

What is R?



- R is a language and environment for gathering and manipulating data, applying statistical and numerical computations, summarizing and visualizing the results.

Website: <http://www.r-project.org/>

- R provides a wide variety of statistical (linear and nonlinear modeling, classical/Bayesian statistical tests, graphics, phylogenetic methods, genomic analyses, superb data handling).
- The R language provides an Open Source route to participation in that activity (more than 6135 available contributed packages).

List of packages: <http://cran.r-project.org/> #see under software (left column) > Packages

- R is available as a Free Software. It runs on most platforms: Linux, Windows and Mac OS.
- You can write your own functions for specific needs

Basic Functions in R: Preparing data

- To explore some basic R-functions, we need a dataset that can be read in R
- Open the .xlsx file that is in the course website ([data1](#)):

Birds_mass_BM_McNab_2009_class.xlsx

- **Suggestions:** use descriptive names for your data files and avoid 'blank' spaces
- This data is the from a review on metabolic rates across all birds (533 species) that include information about body mass, BMR (basal metabolic rate), time (diel behavior), torpor (animal dormancy during reduced food availability), climate preference, migration, distribution (islands, mountains), ability to flight, food preferences and original references.

See [ref1](#) for Jan 8

Comparative Biochemistry and Physiology, Part A 152 (2009) 22–45

Contents lists available at ScienceDirect

ELSEVIER

Comparative Biochemistry and Physiology, Part A

journal homepage: www.elsevier.com/locate/cbpa

CBP

Ecological factors affect the level and scaling of avian BMR

Brian Keith McNab*

Department of Zoology, University of Florida, Gainesville, FL 32611, USA

ARTICLE INFO

Article history:
Received 26 May 2008
Received in revised form 14 August 2008
Accepted 20 August 2008
Available online 27 August 2008

Keywords:
Avian flight
Ecological factors
Energetics
Geography
Mammalian flight
Migration
Non-passerines
Passerines
Phylogeny

