Introduction to Programming in R

Matthew K. Lau

Cottonwood Ecology Group
Department of Biological Sciences
Northern Arizona University

July 12, 2011
Introductions
Why learn R?
On a recent flight from Tokyo to Beijing, at around the time my lunch tray was taken away, I remembered that I needed to learn Mandarin. "...dammit," I whispered, "I knew I forgot something."
– David Sedaris (New Yorker, July 2011)
Three Reasons

Software does not *have* to limit analysis.
Three Reasons

- Software does not *have* to limit analysis.
- Scripting can save time and effort.
Three Reasons

- Software does not have to limit analysis.
- Scripting can save time and effort.
- Free and Free
Three Reasons

- Software does not have to limit analysis.
- Scripting can save time and effort.
- Free = No Cost and Free = Open Source
Three Reasons

▶ Software does not *have* to limit analysis.
▶ Scripting can save time and effort.
▶ Free = No Cost and Free = Open Source
What can R do?

- Math (e.g. linear algebra)
- Basic Statistics
- Publication quality plots
- Simulations
- Database interfacing
- GIS

- Phylogenetics
- Multivariate statistics
- Network analysis
- Bayesian statistics
- Animations
- Make julienned fries
- and much MUCH MORE...
Requested Topics

- The basics (What the heck!)
- Data input and management
- Bayesian statistics (interfacing with Win-Bugs)
- Growth curve analyses
- Summary statistics from large datasets
- GLM, GAM and Mixed Models
Lasciate ogne speranza, voi ch’intrate.
– Dante Aligheri (The Inferno)
Abandon all hope, ye who enter here.
– Dante Aligheri (The Inferno)
The Basics

Work Flow

Data Management

Analysis

Plotting

Packages

Resources

Advanced
Getting Started

- Open R.
- Say hello to the command line.
Think Intuitively

Remember, no one’s listening but R.
Math in R

What do you get when you add two and two?

Matthew K. Lau

http://dana.ucc.nau.edu/~mkl48/bio/home.html
Math in R

What do you get when you add two and two?

> 2 + 2

[1] 4
Getting Help.

- ?, ??, help
- Googling
- Manuals and Books

Matthew K. Lau

http://dana.ucc.nau.edu/~mk148/bio/home.html

Introduction to Programming in R
How do you use `?` to learn about `help`?
How do you use ? to learn about help?

> ? help
What other math commands are there?
What other math commands are there?

> ? '+'

Matthew K. Lau

http://dana.ucc.nau.edu/~mkl48/bio/home.html

Introduction to Programming in R
Objects

How do you create an object?  

> \( x = 2 + 2 \)
Objects

What is the difference between `x` and `'x'`?

```r
> x
[1] 4
> "x"
[1] "x"
```
Can you do operations with objects?
Can you do operations with objects?

\[
\begin{align*}
&> x = 2 + 2 \\
&> y = 2 + 2 \\
&> x + y \\
&\quad [1] 8 \\
&> x * y \\
&\quad [1] 16 \\
&> x/y \\
&\quad [1] 1
\end{align*}
\]
Objects

What can I name objects?

- Letters
- Symbols (e.g. "." and "_")
- Numbers (following a letter)
- Keep them short (\leq 7 characters)
- Avoid function names (e.g. data, factor, sqrt)
Objects

How do I know if I have created an object already? How do you I get rid of them?

> ls()
> rm()

*NOTE: rm(ls()) will remove all objects in your workspace.
Functions

function(\texttt{arguments},...)
What do you think the functions are for the mean and standard deviation?
Apply them to our object x.
Functions - Cognates

What do you think the functions are for the mean and standard deviation?
Apply them to our object \( x \).

\[
\begin{align*}
> \text{mean}(x) \\
[1] & \quad 4 \\
> \text{mean}(%(\text{mean}(x))) \\
[1] & \quad 4 \\
> \text{sd}(x) \\
[1] & \quad \text{NA}
\end{align*}
\]
Comprehensive R Archive Network (CRAN)

- How is CRAN organized?
- CRAN.r-project.org to download R.
- Also a resource for help files, manuals and other resources.
Work Flow Outline

1. Create a project working directory.
2. Create a data folder.
3. Place data in data folder.
4. Open and save a new script in the working directory.
5. Set the working directory.
6. Call files by 'filename' or data by data/'filename'.

