1. 改变R的语言：右击快捷方式图标，选择属性，在target field后面加上LANGUAGE=en
2. ls(pat="^m")名称中带m 且以m开头的
3. R中用分号来隔开同一行中的不同命令语句
4. ls.str()展示内存中所有对象的详细信息 pat在些也适用
5. max.level规定显示对象的详细级别
6. rm()删除某个对象 rm(list=ls())删除所有对象
7. ?lm=help(lm)=help("lm") lm是线性函数模型;但是要查询的是某特殊语法意义字符时就必须用最后一种，如\*，<-等
8. help(bs)出错，因为所属的包没有打开 help("bs", try.all.packages = TRUE)搜索出bs所在的包，try.all.packages在默认状态下是等于FALSE的，因此要TRUE（必须大写）; help("bs", package=splines)直接指定package=splines,不用打开splines包也能显示bs的帮助文件; help.start(), help.search()可以搜索关键词
9. rebuild用来刷新数据库
10. 对象的类型和长度可以分别通过函数mode和length得到
11. Inf表示正无穷，-Inf表示负无穷; x=5/0=Inf
12. x=exp(1)=2.718, x=2^4=16, x=1e2=100(在这里e2表示10的平方，注意与exp区分。前面的1不能省去！）NaN表示非数字
13. x表示字符型变量时必须加上双引号或单引号，但是在双引号或单引号里面再引用双引号或单引号时，须在前面加上\, 如x <- 'Double quotes " delimitate R\'s strings.'（因为它不能判定哪个单引号或双引号是一对的，因此要在语句引用的单引号前加上\,否则会出错）
14. 向量是一个变量，其意思也即人们通常认为的那样；因子是一个分类变量；数组是一个k维的数据表；矩阵是数组的一个特例，其维数k = 2。注意，数组或者矩阵中的所有元素都必须是同一种类型的；数据框是由一个或几个向量和（或）因子构成，它们必须是等长的，但可以是不同的数据类型；“ts”表示时间序列数据，它包含一些额外的属性，例如频率和时间；列表可以包含任何类型的对象，包括列表！对于一个向量，用它的类型和长度足够描述数据；而对其它的对象则另需一些额外信息，这些信息由外在的属性给出。这些属性中的是表示对象维数的dim，比如一个2行2列的的矩阵，它的dim是一对数值[2,2]，但是其长度是4。
15. getwd()获得工作目录，setwd("c:/data")或者setwd("/home/paradis/R") 来改变目录。
16. read.table读取文件 scan也可以读取，且有更多功能;write.table可以在文件中写入一个对象
17. 算子“：”的优先级是最高的

x=1:10\*3

3 6 9 12 15 18 21 24 27 30

1. seq(1, 5, 0.5)与seq(length=9, from=1, to=5)等价，输出的结果是

[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0

1. rep用来创建一个所有元素都相同的向量，rep(1,5)=1 1 1 1 1
2. sequence创建一系列连续的整数序列(注意seq与sequence是两个不同的函数)

x=sequence(5:2);y=sequence(c(5,3,2));b=c(5,4,1);z=sequence(b)

x;y;z

[1] 1 2 3 4 5 1 2 3 4 1 2 3 1 2

[1] 1 2 3 4 5 1 2 3 1 2

[1] 1 2 3 4 5 1 2 3 4 1

sequence只可以跟表达式连接

1. gl用于产生规则的因子序列。

gl(3, 5, length=30)

[1] 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3

Levels: 1 2 3

gl(2, 6, label=c("Male", "Female"))

[1] Male Male Male Male Male Male

[7] Female Female Female Female Female Female

Levels: Male Female

1. expand.grid( )创建一个数据框，结果是把各参数的各水平完全搭配：

expand.grid(h=c(60,80), w=c(100, 300), sex=c("Male", "Female"))

h w sex

1 60 100 Male

2 80 100 Male

3 60 300 Male

4 80 300 Male

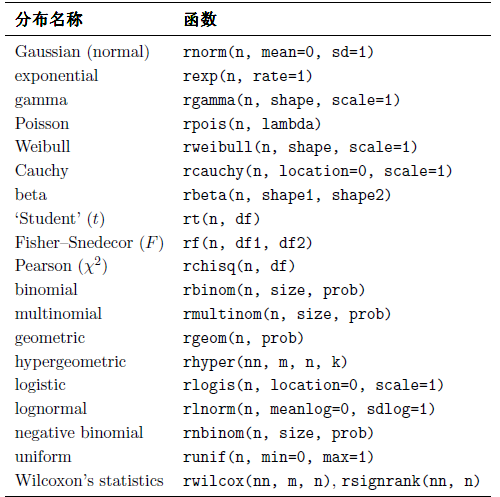
5 60 100 Female

6 80 100 Female

7 60 300 Female

8 80 300 Female

1. statistical distribution and the R functions:



1. 也可以用这种方法来创建一个矩阵 x=1:15; dim(x) <-c(5, 3)
2. yy%\*%xx是矩阵相; yy\*yy矩阵内相应元素相乘
3. cbind组合多个向量，矩阵或数据框
4. solve(a,b)解矩阵方程，a%\*%x=b, 如果b缺失，则默认为单位矩阵
5. binned variogram and stores the cloud

vario.bc <- variog(s100, max.dist=1, bin.cloud=TRUE)

