icebreakeR: An Introduction to R

Andrew Robinson

Department of Mathematics & Statistics
University of Melbourne
Introduction
What is R?

- R is a programming language that has been optimized for data analysis and modeling.
- R can be used as an object-oriented programming language, or as a statistical environment within which sets of instructions can be performed automatically.
Why use R?

1. R runs on Windows, Mac-OS, and Unix variants;
2. R provides a vast number of useful statistical tools;
   1. many of which have been painstakingly tested;
3. R produces publication-quality graphics in a variety of formats;
4. R plays well with \LaTeX{} via the Sweave package;
5. R plays well with FORTRAN, C, and shell scripts;
6. R scales, making it useful for small and large projects;
7. R is object-oriented;
8. R eschews the GUI.
Why avoid R?

Frustration!

1. R cannot do everything;
2. R will not hold your hand;
3. The documentation can be opaque;
4. R can drive you crazy, or age you prematurely;
5. The contributed packages have been exposed to varying degrees of testing and analysis;
6. R stores objects in RAM;
7. R eschews the GUI.
A contrast

1. R is object-oriented;
2. SAS is PROCedure-oriented;
**File Structure: Project**

- data, documents, graphics, images, notes, scripts.

**Exercise 1**
GETTING GOING AND STOPPING

**GETTING GOING**
- Starting R
- Adding Packages

Exercise 2

**STOPPING**
- Cancelling operations
- Quitting

Exercise 3
Where it all begins.

**Relevant commands**

- `getwd()`
- `setwd("new working directory")`

**Exercise 4**
Work through the exercise in the Showcase chapter. Reflect on the commonalities between the exercise and what you need from R.
R comes with internal help. Use the examples.

**Relevant Commands**

- `help(object)`
- `?object`
- `help.search("phrase")`
- `help.start()`

**Exercise 5**
The Internet is a vast repository of advice. Most of it is good.

**RELEVANT COMMANDS**

- `RSiteSearch()`
- Google
  - R-help...
Responsive Help

A list-server exists. Information is available at:
https://stat.ethz.ch/mailman/listinfo/r-help

Relevant issues
- Post questions as a last resort.
- http://www.r-project.org/posting-guide.html

Exercise 6
Why write scripts?

**Relevant issues**

- `source("script-name.R", echo=TRUE)`
- Comments: `#`

**Exercise 7**
Work Spaces

The workspace is the container of all your objects.

Relevant commands

- `ls()`
- `rm(object)`
- `rm(list=ls())`
- `save.image()`
- `save(object)`
- `load()`

Exercise 8
The past.

**RELEVANT COMMANDS**

- `savehistory()`
- `loadhistory()`

**Exercise 9**
Extensibility

R starts speedily.

**Relevant commands**

- `require(package)`
- `installed.packages()`
- `available.packages()`
- `install.packages(package)`
**Import**

○ `read.xxx()`

**Exercise 10**

**Export**

○ `write.xxx()`

○ `pdf("pdf-name.pdf") ... dev.off()`

**Exercise 11**
R is hard work

Work hard!
Work hard!

**That means:**
- Read widely.
- Use the resources available.
- Experiment patiently and flexibly.
- Keep scripts.
- Comment generously.
What is an Object?

Characteristics

- Everything.
- Objects are realizations of a class.
- All objects have attributes.
Why use Objects?

Using objects simplifies many complicated problems.

1. Communication.
2. Comparison.

But you have to put them somewhere!
Every object that is followed by () is a **function**, being called. Every object that is followed by [] is being **sub-** or **super-setted**.

**Exercise 12**
Object Types

Atomistic

- Numeric.
- String.
- Factor.
- Integer.
- Logical.
- Missing.

Exercise 13
**Object Types**

**Containers**
- Vector
  - Vectorization
- Dataframe.
- Matrix.
- Array.
- List.

**Exercise 14**

**Exercise 15**
Messing with Data

merge()
reshape()
order()
Programming

Write your own functions.

- Flow control
- Scoping
- Class control

Exercise 16
Simple data descriptions are readily available.

- **Univariate**
  - Numerical
  - Categorical

- **Multivariate**
  - Numerical/Numerical
  - Numerical/Categorical
  - Categorical/Categorical

**Exercise 17**
# ALL THE LITTLE PIECES ...

- `plot(x, y, ...)`
- `xlim=`, `ylim=``
- `xlab=`, `ylab=``
- `main=``
- `col=``
- `pch=`, `lty=`, `...`
**Graphics: Organization**

- `par(...)`
- `las=1`
- `mfrow=c(2,2)`
- `mar=c(4,4,3,2)`
- `new=TRUE`
Graphics: Augmentation

- `plot(x, y, ...)`
- `points()`
- `axis()`
- `mtext()`
- `box()`
- `legend()`
Right-click the graphic, copy as windows metafile, and paste to a document. Or …

- `pdf("pdf-name.pdf")`
- `plot(x, y, ...)`
- `dev.off()`
Graphics: Challenges

- Error Bars
- Colour by groups
Graphics: Contributions

- trellis (lattice package)
- grammar of graphics (ggplot, ggplot2 packages)
Exercise 18