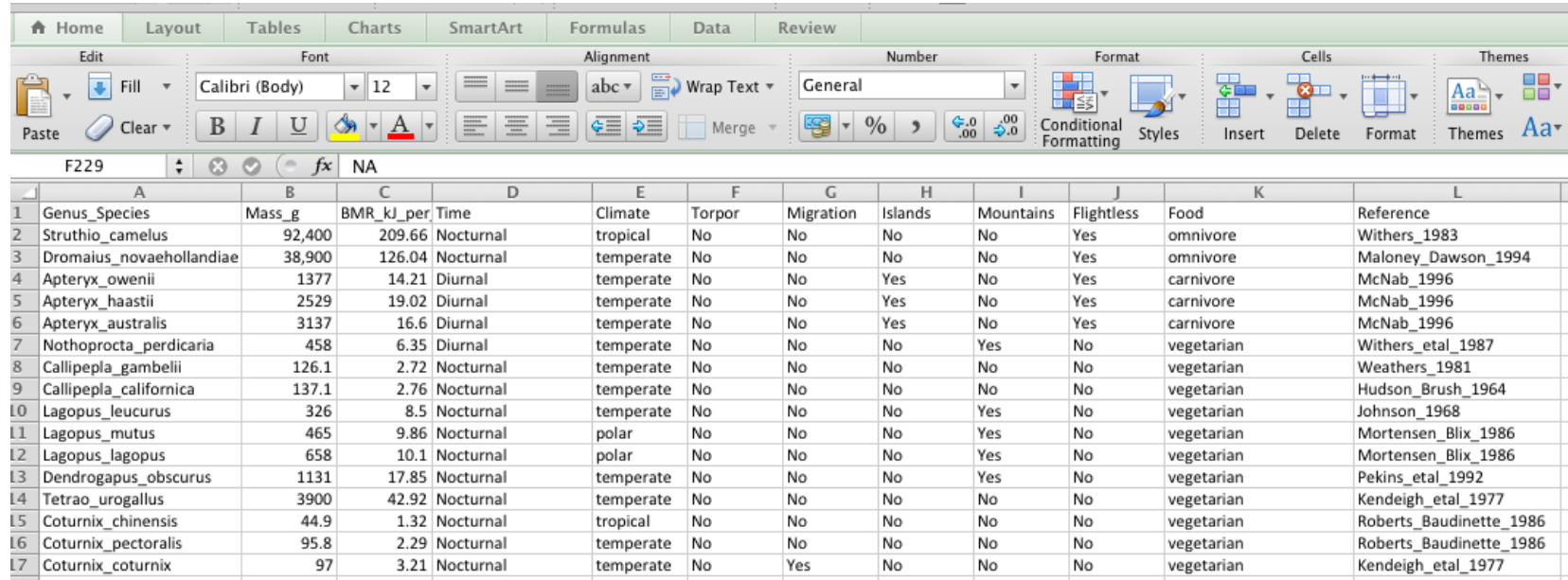
ABSTRACT

The basal rate of metabolism (BMR) in 533 species of birds, when examined with ANCOVA, principally correlates with body mass, most of the residual variation correlating with food habits, climate, habitat, a volant or flightless condition, use or not of torpor, and a highland or lowland distribution. Avian BMR also correlates with migratory habits, if climate and a montane distribution is excluded from the analysis, and with an occurrence on small islands if a flightless condition and migration are excluded. Residual variation correlates with membership in avian orders and families principally because these groups are behaviorally and ecologically distinctive. However, the distinction between passerines and other birds remains a significant correlate of avian BMR, even after six ecological factors are included, with other birds having BMRs that averaged 74% of the passerine mean. This combination of factors accounts for 97.7% of the variation in avian BMR. Yet, migratory species that belong to Anseriformes, Charadriiformes, Pelecaniformes, and Procellariiformes and breed in temperate or polar environments have mass-independent basal rates equal to those found in passerines. In contrast, penguins belong to an order of polar, aquatic birds that have basal rates lower than passerines because their flightless condition depresses basal rate. Passerines dominate temperate, terrestrial environments and the four orders of aquatic birds dominate temperate and polar aquatic environments because their high BMRs facilitate reproduction and migration. The low BMRs of tropical passerines may reflect a sedentary lifestyle as much as a life in a tropical climate. Birds have BMRs that are 30–40% greater than mammals because of the commitment of birds to an expensive and expansive form of flight.

© 2008 Elsevier Inc. All rights reserved.

Basic Functions in R: Preparing data

- Let's explore the .xlsx file and notice the codification of entries:



The screenshot shows the Microsoft Excel interface with a data table. The table has 13 columns (A-L) and 17 rows (1-17). The first row is the header, and the subsequent rows contain data for various species. The data is as follows:

	A	B	C	D	E	F	G	H	I	J	K	L
1	Genus_Species	Mass_g	BMR_kj_per	Time	Climate	Torpor	Migration	Islands	Mountains	Flightless	Food	Reference
2	Struthio_camelus	92,400	209.66	Nocturnal	tropical	No	No	No	No	Yes	omnivore	Withers_1983
3	Dromaius_novaehollandiae	38,900	126.04	Nocturnal	temperate	No	No	No	No	Yes	omnivore	Maloney_Dawson_1994
4	Apteryx_owenii	1377	14.21	Diurnal	temperate	No	No	Yes	No	Yes	carnivore	McNab_1996
5	Apteryx_haastii	2529	19.02	Diurnal	temperate	No	No	Yes	No	Yes	carnivore	McNab_1996
6	Apteryx_australis	3137	16.6	Diurnal	temperate	No	No	Yes	No	Yes	carnivore	McNab_1996
7	Nothoprocta_perdicaria	458	6.35	Diurnal	temperate	No	No	No	Yes	No	vegetarian	Withers_etal_1987
8	Callipepla_gambelii	126.1	2.72	Nocturnal	temperate	No	No	No	No	No	vegetarian	Weathers_1981
9	Callipepla_californica	137.1	2.76	Nocturnal	temperate	No	No	No	No	No	vegetarian	Hudson_Brush_1964
10	Lagopus_leucurus	326	8.5	Nocturnal	temperate	No	No	No	Yes	No	vegetarian	Johnson_1968
11	Lagopus_mutus	465	9.86	Nocturnal	polar	No	No	No	Yes	No	vegetarian	Mortensen_Blix_1986
12	Lagopus_lagopus	658	10.1	Nocturnal	polar	No	No	No	Yes	No	vegetarian	Mortensen_Blix_1986
13	Dendrogapus_obscurus	1131	17.85	Nocturnal	temperate	No	No	No	Yes	No	vegetarian	Pekins_etal_1992
14	Tetrao_urogallus	3900	42.92	Nocturnal	temperate	No	No	No	No	No	vegetarian	Kendeigh_etal_1977
15	Coturnix_chinensis	44.9	1.32	Nocturnal	tropical	No	No	No	No	No	vegetarian	Roberts_Baudinette_1986
16	Coturnix_pectoralis	95.8	2.29	Nocturnal	temperate	No	No	No	No	No	vegetarian	Roberts_Baudinette_1986
17	Coturnix_coturnix	97	3.21	Nocturnal	temperate	No	Yes	No	No	No	vegetarian	Kendeigh_etal_1977

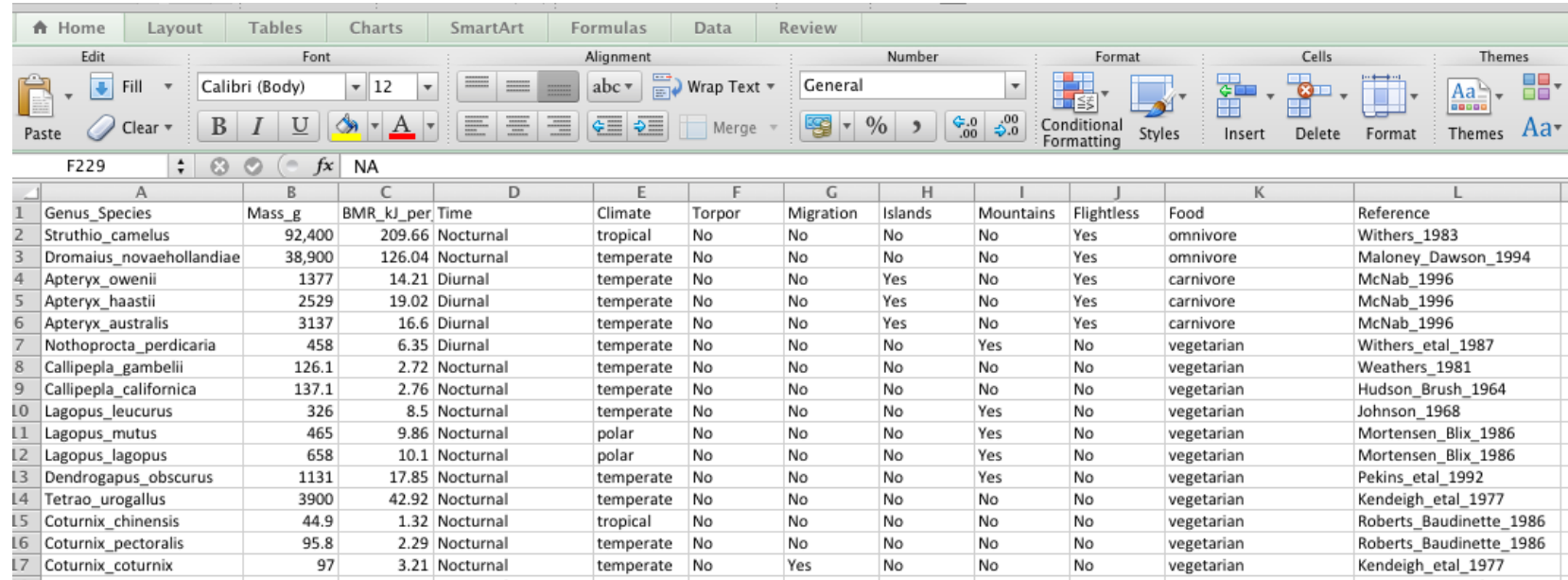
Header: first line in data file with descriptive variable names, no commas, no parenthesis, no blank spaces (use underscores “_”) and no unusual ASCII text characters. Notice that I added the information about the units for some variables such as _g for grams