1. Variofit: Estimate covariance parameters by fitting a parametric model to a empirical variogram
2. Greece letters:



1. Box-Cox transformation: 将非正态分布的数据转化为正态分布; lamda=1时不转换，lamda=0时转换为对数，  box-cox的用法，举几个例子

     λ=-2，  就用 1/(Y平方)进行转换，

     λ=0，直接LOG(Y)转化，

     λ=2, 就用Y平方转化，

     λ=0.04 就用Y的0.04方进行转化，

1. demo(graphics) 展示graphics package 的功能！
2. vv6 <- rep(1:3,each=2) # repeats each element of the sequence "each" times

vv7 <- rep(1:3,times=2) # repeats the sequence "times" times ,也可以把times=2简写成2

1. 用dimnames = list给行列命名：

mm7 <- matrix(letters[1:4],2,2,dimnames = list(c("row1", "row2"),c("C.1", "C.2")))

1. 用对角线矩阵来创建单位矩阵：I=diag(n)
2. 求各行的和：rowSums(data)

1. read.gal相当于read.gwt2nb(一个是产生queen权重, 一个是产生distance或k nearest neighbors的权重), read.gal跟poly2nb的作用几乎一样，read.gal的ID从文件中读取ID，poly2nb 的ID通常从0开始；read.gal从gal文件读取数据，poly2nb从shp 文件中读取数据plot(queen.3x6, xy.q)是3列6行；跟一般的矩阵表示的方法不一样；刚好反过来
2. k nearest neighbor要通过两步才可以plot, 一是利用knearneigh提取k个最近的neighbor的信息，二是利用knn2nb将其转化为nb类型的数据；dnearneigh则不用转化，可以直接plot
3. 搜索column的值，如果是某一值，就在新的column赋值

m <- c(grep(", KS" , aa$AreaName), grep(", OK", aa$AreaName))

aa$Region[m] <- "SPP"

1. file.lists() list all the files in the working directory
2. a[[i]]是list形式的数据，a[[i]][m,n]表示a[[i]]的m行n列；a[[i]] <- a[[i]][ ,-c(8:17,19:24)]就是表示去掉a[[i]]的8到17以及19到24列
3. avoid using # in the data, will cause a lot of problems when reading or writing. 如果文件中包含有#的数据，用comment.char=""来读入，这时#不会被看成是commend，而是一个普通符号；但是读入文件后，对文件进行操作，#号仍然会被看成是command而不进行操作；如nrow(aa),就不会计算带有#号的行
4. Data中不能包含单引号（‘），否则会引起此类错误Error in scan(file, what, nmax, sep, dec, quote, skip, nlines, na.strings,: number of items read is not a multiple of the number of columns.  英文很容易出现这种情况，特别指示人名相关名品时；如 Mary’s company
5. 作图，加等值线，x 及 y 值的范围

x <- 10\*(1:nrow(volcano))

y <- 10\*(1:ncol(volcano))

1. Error in centroids$x : $ operator is invalid for atomic vectors，可以将centroids变为data frame，as.data.frame(centroids)，可以解决很多类似问题
2. How to turn the NA to 0:

x <- data.frame(a = c(0,1,2,NA), b = c(0,NA,1,2), c = c(NA, 0, 1, 2))

x$a[is.na(x$a)] <- 0

x$c[is.na(x$c)] <- 0

1. >>m <- matrix(paste(letters[1:4], "does not work."),  
   >>m  
   [1,] "a does not work." "c does not work."  
   [2,] "b does not work." "d does not work."  
   >>gsub("does not work.", "", m) ###use gsub to replace some characters  
   [1] "a " "b " "c " "d "*##replace “does not work” with “”*
2. mis[[m]] <- dif[[m]][dif[[m]][ , 2]!= 0,1] ##send the value in the first column of dif[[m]] to mis[[m]], if the values in the second column of dif[[m]] is not equal to 0
3. Subsetting rows selecting on more than one value. We use the %in% notation when we want to subset on multiple values of y. The x.sub5 data frame contains only the observations for which the values of variable y are equal to either 1 or 4.

x.sub5 <- x.df[x.df$y %in% c(1, 4), ]

To choose the rows not in the range:

EIA2002 <- eia2002.1[!eia2002.1$ORISPL%in%c(99999, camd0$ORISPL), ]

1. la.test <- la[which((la$ORISPL\_CODE == 315)|(la$ORISPL\_CODE == 356)|(la$ORISPL\_CODE == 56472)), ]

Can’t use la.test <- la[which(la$ORISPL\_CODE == 315|356|56472), ]

1. How to split data: <http://tolstoy.newcastle.edu.au/R/e2/help/07/01/8810.html>
2. to change the sequence of columns, we can use the following method

RA\_EIA\_CAMD2 <- EIA\_CAMD2[,c(1,8,2,10,3,4,11,6,7,9,5)]

1. How to do multiple variables’ linear regression in R :  
   <http://www.statmethods.net/stats/regression.html>
2. how to draw several pictures in the same box: par(mfrow=c(2,2))
3. how to get the split levels’ names:

