INTRODUCTION TO R

ComputeFest 2012

Alex Storer

Institute for Quantitative Social Science at Harvard
astorer@iq.harvard.edu

http://www.people.fas.harvard.edu/~astorer/computefest2012/
Introduction

• Research Technology Consulting at IQSS
• Dataverse
• Research Computing Environment
• OpenScholar
• Packages for computational text analysis, multiple imputation, and statistics modeling in R
• Trainings on R, Stata, SAS

http://www.iq.harvard.edu
Goals

1. Why should anyone use R?
2. Basic Familiarity with R
3. Getting Help and Installing Packages
4. Data Input and Output
5. Making Plots
6. Performing Statistical Analysis
7. Functions, loops and building blocks of programming

A foundation to take the next step.

We'll learn the basics as we go while working on real problems.
Why should I use R?

“I already know Matlab. Why would I want to use R?”

1. Matlab is expensive.
2. R is open-source and community developed.
3. R is platform independent (and can even be run online)
4. Beautiful graphics with clear syntax
5. The most actively developed cutting-edge statistics applications can be installed *from inside R!"
Why should I use R?

“I already know Python. Why would I ever want to learn a new language that already does a subset of what Python already does?”

1. R is designed to do data-processing and statistical analysis, and it makes it easy.
2. More support in R (e.g., the Use R! series)
3. More active development in R for data analysis
4. Who are you sharing your code with?
Starting R

```
1 library(ggplot2)
2 view(diamonds)
3 summary(diamonds)
4 aveSize <- round(mean(diamonds$carat), 4)
5 clarity <- levels(diamonds$clarity)
6 p <- qplot(carat, price, data=diamonds, color=clarity,
7 xlab="Carat", ylab="Price",
8 main="Diamond Pricing")
9 > summary(diamonds$price)
10 Min.  : 0.000  1st Qu.: 4.710  Median : 5.731  Mean    : 5.731  3rd Qu.: 6.540  Max.   : 3.933
11 > aveSize <- round(mean(diamonds$Carat), 4)
12 > p <- qplot(carat, price, data=diamonds, color=clarity,
13 xlab="Carat", ylab="Price",
14 main="Diamond Pricing")
15 > format.plot(plot=p, size=23)
16```

Diamond Pricing

Price

Carat
# Syntax Notes

<table>
<thead>
<tr>
<th>R</th>
<th>Matlab</th>
<th>Python</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>x &lt;- seq(1,10)</code> # or <code>x &lt;- 1:10</code> # or <code>x = 1:10</code></td>
<td><code>x = 1:10</code> %a less flexible %version of linspace</td>
<td><code>x = range(1,11)</code> # indices start at 0</td>
</tr>
<tr>
<td><code>for (i in x) {print(&quot;hello&quot;)}</code></td>
<td><code>for (i = x) disp(&quot;hello&quot;)</code> <code>end</code></td>
<td><code>for i in x: print(&quot;hello&quot;)</code></td>
</tr>
<tr>
<td><code>foo.bar &lt;- 10</code> &gt; <code>foo.bar [1] 10</code></td>
<td><code>foo.bar = 10</code></td>
<td><code>foo.bar = 10</code></td>
</tr>
<tr>
<td></td>
<td><code>foo.bar</code> <code>foo = bar: 10</code></td>
<td><code>NameError: name 'foo' is not defined</code></td>
</tr>
</tbody>
</table>
Vectors in R

```r
> x <- c(1,1,2,3,5,8,13,21)
> length(x)
[1] 8
> x[5]
[1] 5
```

The `c()` command concatenates items into a vector of the same type!

*Note that R counts from 1.*
Logical Indexing, Growing Vectors

> x<5
[1]  TRUE  TRUE  TRUE  TRUE  FALSE  FALSE  FALSE
> x[x<5]
[1] 1 1 2 3
> x[10] = 55
> x
 [1] 1 1 2 3 5 8 13 21 NA 55
> x %% 2
[1] 1 1 0 1 1 0 1 1 1 NA 1
Exercise 1

1. Use logical indexing to find the area of Georgia. Use the included vectors state.name and state.area.

2. What happens when you combine 2 and "2" using the c() command?

3. Make a vector containing 100 numbers, and use plot to visualize a mathematical function of your choice!
One more function…

> help("foo")

> ?foo
Exercise 2

1. Which is bigger, $\log \sqrt{\pi}$ or $\sqrt{\log \pi}$?
2. What does the function rep do?
3. Make a vector containing 50 ones and call it fib
4. Use a for loop to compute the first 50 Fibonacci numbers (store them in fib)
5. What does the function table do?
6. How many of the first 50 Fibonacci numbers are divisible by 3? (Recall: a \( \%\% \) b)
7. What is the mean of the first 15 Fibonacci numbers?
**Data Frames**

- **Data frames** are the most common way to store data in R
- Like a matrix, but columns and rows can have names