Exercise 19
Linear Models

```r
name <- lm(y ~ x)
```

**Useful Options**
- `data = dataframe`
- `na.action = na.exclude`

**Occasional Options**
- `subset = logical or index`
- `weights = weights`
- `formula = terms(y ~ x, keep.order = TRUE)`
plot(model)
Diagnostics Interpretation

- Residuals vs Fitted
- Normal Q–Q
- Scale–Location
- Residuals vs Leverage

Andrew Robinson (University of Melbourne)  icebreakerR: An Introduction to R
> summary(hd.lm)

Call:
\texttt{lm(formula = height.m \sim dbh.cm, data = ufc)}

Residuals:
\[
\begin{array}{cccccc}
\text{Min} & 1Q & \text{Median} & 3Q & \text{Max} \\
-33.5257 & -2.8619 & 0.1320 & 2.8512 & 13.3206 \\
\end{array}
\]

Coefficients:
\[
\begin{array}{cccccc}
\text{Estimate} & \text{Std. Error} & t \text{ value} & \text{Pr}(>|t|) \\
(\text{Intercept}) & 12.67570 & 0.56406 & 22.47 & <2e-16 & *** \\
dbh.cm & 0.31259 & 0.01388 & 22.52 & <2e-16 & *** \\
\end{array}
\]

---

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.941 on 389 degrees of freedom
Multiple R-Squared: 0.566, Adjusted R-squared: 0.5649
F-statistic: 507.4 on 1 and 389 DF, p-value: < 2.2e-16
**Prediction**