Variables: Several types of variables, continuous (e.g., Mass_g), discrete binary (e.g., Torpor), categorical (e.g., Genus_species, Food).

Missing values: The correct way to entry such values is NA

Basic Functions in R: Preparing data

- Let's explore the .xlsx file and notice the codification of entries:



The screenshot shows an Excel spreadsheet with the following data:

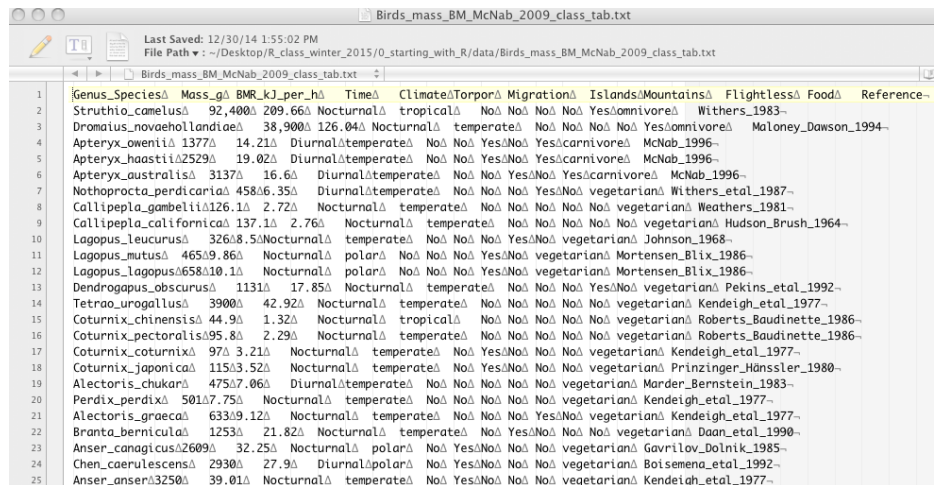
	A	B	C	D	E	F	G	H	I	J	K	L
	Genus_Species	Mass_g	BMR_kj_per	Time	Climate	Torpor	Migration	Islands	Mountains	Flightless	Food	Reference
2	Struthio_camelus	92,400	209.66	Nocturnal	tropical	No	No	No	No	Yes	omnivore	Withers_1983
3	Dromaius_novaehollandiae	38,900	126.04	Nocturnal	temperate	No	No	No	No	Yes	omnivore	Maloney_Dawson_1994
4	Apteryx_owenii	1377	14.21	Diurnal	temperate	No	No	Yes	No	Yes	carnivore	McNab_1996
5	Apteryx_haastii	2529	19.02	Diurnal	temperate	No	No	Yes	No	Yes	carnivore	McNab_1996
6	Apteryx_australis	3137	16.6	Diurnal	temperate	No	No	Yes	No	Yes	carnivore	McNab_1996
7	Nothoprocta_perdicaria	458	6.35	Diurnal	temperate	No	No	No	Yes	No	vegetarian	Withers_etal_1987
8	Callipepla_gambelii	126.1	2.72	Nocturnal	temperate	No	No	No	No	No	vegetarian	Weathers_1981
9	Callipepla_californica	137.1	2.76	Nocturnal	temperate	No	No	No	No	No	vegetarian	Hudson_Brush_1964
10	Lagopus_leucurus	326	8.5	Nocturnal	temperate	No	No	No	Yes	No	vegetarian	Johnson_1968
11	Lagopus_mutus	465	9.86	Nocturnal	polar	No	No	No	Yes	No	vegetarian	Mortensen_Blix_1986
12	Lagopus_lagopus	658	10.1	Nocturnal	polar	No	No	No	Yes	No	vegetarian	Mortensen_Blix_1986
13	Dendrogapus_obscurus	1131	17.85	Nocturnal	temperate	No	No	No	Yes	No	vegetarian	Pekins_etal_1992
14	Tetrao_urogallus	3900	42.92	Nocturnal	temperate	No	No	No	No	No	vegetarian	Kendeigh_etal_1977
15	Coturnix_chinensis	44.9	1.32	Nocturnal	tropical	No	No	No	No	No	vegetarian	Roberts_Baudinette_1986
16	Coturnix_pectoralis	95.8	2.29	Nocturnal	temperate	No	No	No	No	No	vegetarian	Roberts_Baudinette_1986
17	Coturnix_coturnix	97	3.21	Nocturnal	temperate	No	Yes	No	No	No	vegetarian	Kendeigh_etal_1977