```r
> head(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
```

- You can access columns by name using the `$` operator
- e.g., `count(mtcars$gear)`
Data Frames: viewing and manipulating

```r
> summary(mtcars)
> write.csv(mtcars, '/tmp/blah.csv')
> mtcars$logmpg <- log(mtcars$mpg)
> rownames(mtcars)
> colnames(mtcars)
```
Data Frames: viewing and manipulating

```r
> mtcars["Valiant",]
> mtcars["Valiant",c("hp","gear")]
> mtcars[c(1,8,3,5,2),c("hp","gear")]
> guzzlers <- subset(mtcars,mpg<15)
```
Exercise 2

1. What is the median mpg for cars with horsepower ("hp") less than 100?

2. Use the order command to print the state.name sorted by state.area

3. Use the order command and write.csv to save the mtcars dataset sorted by horsepower

4. What is the name of the car with the most horsepower out of all cars with 4 cylinders ("cyl")?
The best thing about R...

install.packages(""")

Curated Lists of Packages:
- Chemometrics and Computational Physics
- Medical Image Analysis
- Phylogenetics
- Ecological and Environmental Data

3546 Possibilities.

http://cran.r-project.org/web/views/
Example: Rainfall Anomalies

• http://jisao.washington.edu/data/doe_prec/

Goals: Make a 'movie' of rainfall anomalies on a map of the Earth and compare data in Boston and Tokyo

• This is a long example!
• We'll look a lot of R tricks along the way
• Most important: how to solve general "how do I do" problems!
Example: Rainfall Anomalies

Sub-goal: Just load the data into R

Data is in **NetCDF** format…

...there's a package for that!

```r
> install.packages("ncdf")
> library("ncdf")
```

We only need to install the package once.

We have to load it each time we wish to use it.
Example: Rainfall Anomalies

Sub-goal: Just load the data into R

Let's just get the data!

```r
> sfile <- "http://jisao.washington.edu/data/doe_prec/precipanom18511989.nc"
> download.file(sfile,"/tmp/recip.nc")
> dat <- open.ncdf("/tmp/recip.nc")
> dat
```

Notice the use of '/ ' (this can trip you up in Windows!)

```
[1] "file /tmp/recip.nc has 3 dimensions:"
[1] "lat  Size: 44"
[1] "lon  Size: 72"
[1] "time  Size: 556"
[1] "-------------------"
[1] "file /tmp/recip.nc has 1 variables:"
[1] "short data[lon, lat, time] Longname: djf, mam, jja, son mm. Missval: 32767"
```
Example: Rainfall Anomalies

**Sub-goal**: Just load the data into R

The data is in R, but we have to load the variables.

```r
> rain.data <- get.var.ncdf(dat, 'data')
> rain.lat <- get.var.ncdf(dat, 'lat')
> rain.lon <- get.var.ncdf(dat, 'lon')
> rain.time <- get.var.ncdf(dat, 'time')
```
Aside: a few more R basics

We can examine the variables in our workspace:

```r
> ls()
[1] "dat"       "rain.data" "rain.lat"
[2] "rain.lon"  "rain.time"
```

...guesses on how to remove a variable?
Aside: a few more R basics

R is **object oriented** and each object's class can be queried.

```
> class(dat)
[1] "ncdf"
> class(rain.data)
[1] "array"
```

Some basic R commands are **overloaded**:
- e.g., `plot` can plot classes "array" and "data.frame"

Most **analyses can be stored as an object** that can be summarized, plotted and contain relevant information
Exercises 3

1. Use the `diff` command to investigate the spacing of the data points in time.

2. Make a new variable called `rain.lat.inc` that contains the latitude values in increasing order. (Hint: can you reverse the array? Use the `seq` function to index the array from the end to the beginning.)

3. Use the `ls` command to verify that this variable now exists and check its object type using `class`.

4. Make an array containing all points at time 300.

5. **Bonus!** Figure out how to write a function that reverses an array. (Try `?function`)
Example: Rainfall Anomalies

Sub-goal: Just plot one time point

> image(rain.data[,,300])

Axes are strange…
Example: Rainfall Anomalies

Sub-goal: Fix our plot

```r
> image(rain.lon, rain.lat.inc, rain.data[,,300])
```
Example: Rainfall Anomalies

**Sub-goal:** Fix our plot

```r
revinds <- length(rain.lat):1
image(x=rain.lon,y=rain.lat.inc,rain.data[,revinds,300],
     xlab="Longitude",ylab="Latitude",
     main="Rainfall Anomalies")
```
Package: maps

> install.packages("maps")
> library("maps")
> map("world")
> map("world2")
Example: Rainfall Anomalies

**Sub-goal:** Add map to our plot

