Zoom Export Publish



STAMPS_Intro2R_OptionB.R STAMPS_Intro2R_OptionA.R STAMPS_Intro2R_OptionC.R

Source on Save Run Source

```r
# The range of this histogram is huge, so we want to focus on the distribution o
# OTUs that were observed once or more but less than 60 times, we would do
ggplot(subset(abundances, JPA_Jan > 0 & JPA_Jan < 60), aes(JPA_Jan)) +
  geom_histogram()

# a scatterplot shows how correlated the abundances are between january and febr
ggplot(subset(abundances, JPA_Jan > 0 & JPA_Feb > 0),
       aes(JPA_Jan, JPA_Feb)) +
  geom_point()
# again, ggplot() specifies what we want plotted
# geom_point() specifies that we want a scatterplot

ggplot(subset(abundances, JPA_Jan > 0 & JPA_Feb > 0),
       aes(JPA_Jan, JPA_Feb)) +
  geom_point() +
  theme_classic() # changes the styling

ggplot(abundances, aes(JPA_Jan, JPA_Feb)) +
  geom_point() +
  labs(title="My first scatterplot") + # adds a title
  xlab("January OTU counts") + # adds labels
  ylab("February OTU counts")

# one tedious thing about ggplot is that all of the information that
# you want to plot must be in the same data frame
# So incorporating Season (from the data frame "covariates")
# to colour the points our scatterplot is not trivial

# We need to create a new data frame that contains all this information
# together...
```

116:1 (Untitled) R Script

Environment History

R PACKAGES

- R can't cluster your sequences, make nice graphs, estimate species richness, order a pizza…
- Enter: packages!
  - DADA2 can cluster your sequences
  - ggplot can make nice graphs
  - breakaway can estimate species richness
  - pizza?

R PACKAGES

```
> library("dada2", lib.loc="~/Library/R/3.3/library")
Loading required package: Rcpp
Warning message:
package ‘dada2’ was built under R version 3.3.3
>
```

CREATING VARIABLES

```
> x <- 5
> x
[1] 5
> y = 10
> y
[1] 10
> x + y
[1] 15
>
```

2 different ways to assign variables values in R

```
> covariates <- read.table("FWS_covariates.txt", header=TRUE, sep="\t", as.is=TRUE)
> covariates$SampleName
 [1] "JPA_Jan" "JPA_Feb" "JPA_May" "JPA_Jun"
 [5] "LOP_Jan" "LOP_Feb" "LOP_Jun" "LOP_Jul"
 [9] "MBL_Jan" "MBL_Feb" "MBL_Jun" "MBL_Aug"
```

USING FUNCTIONS

```
> x <- 5
> x
[1] 5
> y = 10
> y
[1] 10
> x + y
[1] 15
>
> covariates <- read.table("FWS_covariates.txt", header=TRUE, sep="\t", as.is=TRUE)
> covariates$sampleName
 [1] "JPA_Jan" "JPA_Feb" "JPA_May" "JPA_Jun"
 [5] "LOP_Jan" "LOP_Feb" "LOP_Jun" "LOP_Jul"
 [9] "MBL_Jan" "MBL_Feb" "MBL_Jun" "MBL_Aug"
```

CREATING FUNCTIONS

```
# Write a function that takes in an OTU table
# and returns the relative abundance table
counts_to_abundances <- function(otu) {
  apply(otu, 2, function(x) x/sum(x))
}
```