Suggestions:

- A column of data should contain only one data type (i.e., numbers or characters, not both). R will interpret any column with even a single character as non-numeric.
- Create a relational variables that allow to merge different data files. For example, you can create one file that include the species name and data about morphology. Another file that has the same species name and physiology variables. An associated phylogeny should also have the same species name.
- Maintain effective metadata (data about the data or field notes as verbatim)

Basic Functions in R: Preparing data

- To export your data you can use excel: File>Save as>Format (choose: Tab delimited Text .txt).
- I prefer to copy and paste directly in a text editor such as TextWranger or BBEdit. Notice that I updated the editor preferences to show tab stops, line numbers, invisibles (blank spaces, returns, tabs). See last class for information.



	Genus	Species	Mass_g	BMR_kJ_per_h	Time	Climate	Torpor	Migration	Islands	Mountains	Flightless	Food	Reference	
1	Struthio	camelus	92,400	209.66	Nocturnal	tropical	No	No	No	Yes	omnivore		Withers_1983-	
2	Dromaius	novaeollandiae	38,900	126.04	Nocturnal	temperate	No	No	No	Yes	omnivore		Maloney_Dawson_1994-	
3	Apteryx	owenii	1377	14.21	Diurnal	temperate	No	No	Yes	No	Yes	carnivore	McNab_1996-	
4	Apteryx	haastii	2529	19.02	Diurnal	temperate	No	No	Yes	No	Yes	carnivore	McNab_1996-	
5	Apteryx	australis	3137	16.6	Diurnal	temperate	No	No	Yes	No	Yes	carnivore	McNab_1996-	
6	Nothoprocta	perdicaria	458	6.35	Diurnal	temperate	No	No	Yes	No	vegetarian		Withers_etal_1987-	
7	Callipepla	gambelii	126.1	2.72	Nocturnal	temperate	No	No	No	No	vegetarian		Weathers_1981-	
8	Callipepla	californica	137.1	2.76	Nocturnal	temperate	No	No	No	No	vegetarian		Hudson_Brush_1964-	
9	Lagopus	leucurus	326	8.5	Nocturnal	temperate	No	No	Yes	No	vegetarian		Johnson_1968-	
10	Lagopus	mutus	465	9.86	Nocturnal	polar	No	No	Yes	No	vegetarian		Mortensen_Blix_1986-	
11	Lagopus	lagopus	658	10.1	Nocturnal	polar	No	No	Yes	No	vegetarian		Mortensen_Blix_1986-	
12	Dendrogapus	obscurus	1131	17.85	Nocturnal	temperate	No	No	No	Yes	No	vegetarian		Pekins_etal_1992-
13	Tetrao	urogallus	3900	42.92	Nocturnal	temperate	No	No	No	No	vegetarian		Kendeigh_etal_1977-	
14	Coturnix	chinensis	44.9	1.32	Nocturnal	tropical	No	No	No	No	vegetarian		Roberts_Baudinette_1986-	
15	Coturnix	pectoralis	95.8	2.29	Nocturnal	temperate	No	No	No	No	vegetarian		Roberts_Baudinette_1986-	
16	Coturnix	coturnix	97	3.21	Nocturnal	temperate	No	Yes	No	No	vegetarian		Kendeigh_etal_1977-	
17	Coturnix	japonica	115	3.52	Nocturnal	temperate	No	Yes	No	No	vegetarian		Prinzinger_Haessler_1980-	
18	Alectoris	chukar	475	7.06	Diurnal	temperate	No	No	No	No	vegetarian		Marder_Bernstein_1983-	
19	Perdix	perdix	501	7.75	Nocturnal	temperate	No	No	No	No	vegetarian		Kendeigh_etal_1977-	
20	Alectoris	graeca	633	9.12	Nocturnal	temperate	No	No	Yes	No	vegetarian		Kendeigh_etal_1977-	
21	Branta	bernicula	1253	21.82	Nocturnal	temperate	No	Yes	No	No	vegetarian		Daan_etal_1990-	
22	Anser	canagicus	2609	32.25	Nocturnal	polar	No	Yes	No	No	vegetarian		Gavrilov_Dolnik_1985-	
23	Chen	caerulescens	2930	27.9	Diurnal	polar	No	Yes	No	No	vegetarian		Boisemena_etal_1992-	
24	Anser	anser	3250	39.01	Nocturnal	temperate	No	Yes	No	No	vegetarian		Kendeigh_etal_1977-	
25														

- TextWranger allows to change characters, words, numbers.
Change the tabs (tab delimited file) for commas (csv file).
Search>Find>in the window (Find: paste a 'tab' in Replace: paste or type a ',')
Click 'Replace all'
- Save file as 'Birds_mass_BM_McNab_2009_class_csv.txt' (see course website)

The R environment



```
R Console
R version 3.1.2 (2014-10-31) -- "Pumpkin Helmet"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin10.8.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.65 (6833) x86_64-apple-darwin10.8.0]

[History restored from /Users/jcsantos/.Rhistory]

>
```

• The R environment allows an integrated suite for data manipulation, calculation and graphical display.