Matthew K. Lau
http://dana.ucc.nau.edu/~mkl48/bio/home.html

Introduction to Programming in R
1. Long tasks are awkward in the command line.
2. Command line entries are not saved.
3. Scripted code can be annotated.
Scripting

1. Open a new script window.
2. Save it to your working directory.
3. What happens when you run "# 2+2" using CTRL R?
1. Open a new script window.
2. Save it to your working directory.
3. What happens when you run 
"# 2+2" using CTRL R?

> # 2+2
>
Scripting Tips

1. Start each script with meta-data.
2. Annotate each section and individual lines if possible.
3. Be descriptive but succinct, like a lab journal.

#Matthew K. Lau 10July2011
#Script from Introduction to R class at UNCW.

#How to do basic math operations.
2+2 #addition
2-2 #subtraction
2*2 #multiplication
Scripting Tips

1. Start each script with meta-data.
   #Matthew K. Lau 10July2011
   #Script from Introduction to R class at UNCW.

2. Annotate each section and individual lines if possible.
   #How to do basic math operations.
   2+2 #addition
   2−2 #subtraction
   2*2 #multiplication

3. Be descriptive but succinct, like a lab journal.

*You’ll thank yourself later when you easily remember what you were doing when you wrote your script.

Matthew K. Lau

http://dana.ucc.nau.edu/~mk148/bio/home.html

Introduction to Programming in R
Importing

- Entering by hand (:, c)
- Importing from a file (read.csv)
How do you create a vector of integers from 1:5?

```r
> 1:5
[1] 1 2 3 4 5
```
Entering data by hand (:, c)

How do you create a vector of integers from 1:5?

```r
> c(1, 2, 3, 4, 5)
[1] 1 2 3 4 5
```
How do you create a vector of integers from 1:5?
(Hint: Create an object)
How do you create a vector of integers from 1:5? (Hint: Create an object)

NOTE: Our object x was an integer, but now it’s a vector.

```r
> x = 1:5
> x
[1] 1 2 3 4 5
```
Data Types

- (Scalars)
- Vectors (numeric, character)
- Matrices and Arrays (matrix, array)
- Data Frames (data.frame)
- Lists (list)
- ...

Matthew K. Lau
http://dana.ucc.nau.edu/~mkl48/bio/home.html

Introduction to Programming in R
Vectors (numeric, character)

How do you know what type of data you have?

```r
> class(x)
[1] "integer"
> mode(x)
[1] "numeric"
> y = c(1, 2, 2.5, 3, 5)
> class(y)
[1] "numeric"
> mode(y)
[1] "numeric"
```
Vectors (numeric, character)

How do change data types?

```r
> x = as.character(x)
> class(x)
[1] "character"
> mode(x)
[1] "character"
```
Matrices and Arrays

How do you create a matrix?

> M = matrix(data = 1:9, nrow = 3, ncol = 3)
> M

[,1] [,2] [,3]
[1,]  1  4  7
[2,]  2  5  8
[3,]  3  6  9
Matrices and Arrays

How do you create an array?
Matrices and Arrays

How do you create an array?

```r
> A = array(1:9, dim = c(3, 3))
> A

[,1] [,2] [,3]
[1,] 1 4 7
[2,] 2 5 8
[3,] 3 6 9

> class(A)
[1] "matrix"

> mode(A)
[1] "numeric"
```
### Matrices and Arrays

