Biometry. Lecture 11

Alexey Shipunov

Minot State University

March 2, 2016

Outline

1. Inside R: Matrices, lists and data frames
   - Data frames

2. One-dimensional data
   - Central tendency
   - Range

Outline

1. Inside R: Matrices, lists and data frames
    - Data frames
2. One-dimensional data
    - Central tendency
    - Range

Starting...

```
> setwd("<working folder>")
```

or

“`Change dir`”

in menu!

On Mac, be sure that startup option is working: `getwd()` (`getwd()` checks if R is in working folder, `dir()` checks the folder content)

Inside R: Matrices, lists and data frames

Data frames

Selection by condition

```
> # recreate "d" data frame from the previous lecture script!
> d[d$sex=="female",] # will select only women
> d[d$sex!="female",] # will select all other genders ;)
```

`==` is “equal?”, `&` “and”, `|` “or” and `!` is “not”

Sorting and ordering

```
> sort(x) # ascending
> rev(sort(x)) # descending
> d[order(d$sex, d$height), ] # sort by sex then by height
```

One-dimensional data

Central tendency

Mean and median

- These are two most frequently used characteristics of the central tendency.
- Median is more robust than mean.

Mean and median

```
> salary <- c(21, 19, 27, 11, 102, 25, 21)
> mean(salary); median(salary)
> median(1:3); median(1:4)
```

When number of elements is odd, median is a central value; if even—median is the average between two centrals.

Median is the third quartile

Quartiles take out 0% (minimum, `min()`), 25% (lower hinge) , 50%, 75% (upper hinge) and 100% (maximum, `max()`) of ordered data. Median is simply a 50% (third) quartile.

```
> fivenum(salary)
> summary(salary)
```

Technical: how to calculate medians for all columns

```
> sapply(trees, median)
```

Commands of `*apply()` family (`sapply()`, `apply()`, `lapply()`, `mapply()`, `tapply()`) are most powerful in R

Mode

Mode is the most frequent value:

```
> t.sex <- table(sex)
> mode <- t.sex[which.max(t.sex)]
> mode
> names(mode)
```

One-dimensional data

Range

Standard deviation, variance and IQR

- Variance is a sum of square differences between each value and mean divided by number of degrees of freedom (so-called “Bessel’s correction”)
- Standard deviation is a square root from variance
- IQR (inter-quartile range) is simply a difference between fourth and second quartiles. It is more robust than standard deviation.

Standard deviation, variation and IQR

```
> sd(salary); var(salary); IQR(salary)
```

Coefficient of variation

Coefficient of variation (CV) is a standardized (by mean) standard deviation

```
> cv.trees <- 100*sapply(trees, sd)/colMeans(trees)
> cv.trees
```

Which variable does variate most?

What is a function

```
> Sum <- function(a, b)
+ {
+ a + b
+ }
> Sum(2, 3)
```

Function is a piece of code which may run independently. All R commands are functions. Please note that every functions requires two parts: **arguments** in *round brackets* and **body** in *curly brackets*. It is too boring to enter functions line-by-line. Instead, it is better to copy function from external editor. If function contains mistake(s), one may use `fix()` command.

Let us create a simple useful function

R has no command for coefficient of variation, we will create it ourselves:

```
> CV <- function(x)
+ {
+ (sd(x) / mean(x)) * 100
+ }
> CV(trees[,3])
> CV(trees$Volume)
> sapply(trees, CV)
```

We can then run `fix(CV)` and add `round(..., 2)` function to make numbers more readable.

Boxplots

Boxplots (invented by John Tukey) are one of the best representations of data central tendency and range.

```
> boxplot(salary)
> boxplot(trees)
> eq <- read.table("http://ashipunov.info/data/eq.txt",
+ h=TRUE)
> boxplot(DL.R ~ SPECIES, data=eq)
```

Boxplots do not show mean and standard deviation.

Stacked boxplots

```
> seq <- scale(eq[,2:9]) # scale all except SPECIES
> boxplot(seq[eq$SPECIES=="arvense",],
+ at=1:8-0.2, boxwex=0.3, # shift and narrow
+ xaxt="n", yaxt="n", # no labels
+ main="Horsetails, scaled and paired characters")
> boxplot(seq[eq$SPECIES=="fluviatile",],
+ at=1:8+0.2, boxwex=0.3,
+ xaxt="n", yaxt="n",
+ add=T, col="green") # overlay and colorize
> axis(1, 1:8, names(eq)[2:9], cex.axis=.75) # labels
> legend("topleft", c("arvense","fluviatile"),
+ fill=c("white","green"))
```

This stuff is better to remember in separate *.r script and run it with `source()` command when you need to update your plot.

Histograms

Histograms show the frequency of every data interval:

```
> hist(salary)
> hist(trees[,1])
```

Density plots

Density plot smooths the histogram:

```
> plot(density(trees[,3]))
> plot(density(rnorm(1000))) # 100000 is even better!
```

Density plots looks prettier but may lead to wrong conclusions especially if sample is small.

`summary()`

`summary()` is a “smart” (generic) function which gives the most appropriate description of data. In many cases, it will give quantiles + mean:

```
> summary(salary)
> summary(trees)
> summary(sex)
```

`summary()` and “bad” data

summary() is very useful when one needs to check a reliability of data:

```
> err <- read.table("http://ashipunov.info/data/errors.txt",
+ h=TRUE, sep="\t")
> str(err)
> summary(err)
```

`AGE` became a factor (erroneous “a”), empty name (instead of `NA`), and impossible minimal height.

Finishing...

Save your commands!

(`savehistory(<todaysdate>.r)` or `File -> Save as...` on Mac)

Summary: most important commands

- `median()`—returns a median value
- `IQR()`—returns robust range
- `boxplot()`— draws a Tukey's boxplot

For Further Reading

A. Shipunov.
Biometry [Electronic resource].
2012—onwards.
Mode of access:
http://ashipunov.info/shipunov/school/biol_240

A. Shipunov, and many others.
Visual statistics. Use R!
2016—onwards.
Mode of access: http://ashipunov.info/shipunov/school/biol_240/en/visual_statistics.pdf