**How to start analyzing data with R for beginners!**

**By Haydee Peralta-Vazquez**

**Introduction**

Data analysis could be an intimidating activity if we are not familiar with data management and visualization. A basic definition of data management simply refers to the ability of organizing any information that was produced from experiments, direct observations and/or models into unique data items (e.g. images, maps, numerical datasets, etc.). Well-organized numerical datasets, for example, simplify the production of diagrams such as boxplots or scatterplots. Such diagrams are often useful to detect or predict trends and patterns from categorical variables (e.g. boxplots showing differences that are due to sex) or correlations between continuous variables (e.g. a smooth scatterplot that shows weight as a function of volume); and are also a succinct way to summarize results from a research experiment. This is ultimately the most important aspect of data analysis: discovering how the occurrence of certain events (or observations) explains some previously detected phenomena (usually the motivation for scientific research). The aim of this tutorial is to introduce a first-time-R user how to start analyzing data by simplifying and visualizing a large data set using dplyr, ggplot2 and lme4.

**First thing first**

Lets first clean the R environment and then call the packages. Note that I also call for updates before I actually call for the packages just to make sure everything will work fine.

rm(list=ls(all=TRUE))

# Updating and calling packages

update.packages(ask = FALSE, repos = 'http://cran.rstudio.org')

library(dplyr)

library(ggplot2)

library(lme4)

setwd("/Users/Haydeeakin/Desktop/Flight mill raw data for R")

flight <- read.csv("flightmill complete.csv", header=TRUE)

**First glance at the data**

As a first step raw data should be transformed into a comprehensible data file with an intuitive and well-organized structure. Raw data should be understood as any dataset that doesn’t have a concise and clear organization of its observations into a logical table or data frame.

Let’s have a look to one example of a disorganized dataset:

> str(flight)

'data.frame': 4914 obs. of 44 variables:

$ beetle\_id : Factor w/ 147 levels "L1016J14-25",..: 2 2 2 2 2 2 2 2 2 2 ...

$ julian\_emergence\_date : int 169 169 169 169 169 169 169 169 169 169 ...

$ sex : Factor w/ 2 levels "f","m": 1 1 1 1 1 1 1 1 1 1 ...

$ log\_num : int 10 10 10 10 10 10 10 10 10 10 ...

$ pre\_flight\_weight\_mg : num 9.8 9.8 9.8 9.8 9.8 9.8 9.8 9.8 9.8 9.8 ...

$ length\_mm : num 5.06 5.06 5.06 5.06 5.06 ...

$ width\_mm : num 2 2 2 2 2 2 2 2 2 2 ...

$ pre\_flight\_tarso : int 0 0 0 0 0 0 0 0 0 0 ...

$ pre\_flight\_procto : int 0 0 0 0 0 0 0 0 0 0 ...

$ preflight\_total\_mites : int 0 0 0 0 0 0 0 0 0 0 ...

$ p\_a\_mites\_preflight : int 0 0 0 0 0 0 0 0 0 0 ...

$ post\_flight\_weight.\_mg : num 9.03 9.03 9.03 9.03 9.03 9.03 9.03 9.03 9.03 9.03 ...

$ post\_flight\_tarso : int 0 0 0 0 0 0 0 0 0 0 ...

$ post\_flight\_procto : int 0 0 0 0 0 0 0 0 0 0 ...

$ post\_flight\_total\_mites : int 0 0 0 0 0 0 0 0 0 0 ...

$ p\_a\_mites\_postflight : int 0 0 0 0 0 0 0 0 0 0 ...

$ post\_flight\_tarso\_lost : int 0 0 0 0 0 0 0 0 0 0 ...

$ post\_flight\_procto\_lost : int 0 0 0 0 0 0 0 0 0 0 ...

$ post\_flight\_total\_mite.\_lost: int 0 0 0 0 0 0 0 0 0 0 ...

$ wet\_weight\_lost\_mg : num 0.77 0.77 0.77 0.77 0.77 0.77 0.77 0.77 0.77 0.77 ...

$ u\_elytra\_tarso : int 0 0 0 0 0 0 0 0 0 0 ...