What about connecting two vectors together into columns?

```r
> x = 1:5
> y = 1:5
> cbind(x, y)
```

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Matthew K. Lau

http://dana.ucc.nau.edu/~mkl48/bio/home.html
What about connecting two vectors together into rows?

```r
> x = 1:5
> y = 1:5
> rbind(x, y)

x 1  2  3  4  5
y 1  2  3  4  5
```
Data Frames

How do you create a data frame from numeric and character vectors?
Data Frames

How do you create a data frame with a numeric and a character vector?

```r
> x = 1:3
> y = c("a", "b", "c")
> data.frame(x, y)

   x y
1 1 a
2 2 b
3 3 c
```
Lists

How do you create a list with m and a?
How do you create a list with m and a?

> list(M, A)

[[1]]
[,] [,2] [,3]
[1,] 1 4 7
[2,] 2 5 8
[3,] 3 6 9

[[2]]
[,] [,2] [,3]
[1,] 1 4 7
[2,] 2 5 8
[3,] 3 6 9
Importing a Matrix

How do you import a matrix?
Data can be found at http://perceval.bio.nau.edu/downloads/igert/IntroR-Course_Notes/CommData.csv

> com = read.csv("data/CommData.csv")
Quick Summaries of Data

```r
> head(com)

1 1 321 179 179 143 36 179 179 179 250 179 0 0 71 107 36 0 0 143
2 1 250 143 36 214 214 214 250 214 0 107 0 36 107 36 0 71 36 0
3 1 71 250 107 179 143 107 179 250 143 179 0 0 36 36 107 36 36 36
4 1 36 250 71 107 71 179 143 250 36 357 0 71 0 36 36 36 36 36
5 1 179 107 143 214 0 143 107 179 143 250 36 36 36 0 0 36 36 36
6 1 357 107 143 286 179 179 179 179 250 286 0 36 143

1 0 71 0 71 36 0 36 36 0 0 107 36
2 36 0 36 36 36 71 71 36 0 143 71 36
3 36 36 36 71 36 143 36 0 36 71 0 0
4 107 71 36 36 0 36 36 36 0 0 36 36
5 36 71 0 36 36 36 0 71 36 71 36 143
```
### Quick Summaries of Data

```r
> summary(com)
```

<table>
<thead>
<tr>
<th></th>
<th>env</th>
<th>V1</th>
<th>V2</th>
<th>V3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>1.0</td>
<td>0.0</td>
<td>36.0</td>
<td>36.0</td>
</tr>
<tr>
<td>Median</td>
<td>1.5</td>
<td>71.0</td>
<td>107.0</td>
<td>89.0</td>
</tr>
<tr>
<td>Mean</td>
<td>1.5</td>
<td>107.1</td>
<td>109.0</td>
<td>92.9</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>2.0</td>
<td>187.8</td>
<td>152.0</td>
<td>152.0</td>
</tr>
<tr>
<td>Max.</td>
<td>2.0</td>
<td>357.0</td>
<td>250.0</td>
<td>250.0</td>
</tr>
</tbody>
</table>

|       | V4         | V5         | V6         | V7         | V8         |
|-------|------------|------------|------------|------------|
| Min.  | 0.0        | 0.00       | 0.0        | 0.0        | 0.0        |
| 1st Qu.| 36.0      | 36.00      | 36.00      | 36.00      |
| Median | 107.0     | 107.00     | 107.00     | 107.00     |
| Mean   | 96.45     | 98.35      | 103.8      | 110.7      |
| 3rd Qu.| 143.00    | 152.00     | 152.00     | 152.00     |
| Max.   | 286.0     | 250.0      | 286.0      | 321.0      | 321.0      |

Matthew K. Lau

http://dana.ucc.nau.edu/~mkl48/bio/home.html

Introduction to Programming in R
Manipulating

How do you pull out a single value from a vector?

```r
> x = 1:5
> x[3]
[1] 3
```
Manipulating

How do you isolate multiple values?

```r
> x = 1:5
> c(1, 2, 3)
[1] 1 2 3
> x[c(1, 2, 3)]
[1] 1 2 3
```
How do you isolate multiple values? Is there a simpler way?
Manipulating

How do you isolate multiple values? Is there a simpler way?

```r
> x = 1:5
> 1:3
[1] 1 2 3
> x[1:3]
[1] 1 2 3
```
Manipulating

How do you subset a vector?

