

# UseRs! An introduction to R

## Class 4 – Modeling notes

R is like a box of legos – some packages come standard, others you have to add. Your own scripts will start to feel like lego blocks as well, once you figure out how to do something, you'll be able to quickly repeat it by returning to a saved script and editing it as necessary.

Often something that you want to do will require using a non-standard package. How do you find it?

A place to start: the CRAN view pages: <http://cran.r-project.org/> then click on “Views” in the left column. These give an overview of many (probably not all) packages and puts them in context with other packages to help you decide what may be useful.

An example: The Spatial View  
Spatial classes: package sp.  
The reference manual?  
Read the vignette (when it exists)

Google! Other sources of help – the list-servs! Don't underestimate the information contained in them. The challenge is finding useful search terms.

The contents of a package will not be available in your search function until you install that package, so you'll have to go to the internet to learn about them.

Susan's example:

Compare 2 linear regressions and make a plot with two panels

```
setwd("/media/Data/Rcourse")
### load data file. Don't forget the file extension and the h=T.
data=read.table("polyvsres.txt",h=T)

### linear regression
lm1=lm(resNRI~polNRI,data=data)
lm2=lm(resNTI~polNTI,data=data)

par(mfrow=c(2,1),mai=c(0.1,1.5,.75,.5),mgp=c(4,1,0))
plot(data$resNRI~data$polNRI,
las=1,cex.axis=1.5,cex.lab=1.5,ylab="NRI",
xlab="",main="Phylogenetic Structure and
Resolution",cex.main=1.5,pch=16,xlim=c(-3,1.5),xaxt="n")

#draw the regression line
abline(lm1)

#get predictions out
predict1=predict(lm1,data.frame(polNRI=sort(data
$polNRI)),interval="confidence")
#draw the upper and lower predictions
```

```

lines(sort(data$polNRI),predict1[,2],col="grey",lty="dashed")
lines(sort(data$polNRI),predict1[,3],col="grey",lty="dashed")

#add some annotation
text(1,-2.5,bquote(paste(r^2," = ",.(round(summary(lm1)$r.squared,
2))))),cex=1.4)
#more annotation
text(-1,0,"Check out that fit!",cex=1.4)

## add second plot
par(mai=c(1,1.5,0,.5))
plot(data$resNTI~data
$polNTI,las=1,cex.axis=1.5,cex.lab=1.5,ylab="NTI",xlab="Resolution",
pch=16,xlim=c(-3,1.5))

abline(lm2)

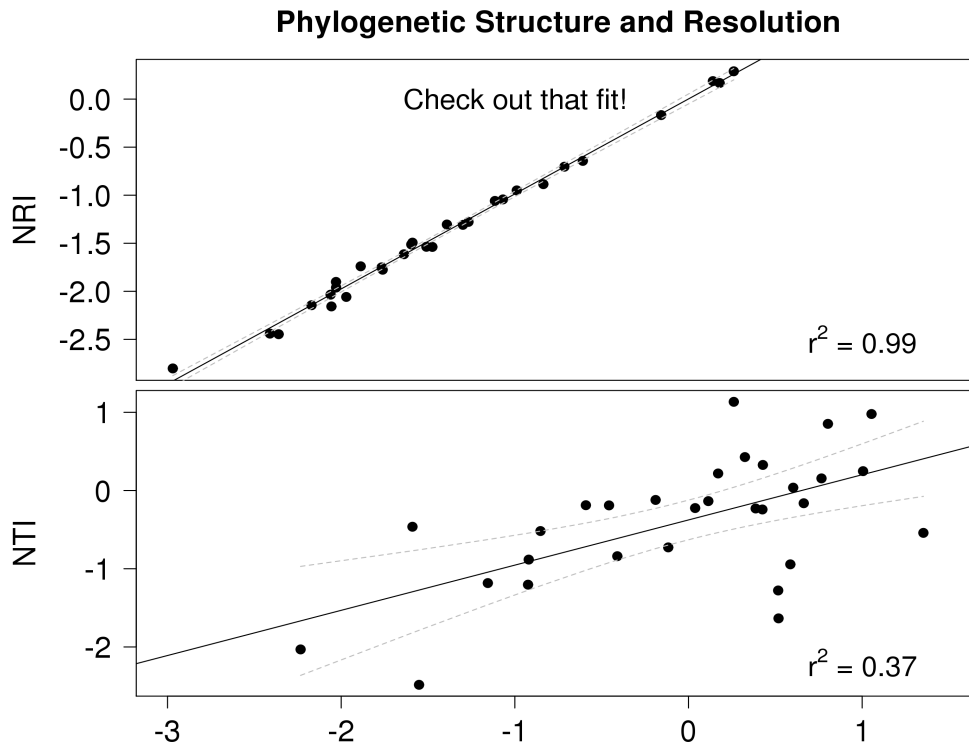
predict2=predict(lm2,data.frame(polNTI=sort(data
$polNTI)),interval="confidence")

lines(sort(data$polNTI),predict2[,2],col="grey",lty="dashed")
lines(sort(data$polNTI),predict2[,3],col="grey",lty="dashed")

text(1,-2.2,bquote(paste(r^2," = ",.(round(summary(lm2)$r.squared,
2))))),cex=1.4)

```

This should result in a plot that looks something like this:



## Suegene's example: Anova analysis of lacewing call-response data

```
setwd("/media/Data/RCourse/")
newdata=read.table("Suegene.txt", h=T)

#graphical
library(lattice)
bwplot(type~cross|time, data=newdata)

#2-way aov (ANOVA)
m1=aov(type~cross*time, data=newdata)
summary(m1)

#linear model
m2=lm(type~cross*time, data=newdata)
summary(m2)
drop1(m2,~,test="F") # type III SS and F Tests
anova(m2)

#mixed effects linear model (repeated measures)
library(nlme)
m3=lme(type~time*cross, random=~1|ID, data=newdata)
summary(m3)
anova(m3)
```

Generate some random data and plot it.

```
x=rnorm(100,mean=10,sd=2)
y=rnorm(100,mean=10,sd=2)
plot(x,y)
lm1=lm(y~x)
summary(lm1)

y2=rnorm(100,mean=x,sd=2)
lm2=lm(y2~x)
plot(x,y2)
summary(lm2)
```

Two more classes:

Functions and loops

Spatial Analysis – Trina's example

What are we missing?