- (1) Data handling and storage facility
- (2) Operators for calculations on: arrays, matrices, and data frames
- (3) Plotting functions for display either on-screen (e.g., jpg, tiff) or on hardcopy (e.g., pdf, png)
- (4) Automatization and programming language which includes conditionals, loops, recursive functions and input and output facilities

- We are going to use the R-reference card from Matt Baggott: 'Baggott_refcard_v2.pdf'
*you can also get this card from: <http://cran.r-project.org/other-docs.html>

The R environment: Installing Packages and Help



- Check installed version of R. Any text after `#` allows to comment (i.e., this text will not run or include commands readable by the software)

```
> sessionInfo() # What version and the platform do you have?
```

- Let's install some packages: 'plyr' and 'reshape' both packages are facilitate data manipulation and transformation.

```
> install.packages("plyr") # you might need to retype the quotation marks
```

- A window will appear asking for closest CRAN mirror (i.e., the repository of all contributed packages of R). Select: [USA \(CA 1\)](#)

```
> install.packages("plyr")
--- Please select a CRAN mirror for use in this session ---
trying URL 'http://cran.cnr.Berkeley.edu/bin/macosx/contrib/3.1/plyr_1.8.1.tgz'
Content type 'application/x-gzip' length 837044 bytes (817 Kb)
opened URL
=====
downloaded 817 Kb
```

The R environment: Installing Packages and Help



- Load library in the R environment

```
> library(plyr) # Repeat the same process for 'reshape' and check if both packages are  
# loaded using sessionInfo()
```

- All packages in CRAN have its own manual of the functions that they include. You can 'google' the name of the package (e.g., reshape + CRAN)

For plyr: <http://cran.r-project.org/web/packages/reshape/index.html>

See the direct links in the course website

- You can find information for a function of an specific package that has been loaded in R-environment using a '?' and the name of the function (e.g., reshape)

```
> ?reshape # A new window will appear with the same information as is in the pdf manual
```

- This information will include a brief description, usage, arguments (input from the user), specific details on the function use, value (the the type of output of the function) and examples of its use.

- **Tip:** If you press the top arrow, the previous typed command will appear



The R environment: Reading Tables

- In order to read our data files, we need to indicate to R where to find our files and output any resulting analyses. You can do as follows:

R top menus: Misc>Change Working Directory (select the directory that has our file)

- Alternatively, you can set your working directory with the command:

```
>setwd(dir)# where dir is the path to your folder, you can get it using terminal (Mac OS)
```

- Once the working directory is set. We can read data files using the `read.table` command. The options `header` indicates that the table first line as variable names and `sep` contains 'tab' in quotations. For a comma delimited file you change `sep = ","`

```
> read.table (file="Birds_mass_BM_McNab_2009_class_tab.txt", header = TRUE, sep = " ")
```

- We can assign this table to an object with '`<-`' arrow

```
> birds_MR <- read.table("Birds_mass_BM_McNab_2009_class_tab.txt", header = TRUE, sep = " ")  
> birds_MR
```

- This matrix has 533 entries so we might want to check only the first 6 rows

```
> head(birds_MR)
```

The R environment: Data objects (Numeric Vectors)



- A numerical vector consists of numbers and it can be extracted from our data table by selecting only one column that includes numbers

```
> Mass <-birds_MR$Mass_g # we can assign a column of the table by using '$' which says
                        #assign only column Mass_g of 'birds_MR' to vector 'Mass'
> str(Mass)# it will give you the structure of the object: num [1:533] 92400 ...
> class(Mass)# it will give you the class of the object: [1] "numeric"
> length (Mass)# it will give you the number of entries: [1] 533
```

Repeat the same process with the BMR_kJ_per_h variable.

```
> BMR <-birds_MR$BMR_kJ_per_h
```

- We can do different numerical or logical (i.e., true or false) calculations with these vectors

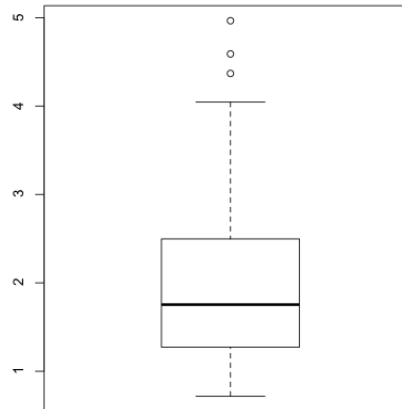
```
> log10(Mass) #logarithm base 10
> Mass+1 #adds 1 to all entries in Mass
> Mass[12] # prints the element 12 in the vector Mass: [1] 1131
> Mass[1:12] # prints elements 1 to 12 in the vector Mass
> Mass [Mass > 1000] # prints elements with more than 1000 g
> Mass [Mass == 1377] # prints elements exactly 1377 g
> Mass [Mass >= 1377] # prints elements greater than or equal 1377 g
> unique(Mass) #list unique elements that are not repeated
> duplicated(Mass) #logical is each element duplicated? True or False
> Mass_specific_BMR_kj_per_h_g <- BMR/Mass #calculations between numeric vectors
```

The R environment: Data objects (Numeric Vectors)



- We can make a box plot of the elements of the numerical vector

```
> boxplot(log10(Mass))
```



- We can make other numeric vectors with some elements of Mass object

```
> Mass_first_10 <- Mass[1:10] #first 10 elements
> Mass_some_elements <- Mass[c(1,5,15,3,4,400)] #some specific elements
> Mass_conditional <- Mass[Mass < 1000 & Mass > 100] # vector with elements greater than
# 100 g but less than 1000 g
> Mass_duplicated <- Mass[duplicated(Mass)] #duplicated elements in Mass
> Mass_non_duplicated <- Mass[!duplicated(Mass)] #similar to unique(), uses ! to indicate
#the opposite to the logical duplicate
```

Suggestion: Notice that I have given to each new object a new name. Every time that the same name object is overwritten (i.e., assigned with <-), previous contents are deleted and this process is irreversible.