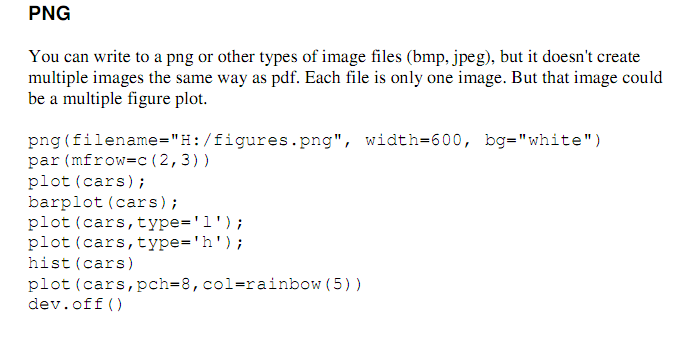
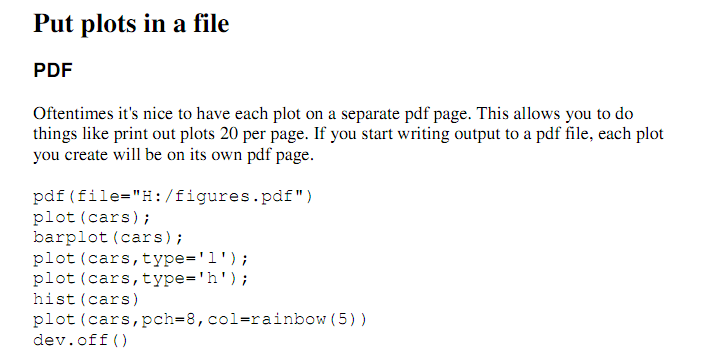
a <- summary(split)

rownames(a)

1. file.choose() opens up an explorer window and allows us to pick the file from our computer.
2. y ~ x1 \* x2 y is modelled by x1, x2 and also by the interaction between them
3. Using R for statistical analysis:

<http://www.gardenersown.co.uk/Education/Lectures/R/data2.htm#nav>

1. If you are going to Merge a lot of files, the columns of the files should be different, or else, there will be errors like” the name doesn’t match with previous names”. You can change the columns name according to year. Like V1\_2001, Vn\_2010…. So that the corresponding columns have different names.
2. Put plots in a file



1. Graphical Parametes:

<http://www.statmethods.net/advgraphs/parameters.html>

Box Plot: <http://www.statmethods.net/graphs/boxplot.html>

advanced graph types: <http://www.statmethods.net/advgraphs/index.html>

1. fix( ) can open the variables in a table, just like the matlab. Then you can fix the values
2. names(a) to get the names of the lists
3. Sys.time and Sys.Date returns the system's idea of the current date with and without time. <http://127.0.0.1:23087/library/base/html/Sys.time.html>
4. get(a): get the contents of variables named “a”
5. Adding time to a plot and adventure smoothing:

<http://personality-project.org/r/r.plottingdates.html>

1. yearmon(a) turn the (2010-09) format data into the Sep 2010 data, that system can identify
2. A lot of R plotting methods and examples:

<http://www.oga-lab.net/RGM2/images.php?query=%2A%3A%2A&ctv=Environmetrics&start=0&facet=true&facet_missing=true&facet_query=ctv%3A%5CN&pageID=5>

1. if pdf() is put after par(), every plotting picture will take up one page in the pdf file; if it is put in front of the par(), all plotting pictures will share the same page in pdf file.
2. quantile() see the quantile of the data
3. Data Mining Algorithms In R/Clustering/Fuzzy Clustering - Fuzzy C-means:

<http://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Clustering/Fuzzy_Clustering_-_Fuzzy_C-means>

1. fanny( ): Computes a fuzzy clustering of the data into k clusters.
2. When using merge( ) function, the by = by must be quoted with “ ”

merge(la, EIA, by.x = "ORISPL\_CODE", by.y = "ORISPL")

## A Short Introduction to the [*XML*](file:///S:\package(%22XML%22)) package for R

<http://www.omegahat.org/RSXML/shortIntro.html>

1. strsplit can split the characters into several parts. Just check the ?strsplit function.
2. color chart: <http://research.stowers-institute.org/efg/R/Color/Chart/>
3. Insert the row before the 5th row:

rbind(mat[1:4, ], c(6, 11), mat[5, ])

[,1] [,2]

[1,] 1 6

[2,] 2 7

[3,] 3 8

[4,] 4 9

[5,] 6 11

[6,] 5 10

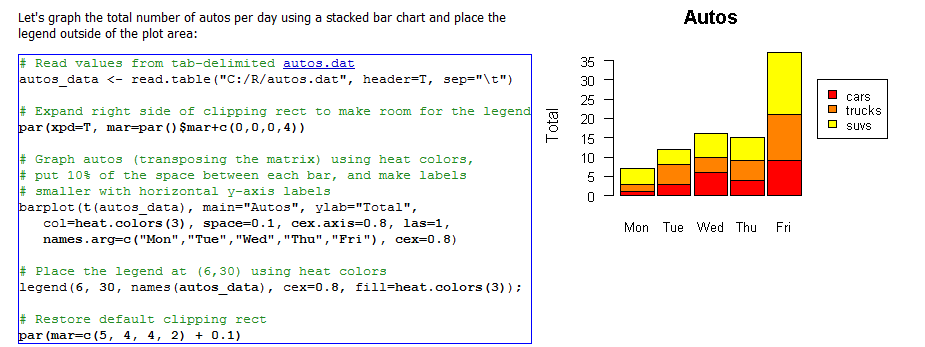
1. str() can get some conclusion of the data
2. very useful discussion about matrix, array, list, data.frame

<http://r.789695.n4.nabble.com/Data-frame-Vs-Matrix-Vs-Array-Definitions-Please-td3014717.html>