> x = 1:5
> c(FALSE, FALSE, FALSE, TRUE, FALSE)

[1] FALSE FALSE FALSE TRUE FALSE

> x[c(FALSE, FALSE, FALSE, TRUE, FALSE)]

[1] 4
Manipulating

How do you subset a vector?

> x == 4

[1] FALSE FALSE FALSE TRUE FALSE

> x[x == 4]

[1] 4

> x != 4

[1] TRUE TRUE TRUE FALSE TRUE FALSE TRUE
Manipulating

How do you isolate all of the values of $x$ greater than or equal to 2?
Manipulating

How do you isolate all of the values of \( x \) greater than or equal to 2?

\[
> x[x >= 2]
\]

\[
[1] 2 3 4 5
\]
Manipulating

How do you isolate one number from a matrix?

\[
\begin{array}{ccc}
[1,] & 1 & 4 & 7 \\
[2,] & 2 & 5 & 8 \\
[3,] & 3 & 6 & 9
\end{array}
\]

> M[2, 3]

[1] 8
### Manipulating

How do you isolate a whole row or column from a matrix?

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[,1]</td>
<td>[,2]</td>
<td>[,3]</td>
<td></td>
</tr>
<tr>
<td>[1,]</td>
<td>1</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>[2,]</td>
<td>2</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>[3,]</td>
<td>3</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>

```r
> M[1, ]
[1] 1 4 7
```

```r
> M[, 1]
[1] 1 2 3
```
Manipulating

How can you get all of the values of \( M[1,] \) that are greater than 2?
Manipulating

How can you get all of the values of $M[1,]$ that are greater than 2?

```
[,1] [,2] [,3]
[1,] 1 4 7
[2,] 2 5 8
[3,] 3 6 9

> M[1, ]
[1] 1 4 7

> M[1, ] > 2
[1] FALSE TRUE TRUE

> M[1, M[1, ] > 2]
[1] 4 7
```
Manipulating

How do you sort a matrix by the first column?

```r
> order(M[, 1])
[1] 1 2 3
> order(M[, 1], decreasing = TRUE)
[1] 3 2 1
```
Manipulating

How do you sort a matrix by the first column?

```r
> M[order(M[, 1]), ]
[,1] [,2] [,3]
[1,] 1 4 7
[2,] 2 5 8
[3,] 3 6 9

> M[order(M[, 1], decreasing = TRUE), ]
[,1] [,2] [,3]
[1,] 3 6 9
[2,] 2 5 8
[3,] 1 4 7
```
Manipulating

How do you sort a matrix by the first row?
Manipulating

How do you sort a matrix by the first row?

```r
> M[1, ]
[1] 1 4 7
> order(M[1, ])
[1] 1 2 3
> M[, order(M[1, ])]
[,1] [,2] [,3]
[1,] 1  4  7
[2,] 2  5  8
[3,] 3  6  9
```
How would you sort our matrix (com) by the column names?
Manipulating

> colnames(com)

[1] "env" "V1" "V2" "V3" "V4" "V5" "V6" "V7" "V8"
[13] "V12" "V13" "V14" "V15" "V16" "V17" "V18" "V19" "V20"
[25] "V24" "V25" "V26" "V27" "V28" "V29" "V30"
How would you sort our matrix \( \text{com} \) by the column names?

\[
> \text{com}[, \text{order} \left( \text{colnames} \left( \text{com} \right) \right)]
\]
Manipulating

Can I refer to the columns by name?

```r
> colnames(com)[1:3]
[1] "env" "V1" "V2"

> attach(com)

> com[order(V1), ]

> detach(com)

> com$V1

> com$V2
```
Watch Out!