The R environment: Data objects (Categorical Vectors)



- Character vector contain words and character values. So, vectors that have elements with numbers and letters will be consider a character vector.

```
> food <-birds_MR$Food # factor vector food
> str(food)# it will give you the structure of the object: Factor w/ 3 levels...
> class(food)# it will give you the class of the object: [1] "factor"
> length (food)# it will give you the number of entries: [1] 533
```

```
> food_character <- as.character(food) #converts factor to character vector
```

- We can do counting calculations with character vectors

```
> length (which(food_character == "carnivore")) #how many times "carnivore" is repeated
> carnivore_n <- length (which(food_character == "carnivore")) #how many times
#'carnivore' is repeated
> omnivore_n <- length (which(food_character == "omnivore"))
> vegetarian_n <- length (which(food_character == "vegetarian"))

> vector_food <- c(carnivore_n, omnivore_n, vegetarian_n)
> bplt <- barplot(vector_food, names.arg = c("carnivore", "omnivore", "vegetarian") ,
col = c("red", "yellow", "green"), cex.axis = 2, cex.names = 2)
> text(x= bplt, y= vector_food+5, labels=as.character(vector_food), xpd=TRUE, cex = 2)
```

The R environment: Data objects (Categorical Vectors)

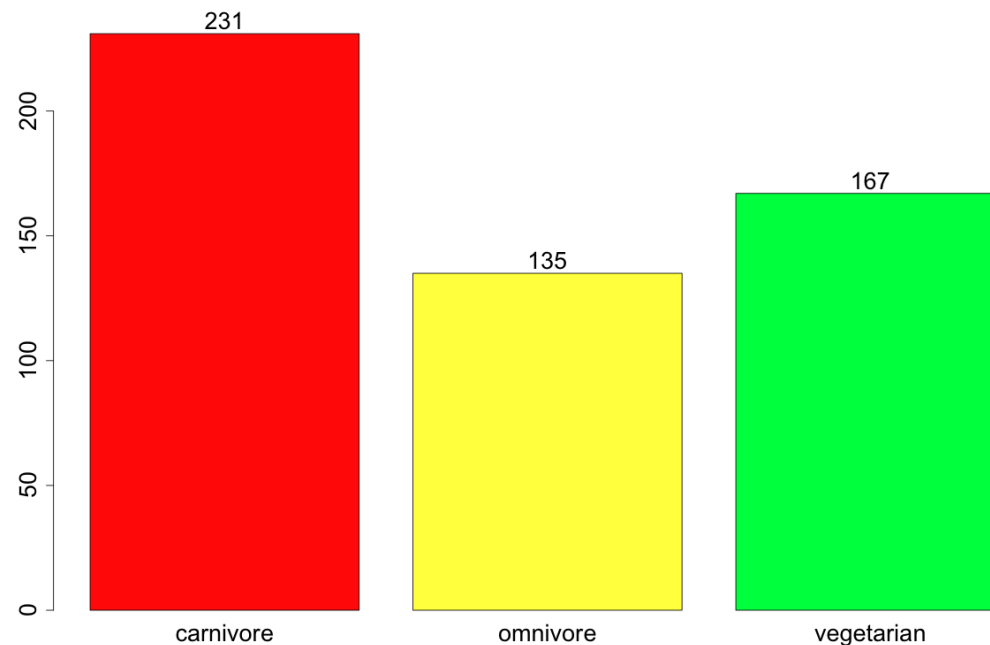


- Character vector contain words and character values. So, vectors that have elements with numbers and letters will be consider a character vector.

```
> food <-birds_MR$Food # factor vector food
> str(food)# it will give you the structure of the object: Factor w/ 3 levels...
> class(food)# it will give you the class of the object: [1] "factor"
> length (food)# it will give you the number of entries: [1] 533

> food_character <- as.character(food) #converts factor to character vector
```

- We can do counting calculations with character vectors





The R environment: Data frames

- Data frames are the single most important object in the R environment and most packages use data frames to do their calculations

```
> str(birds_MR)# Our input data is actually a data frame
'data.frame':  533 obs. of  12 variables:
 $ Genus_Species: Factor w/ 533 levels "Accipiter_cooperi",...: 472 184 54 53 52 331 88 87 264 265 ...
 $ Mass_g       : num  92400 38900 1377 2529 3137 ...
 $ BMR_kJ_per_h : num  209.7 126 14.2 19 16.6 ...
 $ Time        : Factor w/ 4 levels "Diurnal","Diurnal_Nocturnal",...: 4 4 1 1 1 1 4 4 4 4 ...
 $ Climate     : Factor w/ 4 levels "polar","temperate",...: 4 2 2 2 2 2 2 2 1 ...
 $ Torpor      : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
 $ Migration   : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
 $ Islands     : Factor w/ 2 levels "No","Yes": 1 1 2 2 2 1 1 1 1 1 ...
 $ Mountains  : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 1 1 2 2 ...
 $ Flightless  : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 1 1 1 1 1 ...
 $ Food        : Factor w/ 3 levels "carnivore","omnivore",...: 2 2 1 1 1 3 3 3 3 3 ...
 $ Reference   : Factor w/ 130 levels "Adams_Brown_1984",...: 127 64 71 71 71 128 114 45 47 84 ...
```