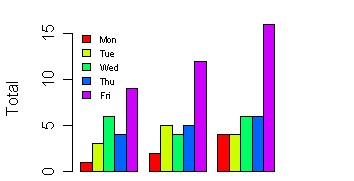
1. gc() : A call of gc causes a garbage collection to take place
2. counts <- table(mtcars$gear), table function is used to count the number of the different factors and list them as table
3. How to make the stacked barplots and make the legend outside of plots:

par(xpd=T, mar=par()$mar+c(0,0,0,4))

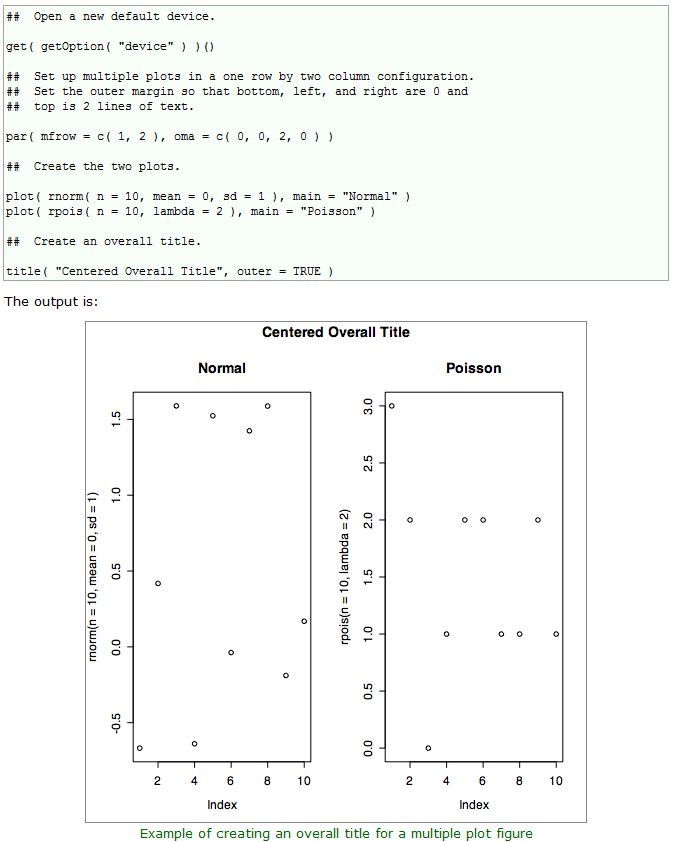
1. barplot(t(AggrPt0906[,2:5]), main = "Plant", col = 2: 5, beside = T,legend.text = T ) can get the plot and legend in the same time, but the position and style of the legend is not as flexible as using the legend function saperately.



Add the argument beside = T to get the juxtaposed bars



1. Creating an Overall title for a multiple plot figure:



1. Legend: xjust how the legend is to be justified relative to the legend x location. A value of 0 means left justified, 0.5 means centered and 1 means right justified. yjust is the same as xjust for the legend y location.
2. Introduction to String Matching and Modiﬁcation in R Using Regular Expressions: <http://biostat.mc.vanderbilt.edu/wiki/pub/Main/SvetlanaEdenRFiles/regExprTalk.pdf>

# [R Grouping functions: sapply vs. lapply vs. apply. vs. tapply vs. by vs. aggregate vs.](http://stackoverflow.com/questions/3505701/r-grouping-functions-sapply-vs-lapply-vs-apply-vs-tapply-vs-by-vs-aggrega)

<http://stackoverflow.com/questions/3505701/r-grouping-functions-sapply-vs-lapply-vs-apply-vs-tapply-vs-by-vs-aggrega>

1. you can expand the data.frame, but not the matrix:

CO2.geo <- data.frame(0,0)

CO2.geo[2, ] <- c(1,1)

CO2.geo[3, ] <- c(1,1)

# Great-circle distance calculations in R (calculate the point distance with coordinates)

<http://pineda-krch.com/2010/11/23/great-circle-distance-calculations-in-r/>

1. head(letters)

head(letters, n = -6L)

head(freeny.x, n = 10L)

head(freeny.y)

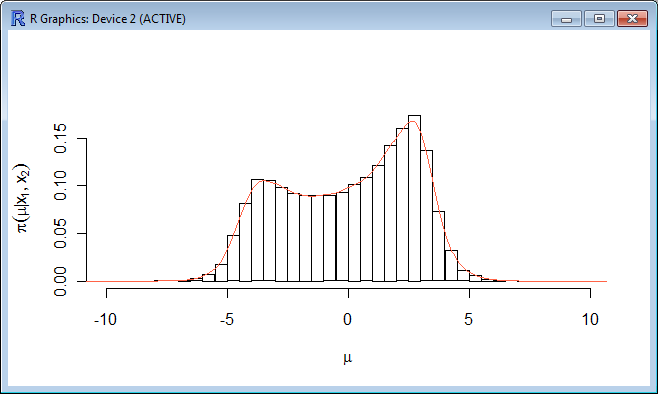
tail(letters)

tail(letters, n = -6L)

tail(library)

1. hist(mu[-1,],breaks=30,prob=TRUE,main=NULL,

xlab=expression(mu),ylab=expression(pi(paste(mu,"|",x[1]),x[2])))



then you can get this expression of x and y label in the figure

1. Use rlm() function to do robust regression. <http://www.ats.ucla.edu/stat/r/dae/rreg.htm>
2. boxplot.n( ) function can plot the boxplot with numbers on the bottom or top of the figures
3. Introduction to data technologies: <http://www.stat.auckland.ac.nz/~paul/ItDT/HTML/itdt.html>

