Medical Image Analysis with R

Brian B. Avants

PENN Image Computing & Science Laboratory

Dept. of Radiology, University of Pennsylvania

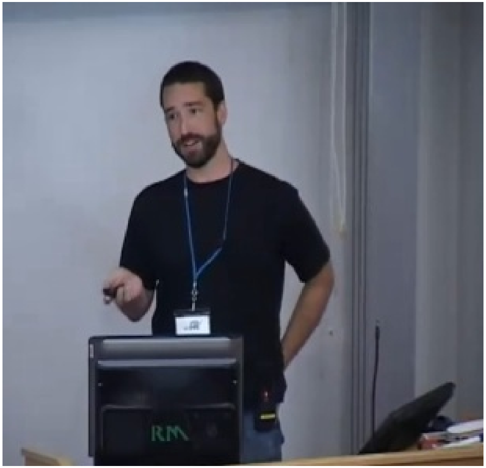
Philadelphia, PA, 19104 [1]

September 23, 2013

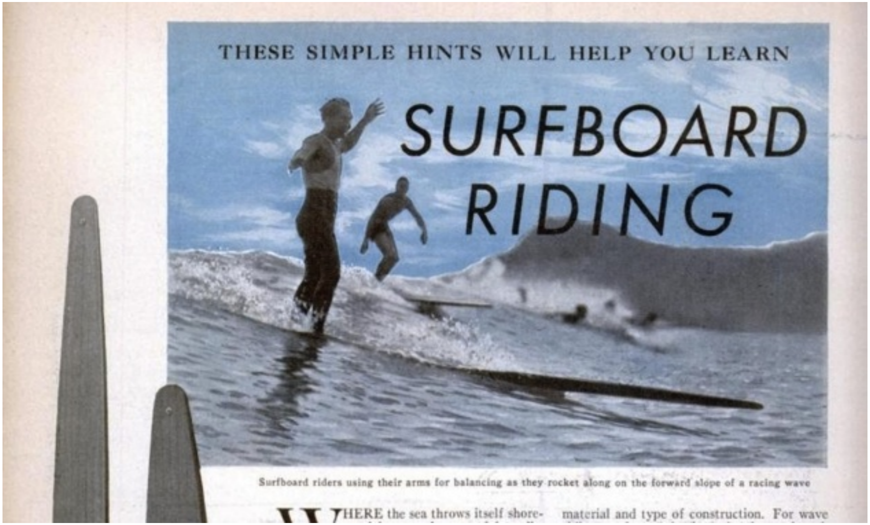
[1] Thank you for participating!

Many Thanks for Contributions from:

Brandon Whitcher, Ph.D.
Pfizer
Cambridge, MA, USA



How much *R* experience in the room?



THESE SIMPLE HINTS WILL HELP YOU LEARN

SURFBOARD RIDING

Surfboard riders using their arms for balancing as they rocket along on the forward slope of a racing wave

WHERE the sea throws itself shore- material and type of construction. For wave

Schedule

- Introduction
- Basic Examples
- 1/2 Hour Break
- Longitudinal Analysis and Model Selection (P. Thomas Fletcher)
- Wrap-up / discussion

Schedule

- Introduction
- Basic Examples
- 1/2 Hour Break
- Longitudinal Analysis and Model Selection (P. Thomas Fletcher)
- Wrap-up / discussion

Schedule

- Introduction
- Basic Examples
- 1/2 Hour Break
- Longitudinal Analysis and Model Selection (P. Thomas Fletcher)
- Wrap-up / discussion

Schedule

- Introduction
- Basic Examples
- 1/2 Hour Break
- Longitudinal Analysis and Model Selection (P. Thomas Fletcher)
- Wrap-up / discussion

Schedule

- Introduction
- Basic Examples
- 1/2 Hour Break
- Longitudinal Analysis and Model Selection (P. Thomas Fletcher)
- Wrap-up / discussion

Tutorial Goals

- Learn about *R* in general (operations, dataframes, models)
- Understand basics of *R* image-based statistics (I/O, accessing values, structure, function)
- Practice some example reproducible studies ...
- Identify opportunities for innovation/future work
- A good source on data analysis with *R* examples: "Advanced Data Analysis from an Elementary Point of View" here.

Tutorial Goals

- Learn about *R* in general (operations, dataframes, models)
- Understand basics of *R* image-based statistics (I/O, accessing values, structure, function)
- Practice some example reproducible studies ...
- Identify opportunities for innovation/future work
- A good source on data analysis with *R* examples: "Advanced Data Analysis from an Elementary Point of View" here.

Tutorial Goals

- Learn about *R* in general (operations, dataframes, models)
- Understand basics of *R* image-based statistics (I/O, accessing values, structure, function)
- Practice some example reproducible studies ...
- Identify opportunities for innovation/future work
- A good source on data analysis with *R* examples: "Advanced Data Analysis from an Elementary Point of View" here.

Tutorial Goals

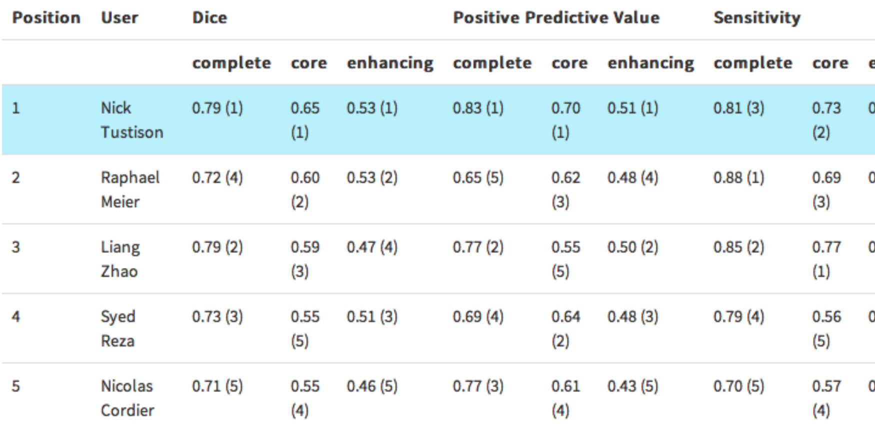
- Learn about *R* in general (operations, dataframes, models)
- Understand basics of *R* image-based statistics (I/O, accessing values, structure, function)
- Practice some example reproducible studies ...
- Identify opportunities for innovation/future work
- A good source on data analysis with *R* examples: "Advanced Data Analysis from an Elementary Point of View" here.

Tutorial Goals

- Learn about *R* in general (operations, dataframes, models)
- Understand basics of *R* image-based statistics (I/O, accessing values, structure, function)
- Practice some example reproducible studies ...
- Identify opportunities for innovation/future work
- A good source on data analysis with *R* examples: "Advanced Data Analysis from an Elementary Point of View" here.

R is Relevant to Your Success

We entered a competition with R in our holster.



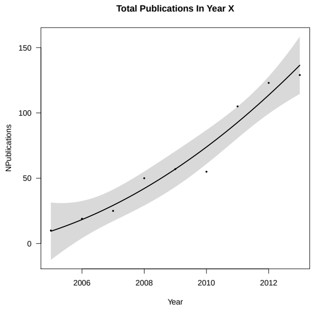
Position	User	Dice			Positive Predictive Value			Sensitivity		
		complete	core	enhancing	complete	core	enhancing	complete	core	e
1	Nick Tustison	0.79 (1)	0.65 (1)	0.53 (1)	0.83 (1)	0.70 (1)	0.51 (1)	0.81 (3)	0.73 (2)	0
2	Raphael Meier	0.72 (4)	0.60 (2)	0.53 (2)	0.65 (5)	0.62 (3)	0.48 (4)	0.88 (1)	0.69 (3)	0
3	Liang Zhao	0.79 (2)	0.59 (3)	0.47 (4)	0.77 (2)	0.55 (5)	0.50 (2)	0.85 (2)	0.77 (1)	0
4	Syed Reza	0.73 (3)	0.55 (5)	0.51 (3)	0.69 (4)	0.64 (2)	0.48 (3)	0.79 (4)	0.56 (5)	0
5	Nicolas Cordier	0.71 (5)	0.55 (4)	0.46 (5)	0.77 (3)	0.61 (4)	0.43 (5)	0.70 (5)	0.57 (4)	0

The first 3 rules of statistics: 'Draw a picture, Draw a picture, Draw a picture.'—Michael Starbird.

R in medical imaging?

Search "r-project.org + medical + imaging"

```r
dd <- read.csv("data/RMI.csv")
mdl <- lm(NPublications ~ Year + I(Year^2), data = dd)
visreg(mdl, main = "Total Publications In Year X")
```



R contains virtually all popular statistical and machine learning algorithms, including Boosting, the LASSO, and random forests, often contributed by the inventors.

Why use *R* in medical imaging?

- It's free — not "free" like Matlab/SPM but really free
  - It is the *de facto* standard for statistical computing
  - a New York Times article from 2009 estimated that there are at least 250,000 active R users
- Why not use Python?
  1. IMHO, *R* is easier to compile/maintain/install
  2. Visualization in *R* is as good or better than Python
  3. Most importantly—*statisticians contribute directly to R*
  4. Because of this, many Python users rely on *R*
- *R* facilitates reproducible research:
  1. CRAN Task View (link)
  2. *Biostatistics (link)*
  3. *jstatsoft.org*
- *R* assists reproducible medical image analysis but it's still effortful to do it correctly. See our comments in Frontiers.

Why use *R* in medical imaging?