1. x and 'x'
2. x == x and x=x
3. T and F are TRUE and FALSE by default, unless you change them.
Exporting

1. Pick a file format (I recommend .csv).
2. Decide on whether you want to include row names and column names.
3. Write your object using `write.csv`, customizing output using the `row.names=FALSE` argument to exclude row names.

```r
> write.csv(com,'data/mymatrix',row.names=FALSE)
```
INTERMISSION
How do you calculate summary statistics from vectors?

```r
> x = 1:3
[1] 1 2 3
> y = 2:4
[1] 2 3 4

> length(x)
[1] 3
> mean(x)
[1] 2
> sqrt(x)
[1] 1.000000 1.414214 1.732051
```
Summary Statistics

How do you calculate summary statistics from vectors?

\[
\begin{align*}
&> x = 1:3 \\
&> y = 2:4 \\
&> sd(x) \\
&[1] 1 \\
&> var(x) \\
&[1] 1 \\
&> cor(x, y) \\
&[1] 1
\end{align*}
\]
Summary Statistics

How do you calculate summary statistics from matrices, such as the mean for all observations (i.e. for each row)?

```r
> M

[,1] [,2] [,3]
[1,] 1  4  7
[2,] 2  5  8
[3,] 3  6  9

> apply(M, MARGIN = 1, FUN = mean)

[1] 4 5 6
```
Summary Statistics

How about for all columns?

> M

[,1] [,2] [,3]
[1,] 1  4  7
[2,] 2  5  8
[3,] 3  6  9

> apply(M, MARGIN = 2, FUN = mean)

[1] 2 5 8
Summary Statistics

What if I want the mean for a given variable \( y \) for each level of a factor \( x \)?

\[
\begin{align*}
&> y = \text{com}\$V1 \\
&> x = \text{com}\$env \\
&> 	ext{tapply}(y, INDEX = x, fun = \text{mean})
\end{align*}
\]

\[ [1] \hspace{1em} 1 \hspace{0.5em} 1 \hspace{0.5em} 1 \hspace{0.5em} 1 \hspace{0.5em} 1 \hspace{0.5em} 1 \hspace{0.5em} 1 \hspace{0.5em} 1 \hspace{0.5em} 1 \hspace{0.5em} 1 \hspace{0.5em} 2 \hspace{0.5em} 2 \hspace{0.5em} 2 \hspace{0.5em} 2 \hspace{0.5em} 2 \hspace{0.5em} 2 \hspace{0.5em} 2 \hspace{0.5em} 2 \hspace{0.5em} 2 \hspace{0.5em} 2 \hspace{0.5em} 2 \hspace{0.5em} 2 \hspace{0.5em} 2 \hspace{0.5em} 2 \hspace{0.5em} 2 \hspace{0.5em} 2 \hspace{0.5em} 2 \]
What about getting the standard deviation for $y$ given the levels of $x$?

```r
> y = com$V1
> x = com$env
```
What about getting the standard deviation for \( y \) given the levels of \( x \)?

\[
\begin{align*}
> & \ y = \text{com}$V1 \\
> & \ x = \text{com}$env \\
> & \ \text{tapply}(y, \ INDEX = x, \ fun = \text{sd}) \\
\end{align*}
\]

\[
[1] 1 1 1 1 1 1 1 1 1 1 2 2 2
\]
Summary Statistics

How about getting the sum of all of the values in the matrices in a list?