R Graphic and codes from Paul Murrell: <http://www.stat.auckland.ac.nz/~paul/RG2e/>

1. Converting grads data to netCDF data in R using CDO:

<http://www.numbertheory.nl/2012/01/30/converting-grads-files-to-netcdf-from-within-r-using-cdo/>

<http://r-sig-geo.2731867.n2.nabble.com/reading-grd-file-in-R-td5616830.html>

1. Working with Spatial Data—A collection of notes, examples, references, and thoughts on working with spatial data: <http://casoilresource.lawr.ucdavis.edu/drupal/book/export/html/519>
2. x.num <- as.numeric(levels(x)[x]) ##convert factors to numeric type
3. extend a vector:

x <- c()

for (i in 1:5) x[length(x)+1] <- i

extend a matrix/data frame:

x <- matrix(, ncol = 5, nrow = 0)

for (I in 1:5) x[nrow(x)+1] <-

1. \*\*ply

##apply##

M <- matrix(seq(1,16), 4, 4)

apply(M, 1, min)

apply(M, 2, max)

# 3 dimensional array

M <- array( seq(32), dim = c(4,4,2))

# Apply sum across each M[\*, , ] - i.e Sum across 2nd and 3rd dimension

apply(M, 1, sum)

# Apply sum across each M[\*, \*, ] - i.e Sum across 3rd dimension

apply(M, c(1,2), sum)

# Result is two-dimensional

##lapply, both input and output are list

x <- list(a = 1, b = 1:3, c = 10:100)

lapply(x, FUN = length)

lapply(x, FUN = sum)

##sapply, input list, output vector

x <- list(a = 1, b = 1:3, c = 10:100)

sapply(x, FUN = length)

sapply(x, FUN = sum)

##output matrix

sapply(1:5,function(x) rnorm(3,x))

sapply(1:5,function(x) matrix(x,2,2))

sapply(1:5,function(x) matrix(x,2,2), simplify = "array")

##mapply apply the function to multiple data structures

#Sums the 1st elements, the 2nd elements, etc.

mapply(sum, 1:5, 1:5, 1:5)

##tapply - For when you want to apply a function to subsets of a

##vector and the subsets are defined by some other vector,

##usually a factor.

x <- 1:20

y <- factor(rep(letters[1:5], each = 4))

tapply(x, y, sum)

1. Time Series analysis and its applications: with R examples:

<http://www.stat.pitt.edu/stoffer/tsa3/>

Codes used in the text examples: <http://www.stat.pitt.edu/stoffer/tsa3/Rexamples.htm>

# Some R Time Series Issues (problems): <http://www.stat.pitt.edu/stoffer/tsa2/Rissues.htm>

1. The ["forecast"](http://cran.r-project.org/web/packages/forecast/index.html) package in [R](http://en.wikipedia.org/wiki/R_(programming_language)) can automatically select an ARIMA model for a given time series with the auto.arima() function. The package can also simulate seasonal and non-seasonal ARIMA models with its simulate.Arima() function. It also has a function Arima(), which is a wrapper for the arima from the "stats" package.
2. installed.packages()[,1:3]

gives the first 3 columns of info: the [1] package name, [2] library path, [3] version number.

1. To make sure you have the current version of your R packages, periodically update them by issuing the command   
   update.packages(ask=FALSE)
2. remove xlab at the original plot, and add self-defined axis

plot(FT$ELE\_GEN ~ FT$YM, xlab = i,xaxt = "n")

Axis(side=1, at=seq(2001, 2011, by=1), labels=T)

1. data matching/difference

union(x, y)

intersect(x, y)

setdiff(x, y)

setequal(x, y)

I s.element(el, set)

1. Install R and Rstudio in Ubuntu: <http://www.rstudio.com/ide/docs/server/management>

<http://blog.cloud-mes.com/2012/09/16/install-r-and-rstudio-in-ubuntu/>

1. Excellent R and statistics blog: http://yihui.name/cn/
2. How to plot netCDF file:

b = brick("C:\\study\\important document\\org data\\NARR\_monolevel\_2m\\air.2m.2010.nc", varname= 'air')

plot(b, 1511)

#or

r = raster("C:\\study\\important document\\org data\\NARR\_monolevel\_2m\\air.2m.2010.nc",

varname= 'air', band=1)

plot(r)

#or

sp = as(b, 'SpatialGridDataFrame')

1. Assign Closest Points to Members of second point list:

<http://www.nceas.ucsb.edu/scicomp/usecases/AssignClosestPointsToPoints>

1. plot CAMD powerplants on Google Earth

cd.CAMD <- CAMD.org[c(ncol(CAMD.org), ncol(CAMD.org)-1)]

sp.CAMD <- SpatialPointsDataFrame(coords = cd.CAMD, data = CAMD.org[,1:11])

proj4string(sp.CAMD) <- CRS("+proj=longlat +datum=WGS84")

plotKML(sp.CAMD)

1. Plot amazing and interactive map/figures/tables with R by google visualization:

<http://code.google.com/p/google-motion-charts-with-r/wiki/GadgetExamples>

1. R 时代，你要怎样画地图： <http://cos.name/2013/01/drawing-map-in-r-era/#more-6829>

Introduce several very useful R packages for mapping and geocoding:

ggmap: <https://sites.google.com/site/davidkahle/ggmap>

<http://www.stanford.edu/~cengel/cgi-bin/anthrospace/great-circles-on-a-recentered-worldmap-in-ggplot>