$ u\_elytra\_procto : int 0 0 0 0 0 0 0 0 0 0 ...

$ total\_u\_elytra\_mites : int 0 0 0 0 0 0 0 0 0 0 ...

$ dry\_weight\_mg\_post\_flight : num 3.68 3.68 3.68 3.68 3.68 3.68 3.68 3.68 3.68 3.68 ...

$ volume\_mm3 : num 10.6 10.6 10.6 10.6 10.6 ...

$ condition\_mg\_incorrect : num -0.331 -0.331 -0.331 -0.331 -0.331 ...

$ beettle\_age\_days : int 15 15 15 15 15 15 15 15 15 15 ...

$ flight\_mill : int 1 1 1 1 1 1 1 1 1 1 ...

$ date\_flight : Factor w/ 19 levels "July 10 2014",..: 9 9 9 9 9 9 9 9 9 9 ...

$ month\_flight : Factor w/ 2 levels "July","Oct": 1 1 1 1 1 1 1 1 1 1 ...

$ julian\_flight\_date : int 184 184 184 184 184 184 184 184 184 184 ...

$ accumulated\_intervals : int 2 1 2 2 2 2 1 2 1 2 ...

$ counts\_bout : int 14 7 8 8 8 7 6 7 7 7 ...

$ time\_bout : int 32 16 32 32 32 32 16 32 16 32 ...

$ start\_time : int 102 2310 7430 8358 8630 9526 9782 10006 10230 10358 ...

$ raw\_num\_this\_file : int 1 2 3 4 5 6 7 8 9 10 ...

$ raw\_num\_original\_files : int 5174 5311 5632 5690 5707 5763 5778 5793 5806 5815 ...

$ num\_circunferences : num 7 3.5 4 4 4 3.5 3 3.5 3.5 3.5 ...

$ distance\_cm : num 704 352 402 402 402 ...

$ speed : num 22 22 12.6 12.6 12.6 ...

$ speed\_km\_hr : num 0.792 0.792 0.452 0.452 0.452 ...

$ treatment : Factor w/ 2 levels "mite","no mite": 2 2 2 2 2 2 2 2 2 2 ...

$ distance\_km : num 0.00704 0.00352 0.00402 0.00402 0.00402 ...

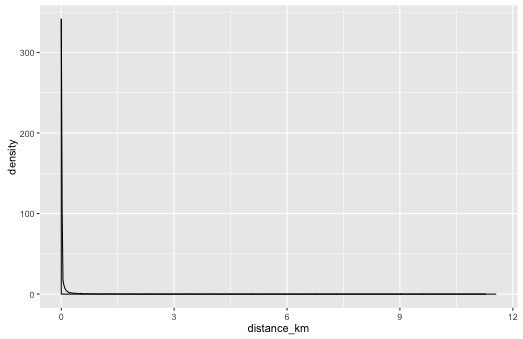
$ distance\_km.t : num 1.01 1 1 1 1 ...

As we can see, this dataset contains 44 variables and 4914 observations. It comes from a behavioral study that focused on testing whether ectosymbiont load affects how well mountain pine beetle dispersed over an eight hours period and whether beetle condition predicts better dispersers. Beetles are adult males and females that grew under different natal environments (e.g. log\_num for this specific dataset) and treatments (e.g. mite, no mite). The study allowed beetles to disperse for eight hours. This dataset only includes beetles that flew at leas once during those eight hours. Therefore there is more than one flight event for every beetle, in other words there are several flight observations per beetle and the length of observations is unequal for every beetle. This is a relevant piece of information that will help to decide which variables should be included in a more comprehensible dataset. The data set as it is only allows for certain type of graphs; for example those that examine general data distributions.