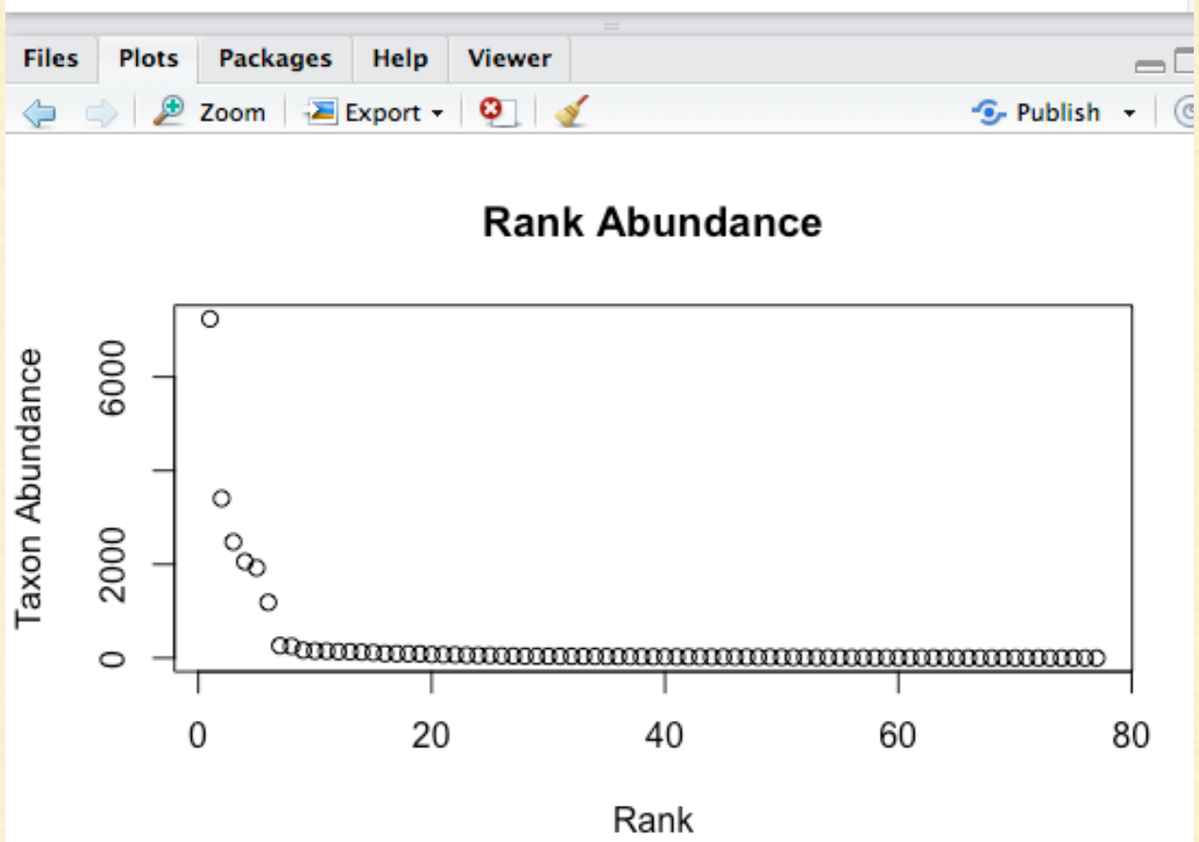
DATA TYPES

```
> a <- 5
> class(a)
[1] "numeric"
> b <- "hello!"
> class(b)
[1] "character"
> c <- matrix(c(1:4), nrow = 2)
> class(c)
[1] "matrix"
> d <- data.frame(a, b)
> class(d)
[1] "data.frame"
> e <- list(a, b)
> class(e)
[1] "list"
>
```

PLOTTING

```
> ggplot(abundances, aes(JPA_Jan, JPA_Feb)) +
+   geom_point() +
+   labs(title="My first scatterplot") + # adds a title
+   xlab("January OTU counts") + # adds labels
+   ylab("February OTU counts")
>
```


```
> plot(myRankAbund, main="Rank Abundance", xlab="Rank",
ylab="Taxon Abundance")
>
```



SAVING VARIABLES

```
#####################################
#
#  Save your work!
#
#####################################

# this saves all of the objects (functions and variables)
# that you created
save.image("optionA-objects.Rdata")


# next time, when you want to load these objects:
load("optionA-objects.Rdata")

# you can see how this works by first removing an object
rm(abundances) # remove this matrix
abundances # it no longer exists
load("optionA-objects.Rdata") # reload the saved data
abundances[1:5,1:5] # look! it exist again
```

THE IMPORTANCE OF SCRIPTING

- R scripts are files of R commands
- Great for rerunning your analyses with new data
- Great for returning to months later when you write your Methods section

THE IMPORTANCE OF SCRIPTING

This ***will*** happen to you:

Your Colleague 📎 August 4, 2014 9:30 AM
To: Susan Huse
updated repeat donor metadata Inbox - Gmail 1

Sue,

Here is the updated metadata for the repeat semen donors. The density for patient #365 has been updated, while all the remaining data has remained the same.

Hopefully you can re-run the analysis with the updated value fairly quickly and effortlessly.

Thanks,
Ed

CSA: COMMUNITY SUPPORTED ARGH

```
> ?ggplot
>
```

Files | Plots | Packages | Help | Viewer

R: Create a new ggplot Find in Topic

ggplot {ggplot2} R Documentation

Create a new ggplot

Description

`ggplot()` initializes a ggplot object. It can be used to declare the input data frame for a graphic and to specify the set of plot aesthetics intended to be common throughout all subsequent layers unless specifically overridden.