1. Mapping the world’s biggest Airlines:

<http://spatialanalysis.co.uk/2012/06/mapping-worlds-biggest-airlines/>

1. how to extract the original data of figures from website: <http://cos.name/cn/topic/107729>
2. R资料下载：<http://xccds1977.blogspot.com/p/r.html>

## str\_trim {stringr}: Trim whitespace from start and end of string.

1. snowfall is the new version of snow package, which is easier to use.
2. Parallel Computation with R (Snow/Snowfall): with some examples

<http://blog.nguyenvq.com/2009/06/20/parallel-computing-in-r-snowfallsnow/>

<http://stat.ethz.ch/R-manual/R-devel/library/parallel/doc/parallel.pdf>

1. When using the parallel computation, unless the computation of each input is time-consuming, it will take even longer than using single core. Because it takes some time to assign jobs.
2. For parallel apply: the function parRapply /parCapply functions in the snow package is fast

cl <- makeCluster(64, type = "SOCK")

t2 <- proc.time()

prl <- parRapply(cl, aa, fun = "sum")

stopCluster(cl)

t2 <- proc.time() - t2

But for lapply, the function mclapply in multicore package is fast:

t4 <- proc.time()

mcl <- mclapply(aa, FUN = "sum", mc.cores = 64)

t4 <- proc.time() - t4

1. multicore is only available in Linux version, not windows.
2. Covert matrix to list by column:

aa <- matrix(runif(100), ncol = 10)

bb <- as.list (as.data.frame(aa))

##note that: cc <- as.list(aa) will turn aa into list one by one, not by row

1. Some google earth packages:

RKML <http://blog.revolutionanalytics.com/2010/08/how-to-animate-google-earth-with-r.html>

R2G2: <http://www.r-bloggers.com/displaying-your-data-in-google-earth-using-r2g2/>

1. R 与高性能运算： <http://cos.name/wp-content/uploads/2011/05/01-Li-Jian-HPC.pdf>

compiler package can compile R language, enhance the efficiency.

1. How to develop R packages: <http://cos.name/2011/05/write-r-packages-like-a-ninja/>
2. Course About data visualization with R, provided by the professor who created ggplot:

<http://had.co.nz/stat645/>

1. It will takes about twice of memory to use snowfall functions. What is more, the rm() and gc()

function can only release half memory. Have to quit R to free all memory…..

1. This article introduces some fundamental settings for R:

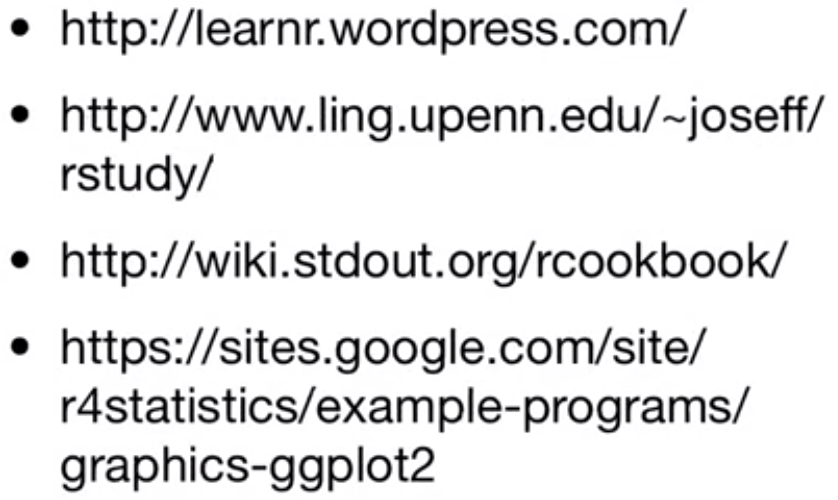
<https://github.com/yihui/r-ninja/blob/master/01-setup.md>

1. rm all variables but “x”: rm(list=setdiff(ls(), "x"))
2. The subtle distinction between na.action=na.omit and na.rm=TRUE is that na.omit will remove the entire row of data when it encounters a missing value, while the na.rm=TRUE   
   argument will remove missing values separately from each variable.
3. Run Python code in R:

<http://stackoverflow.com/questions/10155703/call-python-with-system-in-r-to-run-a-python-script-emulating-the-python-conso>

1. ggplot2 documents: <http://docs.ggplot2.org/current/>
2. mode(x) <- “numeric” can covert x to numeric type.
3. ggplot2 community:

<http://www.ling.upenn.edu/~joseff/rstudy/summer2010_ggplot2_intro.html>



1. Data.table package in R: <http://rwiki.sciviews.org/doku.php?id=packages:cran:data.table>

Can deal with data very fast <http://datatable.r-forge.r-project.org/>

<http://rwiki.sciviews.org/doku.php?id=packages:cran:data.table>

1. how to restore the default plotting settings

old<-par(no.readonly = TRUE) ##or par(), but par() will get warnings

par(mfrow = c(3,1))

plot(1:10)

par(old)

1. check object.size for each variable, and plot them:

all.object.sizes <- function()

{ return(rev(sort(sapply(ls(envir=.GlobalEnv), function (object.name)

object.size(get(object.name))))))