- It's free — not "free" like Matlab/SPM but really free
  - It is the *de facto* standard for statistical computing
  - a New York Times article from 2009 estimated that there are at least 250,000 active R users
- Why not use Python?
  1. IMHO, *R* is easier to compile/maintain/install
  2. Visualization in *R* is as good or better than Python
  3. Most importantly—*statisticians contribute directly to R*
  4. Because of this, many Python users rely on *R*
- *R* facilitates reproducible research:
  1. CRAN Task View (link)
  2. *Biostatistics (link)*
  3. *jstatsoft.org*
- *R* assists reproducible medical image analysis but it's still effortful to do it correctly. See our comments in Frontiers.

Why use *R* in medical imaging?

- It's free — not "free" like Matlab/SPM but really free
  - It is the *de facto* standard for statistical computing
  - a New York Times article from 2009 estimated that there are at least 250,000 active R users
- Why not use Python?
  1. IMHO, *R* is easier to compile/maintain/install
  2. Visualization in *R* is as good or better than Python
  3. Most importantly—*statisticians contribute directly to R*
  4. Because of this, many Python users rely on *R*
- *R* facilitates reproducible research:
  1. CRAN Task View (link)
  2. *Biostatistics (link)*
  3. *jstatsoft.org*
- *R* assists reproducible medical image analysis but it's still effortful to do it correctly. See our comments in Frontiers.

Why use *R* in medical imaging?

- It's free — not "free" like Matlab/SPM but really free
  - It is the *de facto* standard for statistical computing
  - a New York Times article from 2009 estimated that there are at least 250,000 active R users
- Why not use Python?
  1. IMHO, *R* is easier to compile/maintain/install
  2. Visualization in *R* is as good or better than Python
  3. Most importantly—*statisticians contribute directly to R*
  4. Because of this, many Python users rely on *R*
- *R* facilitates reproducible research:
  1. CRAN Task View (link)
  2. *Biostatistics (link)*
  3. *jstatsoft.org*
- *R* assists reproducible medical image analysis but it's still effortful to do it correctly. See our comments in Frontiers.

Why use *R* in medical imaging?

- It's free — not "free" like Matlab/SPM but really free
  - It is the *de facto* standard for statistical computing
  - a New York Times article from 2009 estimated that there are at least 250,000 active R users
- Why not use Python?
  1. IMHO, *R* is easier to compile/maintain/install
  2. Visualization in *R* is as good or better than Python
  3. Most importantly—*statisticians contribute directly to R*
  4. Because of this, many Python users rely on *R*
- *R* facilitates reproducible research:
  1. CRAN Task View (link)
  2. *Biostatistics (link)*
  3. *jstatsoft.org*
- *R* assists reproducible medical image analysis but it's still effortful to do it correctly. See our comments in Frontiers.

Why use *R* in medical imaging?

- It's free — not "free" like Matlab/SPM but really free
  - It is the *de facto* standard for statistical computing
  - a New York Times article from 2009 estimated that there are at least 250,000 active R users
- Why not use Python?
  1. IMHO, *R* is easier to compile/maintain/install
  2. Visualization in *R* is as good or better than Python
  3. Most importantly—*statisticians contribute directly to R*
  4. Because of this, many Python users rely on *R*
- *R* facilitates reproducible research:
  1. CRAN Task View (link)
  2. *Biostatistics (link)*
  3. *jstatsoft.org*
- *R* assists reproducible medical image analysis but it's still effortful to do it correctly. See our comments in Frontiers.

Why use *R* in medical imaging?

- It's free — not "free" like Matlab/SPM but really free
  - It is the *de facto* standard for statistical computing
  - a New York Times article from 2009 estimated that there are at least 250,000 active R users
- Why not use Python?
  1. IMHO, *R* is easier to compile/maintain/install
  2. Visualization in *R* is as good or better than Python
  3. Most importantly—*statisticians contribute directly to R*
  4. Because of this, many Python users rely on *R*
- *R* facilitates reproducible research:
  1. CRAN Task View (link)
  2. *Biostatistics (link)*
  3. *jstatsoft.org*
- *R* assists reproducible medical image analysis but it's still effortful to do it correctly. See our comments in Frontiers.

Why use *R* in medical imaging?

- It's free — not "free" like Matlab/SPM but really free
  - It is the *de facto* standard for statistical computing
  - a New York Times article from 2009 estimated that there are at least 250,000 active R users
- Why not use Python?
  1. IMHO, *R* is easier to compile/maintain/install
  2. Visualization in *R* is as good or better than Python
  3. Most importantly—*statisticians contribute directly to R*
  4. Because of this, many Python users rely on *R*
- *R* facilitates reproducible research:
  1. CRAN Task View (link)
  2. *Biostatistics (link)*
  3. *jstatsoft.org*
- *R* assists reproducible medical image analysis but it's still effortful to do it correctly. See our comments in Frontiers.

Why use *R* in medical imaging?

- It's free — not "free" like Matlab/SPM but really free
  - It is the *de facto* standard for statistical computing
  - a New York Times article from 2009 estimated that there are at least 250,000 active R users
- Why not use Python?
  1. IMHO, *R* is easier to compile/maintain/install
  2. Visualization in *R* is as good or better than Python
  3. Most importantly—*statisticians contribute directly to R*
  4. Because of this, many Python users rely on *R*
- *R* facilitates reproducible research:
  1. CRAN Task View (link)
  2. *Biostatistics (link)*
  3. *jstatsoft.org*
- *R* assists reproducible medical image analysis but it's still effortful to do it correctly. See our comments in Frontiers.

Why use *R* in medical imaging?

- It's free — not "free" like Matlab/SPM but really free
  - It is the *de facto* standard for statistical computing
  - a New York Times article from 2009 estimated that there are at least 250,000 active R users
- Why not use Python?
  1. IMHO, *R* is easier to compile/maintain/install
  2. Visualization in *R* is as good or better than Python
  3. Most importantly—*statisticians contribute directly to R*
  4. Because of this, many Python users rely on *R*
- *R* facilitates reproducible research:
  1. CRAN Task View (link)
  2. *Biostatistics (link)*
  3. *jstatsoft.org*
- *R* assists reproducible medical image analysis but it's still effortful to do it correctly. See our comments in Frontiers.

Why use *R* in medical imaging?

- It's free — not "free" like Matlab/SPM but really free
  - It is the *de facto* standard for statistical computing
  - a New York Times article from 2009 estimated that there are at least 250,000 active R users
- Why not use Python?
  1. IMHO, *R* is easier to compile/maintain/install
  2. Visualization in *R* is as good or better than Python
  3. Most importantly—*statisticians contribute directly to R*
  4. Because of this, many Python users rely on *R*
- *R* facilitates reproducible research:
  1. CRAN Task View (link)
  2. *Biostatistics (link)*
  3. *jstatsoft.org*
- *R* assists reproducible medical image analysis but it's still effortful to do it correctly. See our comments in Frontiers.

Why use *R* in medical imaging?

- It's free — not "free" like Matlab/SPM but really free
  - It is the *de facto* standard for statistical computing
  - a New York Times article from 2009 estimated that there are at least 250,000 active R users
- Why not use Python?
  1. IMHO, *R* is easier to compile/maintain/install
  2. Visualization in *R* is as good or better than Python
  3. Most importantly—*statisticians contribute directly to R*
  4. Because of this, many Python users rely on *R*
- *R* facilitates reproducible research:
  1. CRAN Task View (link)
  2. *Biostatistics (link)*
  3. *jstatsoft.org*
- *R* assists reproducible medical image analysis but it's still effortful to do it correctly. See our comments in Frontiers.

Why use *R* in medical imaging?

- It's free — not "free" like Matlab/SPM but really free
  - It is the *de facto* standard for statistical computing
  - a New York Times article from 2009 estimated that there are at least 250,000 active R users
- Why not use Python?
  1. IMHO, *R* is easier to compile/maintain/install
  2. Visualization in *R* is as good or better than Python
  3. Most importantly—*statisticians contribute directly to R*
  4. Because of this, many Python users rely on *R*
- *R* facilitates reproducible research:
  1. CRAN Task View (link)
  2. *Biostatistics (link)*
  3. *jstatsoft.org*
- *R* assists reproducible medical image analysis but it's still effortful to do it correctly. See our comments in Frontiers.

Set up for Medical Image Analysis

Let's assume you downloaded the latest 3.x version of *R* from CRAN.
We now open *R* and install MIA-relevant packages.

```
pkgnames <- c("visreg", "boot", "rgl", "knitr", "ggplot2", "oro.nifti",
    "candisc", "pheatmap")
k <- length(pkgnames)
```

Next actually install the packages.

```
install.packages(pkgnames)
```

We installed 8 packages. (knitr lets us use `\Sexpr{}` to refer to *R* variables in LaTeX.)

Install *ANTsR*

ANTs + *R*

- Operating System: Linux, OSX
- See: Install ANTsR (link)
- Will install ⋆everything⋆ you need if you want it to, including R .... otherwise will just intall *ANTsR* dependencies/utils.

OSX NOTES

- Requires: Xcode (link) and its command line tools (google install instructions)
- Requires: a clean Homebrew ("brew doctor" does not complain)
- you may want to comment out lines like: brew install ...X... if you already have software X around.

What is *knitr* ?

- Yihui Xie's system *knitr* for making documents that compute
- *knitr* lets you write a document that employs *R* directly
- *R* evaluates code when the document is compiled
- The user controls when this does / does not happen ...
- Creates, figures, statistics etc that are embedded in rst, html, latex, pretty much any common document format is doable.

