BioEE 758
R Programming
Week 8 handout/homework

Week 8 objectives:

1) Increase comfort level with functions

2) Learn about handling data with functions

3) Think *hard* about final project!

Next week: all things strings

Using lists to store function output

As we've discussed previously, functions can only have a single return value. However, the return value can be a vector, a list, or a dataframe (which is actually a certain type of list).

If you want to make a list, you start by declaring a variable as a list:

```
result <- list();
```

This allows you to include all sorts of useless information in a single variable, which can be returned from a function:

```
randomJunk <- function()
{
     res <- list();
     res$popstar <- "Britney";
     res$food <- "pork rind";
     res$lucky_stars <- c('N329', 'orion', 'beetlejuice', 'betelgeuse');
     res$lucky_numbers <- NA;
     res$life_expectancy <- runif(1, min=10, max = 80);
     res$super_lotto <- c(15, 36, 24, 11, 86);
     return(res);
}
```

How do you access the components of the list returned by `randomJunk`?

Same as any list or dataframe:

```
z <- randomJunk();
cat(z$popstar, ' rocks my world!\n');
cat(z$lucky_numbers, 'N, Man! Almost a palindrome!\n', sep='');
```

You can also make dataframes (recalling that a dataframe is a special type of list):

```
oneFunctionToBindThem <- function()
{
     res <- list();
     res$alpha <- rnorm(5);
     res$beta <- sample(letters, 5);
     res$gamma <- sample(LETTERS, 5);
     res$moo <- rgamma(5, .25, 1);
     res<-as.data.frame(res);
     return(res);
}
```

OR

```
oneFunctionToBindThem <- function()
{
     alpha <- rnorm(5);
     beta <- sample(letters, 5);
     gamma <- sample(LETTERS, 5);
     moo <- rgamma(5, .25, 1);
     res<-data.frame(a=alpha, b=beta, c=gamma, d=moo);
     return(res);
}
```

Note also the use of the built-in R vectors ‘letters’ and ‘LETTERS’ which might come in handy...

1) Write a function summaryStats() that takes a vector of numbers as an argument and returns a list with the following components: the mean, the variance, the median, the 2.5 % and 97.5% quantiles, and the five largest numbers in the input vector. Quantiles and variance can be obtained using the functions quantile() and var(). Your function should use stop() to return an error message if input is not numeric, or if the input vector does not contain at least 20 numbers.

3) Write a function getCV(...) that takes a vector of numbers as input and returns the coefficient of variation (CV). The CV is just the ratio of the standard deviation to the mean and is a dimensionless index of variability in a set of numbers.

4) Write a function `summarizeLizards(...)` which takes the skinks dataframe as an argument and returns a dataframe with a number of rows equal to the number of species. The columns of the dataframe should contain the following summary statistics for each species:

- *the species name (of course)*

- *the number of males (omit questionables/juveniles)*

- *the number of females (omit questionables/juveniles)*

- *the number of sites where the lizard was captured*

- *the mean snout-vent-length*

- *the largest snout-vent length*

- *the coefficient of variation in snout-vent-length (use getCV from above)*

- *the distance between the MAXIMUM snout-vent recorded and the mean value, expressed in units of standard deviations of snout-vent. Thus, you would calculate this as `(max_of_x – mean_of_x)/std.dev(x).` This might be useful in looking for outliers or extreme values (if for example you suspected that someone might have mis-measured one of the lizards...).*

Do this by declaring a list, then looping over each species name and collecting the relevant information. When you are finished, you can convert the list to a dataframe like this:

```
result <- as.data.frame(result);
```

variable (column) names should be `species, n_male, n_fem, n_sites, mean_svl, max_svl, cv_svl,` and `scaled_max_svl`

Also: remove all rows with NA values from the dataframe before you send it to the function (`x <- na.omit(x)`)

3) In class, we pointed out a problem with *Bounce of Brownie*. It seems that sometimes she manages to escape (briefly) the confines of her room when steps are normally distributed (sd = 3). Explain why this happens. Explain to me – in words – how you made this diagnosis.

5) A real biological problem: certain stochastic models have been widely used in both ecology and evolutionary biology. Consider these two classic problems:

i) The distribution of abundances of species within communities. Ecologists have long been interested in the processes that generate and maintain patterns of differential species richness among species within communities. Robert MacArthur (1957) proposed that resources might be divided randomly among species within a community; thus, some species would have a larger chunk of the 'resource pie' than others. Under this model, the abundance of species was predicted to be proportional to the resource base available to them. Species with a large resource base should be more abundant than species with smaller resource bases. Admittedly, this 'model' doesn't mechanistically explain why resources might be partitioned randomly, but it does make a prediction about the abundances of species within communities.

