Getting Started with R

Xiaojing Dong

September 26, 2017

Download R and R Studio

Visit http://cran.r-project.org/ or google "R language."

CRAN (Comprehensive R Archive Network) is a network of mirror sites that allow you to download precompiled binary versions of R or source. Choose the R version that is suitable to your operating system (Windows, Mac or Linux).

Although optional, you are highly recommended to install **R Studio** as well. After installing R, go to https://www.rstudio.com/products/rstudio/download, choose the RStudio Desktop (free) version to download and install. It provides a nicer user interface together with some useful tools for programming in R.

You are recommended to install R on your computer first, then R-studio. During the installation process, choose the default options, unless you have a good understanding of the other options.

Set Working Directory

It is convenient to set your working directory first. This way, when you load or save a file, R will automatically access the working directory. To do that, use function **setwd()**. Note that the "\" used in windows directory specifications does not work here, it needs to be the other direction "/".

```
setwd("c/users/mydirectory/")
```

You can also check which working directory you are at right now using

```
getwd()
```

Here is the official R manual page about this http://stat.ethz.ch/R-manual/R-patched/library/base/html/getwd.html

Get Packages

One of the reasons that R has become so popular in recent years is the wide selections of R packages created by R users. A package generally contains functions, data, and complied codes, in addition to a user manual.

You can find a full list of the R-packages from https://cran.r-project.org/. It is not recommended that you go through the complete list of packages, unless you really have some time that you want to kill.

To get a package you will need to first learn about its name, then use the function **install.packages("name")** to install the package and the relevant files on your computer. Over the quarter, we will need to install a few packages in addition to the default ones that are installed together with R.

Get Help

There are generally two ways to help on using R. If you know the name of the function that you are using, but would like to learn about its details or parameter definitions, use **help()**, or *?*.

```
?plot
help("plot")
```

Either way, you will get the official R manual for the function plot().

If you do not know the name of the function to use, the best way to find help is through online search. R has become so popular that there are online communities providing really helpful services to R users. There are a few websites that you might find useful to get answers to many of your R programming related questions.

https://stackoverflow.com/ is a community for coders. Not just R, for any programming language, if you have questions, you will find them answered by the community soon.
http://www.statmethods.net/index.html is a nicely organized website helping beginners to get started with R.
https://blog.rstudio.com/ is the Rstudio official blog site. https://www.r-bloggers.com/ is the R blog sit.

This is far from a representative or complete list of websites. One of the best part of using R is that you will never feel lonely. Just type your question online, you will find an answer somewhere on the web pretty soon.

Finally, if you want to read a book about R, here is the link to the pdf version of The Art of R Programming

Review of Basic Statistics

Let's review some of the basic statistics we learned before, using R. First we create a variable using a vector.

```
a = c(1.3, 3, 2, 2.5, 3.8, 2.3, 4, 3.1)
```

Some basic summary statistics:
* mean(a)=2.75

* var(a) =0.8257143
* median(a)=2.75

Q: How is variance calculated?

A: You need to calculate the mean first. Variance measures the average spread or average distance from the mean.

$var(a) = E[(a - \mu_a)^2]$

When calculating the expecation, we use the average calcualtion, depending on whether it is the population variance or the sample variance, the denominator will be different.

Sample variance $var(a) = \frac{\sum_{i=1}^{N}(a_i - \mu_a)^2}{N-1}$

Population variance $var(a) = \frac{\sum_{i=1}^{N}(a_i - \mu_a)^2}{N}$

N represents the sample size, meaning the number of elements in the variable a. If $N \to \infty, N = N - 1$, then sample variance is the same as population variance. In other words, if the size of the data is really small, the difference between the population and sample variances is really big.

Q: How is median calculated?

A: For odd numbers, sort all the values, the one in the middle is the median value. For even numbers, sort all the values, pick the two in the middle, then calculate the average between these two numbers.

In R, you can sort the values using

```
sort(a)
```

```
## [1] 1.3 2.0 2.3 2.5 3.0 3.1 3.8 4.0
```

Q: What's the difference between mean and median? Which one is more sensitive to outliers?