Let’s have the first glance at the distance data. We tell ggplot to plot distance\_km; we assign it to and object and later add the type of graph we want, as shown below (fig. 1).

dpt<-ggplot(flight, aes(distance\_km))

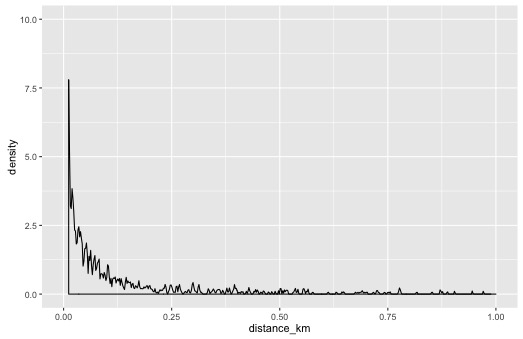
dpt+geom\_density()



Note that the previous figure is difficult to interpret. It shows there were very few beetles that flew much more that the majority, basically that there are outliners. Let’s now have a look at what the majority did instead (fig. 2). To do that we need to modify the limits of the x and y axis.

dp<-ggplot(flight, aes(distance\_km))

dp+geom\_density() + xlim(0, 1) + ylim(0,10)

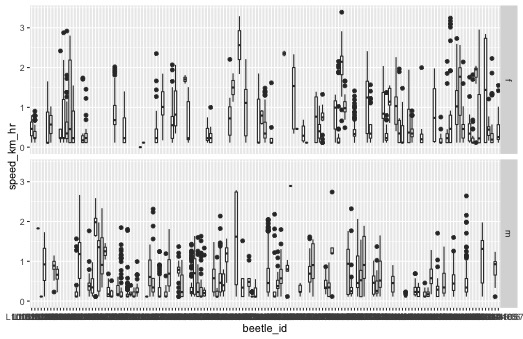


On this second graph most beetles flew short distances of less than 0.25km and few flew almost 1km on a single flight!

What about how fast beetles flew? Let’s plot the distribution of speeds per beetle using a bloxplot graph checking for the simplest distinction between females and males. Note that first line of code is basically to tell ggplot what are the x and y variables and from where these should be called. The second line of code only adds on the desired characteristics for the plot; for example: that this is a bloxplot that should show the distribution of speeds per beetle and that it should be divided in two grids one for males and one for females, see code and figure (fig. 3).

bp <- ggplot(flight, aes(beetle\_id, speed\_km\_hr))

bp + geom\_boxplot() + facet\_grid(sex~.)



**Trimming the raw dataset**

The previously shown dataset isn’t very practical. We need to retrieve those variables that are relevant for now. We use the package dplyr to do that. The following piece of code is shown in three parts to understand the flow of the coding.

First, we select the data and assign it to an object call “temp”.

#1

temp<-select(flight, beetle\_id, log\_num, julian\_emergence\_date, sex, pre\_flight\_weight\_mg, length\_mm, preflight\_total\_mites, distance\_km, speed\_km\_hr, volume\_mm3) %>%

Second, we transform one of the variables into factors using “mutate” then group the dataset by beetle.

#2

mutate(log\_num=factor(log\_num)) %>%

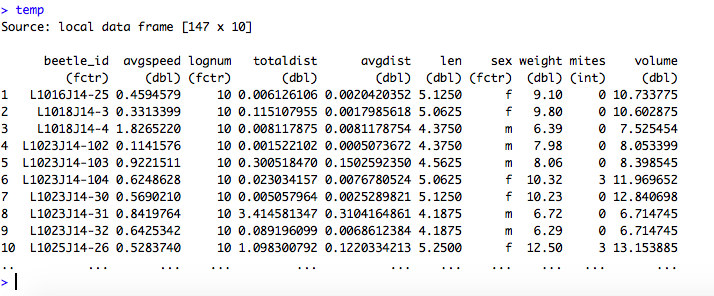
group\_by(beetle\_id) %>%

Finally, we tell R to summarize some variables by certain test statistics - mean (average distance and average speed), sum of all single flights per beetle (total distance) - and keep unique cases for other selected variables. Note that some variables from the raw dataset have duplicated observations per beetle (e.g. weight and sex).