What is *knitr* ?

- Yihui Xie's system *knitr* for making documents that compute
- *knitr* lets you write a document that employs *R* directly
- *R* evaluates code when the document is compiled
- The user controls when this does / does not happen ...
- Creates, figures, statistics etc that are embedded in rst, html, latex, pretty much any common document format is doable.

What is *knitr* ?

- Yihui Xie's system *knitr* for making documents that compute
- *knitr* lets you write a document that employs *R* directly
- *R* evaluates code when the document is compiled
- The user controls when this does / does not happen ...
- Creates, figures, statistics etc that are embedded in rst, html, latex, pretty much any common document format is doable.

What is *knitr* ?

- Yihui Xie's system *knitr* for making documents that compute
- *knitr* lets you write a document that employs *R* directly
- *R* evaluates code when the document is compiled
- The user controls when this does / does not happen ...
- Creates, figures, statistics etc that are embedded in rst, html, latex, pretty much any common document format is doable.

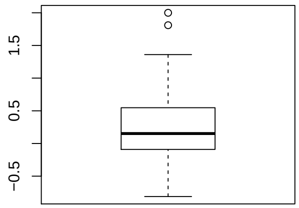
What is *knitr* ?

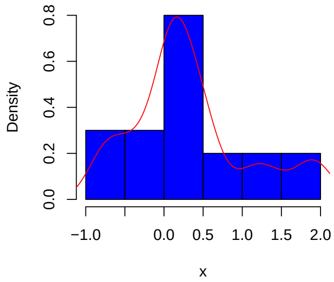
- Yihui Xie's system *knitr* for making documents that compute
- *knitr* lets you write a document that employs *R* directly
- *R* evaluates code when the document is compiled
- The user controls when this does / does not happen ...
- Creates, figures, statistics etc that are embedded in rst, html, latex, pretty much any common document format is doable.

Simple knitr Example

Use knitr to make a couple of plots in our code "chunk":

```
x <- rnorm(20)
boxplot(x)
hist(x, main = "", col = "blue", probability = TRUE)
lines(density(x), col = "red")
```





R ⋆Very⋆ Basics

R organizes data with dataframes, vectors, matrices and arrays (matrices with $\geq$ 3 dimensions).
These can contain missing variables - but you must be careful about type!

```
as.numeric(as.character(c("0.5", 0.1, 0.6, "A")))

## Warning:  NAs introduced by coercion

## [1] 0.5 0.1 0.6  NA
```

A data frame is used for storing data tables. It is a list of vectors of equal length.

R ⋆Very⋆ Basics 2

mtcars is a built-in *R* dataframe

```
mtcars[c(1, 13, 28), 1:6]

##                 mpg cyl  disp  hp drat    wt
## Mazda RX4      21.0   6 160.0 110 3.90 2.620
## Merc 450SL     17.3   8 275.8 180 3.07 3.730
## Lotus Europa   30.4   4  95.1 113 3.77 1.513
```

We analyze the relationship between MPG and other variables.

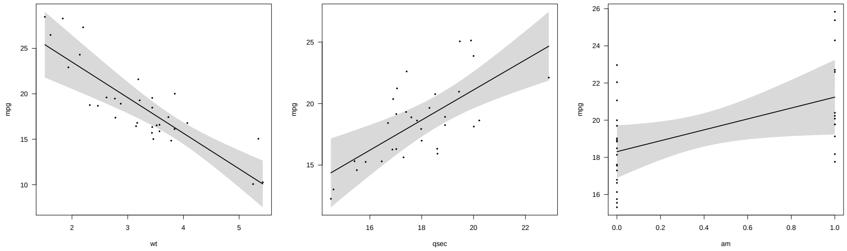
```
myform <- paste(colnames(mtcars)[2:ncol(mtcars)], collapse = "+")
myform <- as.formula(paste("mpg~", myform))
mdl <- lm(myform, data = mtcars)
mdla <- stepAIC(mdl, direction = c("both"))
```

R ⋆Very⋆ Basics 3

```
##
## Call:
## lm(formula = mdla$call$formula, data = mtcars)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.481 -1.556 -0.726  1.411  4.661
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    9.618      6.960    1.38  0.17792
## wt            -3.917      0.711   -5.51    7e-06 ***
## qsec           1.226      0.289    4.25  0.00022 ***
## am             2.936      1.411    2.08  0.04672 *
## ---
## Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
##
## Residual standard error: 2.46 on 28 degrees of freedom
## Multiple R-squared:  0.85, Adjusted R-squared:  0.834
## F-statistic: 52.7 on 3 and 28 DF,  p-value: 1.21e-11
```

R ⋆Very⋆ Basics: Draw a Picture

```
mdl <- lm(mdla$call$formula, data = mtcars)
visreg(mdl, xvar = "wt")
visreg(mdl, xvar = "qsec")
visreg(mdl, xvar = "am")
```

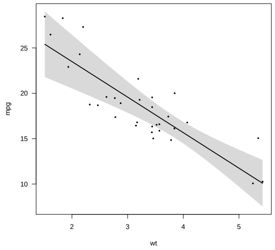


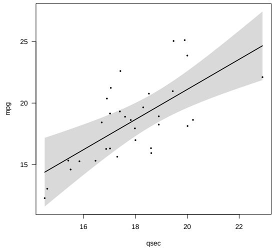
Oops!

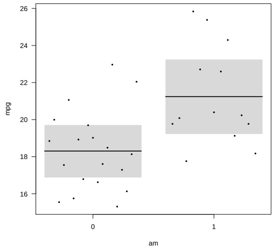
Transmission type should be a factor.

R ⋆Very⋆ Basics: Draw a Picture Fix

```
mtcars$am <- as.factor(mtcars$am)
mdl <- lm(mdla$call$formula, data = mtcars)
visreg(mdl, xvar = "wt")
visreg(mdl, xvar = "qsec")
visreg(mdl, xvar = "am")
```







This is better ...

R ⋆Very⋆ Basics: Draw a Picture 3

```r
coplot(mpg ~ wt | qsec, data = mtcars, panel = panel.smooth,
    rows = 1)
```

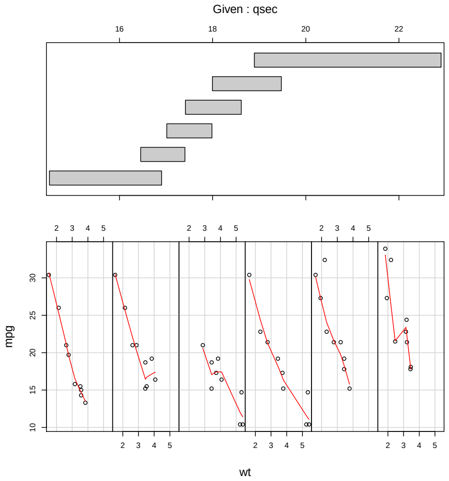
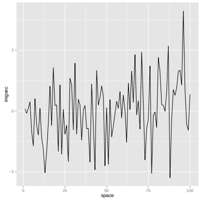


Image Input/Output in *R*

Data Representation in *R*

Represent an image as a *vector* (more on this later). This vector may be derived from a 2 or 3D array of spatially related voxels.

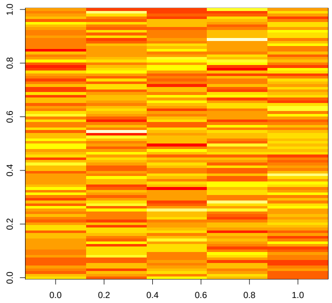
```r
nvox <- 100
imgvec <- rnorm(nvox)
mydat <- data.frame(space = 1:nvox, imgvec = imgvec)
ggplot(data = mydat, aes(x = space, y = imgvec, group = 1)) +
    geom_line()
```



Data Representation in *R*

Represent an image set as a *matrix*.

```
nSubjectsOrTimePoints <- 5
imgmat <- matrix(rep(NA, nSubjectsOrTimePoints * nvox), ncol = nvox)
for (i in 1:nSubjectsOrTimePoints) {
    imgmat[i, ] <- rnorm(nvox)
}
image(imgmat)  # try antsImageWrite( as.antsImage( imgmat ), imgmat.mha )
```



Reading Images

Read em and weep

```
fn <- getANTsRData("ch2", usefixedlocation = FALSE)

## [1] "checksum failure"

print(fn)

## NULL

# oro.nifti
colin <- readNIfTI(fn)

## Error:  File(s) not found!

# antsr
colina <- antsImageRead(fn, 3)

## [1] "filename argument must be of class character and have length 1"
```

Quickly Show Images by oro.nifti

```
orthographic(as.array(colina), oma = rep(2, 4))

## Error:  error in evaluating the argument 'x' in selecting a
method for function 'orthographic':  Error in
as.array.default(colina) :  attempt to set an attribute on NULL
## Calls:  as.array -> as.array -> as.array.default
```

Quickly Show Images w/ANTsR

```r
fn <- "figure/antsrviz.jpeg"
plotANTsImage(as.antsImage(colin), slices = "50x140x5", outname = fn)

## Error:  error in evaluating the argument 'object' in selecting a
method for function 'as.antsImage':  Error:  object 'colin' not
found
```

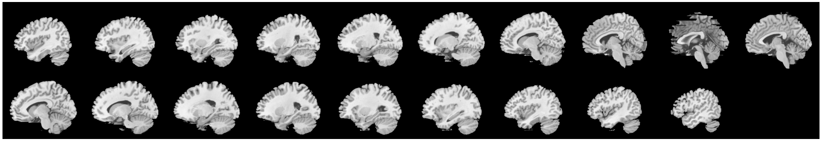


Figure : The *ANTsR* multi-slice output.