> AM.list = list(A, M)
Summary Statistics

How about getting the sum of all of the values in the matrices in a list?

```r
> AM.list = list(A, M)

> lapply(AM.list, FUN = sum)
[[1]]
[1] 45

[[2]]
[1] 45
```
t-test

How do you conduct one- and two-sample t-tests?

```r
> x = com$V1
> y = com$V2
> t.test(x)
```
t-test (one-sample)

> t.test(x)

One Sample t-test

data:  x

  t = 4.1358, df = 19, p-value = 0.0005619
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval: 
  52.89918 161.30082
sample estimates:
  mean of x

  107.1
t-test (one-sample)

> t.test(x, alternative = "two.sided")

One Sample t-test

data:  x
t = 4.1358, df = 19, p-value = 0.0005619
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:  
  52.89918 161.30082
sample estimates:  
mean of x  
  107.1
t-test (one-sample)

> t.test(x, alternative = "greater")

One Sample t-test

data:  x
t = 4.1358, df = 19, p-value = 0.0002810
alternative hypothesis: true mean is greater than 0
95 percent confidence interval: 62.32249 Inf
sample estimates:
mean of x
107.1
t-test (one-sample)

> t.test(x, alternative = "less")

One Sample t-test

data:  x
t = 4.1358, df = 19, p-value = 0.9997
alternative hypothesis: true mean is less than 0
95 percent confidence interval:
   -Inf 151.8775
sample estimates:
    mean of x
107.1
t-test (two-sample)

```r
> t.test(x, y, alternative = "less")

Welch Two Sample t-test

data:  x and y
t = -0.0588, df = 33.738, p-value = 0.4767
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
     -Inf 51.35174
sample estimates:
mean of x mean of y
 107.10    108.95
```
Regression

How do you conduct a regression?

\[
\begin{align*}
> & \quad x = \text{com}\$V1 \\
> & \quad y = \text{com}\$V2 \\
> & \quad \text{xy.fit = lm}(y \sim x) \\
> & \quad \text{summary(xy.fit)}
\end{align*}
\]
Regression

```r
> xy.fit = lm(y ~ x)
> summary(xy.fit)

Call:
  lm(formula = y ~ x)

Residuals:
     Min      1Q  Median      3Q     Max
-100.81  -51.17  -13.80   25.17  157.08

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 84.8026    23.9013   3.548  0.0023 **
x           0.2255     0.1536   1.468  0.1594
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 77.54 on 18 degrees of freedom
Multiple R-squared: 0.1069, Adjusted R-squared: 0.05728
F-statistic: 2.155 on 1 and 18 DF, p-value: 0.1594
```

Matthew K. Lau
http://dana.ucc.nau.edu/~mkl48/bio/home.html

Introduction to Programming in R
How do you conduct an ANOVA?

```r
> x = factor(com$env)
> y = com$V2
> anova.fit = aov(y ~ x)
> summary(anova.fit)
```
> anova.fit = aov(y ~ x)
> summary(anova.fit)

        Df Sum Sq Mean Sq  F value    Pr(>F)
---
x         1  69502   69502 24.20887 0.0001104 ***
Residuals 18 51679   2871  
---

Signif. codes:  0 ^ a˘A¨Y***^ a˘A´Z 0.001 ^ a˘A¨Y**^ a˘A´Z 0.01 ^ a˘A¨Y.^ a˘A´Z 0.05 ^ a˘A¨Y ^ a˘A´Z 1
ANOVA

> unlist(anova.fit)

$\text{coefficients.(Intercept)}$
[1] 167.9

$\text{coefficients.x2}$
[1] -117.9

$residuals.1$
[1] 11.1

$residuals.2$
[1] -24.9

$residuals.3$
> names(anova.fit)

[1] "coefficients"  "residuals"   "effects"     "rank"
[5] "fitted.values" "assign"      "qr"          "df.residual"
[9] "contrasts"    "xlevels"     "call"        "terms"
[13] "model"
> anova.fit$residuals

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>11.1</td>
<td>-24.9</td>
<td>82.1</td>
<td>82.1</td>
<td>-60.9</td>
<td>-60.9</td>
<td>-24.9</td>
<td>82.1</td>
<td>-96.9</td>
<td>11.1</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-50.0</td>
<td>57.0</td>
<td>-50.0</td>
<td>-14.0</td>
<td>-14.0</td>
<td>57.0</td>
<td>21.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
How would you plot a histogram of the residuals?

```r
> anova.res = anova.fit$residuals
> hist(anova.res)
```
How would you make an x-y scatterplot?