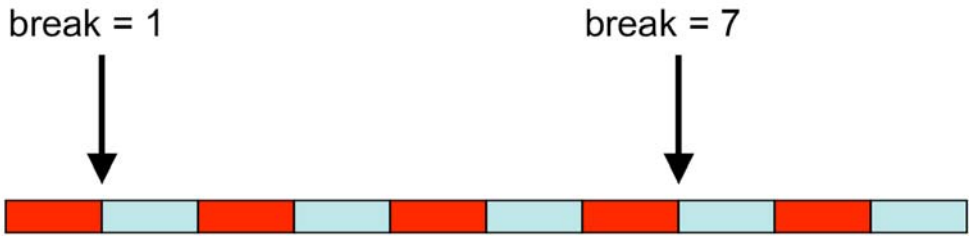
ii) Distribution of species diversity among clades: why do some groups of organisms seem to be much more diverse than other groups? Within mammals, for example, we find clades with only a few species (e.g., egg-laying monotremes, with the platypus and several echidnas), as well as clades with hundreds to thousands of species (e.g., rodents). Do these differences in diversity reflect real differences in rates of evolutionary diversification? Not necessarily: diversification is a stochastic process, so even if the rate of species diversification through time is constant across several groups, we would still expect them to end up with different species diversities after some amount of time (just like in the geometric population growth problem; the growth rate was constant, but results differed among simulations, right?).

Suppose we have some total set of N species divided among K clades (e.g., N = 423 species of orcs in K = 26 families of orcs). It is possible to show theoretically that, if diversification rates have been constant, the total number of species (e.g., all 423 species of orcs) should be divided randomly among the 26 families. If diversification rates have differed among lineages,

we predict that there would be an excess of really large and really small clades and fewer intermediate sized clades.

Both of these scenarios can be modeled with a 'broken stick' distribution. The broken stick distribution is quite intuitive: let's consider those 423 species of orcs, and pretend that they fall into 53 subfamilies. To randomly divide the diversity, we would pretend that we had a stick of length 423. We would let the stick break randomly into 53 pieces, with the length of each piece corresponding to the number of species in each clade.

Suppose we have stick of length 10. To randomly break this into into 3 clades, we need to choose 2 breakpoints. If you were to choose 1 and 7 as your breakpoints, this would divide your stick into the following 3 clades:



This would give you clades of 1, 6, and 3 species.

Stop and think question # 1: **how many breaks do you need to generate K clades?** You'll also have to think about the set of possible breakpoints. Think about the above: if you have 10 species, what breakpoints can you draw that will give you at least 2 clades with non-zero species diversity? 0:10? 1:9? 0:9? 1:10? None of the above? All of the above?

Here is your problem: you will test whether the distribution of species diversity among avian families can be modeled with a broken stick distribution.

Remember: if rates of species diversification have varied significantly among lineages, you will observe too many small and large clades and too few intermediate sized clades. A good summary statistic for the dispersion of clade sizes (or species abundances, if you are thinking ecologically) is the CV, or coefficient of variation (with which you are familiar by now...). You can calculate the CV for avian family sizes and compare it to the distribution

of the CV if the number of species in avian families followed a broken-stick model. A high coefficient of variation would suggest too many clades at the tails of the distribution – perhaps indicating lots of very small and a few very large clades.

This is a real simulation problem. Papers have been published using this model for both the abundance and diversification rate questions.

What I want from you:

a) a brief statement of your hypothesis and how you will test it.

b) pseudocode required to perform the test. You should work out the logic of the stick-breaking before doing any programming.

c) the actual code that performs the test. It is trivial to implement code that does not do this analysis correctly, and you will easily be deceived into thinking that it all is well. **Your code MUST contain an error checking device**: if your broken stick *does not have exactly K clades*, or if your broken stick *does not have exactly N species total*, it must give an error message and **stop()**. You must do this, because I already know in advance that the most common programming error for this problem will result in either too many or too few species or clades. If you start each simulation with a stick of length N, the sum of all species in all clades after the stick is 'broken' must equal N, or something is not working properly

d) In a word document, a nice histogram (with axis labels) of the null distribution. You should also have an arrow or a line or some other indicator of the CV you calculated for the avian family data, such that the reader can visually inspect the position of your observed statistic relative to the null distribution. You can easily change sizes and labels of axes, and even fonts.

e) A legend to go along with the figure interpreting it, as though it were in a publication. You should give the p-value here.

The avian family data can be found in the file 'avian_families.txt', which is posted on the website. You will need to obtain the number of species and families yourself from this data table.

Tip: keep your code flexible. Each simulation should be performed by a

function that takes arguments and returns the relevant statistic (I can only think of two arguments that would be necessary). You may want to use this again; keep it sufficiently general that it would be easy to use in other contexts, such as testing the distribution of species abundances in some ecological community.

Hint: the functions *diff* and *sort* might come in handy.

As for making a decent figure, you should look at Paul Murrell's R Graphics page (with examples and code)

http://www.stat.auckland.ac.nz/~paul/RGraphics/rgraphics.html

As well as the R graph gallery, with hundreds of graphs and code that you can 'steal' from.

http://addictedtor.free.fr/graphiques/