Convert an Image to a Vector

Use *ANTsR* to convert an image to a vector.

```r
imgvec <- colina[colina > 50]
print(length(imgvec))
```

Use oro.nifti to convert an image to a vector.

```r
imgvec <- colin[colin > 50]

## Error:  object 'colin' not found

print(length(imgvec))
```

Both packages enable similar functionality in terms of accessing / converting images to vectors. *ANTsR* allows I/O to files other than nifti such as meta, jpg, dicom, etc, anything ITK reads/writes.

Quantifying Images in *R*

Image Quantification with *R*

It is possible to implement full processing pipelines with *R* for submission to distributed computing systems ...
My knowledge is limited to *ANTsR* .

ANTsR based image quantification

- antsRegistration example (link)
- Atropos segmentation (link)
- phantom population study (link)

The Basic Toolset from outside *R*

Registration: Data is in Examples/Data

```
ANTS 2 -m  CC[r16slice.nii.gz,r64slice.nii.gz,1,4]

 -t SyN[0.25]  -r Gauss[3,0] -o TEST -i 50x40x30
```

Segmentation

```
Atropos -d 2 -a r16slice.nii.gz -x r16mask.nii.gz

-m [0.1,1x1]   -c [10,0]  -i kmeans[3]

-o [Output.nii.gz,Output_prob_%02d.nii.gz]
```

Template building

```
 bash buildtemplateparallel.sh -d 3 -m 30x50x20

-t GR  -s CC -c 1 -o OutPrefix  *ImageName*T1x.nii.gz
```

R Statistical Methods for Imaging

Basic Linear Regression

This is a simple regression study that associates diagnosis (dx) with a local Jacobian-based volume measurement.
We also look at global volume.

```r
predictor <- as.factor(read.csv("data/phantpredictors.csv")$dx)
gvol <- read.csv("data/globalvols.csv")
attach(gvol)
mdl <- lm(vol ~ predictor)
```

This is simulated data ....

Basic Linear Regression Output

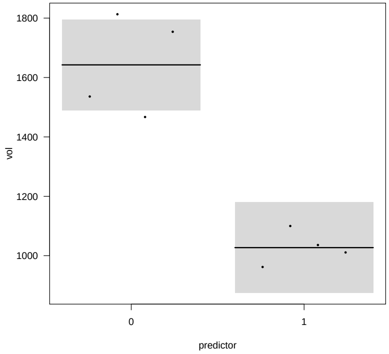
```
summary(mdl)

##
## Call:
## lm(formula = vol ~ predictor)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -175.50  -75.56   -3.75   82.44  170.50
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1642.5       62.4   26.31    2e-07 ***
## predictor1    -615.2       88.3   -6.97  0.00043 ***
## ---
## Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
##
## Residual standard error: 125 on 6 degrees of freedom
## Multiple R-squared:  0.89, Adjusted R-squared:  0.872
## F-statistic: 48.6 on 1 and 6 DF,  p-value: 0.000434
```

Basic Linear Regression Visualization

visreg has easy to use "natural" visualizations for regression ...

```
visreg(mdl)
```



Next apply the global test to the voxelwise morphometry case.

.... voxel-wise statistics in R

```r
mask <- antsImageRead("data/phantmask.nii.gz", 2)
logjac <- read.csv("data/phantomGlogjacs.csv")  # a population of images

attach(logjac)
nvox <- ncol(logjac)
pvals <- rep(NA, nvox)
for (x in c(1:nvox)) {
    voxels <- logjac[, x]
    lmres <- summary(lm(voxels ~ predictor))
    coeff <- coefficients(lmres)
    pval <- coeff[2, 4]
    pvals[x] <- pval
}
qvals <- p.adjust(pvals, method = "BH")
print(min(qvals))
pvali <- antsImageClone(mask)
pvali[mask > 0] <- 1 - qvals
plotANTsImage(mask, functional = list(pvali), threshold = "0.99x1",
    outname = "figure/lmreg.jpeg")
```

Exercise: What happens when you include globalvol as a covariate?

Visualizing voxel-wise statistics in R

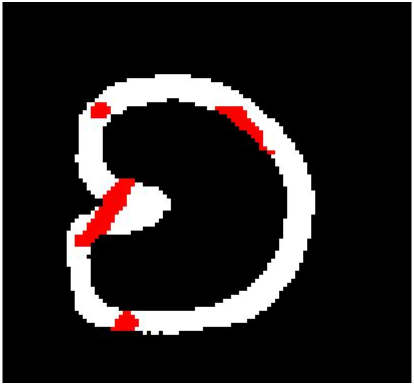


Figure : The regression solution p-values thresholded at 0.01 FDR-corrected.

.... multivariate statistics in R

```r
continuousDX <- 1 - as.numeric(predictor)
continuousDX2 <- gvol
mypreds <- as.matrix(cbind(continuousDX, continuousDX2))
sccan <- sparseDecom2(inmatrix = list(as.matrix(logjac), mypreds),
    inmask = c(mask, NA), sparseness = c(0.25, -1), nvecs = 1,
    its = 2, smooth = 1, perms = 200)
sccansol <- sccan$eig1[[1]]
sccansol[mask > 0] <- sccansol[mask > 0]/max(sccansol[mask >
    0])
plotANTsImage(mask, functional = list(sccansol), threshold = "0.05x1",
    outname = "figure/mvarreg.jpeg")
```

Exercise: What happens when you include globalvol as a covariate?

Visualizing multivariate statistics in R

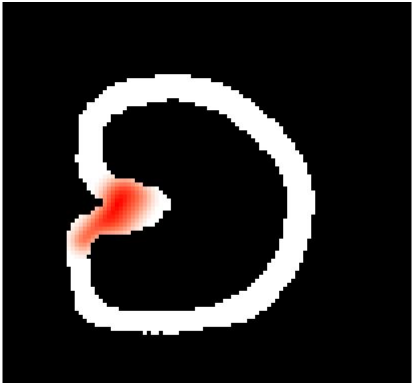


Figure : The sccan solution with p-value 0.

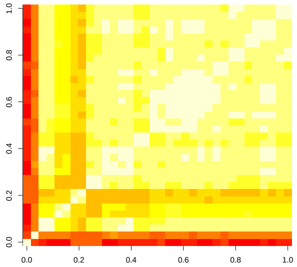
ANOVA 1

How do we assess the importance of multiple predictors acting together within classic regression?

```
nki <- read.csv("data/labelresultsN.csv")
print(names(nki)[1:8])

## [1] "ID"      "SITE"    "SEX"     "AGE"     "VOLUME"
## [6] "LABEL_1" "LABEL_2" "LABEL_3"

image(cor(as.matrix(nki[, 4:37])))
```



ANOVA 2

How do we assess the importance of multiple predictors acting together within classic regression?

```
mdl1 <- lm(LABEL_14 ~ SEX + VOLUME, data = nki)
mdl2 <- lm(LABEL_14 ~ SEX + VOLUME + AGE + I(AGE^2), data = nki)
print(anova(mdl1, mdl2))

## Analysis of Variance Table
##
## Model 1: LABEL_14 ~ SEX + VOLUME
## Model 2: LABEL_14 ~ SEX + VOLUME + AGE + I(AGE^2)
##   Res.Df  RSS Df Sum of Sq    F  Pr(>F)
## 1    183 44.8
## 2    181 34.9  2       9.9 25.7 1.5e-10 ***
## ---
## Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
## pdf
##   2
```

How do we assess the importance of multiple predictors acting together within classic regression?

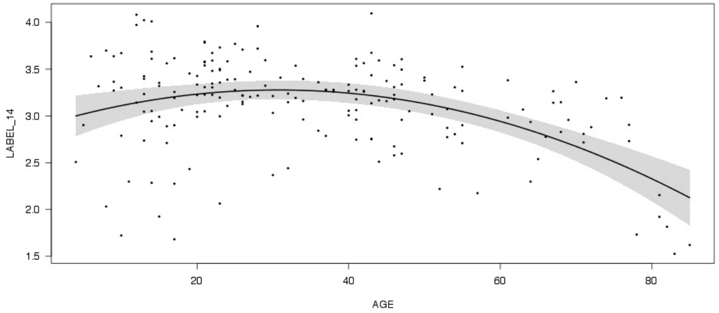


Figure : The quadratic regression of age against thickness while controlling for gender and brain volume

R Multivariate Methods for "Big Data"

first: some brief theory

What is multiple regression?

The solution to a quadratic minimization problem:

Multiple Regression

$$\|y - X\beta\|^2 + \lambda\|\beta\|^2$$

Solved by ordinary least squares methods:

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

with theory for turning β entries into "significance" measurements.

The "ridge" penalty is useful if $p >> n$.