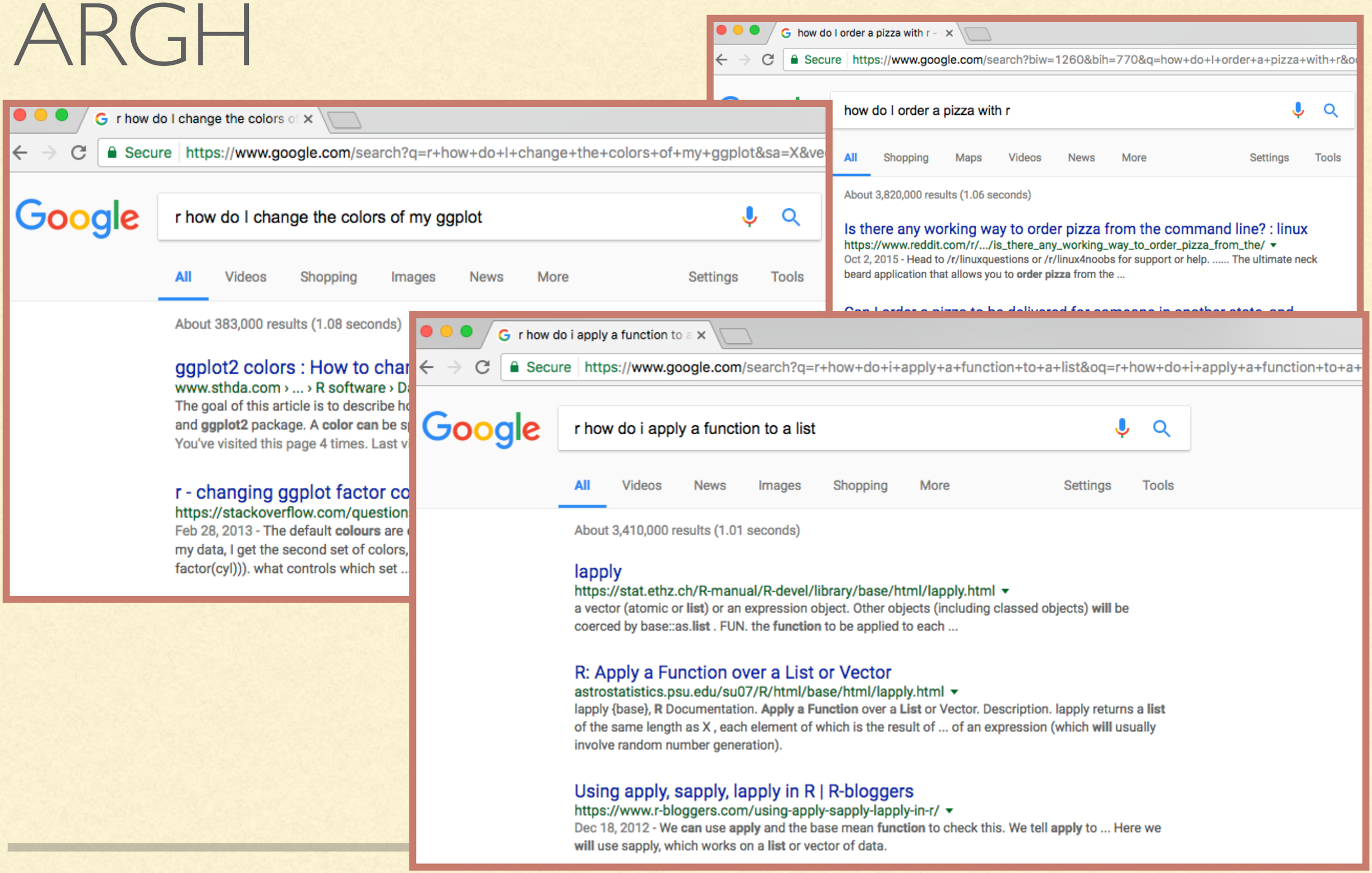
Usage

```
ggplot(data = NULL, mapping = aes(), ..., environment = parent.frame(
```