}

1. barplot(all.object.sizes(), main="Memory usage by object", ylab="Bytes", xlab="Variable name", col=heat.colors(length(object.sizes())))
2. R data mining materials: <http://www.rdatamining.com/>
3. R Reference Card for Data Mining:

<https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxyZGF0YW1pbmluZ3xneDoxN2Y1MDQ0MWRhOGVlMDQy>

1. read.fwf and write.fwf can read and write fixed width files
2. R正则表达式：<http://deerchao.net/tutorials/regex/regex.htm> 用grep 等函数时怎么匹配字符
3. Read the table directly:

x = read.table(textConnection('

X Y

a 10

a 20

a 30

b 100

b 200'), header = TRUE)

1. Read the data from clipboard: dt <- read.table("clipboard")
2. Dates and Times in R: <http://statistics.berkeley.edu/classes/s133/dates.html>
3. Concepts in computing with Data in R (Berkeley): http://statistics.berkeley.edu/classes/s133/
4. 求最大值和次大值：maxandnext= function(x){list(max(x),max(x[-which.max(x)]))}

or use the sort(x, partial = (n-1):n) function, which is much faster than sorting all elements.

1. When using the snowfall packages, it is required to export data and library to each slave. It will take a long time when the data is large. But with the package of multicore, it is not required to export.

mclapply(spl, FUN = WEATHR.fun, mc.cores= 66)

1. time consumptions are almost exactly the same when using foreach or mclapply
2. Reproducible research with R, knitr, Pandoc, and word:

<http://quantifyingmemory.blogspot.co.il/2013/02/reproducible-research-with-r-knitr.html>

<http://www.r-bloggers.com/installing-pandoc-from-r-on-windows/>

1. Shiny Package: Easy web applications in R: <http://www.rstudio.com/shiny/>

<http://rstudio.github.com/shiny/tutorial/#hello-shiny>

1. HTML5 slides: <http://slides.html5rocks.com/#landing-slide>

# Online resources for handling big data and parallel computing in R: <http://www.r-bloggers.com/online-resources-for-handling-big-data-and-parallel-computing-in-r/>

# How to Label all the outliers in a boxplot: <http://www.r-statistics.com/2011/01/how-to-label-all-the-outliers-in-a-boxplot/>

# <http://stackoverflow.com/questions/4787332/how-to-remove-outliers-from-a-dataset>

# x[!x %in% boxplot.stats(x)$out]

# Speed up loop operation in R: <http://stackoverflow.com/questions/2908822/speed-up-the-loop-operation-in-r>

# <http://biostat.mc.vanderbilt.edu/wiki/pub/Main/SvetlanaEdenRFiles/handouts.pdf>

# Is R’s apply family more than syntactic sugar:　 <http://stackoverflow.com/questions/2275896/is-rs-apply-family-more-than-syntactic-sugar>

# R and PBS: <http://users.sdsc.edu/~glockwood/comp/R.php>

# Crime data mapping with R: <http://www.maths.lancs.ac.uk/~rowlings/Teaching/UseR2012/crime.html> , introduce some useful spatial mapping libraries and functions

# [ggplot2: Adding Regression Line Equation and R2 on graph](http://stackoverflow.com/questions/7549694/ggplot2-adding-regression-line-equation-and-r2-on-graph): <http://stackoverflow.com/questions/7549694/ggplot2-adding-regression-line-equation-and-r2-on-graph>

# Use googleVis in shiny: <http://lamages.blogspot.co.uk/2013/02/first-steps-of-using-googlevis-on-shiny.html>

# traceback() to debug error

# test of central limit theorem:

# 

# Permutation test: <http://faculty.washington.edu/kenrice/sisg/SISG-08-06.pdf>

# Discussion of T-test for non-normal When N > 50: <http://stats.stackexchange.com/questions/9573/t-test-for-non-normal-when-n50>

# <http://stats.stackexchange.com/questions/2492/is-normality-testing-essentially-useless>

# Onlinecourse.PSU, paired test: <https://onlinecourses.science.psu.edu/stat500/book/export/html/48>

# R 隐者：<https://github.com/yihui/r-ninja/blob/master/01-setup.md>

# fuzzy match, agrep should do the tricks:

# agrep("biol", "biology")

# [1] 1

# agrep("biolgy", "biology")

# [1] 1

# agrep returns vector indices, so you can easily assign a value by using

# subscripts:animals[agrep("mou", animals)] <- "cheese"

# fancy colorbar legend in ggplot2: <http://kohske.wordpress.com/2011/04/01/alpha-version-of-colorbar-legend-in-ggplot2/>

# findInterval can find the index of one vector x in another, vec, where the latter must be non-decreasing :

# Mathematical Annotation in R: <http://vis.supstat.com/2013/04/mathematical-annotation-in-r/>

# 在R中怎样把字符矩阵转换成对应的数字矩阵

# > m = matrix(c('a', 'b', 'c', 'b', 'a', 'c'), 3) > m      [,1] [,2] [1,] "a"  "b" [2,] "b"  "a" [3,] "c"  "c" > matrix(c(a = 1.5, b = 2, c = 1)[m], nrow(m))      [,1] [,2] [1,]  1.5  2.0 [2,]  2.0  1.5 [3,]  1.0  1.0

# Book, [Forecasting: principles and practice](http://otexts.com/fpp/) : <http://otexts.com/fpp/using-r/>

# Knitr Bootstrap, a framework to create bootstrap styled HTML reports from knitr MD: <https://github.com/jimhester/knitr_bootstrap#knitr-bootstrap>