Principal Component Analysis

Also the solution to a quadratic minimization problem:

PCA: U, V minimize reconstruction error:

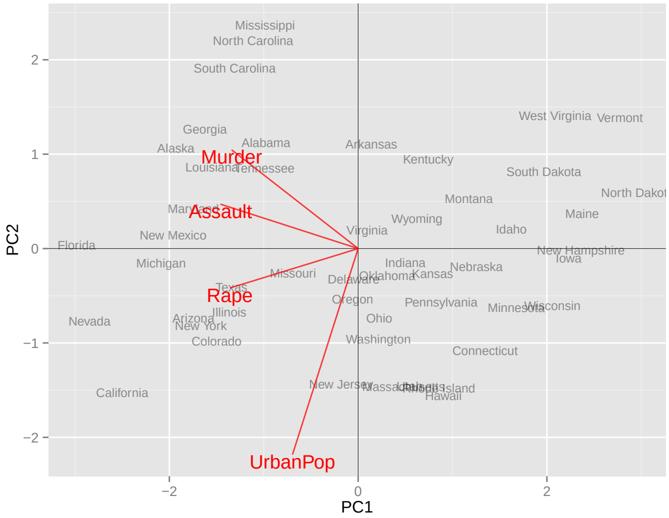
$$\|X - UV^T\|^2 + \sum_k \lambda_k \|V_k\|_1$$

Each of the columns of X is a linear combination of the columns of U. Easy solution in *R* (w/o penalties):

```
mysolution = svd(X)  # or prcomp(X) if X not centered
```

The ℓ_1 penalty is useful if $p >> n$.

Principal Component Analysis Example



Canonical Correlation Analysis

CCA Generalizes Multiple Regression

$$YV + \sum_k \lambda_k \|V_k\|_1 \propto XW + \sum_k \gamma_k \|W_k\|_1$$

where Y, V, X, W are matrices and V, W are canonical variates (the CCA solutions). Also easy in *R* (SVD used internally):

```
enginedata <- mtcars[, c(2, 3, 4, 11)]
outputdata <- mtcars[, c(1, 7)]
mycca <- cancor(enginedata, outputdata)
```

CCA is "symmetric" in that the sets X and Y have equivalent status. A truly multivariate multiple regression.

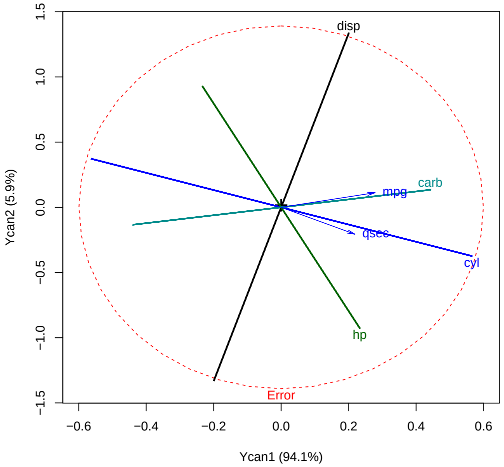
Canonical Correlation Analysis Visualization 1

```
print(mycca)

##
## Canonical correlation analysis of:
##   4   X  variables:  cyl, disp, hp, carb
##   with  2   Y  variables:  mpg, qsec
##
##     CanR CanRSQ  Eigen percent    cum                scree
## 1 0.9319 0.8684 6.5988   94.13  94.13 ********************
## 2 0.5399 0.2915 0.4115    5.87 100.00 *
##
## Test of H0: The canonical correlations in the
## current row and all that follow are zero
##
##    CanR WilksL    F df1 df2 p.value
## 1 0.932  0.093 14.8   8  52  0.0000
## 2 0.540  0.708  3.7   3  27  0.0237
```

Canonical Correlation Analysis Visualization 2

```
heplot(mycca, xpd = TRUE, scale = 0.3)
```



Sparse multiple regression/PCA/CCA

Any of the methods can be made sparse by enforcing the penalties previously highlighted in blue.

Sparse Optimization

- Formulate the problem as a constrained optimization.
- Identify the gradient descent solution—*without sparseness*.
- Use projected gradient descent to solve the optimization—*with sparseness*.
- In imaging, other constraints are valuable too.

PBAC: R ready medical imaging data.

We have training (90084)/testing (90084) data images + psychometrics and analyze the relationship between gray matter and cognition.

```
pbacTRcog[c(1, 13, 28), 1:6]

##    age edu mmse fluency_adj dig_fwd_adj dig_bwd_adj
## 1   72  18   24         4.5         2.5           4
## 13  55  17   29         5.0         2.0           3
## 28  51  16   16         3.0         0.5           1

# also pbac imaging data comes from this mask
mask <- antsImageRead(list.files(path = "./data", pattern = glob2rx("gmask_
    full.names = T), 3)
# with anatomical labels
pbacaal <- antsImageRead(list.files(path = "./data", pattern = glob2rx("pba
    full.names = T), 3)
data("aal", package = "ANTsR")  # description of aal
```

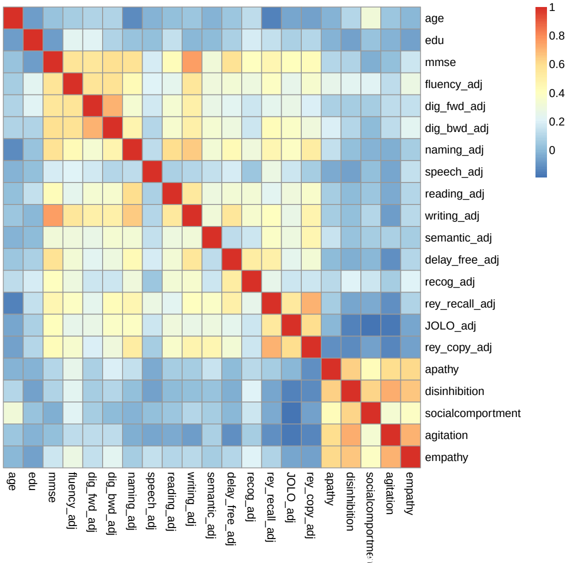
R Multivariate - PBAC ROIs

```
inmask <- mask > 0.5
mylabs <- sort(unique(pbacaal[inmask & pbacaal > 0.5 & pbacaal <
    91 & pbacaal != 51 & pbacaal != 52 & pbacaal != 53 & pbacaal !=
    54]))
roimatrix <- matrix(rep(NA, length(mylabs) * nrow(pbacTRimg)),
    ncol = length(mylabs))
for (i in 1:length(mylabs)) {
    # get vector for this label
    labelVec <- as.numeric(pbacaal[inmask] == mylabs[i])
    roimatrix[, i] <- pbacTRimg %*% (labelVec/sum(labelVec))
}
colnames(roimatrix) <- aal$label_name[mylabs]
mydf <- data.frame(pbacTRcog, roimatrix)
```

Next we will analyze these ROIs and their relationship with demographics.

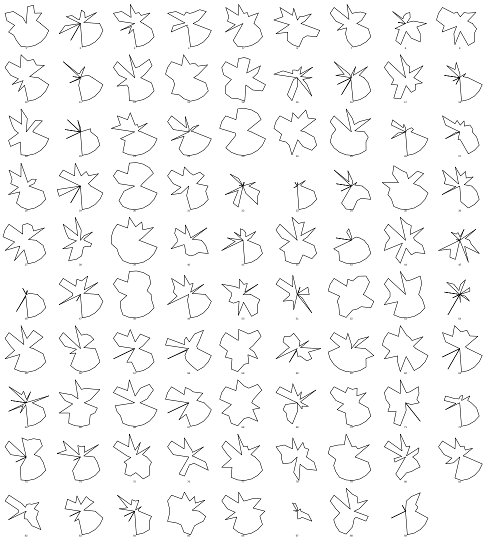
R Multivariate: Inspect Data - PBAC cog

```
pheatmap(cor(pbacTRcog), cluster_rows = F, cluster_cols = F)
```



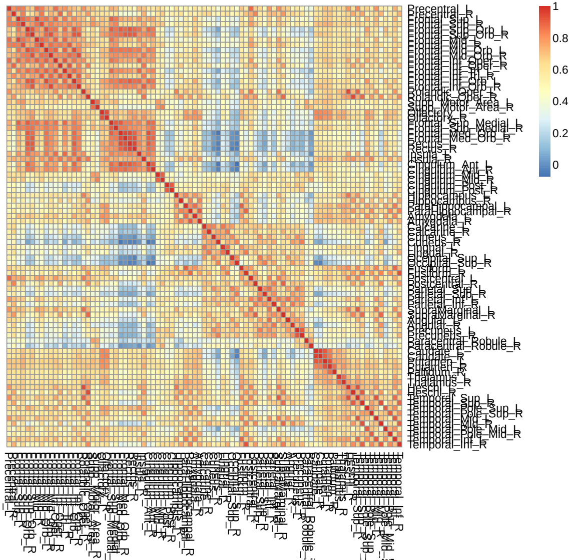
R Multivariate: Inspect Data - PBAC Cog Constellation Plot

Brain Constellation Map of PBAC Cognition

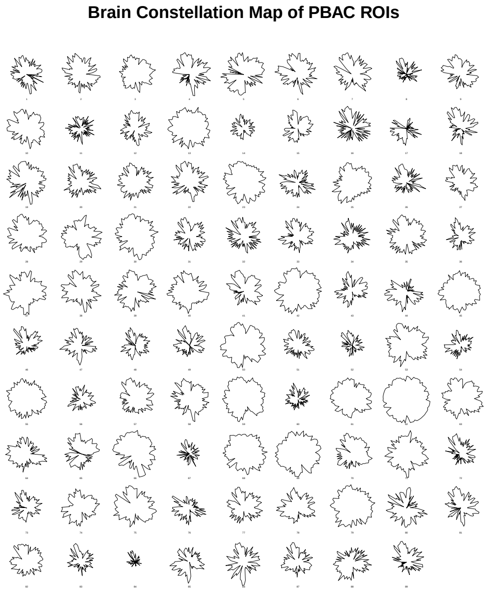


R Multivariate: Inspect Data - PBAC ROI

```r
pheatmap(cor(roimatrix), cluster_rows = F, cluster_cols = F)
```