Covariance and Correlation

Define another variable b

```
b=c(3.2, 4, 3, 3.7, 5, 2.6, 3, 1)
```

Calculate the covariance between a and b
$cov(a, b) = E[(a - \mu_a) \times (b - \mu_b)] = E(ab) - \mu_a\mu_b$

It is calculated as the mean of the product subtracting the product of the means.

Note: usually, when we calculate covariance, it refers to "population" covariance, as calculated above. The difference between sample covariance and population covariance is a scale.

Q: what is the scale difference between population covariance vs. sample covariance?

Correlation can be considered as standized covariance, and is calculated as

$$cor(a,b) = \frac{cov(a,b)}{\sqrt{var(a)var(b))}}$$

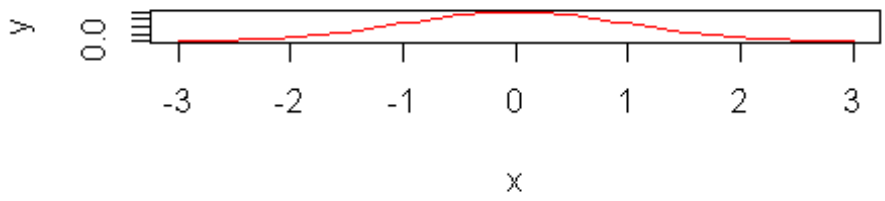
You will find that we will talk about correlations quite often. When two variables have positive correlations, it indicates that there are some co-movement between these two variables. That is when one variable gets bigger, the other one does too. When two variables have negative correlations, it indicates that the two variables have some co-movement, but to the opposite directions. That is when one variable gets bigger, the other one gets smaller.

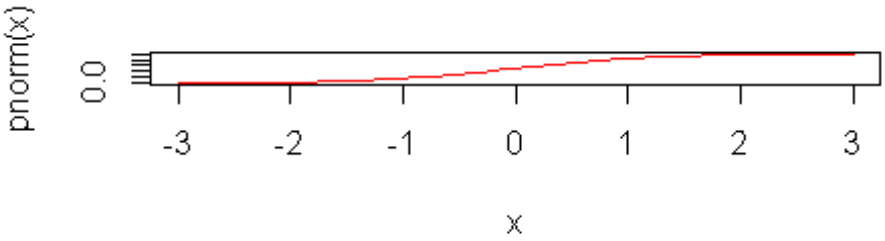
Correlation is a very important statistical concept in data analysis. It helps to find out the relationships among variables. However, a common mistake in data analytics is to interpret the correlation results as if causality exits. Such mistake is especially common when interpreting regression results, as its model setup requires choosing a Y variable and a few X variables. The model estimation from a regression model only represents the correlation between the Y variable and an X variable, but are often interpreted as X causes Y. We will discuss that more in Linear Regression lecture. If you are interested, you are welcome to read an easy-reading paper (pdf) I published together with a colleague.

Normal Distribution

One of the most commonly utilized continous probability distribution is Normal distribution.

```
x = seq(-3,3,0.1)
par(mfrow=c(2,1))
y=dnorm(x)
plot(x,y,type="l",col="red")
plot(x,pnorm(x),type="l",col="red")
```





The top one plots the probability distribution function (PDF) of a standard Normal distribution $x\ N(0,1)$, and the bottom one plots the cumulative distribution function (CDF) of the same normal distribution.

Normal distribution has only two parameters:
* μ is the mean in the normal distribution
* σ^2 refers to the variance in the normal distribution.

The PDF for a normal distribution with parameters (μ, σ^2) can be specified as

$$f(x) = \frac{1}{2\pi\sigma} exp[-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}]$$

The CDF function calculate the area below the curve, up to the value X

$$F(X) = \int_{x=-\infty}^{x=X} f\ (x)dx = \int_{x=-\infty}^{x=X} \frac{1}{2\pi\sigma} exp[-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}]dx$$

Normal distribution has a few features:

1. It is symmetric
2. It has only one mode
3. It has relatively long tail
4. Its CDF function does not have closed form solution