- We can refer to any value, or subset of values, in this data frame

```
> birds_MR [100,3] #print element in row 100 and column 3 (i.e., [m,n] matrix notation)
> birds_MR [100,] #print elements in row 100
> birds_MR [,3] #print elements in column 3 (i.e., BMR_kJ_per_h variable)
> birds_MR [1:10,] #print elements in the first 10 rows
```



The R environment: Data frames

- Accessing data frame information

```
> dim(birds_MR) # dimensions of data frame in rows by column in [m,n] matrix notation
> names(birds_MR) # Get the names of variables in the data frame
> summary(birds_MR) # basic statistics of the variables in the data frame
```

- Dealing with missing data in data frame

```
> birds_MR_incomplete_cases <- birds_MR[!complete.cases(birds_MR),] # rows with missing
#values
> birds_MR_complete_cases <- birds_MR [complete.cases(birds_MR),] # delete rows with
# incomplete data
> birds_MR_complete_cases_2 <- na.omit(birds_MR) # similar to process as above
```

How many species were eliminated?



The R environment: Data frames

- Subsetting data frame into other data frames

```
> birds_nocturnal_temperate <- subset(birds_MR, Time=='Nocturnal' & Climate ==  
'tropical')
```

```
# this will select rows (species) that are both Nocturnal and tropical.  
# Notice the lower and upper case of the names, they need to be exact as is in the data  
# frame
```

```
> birds_big <- subset(birds_MR, Mass_g>1000) # this will select species with more 1000 g
```

```
> birds_big_temperate <- subset(birds_MR, Mass_g > 1000 & Climate == 'temperate')
```

```
> birds_big_not_temperate <- subset(birds_MR, Mass_g > 1000 & !Climate == 'temperate')
```

```
# Notice the ! Indicating that we do not want the species that are temperate
```

```
> birds_temperate_polar <- subset(birds_MR, Climate %in% c('temperate', 'polar'))
```

```
# Notice the %in% that indicate 'nested' in Climate variable states: temperate and polar
```




The R environment: Data frames

- We can create new variables in data frame that include calculations between other variables in data frame

```
> birds_MR$Mass_specific_BMR_kj_per_h_g <- birds_MR$BMR_kJ_per_h/birds_MR$Mass_g
> head(birds_MR) #a new column with the new calculated variable will appear

> birds_MR$size <- ifelse(birds_MR$Mass_g > 1000, "big_bird", "small_bird")

#ifelse function is useful (see ?ifelse). Notice its use ifelse(test, yes, no)

> head(birds_MR) #a new column with the new categorical variable will appear
```

- Some basic statistical functions

```
> mean(birds_MR$Mass_g) # you can get this from the summary() function
> sd (birds_MR$Mass_g) # standard deviation
> min(birds_MR$Mass_g) # minimum value
> max(birds_MR$Mass_g) # maximum value

> birds_MR$log10Mass_g <- log10(birds_MR$Mass_g)
> birds_MR$log10BMR <- log10(birds_MR$BMR_kJ_per_h)

#we needed to transform this variables for log-log calculations necessary for Mass-BMR
regressions
```



The R environment: Data frames

- Some basic statistical functions

```
> cor(birds_MR$log10Mass_g, birds_MR$log10BMR , method = c("pearson"))
```

```
[1] 0.9711327 #?cor to understand the use of this function
```

```
> cor.test( ~ birds_MR$log10Mass_g + birds_MR$log10BMR, data=birds_MR)
```

```
      Pearson's product-moment correlation
```

```
data:  birds_MR$log10Mass_g and birds_MR$log10BMR
```

```
t = 93.8134, df = 531, p-value < 2.2e-16
```

```
alternative hypothesis: true correlation is not equal to 0
```

```
95 percent confidence interval:
```

```
 0.9658656 0.9755971
```

```
sample estimates:
```

```
      cor
```