#3 summarize(avgspeed=mean(speed\_km\_hr),lognum=unique(log\_num),totaldist=sum(distance\_km),avgdist=mean(distance\_km),len=unique(length\_mm),sex=unique(sex),weight=unique(pre\_flight\_weight\_mg),mites=unique(preflight\_total\_mites),volume=unique(volume\_mm3))

temp

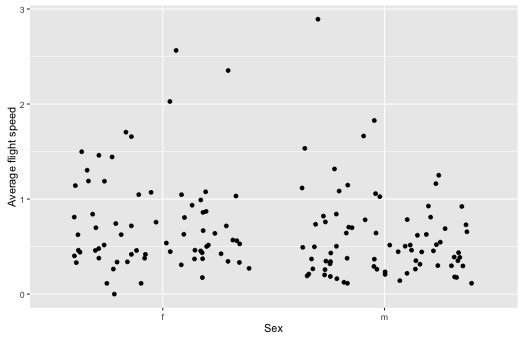
We have now a data frame with 147 rows of information and only ten variables. Each row of information represents a single beetle (fig. 4).



**Describing the data from the trimmed dataset**

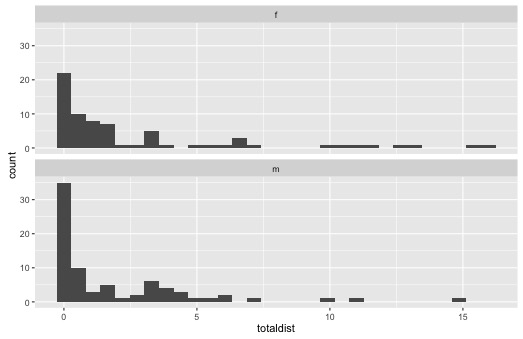
Now we can start describing the data. Let’s check, for example, whether males or females flew faster than the other sex with a graph. For this task we call ggplot, specify the dataset name, specify the variables (sex for the x axis and average speed for the y axis) and define the type of graph. In this case we call for geom jitter. This type of graph handles overplotting produced by smaller datasets (fig. 5).

ggplot(temp,aes(sex,avgspeed))+geom\_jitter()+labs(x='Sex',y='Average flight speed')



As we can see, there is no patter due to sex, in other words neither sex flew faster than the other. But what about who flew more? Let’s check this time with a histogram. In this case we only need to specify one variable that will be grouped by sex. Additionally, we call for totaldist and add two more objects: geom\_histogram and facet\_wrap by sex. We also specify how we want the panels or facets to be displayed by telling R we want the number of columns equals to 1. See line of code below.

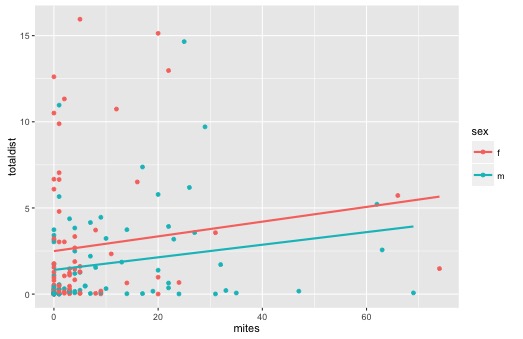
ggplot(temp,aes(totaldist))+geom\_histogram()+facet\_wrap(~sex,ncol=1)



It seems that male and female beetles flew about the same.

Now, lets look at whether mite load affects the dispersal behavior of beetles. For this purpose we first visualize the data and then perform a linear model using the package lm4r that was loaded at the beginning. A scatterplot with the linear regression fit for sex is a good way to visualize the data. We can do this with geom\_point and geom\_smooth method “lm”.

ggplot(temp,aes(mites,totaldist,colour=sex))+geom\_point()+geom\_smooth(method='lm',se=F)

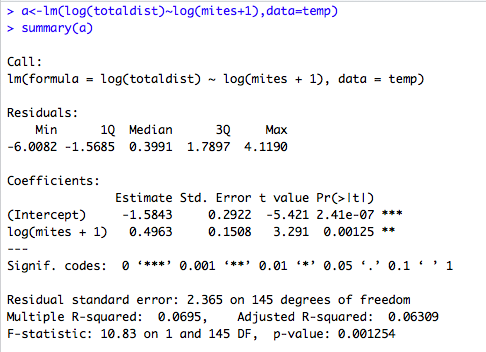


It seems that the more mites beetles had beetles flew more. Let’s now check fitting two linear models to confirm whether mites had an effect or not; one simple linear model with mites as a predictor and total distance flown as the response variable; and a second model that controls for natal environment. There are multiple reasons about why to include the natal environment. One reasonable possibility is that the natal environment beetles grew up had an effect on their quality and may explain some of the variation in the dispersal behavior. We need to fit a linear mixed model where we add log\_num as a random variable. Then we compare both models by dropping the random variable from the second model, the linear mixed model, using “drop1” and specifying the command “test=Chisq”. We are basically using an abbreviated way to compare by means of a Chi square test a model that contains the random variable and one that doesn’t have the random variable.