R Basics: Inspect Data - PBAC ROI Constellation Plot



1200 Subject Constellation Plot

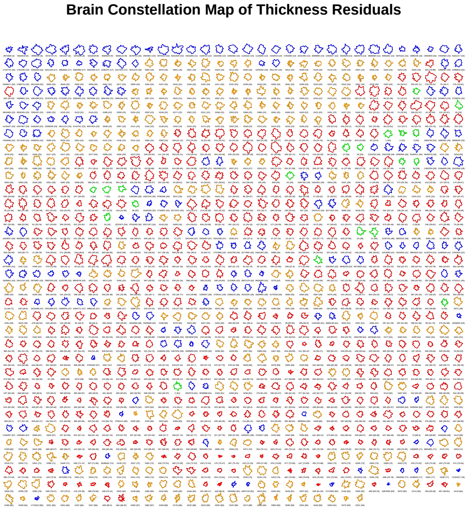
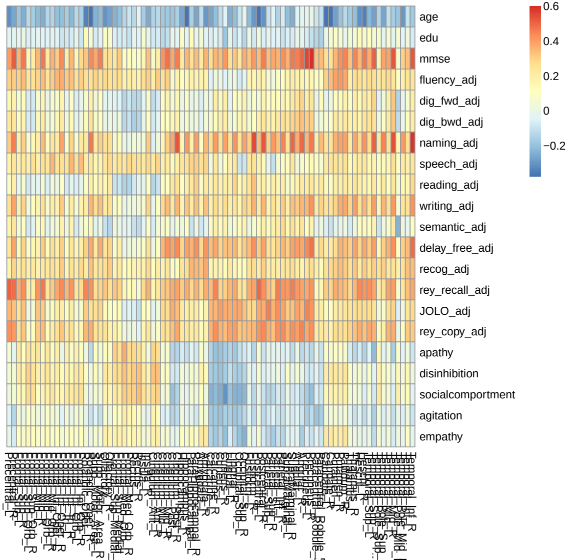


Figure : Data-inspection for a large-scale study.

R Multivariate: Inspect Data - PBAC cog ↔ ROI

```r
pheatmap(cor(pbacTRcog, roimatrix), cluster_rows = F, cluster_cols = F)
```



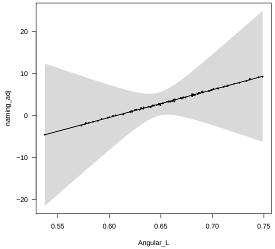
R Multivariate - PBAC ROI Study

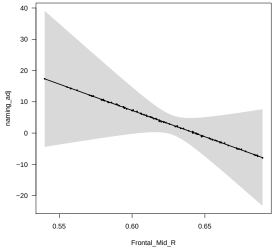
```
myform <- paste(colnames(roimatrix), collapse = "+")
myform <- as.formula(paste("naming_adj~", myform, "+edu"))
mydf <- data.frame(pbacTRcog, roimatrix)
row.names(mydf) <- paste(c(1:nrow(pbacTRcog)), "_", as.character(pbacTRcog$
    sep = "")
mdl <- lm(myform, data = mydf)
mdla <- stepAIC(mdl, direction = c("forward"), k = 20, steps = 20)
ageregions <- gsub("_", "", as.character(mdla$call$formula)[3])
```

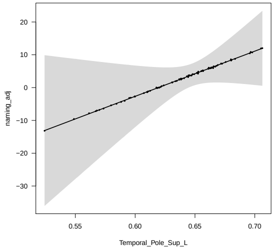
Various brain regions, together, predict naming_adj ... PrecentralL + PrecentralR + FrontalSupL + FrontalSupR + FrontalSupOrbL + FrontalSupOrbR + FrontalMidL + FrontalMidR + FrontalMidOrbL + FrontalMidOrbR + FrontalInfOperL + FrontalInfOperR + FrontalInfTriL + FrontalInfTriR + FrontalInfOrbL + FrontalInfOrbR + RolandicOperL + RolandicOperR + SuppMotorAreaL + SuppMotorAreaR + OlfactoryL + OlfactoryR + FrontalSupMedialL + FrontalSupMedialR + FrontalMedOrbL + FrontalMedOrbR + RectusL + RectusR + InsulaL + InsulaR + CingulumAntL + CingulumAntR + CingulumMidL +

R Multivariate: Draw a Picture - PBAC

```r
visreg(mdla, xvar = "Angular_L")
visreg(mdla, xvar = "Frontal_Mid_R")
visreg(mdla, xvar = "Temporal_Pole_Sup_L")
```

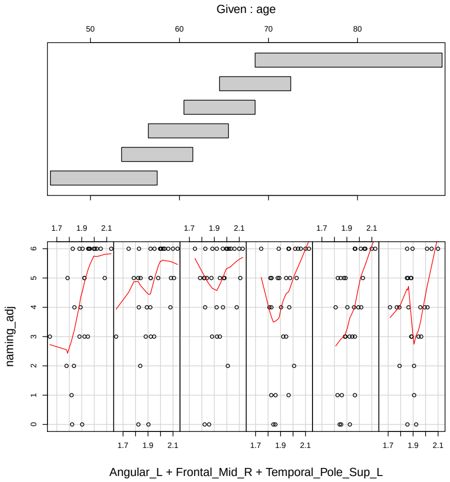






R Multivariate: PBAC Draw a Picture 2

```
coplot(naming_adj ~ Angular_L + Frontal_Mid_R + Temporal_Pole_Sup_L |
    age, data = mydf, panel = panel.smooth, rows = 1)
```



R Multivariate: PBAC SCCAN

Run SCCAN between raw GM data and cognition

```
mysccan <- sparseDecom2(inmatrix = list(as.matrix(pbacTRcog),
    pbacTRimg), inmask = c(NA, mask), smooth = 1, sparseness = c(-0.07,
    0.2), nvecs = nv, its = 3, perms = 0, cthresh = c(0, 250))
```

```
## gm ~ mmse + rey_recall_adj
## [1] "Train Correlation: 1 0.664181703317975"
## gm ~ rey_recall_adj + rey_copy_adj
## [1] "Train Correlation: 2 0.571539338864852"
## gm ~ naming_adj + delay_free_adj
## [1] "Train Correlation: 3 0.586114213255158"
## gm ~ writing_adj + JOLO_adj
## [1] "Train Correlation: 4 0.500195314403038"
## gm ~ delay_free_adj
## [1] "Train Correlation: 5 0.486444226824085"
```

R and Prediction

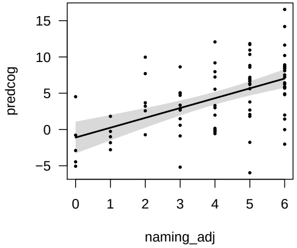
Prediction: PBAC—Cognition from ROIs

Get the testing data ROIs (code hidden).

Predict the naming from test ROI data w/ ordinary regression.

```
predcog <- predict(mdla, newdata = testdf)
print(paste("Test Correlation:", cor.test(pbacTEcog$naming_adj,
    predcog)$est))

## [1] "Test Correlation: 0.522830279263574"
```



Prediction: PBAC—Cognition from Brain

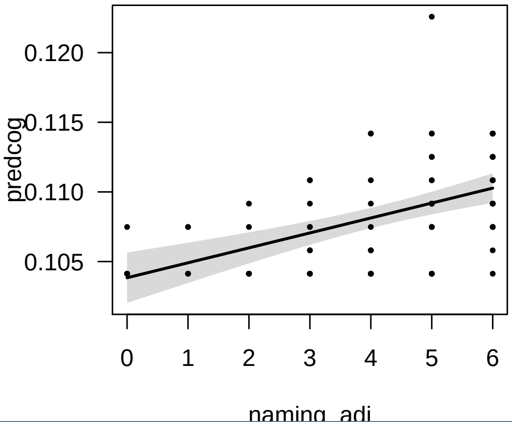
Predict the test voxel data from cognition id'd by SCCAN.

```
## gm ~ mmse + rey_recall_adj
## [1] "Test Correlation: 1 0.607638630247478"
## gm ~ rey_recall_adj + rey_copy_adj
## [1] "Test Correlation: 2 0.450943295749685"
## gm ~ naming_adj + delay_free_adj
## [1] "Test Correlation: 3 0.446546448383557"
## gm ~ writing_adj + JOLO_adj
## [1] "Test Correlation: 4 0.307284063558776"
## gm ~ delay_free_adj
## [1] "Test Correlation: 5 0.504171777768369"
```

The SCCAN model is equally predictive but much more specific.

Prediction: PBAC—Cognition from Brain

```r
predmdl <- lm(predcog ~ 1 + naming_adj, data = pbacTEcog)
visreg(predmdl)
```



Cross-Validation of Diagnosis

boot

```
dx <- as.factor(pbacTRcog$mmse < 26)
dx <- as.factor(pbacTRcog$fluency_adj < 2.6)
traindata <- data.frame(dx = dx, roimatrix)
myform <- paste("dx~", paste(colnames(roimatrix)[1:20], collapse = "+"))
mdl <- glm(as.formula(myform), data = traindata, family = "binomial")
dd <- 0
ntests <- 20
for (i in 1:ntests) dd <- dd + cv.glm(traindata, mdl, K = 5)$delta[1] *
    (1/ntests)
```

Reasonable classification rates.

```
## [1] "prediction % misclassification 6.87711587597379"
```

Prediction: BRATS Challenge

fMRI

fMRI Helper Functions 1

A function for averaging a list of images voxel-wise. Note: It's dimension-free.

```r
avgimg <- function(mylist, mask) {
    avg <- antsImageClone(mylist[[1]])
    avg[mask == 1] <- 0
    for (i in 1:length(mylist)) {
        avg[mask == 1] <- avg[mask == 1] + mylist[[i]][mask ==
            1] * 1/length(mylist)
    }
    return(avg)
}
```

fMRI Helper Functions 2

A function for computing the voxel-wise absolute difference of an image list from its average.

```r
sdimg <- function(mylist, mask) {
    avg <- avgimg(mylist, mask)
    sdi <- antsImageClone(avg)
    sdi[mask == 1] <- 0
    for (i in 1:length(mylist)) {
        sdi[mask == 1] <- sdi[mask == 1] + abs(mylist[[i]][mask ==
            1] - avg[mask == 1]) * 1/length(mylist)
    }
    return(sdi)
}
```

fMRI Helper Functions 3

A function to interleave two R numeric vectors.