> x = com$V1
> y = com$V2
> plot(x, y)
> `plot(x, y)`
Barplots

How would you make a barplot of the means of V1 at each level of env in the com dataset?
How would you make a barplot of the means of V1 at each level of env in the com dataset?

```r
> x = com$env
> y = com$V1
> mu = tapply(y, INDEX = x, fun = mean)
> barplot(mu)
```
Barplots
Customizing Plots

How do you change the axis names?

```r
> x = com$V1
> y = com$V2
> plot(x, y)
> plot(x, y, xlab = "Variable 1", ylab = "Variable 2")
```
Customizing Plots

> plot(x, y, xlab = "Variable 1", ylab = "Variable 2")
Customizing Plots

How do you add a regression line?

```r
> x = com$V1
> y = com$V2
> plot(x, y, xlab = "Variable 1", ylab = "Variable 2")
```
Customizing Plots

> plot(x, y, xlab = "Variable 1", ylab = "Variable 2")
> abline(lm(y ~ x))
Multi-plot Windows

1. Setup the plot window using the `par` function.
2. Change `par(mfrow=c(1,2))`.
3. Build each plot in succession.
Locator - In case you get lost in a plot...

1. Build your plot.
2. Run `locator(1)` in the command line.
3. Click on the plot, R will output the coordinates.
Packages: Installation and Use

1. Find the name of the package you want.
2. Make sure you’re connected to the internet.
3. Use the command, `install.packages('package name')`
4. Use the command, `library('package name')`  
5. If there are conflicting functions in two packages, detach the one you’re not using with the command, `detach(package:'package name')`
Ummm, everyone and their mother uses Excel, can R?

> install.packages('gdata')
> library("gdata")
> read.xls("data/CommData.xls")
**R Commander**

1. R’s version of *JMP* (designed by John Fox)
2. Fully integrated script editor.
3. `install.packages('Rcmdr')`
4. `library('Rcmdr')`
5. Documentation can be found here:  
   http://socserv.mcmaster.ca/jfox/Misc/Rcmdr/
R Commander

Matthew K. Lau
http://dana.ucc.nau.edu/~mkl48/bio/home.html

Introduction to Programming in R
Resources

- **Cheat Sheet**: http://cran.r-project.org/doc/contrib/Short-refcard.pdf
- **Scripting**: http://google-styleguide.googlecode.com/svn/trunk/google-r-style.html
- **SimpleR**: http://www.calvin.edu/~stob/courses/m241/S11/Verzani-SimpleR.pdf
- **Quick-R**: http://www.statmethods.net/index.html
- **Plotting**: http://www.stat.auckland.ac.nz/~paul/RGraphics/rgraphics.html
- **Regression and ANOVA**: http://cran.r-project.org/doc/contrib/Faraway-PRA.pdf
- **Ecological Analyses**: http://ecology.msu.montana.edu/labdsv/R/
  http://www.mpcer.nau.edu/igert/eco_analysis_r.html
- **Network Analyses**: http://erzuli.ss.uci.edu/R.stuff/
Writing Functions

1. Create functions as you would objects using `function`.
2. The fundamental design is:
   ```r
   my.func=function(x,...)’insert things you want done’
   ```
3. Arguments and objects created within the function are not passed out of the function.
Looping

1. Used to repeat a task (can be very powerful, but computationally inefficient).
2. Takes arguments that determine the number of repetitions.
3. Uses the for or while commands.