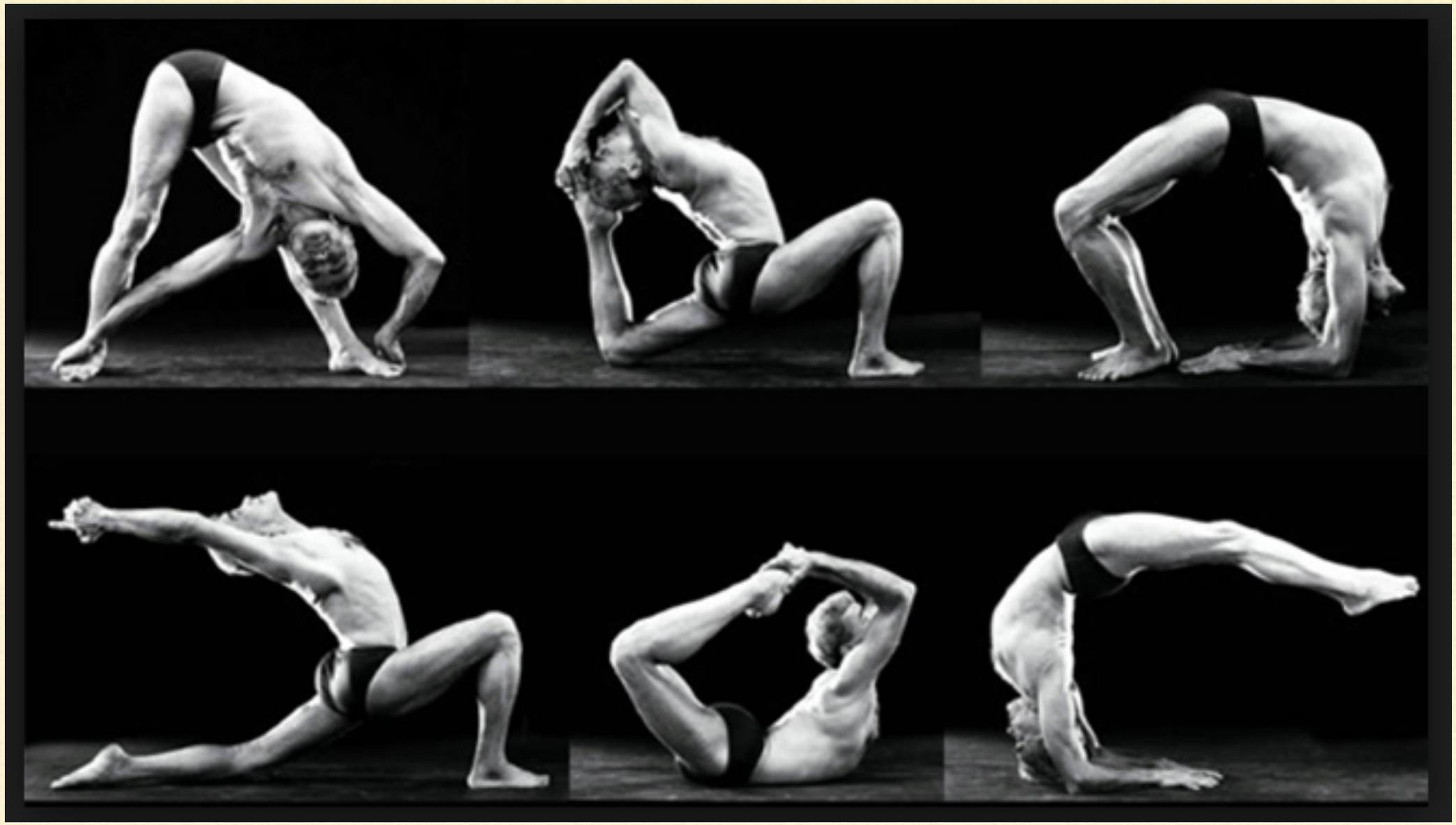
Arguments

`data`	Default dataset to use for plot. If not already a data.frame, will be

CSA: COMMUNITY SUPPORTED ARGH



LAB OPTIONS



LAB OPTION A

- Reading in data
- Introduction to data types
- Creating summaries
- Basic model fitting: lm()
- Plotting with base graphics

LAB OPTION B

- ggplot
- for loops (and how to avoid them)
- apply
- lists
- writing functions

LAB OPTION C

DIY/TIY:

- source() and system()
- reshape: melt() and cast()
- plyr
- ddply
- R Markdown

LAB OPTION(S) D

- Write a package to order Amy a pizza
- Open up the data and start writing up your analysis!
- **Assist your classmates, assist the TAs**
- Peace out and come back after lunch

THE LAB

- Go to **https://github.com/adw96/stamps**
- Download the data files (FWS_OTUs.txt, FWS_covariates.txt) and your choice of script to work through (eg. STAMPS_Intro2R_OptionA.R)
- Open the script of choice in RStudio
- Begin working through, completing the exercises as you go

- Help your neighbours, ask your neighbours for help
- Don't just click through! Cognition required!
- Find your own balance between speed and coverage (90 more minutes tonight, too!)