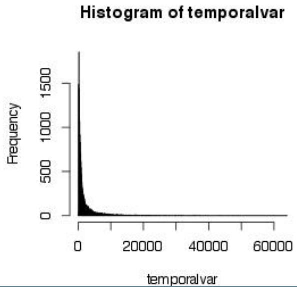
```r
interleave <- function(v1, v2) {
    ord1 <- 2 * (1:length(v1)) - 1
    ord2 <- 2 * (1:length(v2))
    c(v1, v2)[order(c(ord1, ord2))]
}
```

Why might we want this?

BOLD fMRI Motor Tasks 1

"A test-retest fMRI dataset for motor, language and spatial attention functions" — Gigascience, 2013.
Subject: 08143633

```r
fmri <- antsImageRead(fn, 4)
hrf <- hemodynamicRF(scans = dim(fmri)[4], onsets = blockfing,
    durations = rep(12, length(blockfing)), rt = 2.5)
hrf[1:4] <- NA  # first few frames are junk
myvars <- getfMRInuisanceVariables(fmri, moreaccurate = FALSE,
    maskThresh = 100)
```



BOLD fMRI Motor Tasks 2

The previous functions compute *R* friendly variables for fMRI processing: Nuisance, mean, mask, matrix. + the HRF.

```r
mat <- myvars$matrixTimeSeries
avg <- myvars$avgImage
mask <- myvars$mask
nuis <- (myvars$nuisancevariables)
print(colnames(nuis))
antsImageWrite(avg, paste(pre, "avg.nii.gz", sep = ""))
plotANTsImage(myantsimage = avg, functional = list(mask), slices = "12x20x3
    axis = 3, threshold = "0.5x1.5")
```

BOLD fMRI Motor Tasks 3

Use multiple regression to relate a task-design (convolved with HRF) to BOLD activation.

```r
globsig <- myvars$globalsignal
betas <- rep(NA, ncol(mat))
for (i in 1:ncol(mat)) {
    vox <- mat[, i]
    mdl <- lm(vox ~ hrf + globsig + motion1 + motion2 + motion3 +
        compcorr1 + compcorr2 + compcorr3, data = data.frame(nuis))
    betas[i] <- coefficients(summary(mdl))[2, 3]  # probably better way
}
betaimg <- antsImageClone(mask)  # put beta vals in image space
betaimg[mask > 0.5] <- betas
print(max(abs(betas)))  # around 10 or so
# much much faster but i havent figured out how to get
# results out easily
fastResults <- lm(mat[, 1:2] ~ hrf + myvars$globalsignal + motion1 +
    motion2 + motion3 + compcorr1 + compcorr2 + compcorr3, data = data.fram
antsImageWrite(betaimg, paste(pre, "betas.nii.gz", sep = ""))
```

BOLD fMRI Motor Tasks 4

```
gcoords <- getTemplateCoordinates(list(avg, clust), mymni, convertToTal = T
    outprefix = ofn)
print(gcoords$templatepoints)
myregion <- sub("_", "", gcoords$templatepoints$AAL[1])
```

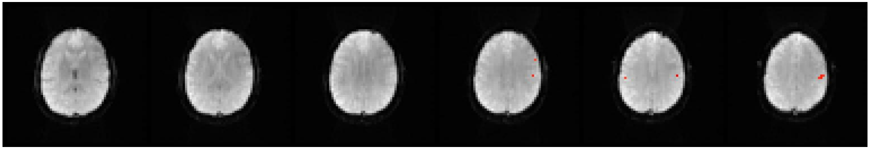


Figure : Univariate results for fingertapping include CentralSulcus.

Is that the "right" location?

BOLD fMRI Motor Tasks 5

We can look at the code for this if it is of interest ...

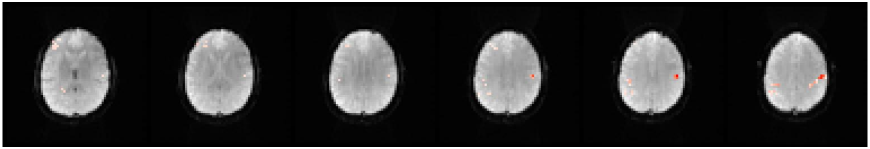


Figure : Multivariate results for fingertapping.

Exercise: Fingertapping repeatability

Repeat with subjects fmri_motor_sub1_s2.nii.gz and fmri_motor_sub2_s1.nii.gz
Evaluate overlap of signal.

BOLD fMRI Processing Inspection

Use multivariate decomposition to help understand both the data and the nuisance variables.

```r
nvecs <- 11
ff <- sparseDecom(rmat[!is.na(hrf), ], mask, 1.25/nvecs, nvecs,
    its = 5, cthresh = 5, smooth = 1, z = -0.9)
for (i in 1:nvecs) {
    print(paste("Test", i))
    mdl <- lm(ff$projections[, i] ~ cblock + myvars$globalsignal[!is.na(hrf
        motion1 + motion2 + motion3 + compcorr1 + compcorr2 +
        compcorr3, data = data.frame(nuis[!is.na(hrf), ]))
    print(summary(mdl))
}
dat <- data.frame(time = ((1:length(hrf[!is.na(hrf)])) * 2.5),
    signal = ff$projections[, 2], nuis = ff$projections[, 3],
    hrf = hrf[!is.na(hrf)])
```

BOLD Decomposition with Regression-Task

Use multivariate decomposition to help understand both the data and the nuisance variables.

```
mdl <- lm(ff$projections[, 2] ~ cblock + myvars$globalsignal[!is.na(hrf)] +
    motion1 + motion2 + motion3 + compcorr1 + compcorr2 + compcorr3,
    data = data.frame(nuis[!is.na(hrf), ]))

##                       (Intercept)
##                         0.8970783
##                            cblock
##                         0.0002709
## myvars$globalsignal[!is.na(hrf)]
##                         0.2735748
##                           motion1
##                         0.6876208
##                           motion2
##                         0.9173405
##                           motion3
##                         0.2948914
##                         compcorr1
##                         0.4551018
##                         compcorr2
```

BOLD Decomposition with Regression-Nuisance

Use multivariate decomposition to help understand both the data and the nuisance variables.

```
mdl <- lm(ff$projections[, 3] ~ cblock + myvars$globalsignal[!is.na(hrf)] +
    motion1 + motion2 + motion3 + compcorr1 + compcorr2 + compcorr3,
    data = data.frame(nuis[!is.na(hrf), ]))

##                        (Intercept)
##                          1.127e-03
##                             cblock
##                          7.895e-01
## myvars$globalsignal[!is.na(hrf)]
##                          2.129e-72
##                            motion1
##                          1.206e-02
##                            motion2
##                          1.652e-16
##                            motion3
##                          1.509e-03
##                          compcorr1
##                          4.179e-06
##                          compcorr2
```

BOLD Decomposition with Regression-Task in the Brain

Use multivariate decomposition to help understand both the data and the nuisance variables.

```r
eigimg <- ff$eigenanatomyimages[[2]]
```

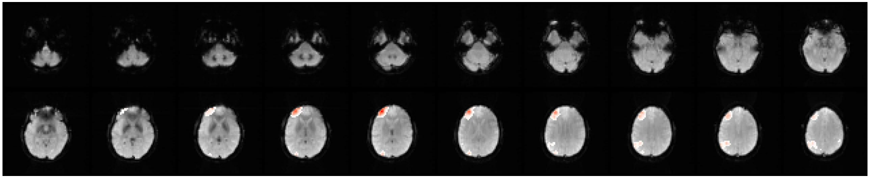


Figure : Multivariate results for fingertapping... task areas

BOLD Decomposition with Regression-Nuis in the Brain

Use multivariate decomposition to help understand both the data and the nuisance variables.

```
eigimg <- ff$eigenanatomyimages[[3]]
```

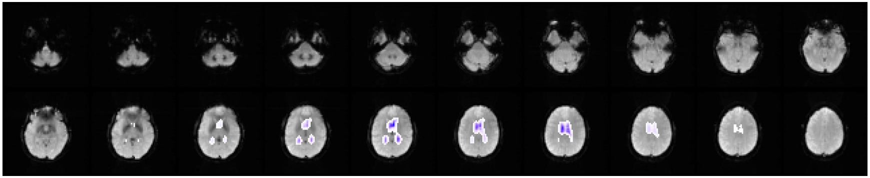


Figure : Multivariate results for fingertapping... nuis areas

BOLD fMRI Signals

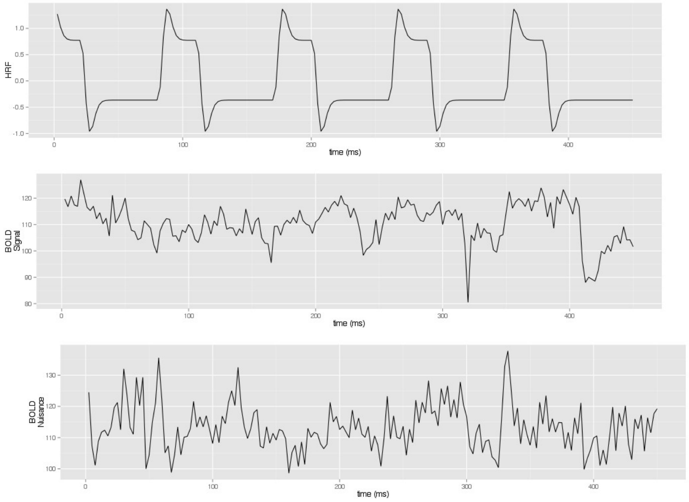


Figure : BOLD signals.