\[
predict(model, \text{newdata}=\text{new-dataframe}, \ldots)
\]
Inference

```r
> anova(hd.lm)
Analysis of Variance Table

Response: height.m

<table>
<thead>
<tr>
<th></th>
<th>Df</th>
<th>Sum Sq</th>
<th>Mean Sq</th>
<th>F value</th>
<th>Pr(&gt;F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbh.cm</td>
<td>1</td>
<td>12388.1</td>
<td>12388.1</td>
<td>507.38</td>
<td>&lt; 2.2e-16 ***</td>
</tr>
<tr>
<td>Residuals</td>
<td>389</td>
<td>9497.7</td>
<td>24.4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```
Exercise 20
**Characteristics**

Mixed-effects models incorporate *two kinds* of predictor variables.

- Fixed effects - speak for themselves.
- Random effects - represent a population.
Necessity

Natural resources data commonly have hierarchical structure.

- Trees within plots within stands within forests.
- Times within trees . . .

Mixed-effects models enable the modeling of correlated data *without* violation of important regression assumptions.
Necessity

Natural resources data commonly have hierarchical structure.

- Trees within plots within stands within forests.
- Times within trees . . .

Mixed-effects models enable the modeling of correlated data *without* violation of important regression assumptions.

**Regression assumptions.**

- True relationship is linear.
- Residuals are normally distributed.
- Residuals have identical distribution (variance).
- Residuals are independent.
Utility

Mixed effects models allow the estimation of useful quantities.

- Variance components.
- Intra-class correlation.
Yet another perspective

Construct a height-diameter relationship using two randomly selected plots in a forest, and that we have measured three trees on each.

Growing conditions are quite different on the plots, leading to a systematic difference between the height-diameter relationship on each.

If we fit a simple regression to the trees then we obtain a residual/fitted value plot.

If we fit a simple regression to the trees with an intercept for each plot then we obtain a residual/fitted value plot.
Note that the model specification implies that:

\[ y_{ij} - \hat{y}_{ij} = \epsilon_{ij} \]  

(1)

and

- The true relationship is linear.
- \( \epsilon_i \sim \mathcal{N}(0, \sigma^2) \)
- The \( \epsilon_i \) are independent.

Clearly not true.
Decomposition 2

What if we could make:

\[ y_{ij} - \hat{y}_{ij} = \hat{b}_i + \hat{\epsilon}_{ij} \]  

(2)

Then we merely need to assume that:

- The true relationship is linear.
- \( b_i \sim \mathcal{N}(0, \sigma^2_b) \)
- \( \epsilon_{ij} \sim \mathcal{N}(0, \sigma^2) \)
- The \( \epsilon_{ij} \) are independent.

Much more tenable!
The assumptions are satisfied because the systematic differences between the plots, which previously produced correlation, are now accounted for by the new random effects.

However, when the time comes to use the model for prediction, we do not need to know the plot identity, as the fixed effects do not require it.
A brief synopsis: a sample of 66 trees was selected in national forests around northern and central Idaho. According to Stage (pers. comm. 2003), the trees were selected purposively.

The habitat type and diameter at 4’6” were also recorded for each tree, as was the national forest from which it came. Each tree was then split, and decadal measures were made of height and diameter inside bark at 4’6”.
**Figure:** Al Stage's Grand Fir stem analysis data: height (m) against diameter (cm). These were dominant and co-dominant trees.
**Figure:** Al Stage’s Grand Fir Stem Analysis Data: height (ft, vertical axes) against diameter (inches, horizontal axes) by National Forest. These were dominant and co-dominant trees.
Model for getting it wrong in R

\[ h_i = \beta_0 + \beta_1 \times d_i + \epsilon_i \]  (3)
Model for getting it wrong in R

\[ h_i = \beta_0 + \beta_1 \times d_i + \epsilon_i \]  

(3)

Regression assumptions.

- True relationship is linear.
- \( \epsilon_i \sim \mathcal{N}(0, \sigma^2) \)
- \( \text{Cov}(\epsilon_i, \epsilon_j) = 0 \) for \( i \neq j \)
**Figure:** Popular regression diagnostics from R.
$h_{it} = \beta_0 + (\beta_1 + b_{1i}) \times d_{it} + \epsilon_{it}$  \hspace{1cm} (4)
Model for getting it less wrong in $\mathbf{R}$

$$h_{it} = \beta_0 + (\beta_1 + b_{1i}) \times d_{it} + \epsilon_{it}$$

(4)

**Regression assumptions.**

- True relationship is linear.
- $b_{1i} \sim \mathcal{N}(0, \sigma_{b_1}^2)$
- $\epsilon_{it} \sim \mathcal{N}(0, \sigma^2)$
- $\text{Cov}(\epsilon_{it}, \epsilon_{jt}) = 0$ for $i \neq j$
- $\text{Cov}(\epsilon_{it}, \epsilon_{ig}) = 0$ for $t \neq g$
Now, the key assumptions that we’re making are that:

1. the model structure is correctly specified
2. the tree and forest random effects are normally distributed,
3. the tree random effects are homoscedastic within the forest random effects.
4. the inner-most residuals are normally distributed,
5. the inner-most residuals are homoscedastic within and across the tree random effects.
6. the innermost residuals are independent within the groups.
**Diagnostics for getting it less wrong in R**

**Figure:** Useful regression diagnostics from R.
**Figure:** More useful regression diagnostics from R.
**Figure**: More useful regression diagnostics from R.
**Diagnostics for getting it less wrong in R**

**Figure:** More useful regression diagnostics from R.
The roles differ

For the design,
- fixed effects represent *themselves*;
- random effects represent a *population*. 
The roles differ

Within the model,
- fixed effects *explain* variation;
- random effects *organize* unexplained variation.
Random effects are effects that common sense says will explain variation, but you don’t want to have to know them in order to be able to apply the model.
Modelling is much more involved

Add a new dimension to your flow chart!
A Modelling Strategy

The modeling strategy depends on the modelers intention.

1. Fit baseline model.
   1. Include the meaningful fixed effects.
   2. Include the design random effects.

2. Check the assumption diagnostics.

3. Add or modify random components until diagnostics are satisfied.
   1. a heteroskedastic variance structure (several candidates)
   2. a correlation structure (several candidates)
   3. extra random effects (e.g. random slopes)

4. Consider adding more fixed effects.

5. Re-examine the diagnostics, add/modify random effects, etc.
**Basic Model Statement**

\[ Y = X\beta + Zb + \epsilon \]

\[ b \sim \mathcal{N}(0, D) \]

\[ \epsilon \sim \mathcal{N}(0, R) \]
**Basic Model Statement**

\[ Y = X\beta + Zb + \epsilon \]

\[ b \sim \mathcal{N}(0, D) \]
\[ \epsilon \sim \mathcal{N}(0, R) \]

**Design Matrices**

- \( X \) allocates the fixed effects.
- \( Z \) allocates the random effects.
Basic Model Statement

\[ Y = X\beta + Zb + \epsilon \]

\[ b \sim \mathcal{N}(0, D) \]
\[ \epsilon \sim \mathcal{N}(0, R) \]

Design Matrices
- \( X \) allocates the fixed effects.
- \( Z \) allocates the random effects.

Covariance Matrices
- \( D \) describes the random effects covariance.
- \( R \) allocates the residuals covariance.
\[ Y = X\beta + Zb + \epsilon \]

\[
\text{Var} (Y \mid X, Z, \beta, b) = R
\]
\[ Y = X\beta + Zb + \epsilon \]

\[
\text{Var}(Y | X, Z, \beta, b) = R
\]

\[
\text{Var}(Y | X, \beta) = ZDZ' + R = V
\]
Log Likelihood

\[
\mathcal{L} (\beta, V | Y, X) = -\frac{1}{2} \ln (|V|) - \frac{n}{2} \ln (2\pi) - \frac{1}{2} (Y - X\beta)' V^{-1} (Y - X\beta)
\]
Profile $\beta$ out

\[ \hat{\beta} = (X'V^{-1}X)^{-1}X'V^{-1}Y \]
\( \beta \text{ IS GONE!} \)

\[
L(\beta, V \mid Y, X) = f(V, Y, X, Z, D, R)
\]
Estimate $\hat{V}$ by maximization and then $\hat{\beta}$ by substitution.
Maximum likelihood estimators of covariance parameters are usually negatively biased.
Briefly, ReML involves applying ML, but replacing

- $Y$ with $KY$;
- $X$ with $0$;
- $Z$ with $K'Z$; and
- $V$ with $K'VK$

where $K$ is such that $K'X = 0$. 