

Biometry. Lecture 9

Alexey Shipunov

Minot State University

February 26, 2014



1 Questions and answers

2 Data

- Data conversion and normalization

3 Inside R: Matrices, lists and data frames

- Matrices
- Lists
- Data frames (tables)



- 1 Questions and answers
- 2 Data
 - Data conversion and normalization
- 3 Inside R: Matrices, lists and data frames
 - Matrices
 - Lists
 - Data frames (tables)



- 1 Questions and answers
- 2 Data
 - Data conversion and normalization
- 3 Inside R: Matrices, lists and data frames
 - Matrices
 - Lists
 - Data frames (tables)



Starting...

```
> setwd("<working folder>")  
or  
"Change dir"  
in menu!
```



Previous final question: the answer

This is how we changed NAs to averages:

```
> h.old <- h
> h[is.na(h)] <- mean(h, na.rm=TRUE)
> h
```

Imagine you have data like `c(1, 0, 0, 2, 3, 0)` but zeros are actually represent NAs. How to change zeros to “real” NAs with similar (or any) command?

```
> a <- c(1, 0, 0, 2, 3, 0)
> a.old <- a
> a[a==0] <- NA
> a
```



Enhancing islets

```

> i <- read.table("http://ashipunov.info/data/islets.txt", h=T)
> sq <- (i[,1]*.6) * (i[,2]*.6)
> pdf("pics/islets.pdf")
> plot(sq, i[,3], main="Small Arctic islands:
+ diversity vs. square",
+ xlab=expression("Square, m"^2),
+ ylab="Number of plant species")
> lines(loess.smooth(sq, i[,3]), lty=1)
> dev.off()

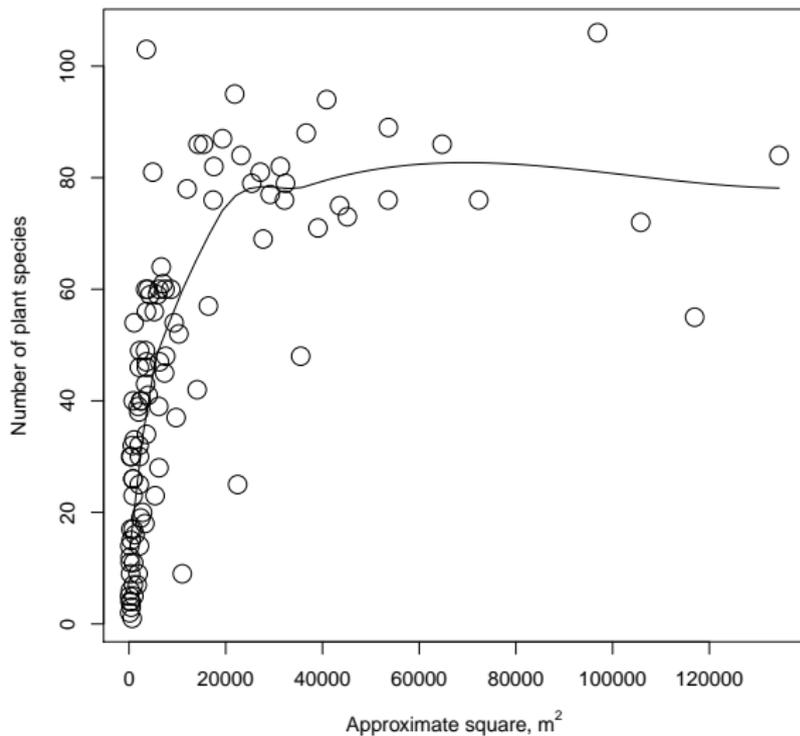
```

I added the estimated curve (command `loess.smooth()` — from LOESS, *locally weighted scatterplot smoothing*) plus better axes labels (note the `expression()` function for superscript) and title.



Enhanced islets

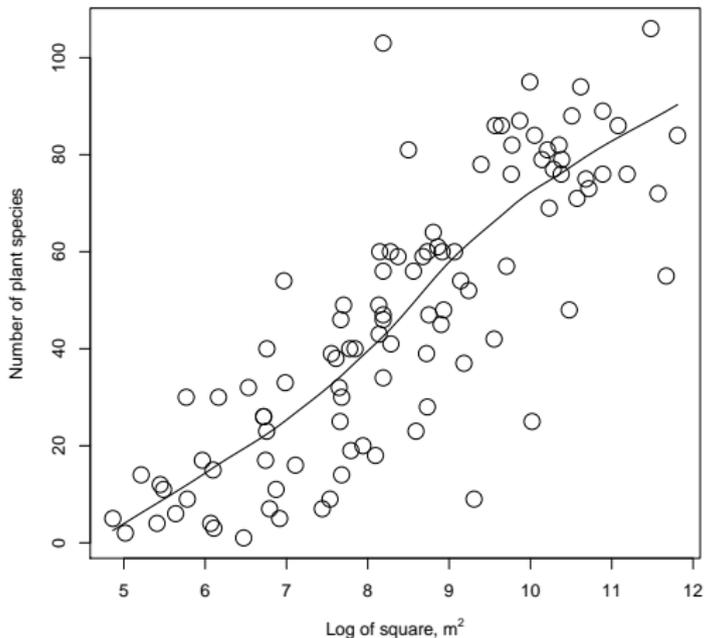
Small Arctic islands: diversity vs. square