BOLD fMRI Language Tasks

Exercise: Check the code below and run the language task studies from the Gigascience article.

```
if (FALSE) {
    fmri <- antsImageRead("data/fmri_covert_verb_generation_sub1_s2.nii.gz"
        4)
    blocko = c(1, 24, 48, 72, 96, 120, 144)
    hrf <- hemodynamicRF(scans = dim(fmri)[4], onsets = blocko,
        durations = rep(12, length(blocko)), rt = 2.5)
    hrf[1:4] <- NA  # first few frames are junk
    myvars <- getfMRInuisanceVariables(fmri, moreaccurate = TRUE,
        maskThresh = 100)
    avg <- myvars$avgImage
    antsImageWrite(avg, "avg_lang.nii.gz")
    mask <- myvars$mask
    mat <- myvars$matrixTimeSeries
    # fmri2<-antsImageClone(fmri) SmoothImage(4,fmri,1,fmri2)
    # mat<-timeseries2matrix( fmri2, mask ) #
    nuis <- (myvars$nuisancevariables)
    print(colnames(nuis))
    plotANTsImage(myantsimage = avg, functional = list(mask),
        slices = "12x20x3", axis = 3, threshold = "0.5x1.5")
```

Simple ASL CBF

Arterial spin labeling (ASL) can measure cerebral blood flow (CBF) non-invasively and more directly than BOLD. It requires specialized processing techniques not widely available.

```
fns <- Sys.glob(file.path("./data/eld*nii.gz"))
asl <- antsImageRead(fns[1], 4)
perf <- aslPerfusion(asl, maskThresh = 300, moreaccurate = FALSE)
param <- list(sequence = "pcasl", m0 = perf$m0)
cbf <- quantifyCBF(perf$perfusion, perf$mask, param)

## Loading required package:  extremevalues

plotANTsImage(cbf$meancbf, slices = "5x17x3", axis = 3, outname = "figure/a
```

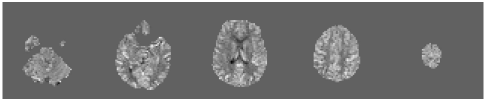


Figure : The *ANTsR* simple CBF estimate with standard regression.

fMRI Boot-Strapping

Load some data already processed.

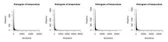
```r
fns <- Sys.glob(file.path("./data/eld*nii.gz"))
asl <- antsImageRead(fns[1], 4)
seg <- antsImageRead(fns[3], 3)
mask <- antsImageClone(seg)
mask[seg > 0] <- 1
mat <- timeseries2matrix(asl, mask)
cbflist <- list()
```

ASL-CBF estimates are unstable and subject to motion artifact.
Idea: We can try resampling methods to estimate both uncertainty and a "true" mean CBF value per voxel.

fMRI Boot-Strapping 2

Luckily, this is easy to implement in *R* .

```
for (i in 1:4) {
    timeinds <- sample(2:nrow(mat), round(nrow(mat)) * 0.3)
    timeinds <- (timeinds%%2) + timeinds
    timeinds <- interleave(timeinds - 1, timeinds)
    aslarr <- as.array(asl)
    aslarr2 <- aslarr[, , , timeinds]
    aslsub <- as.antsImage(aslarr2)
    antsCopyImageInfo(aslsub, asl)
    proc <- aslPerfusion(aslsub, mask = mask, moreaccurate = FALSE,
        dorobust = 0)
    param <- list(sequence = "pcasl", m0 = proc$m0)
    cbf <- quantifyCBF(proc$perfusion, mask, param)
    antsImageWrite(cbf$meancbf, "temp1.nii.gz")
    cbflist <- lappend(cbflist, cbf$meancbf)
}
```



fMRI Boot-Strapping 3

```
avgcbf <- avgimg(cbflist, mask)
sdi <- sdimg(cbflist, mask)
avgcbft <- antsImageClone(avgcbf)
avgcbft[sdi > 25] <- 0
plotANTsImage(avgcbft, slices = "5x17x3", axis = 3, outname = "figure/antsr
plotANTsImage(avgcbft, functional = list(sdi), color = "red",
    slices = "5x17x3", axis = 3, threshold = "20x55", outname = "figure/ant
```

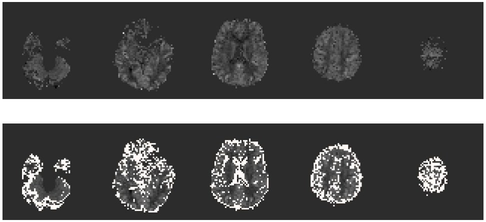


Figure : The *ANTsR* bootstrapped CBF with estimated CBF variance.

Review So Far ...

What Have We Ignored?

- More general linear models: e.g. multinomial, logit ...
- LDA, SVM, advanced visualization, etc.
- functions/packages: *pairs, glmnet, PMA, igraph* ...
- simulation — very valuable.
- too many *R* ⋆tricks⋆ to remember w/o practice.

Review So Far ...

What Have We Ignored?

- More general linear models: e.g. multinomial, logit ...
- LDA, SVM, advanced visualization, etc.
- functions/packages: *pairs, glmnet, PMA, igraph* ...
- simulation — very valuable.
- too many *R* ⋆tricks⋆ to remember w/o practice.

Review So Far ...

What Have We Ignored?

- More general linear models: e.g. multinomial, logit ...
- LDA, SVM, advanced visualization, etc.
- functions/packages: *pairs, glmnet, PMA, igraph* ...
- simulation — very valuable.
- too many *R* ⋆tricks⋆ to remember w/o practice.

Review So Far ...

What Have We Ignored?

- More general linear models: e.g. multinomial, logit ...
- LDA, SVM, advanced visualization, etc.
- functions/packages: *pairs, glmnet, PMA, igraph* ...
- simulation — very valuable.
- too many *R* ⋆tricks⋆ to remember w/o practice.

Review So Far ...

What Have We Ignored?

- More general linear models: e.g. multinomial, logit ...
- LDA, SVM, advanced visualization, etc.
- functions/packages: *pairs, glmnet, PMA, igraph* ...
- simulation — very valuable.
- too many *R* ⋆tricks⋆ to remember w/o practice.

Review So Far ...

Accomplishments!

- Image I/O & ROI analysis
- Morphometry & Regression
- Multivariate methods for Large Imaging Datasets
- Prediction examples ...
- fMRI: univariate & multivariate
- ASL Cerebral Blood Flow
- Various references to valuable *R* ⋆tricks⋆.

Review So Far ...

Accomplishments!

- Image I/O & ROI analysis
- Morphometry & Regression
- Multivariate methods for Large Imaging Datasets
- Prediction examples ...
- fMRI: univariate & multivariate
- ASL Cerebral Blood Flow
- Various references to valuable *R* ⋆tricks⋆.

Review So Far ...

Accomplishments!

- Image I/O & ROI analysis
- Morphometry & Regression
- Multivariate methods for Large Imaging Datasets
- Prediction examples ...
- fMRI: univariate & multivariate
- ASL Cerebral Blood Flow
- Various references to valuable *R* ⋆tricks⋆.

Review So Far ...

Accomplishments!

- Image I/O & ROI analysis
- Morphometry & Regression
- Multivariate methods for Large Imaging Datasets
- Prediction examples ...
- fMRI: univariate & multivariate
- ASL Cerebral Blood Flow
- Various references to valuable *R* ⋆tricks⋆.

Review So Far ...

Accomplishments!

- Image I/O & ROI analysis
- Morphometry & Regression
- Multivariate methods for Large Imaging Datasets
- Prediction examples ...
- fMRI: univariate & multivariate
- ASL Cerebral Blood Flow
- Various references to valuable *R* ⋆tricks⋆.

Review So Far ...

Accomplishments!

- Image I/O & ROI analysis
- Morphometry & Regression
- Multivariate methods for Large Imaging Datasets
- Prediction examples ...
- fMRI: univariate & multivariate
- ASL Cerebral Blood Flow
- Various references to valuable *R* $\star$tricks$\star$.

Review So Far ...

Accomplishments!

- Image I/O & ROI analysis
- Morphometry & Regression
- Multivariate methods for Large Imaging Datasets
- Prediction examples ...
- fMRI: univariate & multivariate
- ASL Cerebral Blood Flow
- Various references to valuable *R* $\star$tricks$\star$.

Example github projects for reproducible research

Example Papers based on R: SCCAN

Example Papers based on R: Eigenanatomy

Example Papers based on R: ASL-CBF

Three steps in an ASL imaging study.

- Normalization / segmentation
- Data inspection
- Analysis
- Visualization See github VisDemo

Resources for Building *R* Packages

Discussion + Future Work