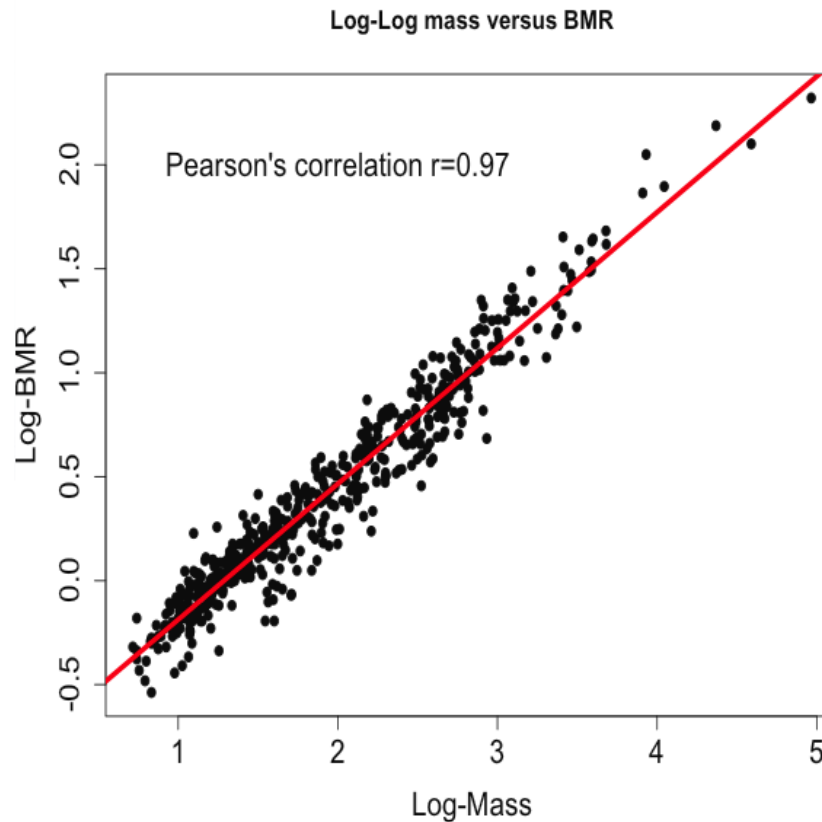
```
0.9711327
```



The R environment: Data frames

- Scatter plot of our relationship between variables

```
> plot(birds_MR$log10Mass_g, birds_MR$log10BMR, main="Log-Log mass versus BMR",  
xlab="Log-Mass", ylab="Log-BMR", pch=19, cex.axis=1.5, cex.lab = 1.5)  
> abline(lm(birds_MR$log10BMR~birds_MR$log10Mass_g ), col="red", lwd = 4) # regression  
line (y~x)  
> text(x= 2, y= 2, labels="Pearson's correlation r=0.97", cex = 1.5)
```





The R environment: Other useful functions

- Column selection from data frame

```
> birds_MR_based_food <- data.frame(birds_MR$Food, birds_MR$log10Mass_g, birds_MR$log10BMR) #this selects specific columns and crates a new data frame
```

```
> birds_MR_based_food_2 <- birds_MR [c("Food", "log10Mass_g", "log10BMR")] #similar results
```

```
> tail(birds_MR_based_food) # check the last rows of this data frame
```

- Renaming columns of the data frame

```
> names(birds_MR_based_food) <- c('food', 'log10Mass', 'log10BMR') #change the name of variables
```

- Writing table to file

```
> write.table(birds_MR, file="Birds_mass_BM_McNab_2009_class_csv_updated.txt", col.names = TRUE, sep = " ,")
```



The R environment: Functions

- You can make your own functions

```
function_name <- function(arguments_1, arguments_2, ...) { expression...do something }
```

- Basic example function

```
> add_two_variables <- function (x,y) {  
      z <- x + y  
      z  
    }  
> add_two_variables(1,2)  
> a <- sample(1:40, 6) #sample 6 numbers randomly from 1 to 40  
> b <- sample(1:40, 6)  
> add_two_variables(a,b)  
> c <- add_two_variables(a,b)  
> results <- data.frame(a,b,c)  
> results
```

- Functions can be applied as loops

```
> lapply (birds_MR, sd) # list of results of applying function in this case 'sd' to our  
data frame
```



The R environment: Exploring beyond

- Even a better correlation plot using 'ggplot2' package

```
install.packages("ggplot2")
install.packages("plyr")

library(ggplot2)
library(plyr)

birds_MR_based_food <- data.frame(birds_MR$Food, birds_MR$log10Mass_g, birds_MR$log10BMR)
head(birds_MR_based_food)
names(birds_MR_based_food) <- c('food', 'log10Mass', 'log10BMR') #change the name of variables

cor_func <- function(x) #we can build our own function to do correlations for specific groups
{ return(data.frame(COR = round(cor(x$log10Mass, x$log10BMR),4))) }

ddply(birds_MR_based_food, .(food), cor_func) #ddply is a loop function that allows to run groups of data
#based on grouping variable (food)

cor_by_food <- ddply(birds_MR_based_food, .(food), cor_func) #ddply is a loop function that allows to run
groups of data based on grouping variable (food)

ggplot(data = birds_MR_based_food, aes(x = log10Mass, y = log10BMR, group=food, colour=food)) + #define x
and y variables, grouping variable, coloring variable
  geom_smooth(method = "lm", se=FALSE, aes(fill = food), formula = y ~ x) + #define the method
for regression in this case lm (least square regression), color by food type
  geom_point(aes(shape = food, size = 2)) +
  annotate("text", x = 1, y = 2.5, label = "carnivore R=", color="red") +
  annotate("text", x = 1.5, y = 2.5, label = as.character(cor_by_food[1,2]), color="red") +
  annotate("text", x = 1, y = 2, label = "omnivore R=", color="darkgreen") +
  annotate("text", x = 1.5, y = 2, label = as.character(cor_by_food[2,2]), color="darkgreen") +
  annotate("text", x = 1, y = 1.5, label = "vegetarian R=", color="blue") +
  annotate("text", x = 1.5, y = 1.5, label = as.character(cor_by_food[3,2]), color="blue")
ggsave("correlations_food.pdf", width = 16, height = 9, dpi = 120) #it will save the graph as a pdf
```



The R environment: Data frames

- Even a better plot using 'ggplot2' package