What if we use logarithm of square?

With `plot(log(sq), ...)` curve becomes almost linear!

Small Arctic islands: diversity vs. log square



Data

Data conversion and normalization



Logarithmic conversion

- Very often the bell-shape curve or linear relation may be achieved by logarithmic conversion
- In R, it is possible to apply `log()` (natural logarithm) function to any vector. The only problem is that the data should not contain zeroes.
- Other mathematical conversions also exist, e.g., square root conversion.



Scaling data

Function `scale()` will convert all columns of data frame to the same scale:

```
> a <- 1:10
> b <- seq(100, 1000, 100)
> d <- data.frame(a, b)
> d
> scale(d)
```

Columns `a` and `b` became identical!



Inside R: Matrices, lists and data frames

Matrices



What we already know of R internals

- We already know integer, numeric, character and logical *vectors*. All vectors could be *named*.
- In addition, R has *factors* (ordered or unordered)



Matrices are vectors

- In R, numeric tables (*matrices*) are simply vectors with two dimensions.
- It is also possible to create multidimensional *arrays*.



Matrix and vector

```
> m <- 1:4
> ma <- matrix(m, ncol=2, byrow=TRUE)
> str(ma)
> str(m)
> mb <- m
> dim(mb) <- c(2,2)
> mb
```

The structure of objects `m` and `ma` are not significantly different, only screen output is not similar (try it!). Function `dim()` will add dimensions to vector transforming it into matrix or array.



Three-dimensional matrix (array)

```
> m3 <- 1:8  
> dim(m3) <- c(2,2,2)  
> m3
```



Inside R: Matrices, lists and data frames

Lists



List is a collection of everything

- List may contain any type of objects
- Moreover, list can contain other lists, and so on



List examples

```
> l <- list("R", 1:3, TRUE, NA, list("r", 4))
> l
> str(l)
> fred <- (name="Fred", wife="Mary", no.children=3,
+ child.ages=c(5,9))
> fred
```



Indexing of vectors, matrices and lists

```
> m[3] # third element of vector m
> ma[2, 1] # second row, first column
> l[1] # lists may be indexed like vectors
> str(l[1]) # it's a list!
> l[[1]] # not the same as l[1]!
> str(l[[1]]) # it's a vector!
```



Names

In R, elements of vectors and lists, columns and rows of matrices may have *names*:

```
> names(fred)
> fred$wife # this is a selection by name
> w <- 60:66
> names(w) <- c("Rick", "Amanda", "Peter", "Alex",
+ "Kathy", "Ben", "George")
> w
> w["Rick"]
> rownames(ma) <- c("a1", "a2")
> colnames(ma) <- c("b1", "b2")
> ma
```



Inside R: Matrices, lists and data frames

Data frames (tables)



More important than any other object

- This is a most important type of object; most of data are represented by data frames
- *Date frame is a list of vectors of same length*



How to create a data frame

```
> x <- 171:177
> sex.f <- c("m", "m", "f", "f", "f", "m", "f")
> m.o <- c("L", "XL", "S", "M", "S", "M", "XL")
> d <- data.frame(weight=w, height=x, size=m.o, sex=sex.f)
> d
> str(d)
```



Selection from data frames

```
> d$weight # by name
> d[[1]] # by number, as list
> d[,1] # by number of column, as matrix
> d["weight"] # by name of column
> d[,2:4] # columns 2, 3, 4
> d[,-1] # all columns except first
> d[-1,] # all rows except first
```



Finishing...

```
>savehistory("20140226.r")
```



Final question (2 points)



Final question (2 points)

How to select from data frame `eq` column which name is `NUM.Z`?



Summary: most important commands

- `[`—selects an element, row or column
- `$`—selects by name from list or data frame



For Further Reading



A. Shipunov.

Biometry [Electronic resource].

2012—onwards.

Mode of access:

http://ashipunov.info/shipunov/school/biol_240



A. Shipunov, and others.

Visual statistics. Use R!

DMK Press, 2012. [Translating from Russian.]