Let’s have a look at the first model

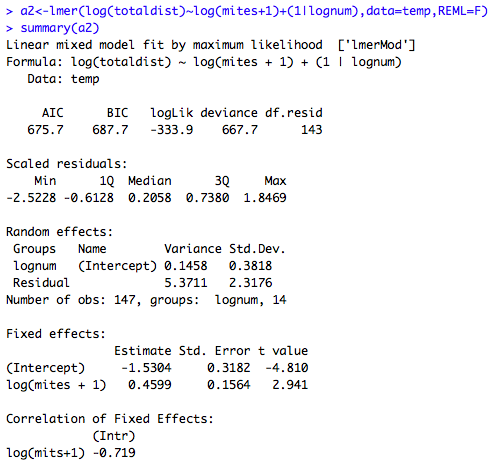
a<-lm(log(totaldist)~log(mites+1),data=temp)

summary(a)



There are four pieces of information in this summary. The first portion reminds the model. The second displays a summary of the residuals. The third and fourth pieces contain the most relevant information to interpret the model. Let’s start with the coefficients. The p-value for the fixed effect “log(mites+1)” seems highly significant. This basically means that mites matter. The more mites the more beetles flew this may seem somewhat odd but there is a biological reason to expect such outcome. That won’t be discussed here. However, this p-value is the same as the one for the overall model that is shown at the very last row of the model summary. This is due to only having one single predictor in the model. Now let’s check the R2 of the model, the fourth piece on the summary. The R2 is a measure of how much variation is explained by the present model. Larger R2 values are desirable however this depends on the type of data and the field. The R2 in this model is very low which basically means that there may be some other variables that could explain the variation in the dispersal behavior.

As mentioned previously, the second model would consider the natal environment as a random variable. To do this, we need to call for a different type of model in this case a linear mixed model. A linear mixed effect model combines both fixed and random effects as predictor variables. The structure for fitting a mix model follows the expression of response ~ fixed predictor + random effect. There are different specific notations or mixed-effect model formulas that won’t be discussed here. For our specific example, we will use the formula (1|g) or random intercept with fixed mean where *g* represents a grouping factor. This means that each grouping factor level has its own random intercept. In this case our grouping factor is log\_num.



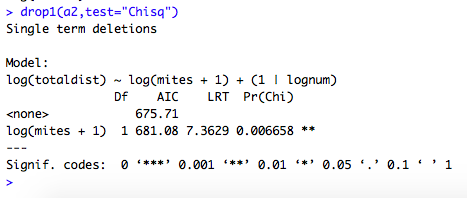
a2<-lmer(log(totaldist)~log(mites+1)+(1|lognum),data=temp,REML=F)

summary(a2)

Mites still matter even when controlling for the natal environment. Note that this second model is quite different to the previous one. There isn’t a straightforward way to compare the two models because the second model didn’t produce a p-value. However, we can compare them by performing a Chi square test.

Let’s have a look at the outcome when using “drop1”.

drop1(a2,test="Chisq")



Now we are able to contrast between the two models by inspecting their AIC values. The model with the lowest AIC value should be considered as a better model. In this case the first model is better than the model with the random variable. This means that natal environment isn’t influential for beetle dispersal.

Other variables that most likely will explain beetle dispersal are beetle weight, beetle length and so on. It is, however, also possible that mite load predicts indirectly other more important explanatory variables for beetle dispersal. This remains to be explored. The appendix of this tutorial contains the complete code and additional data visualization and analysis for the trimmed dataset.

The instructional purpose of this tutorial was the initialization of a-first-time-R user into data analysis using different R packages. The dominion of these packages wasn’t completely explore, however, the current tutorial exemplifies how they can be used and their potential.