```r
image(x=rain.lon, y=rain.lat.inc, rain.data[,revinds,300],
     xlab="Longitude", ylab="Latitude",
     main="Rainfall Anomalies")
map("world2", add=TRUE)
```
Example: Rainfall Anomalies

Goal: Make a "movie"

1. Display all time points one after another
2. Make sure we pause after a plot is displayed!
3. Add a title that tells us what time we're looking at

1. Use a for loop
2. New function: `Sys.sleep(time in sec)`
3. Figure out how to convert the time index to the correct year and season
Example: Rainfall Anomalies

Goal: Make a "movie"

```r
seasons = c("Winter","Spring","Summer","Fall")
year.start = 1855

for (i in 1:length(rain.time)) {
  Sys.sleep(0.05)
  this.season <- seasons[(i %% 4) + 1]
  this.year <- floor(i/4)+year.start
  image(x=rain.lon,y=rain.lat.inc,
       rain.data[,revinds,i],xlab="Longitude",
       ylab="Latitude", main=paste('Rainfall Anomalies',
       '\n',this.year,'::',this.season))
  map("world2",add=TRUE)
}
```
Example: Rainfall Anomalies

**Goal:** Compare Anomalies in Boston and Tokyo

1. Find the lat/lon values of Boston and Tokyo
2. Find the indices for those values
3. Make a time-series for each location

1. Google it!
2. The `which` function
3. R provides a `ts` object
Example: Rainfall Anomalies

Sub-Goal: Return the index to the matrix for a lat/lon value

Boston: Lon 289, Lat 42

> rain.lon
[1]  0  5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 105 110 115 120 125 130 135
[57] 280 285 290 295 300 305 310 315 320 325 330 335 340 345 350 355

Closest Value

Idea: Find the index of the value closest to the truth.
Example: Rainfall Anomalies

**Sub-Goal:** Find the index of a given lat/lon value

```r
boston.lon <- 289; boston.lat <- 42
tokyo.lon <- 139; tokyo.lat <- 35
bos.lon.dist <- abs(boston.lon-rain.lon)
bos.lon.ind = which(bos.lon.dist==min(bos.lon.dist))
bos.lat.ind = which.min(abs(boston.lat-rain.lat))
tok.lon.ind = which.min(abs(tokyo.lon-rain.lon))
tok.lat.ind = which.min(abs(tokyo.lat-rain.lat))
```
Exercise 4

1. Pick a city and find its indices in the array
2. Use `summary` to investigate the range of the data in your city
3. Use `hist` to make a histogram of your city's rainfall data
4. Does this data look normal? Use `qqplot` to investigate.
5. Figure out how to run a Shapiro-Wilk test and interpret its results
Example: Rainfall Anomalies

Sub-Goal: Make a time-series object for Boston and Tokyo

```r
bos.ts <- ts(rain.data[bos.lon.ind, bos.lat.ind, ],
             start=year.start, frequency=4)
tok.ts <- ts(rain.dat[tok.lon.ind, tok.lat.ind, ],
             start=year.start, frequency=4)
```
Time Series Analysis

First Step: Plot your data!

```r
plot(bos.ts,type='l',ylab="Anomalies")
lines(tok.ts,col="red")
```
Missing Data Strategies

Ignore

- Easy to do!
- Precludes some analysis
- Might bias analysis

Replace with the mean

- Still easy
- Analysis still *runs*
- Might bias analysis

Replace using a model

- Called "imputation"
- A good strategy if you have a good model
- Repeating this gives estimates of uncertainty
- Amelia package in R
Missing Data Strategies

- Probably makes sense to ignore
- Many procedures in R will crash by default with NA values!
Is there structure in the Rainfall Anomalies?
Start with just Boston…

```r
> lag.plot(bos.ts, lags=9)
```

- Lag One: "Plot this season versus the previous season"
- Lag Four: "Plot this season versus the same season last year"
- If they're correlated, points will lie along a line
Time Series Analysis

Is there structure in the Rainfall Anomalies?
Start with just Boston…

```r
> acf(bos.ts)
```

• Look at the correlation coefficients all at once
Is there structure in the Rainfall Anomalies?  
Compare Boston to Tokyo  

```r
> ccf(bos.ts,tok.ts,na.action = na.pass)
```

- "Plot this season against last season in Tokyo"
- Look at the cross-correlation coefficients all at once
More Time Series Data

```r
> data(sunspots); plot(sunspots)
```
Exercise!

- Install the forecast package
- Use auto.arima to fit a model to the data
- Use forecast to plot the model prediction
- Can you find a model that gives a better prediction?
Thank You!

• Doing research in social science? support@help.hmdc.harvard.edu

Feedback: bit.ly/cf2012-IQSS