# Markdown写作浅谈: <http://www.yangzhiping.com/tech/r-markdown-knitr.html>

# An Example of R creating a pdf and then delivering with Shiny to Download: <https://github.com/timelyportfolio/shiny-d3-pdf> shiny::runGitHub('shiny-d3-pdf', 'timelyportfolio') <https://github.com/timelyportfolio>

# Global datasets: <http://spatial-analyst.net/wiki/index.php?title=Global_datasets#Lights_at_night_images>

# Cool Shiny examples: <https://groups.google.com/forum/?fromgroups=#!topic/shiny-discuss/RxmZ-JcLqE0>

# Plot google map in knitr: <http://rpubs.com/gallery/googleVis>

# Introduction to github in Chinese: <http://www.worldhello.net/gotgithub/index.html>

# googleVis slides: <http://dl.dropboxusercontent.com/u/7586336/blogger/deck.js/googleVis/index.html#title-slide>

# letters return lower level letters, LETTERS return higher level letters.

# do.call can also be used for data frame: df <- as.data.frame(cbind(letters = letters[1:5], numbers = 1:5)) do.call(paste0, df)

1. Don't R alone! A guide to tools for collaboration with R:

<http://www.noamross.net/blog/2013/1/7/collaborating-with-r.html>

1. a<- sprintf("%06d",1:20)

[1] "000001" "000002" "000003" "000004" "000005" "000006" "000007" "000008"

[9] "000009" "000010"

1. gesture identification system with R: <http://pci.github.io/mocap_js/demo/>
2. Shiny + Gridster + JustGage + Highcharts: drag the box anywhere you want, just like iphone

<http://glimmer.rstudio.com/winstontest/shiny-jsdemo/>

# d3 <- R with rCharts and solidify: <http://www.r-bloggers.com/d3-r-with-rcharts-and-slidify/>

# Companies using R: <http://blog.revolutionanalytics.com/2013/05/companies-using-open-source-r-in-2013.html>

# R中媲美Python Dictionary的神器-hash: <http://equation85.github.io/blog/hash-table-for-r/>

# Interactive slides with googleVis on shiny published on github: <http://www.r-bloggers.com/interactive-slides-with-googlevis-on-shiny/> <http://mages.github.io/googleVis_on_shiny/#1>

# Tweets Miner for stock Markets using R: <http://bpavlyshenko.blogspot.com/2013/05/tweets-miner-for-stock-markets.html>

# Dialect survey results displayed with Shiny: <http://spark-1590165977.us-west-2.elb.amazonaws.com/jkatz/SurveyMaps/>

# Upload data to shiny website: <http://glimmer.rstudio.com/dbets/dBETS/>

<http://jcheng5.github.io/shiny/tutorial/#uploads>

# spreadsheet-like editable data frame input：[https://groups.google.com/forum/?fromgroups=#!searchin/shiny-discuss/input/shiny-discuss/F8aAtv85ZGs/lGOdBkCVFsQJ](https://groups.google.com/forum/?fromgroups=" \l "!searchin/shiny-discuss/input/shiny-discuss/F8aAtv85ZGs/lGOdBkCVFsQJ)

# <http://jsfiddle.net/warpech/hU6Kz/>

# How to save data from user input for later use? <https://groups.google.com/forum/?fromgroups=#!searchin/shiny-discuss/input/shiny-discuss/Olr8m0JwMTo/ZzdOPIrW_MoJ>

# How can we make xkcd style graphs in R? Extrafont: <http://stackoverflow.com/questions/12675147/how-can-we-make-xkcd-style-graphs-in-r>

# Interactive color picker using locator(): <http://www.r-bloggers.com/interactive-color-picker-using-locator/>

1. Displaying time series, spatial and space-time data with R: stories of space and time <http://oscarperpinan.github.io/spacetime-vis/>
2. data.frame(state.name, state.x77) is equal to cbind(state.name, state.x77), but return a data.frame directly, cbind reture matrix
3. replacement trick:   
   m = matrix(c('a', 'b', 'c', 'b', 'a', 'c'), 3)  
   > m  
        [,1] [,2]  
   [1,] "a"  "b"  
   [2,] "b"  "a"  
   [3,] "c"  "c"  
   > matrix(c(a = 1.5, b = 2, c = 1)[m], nrow(m))  
        [,1] [,2]  
   [1,]  1.5  2.0  
   [2,]  2.0  1.5  
   [3,]  1.0  1.0
4. Use R to Bulk-Download Digital Elevation Data with 1" Resolution   
   <http://www.r-bloggers.com/use-r-to-bulk-download-digital-elevation-data-with-1-resolution/>
5. R code used to regrid NARCCAP data: <http://www.narccap.ucar.edu/contrib/tools/R/>
6. spacetime: Spatial-Temporal data in R:   
   <http://cran.r-project.org/web/packages/spacetime/vignettes/jss816.pdf>

<http://www.stat.ncsu.edu/people/fuentes/courses/stwarwick/lectures/spacetime.pdf>

1. colored 3D map: <http://r-nold.blogspot.com/2012/07/colored-3d-map.html>

<http://stackoverflow.com/questions/6901238/overlay-a-map-on-top-of-a-3d-surface-map-in-r>

<http://stackoverflow.com/questions/1896419/plotting-a-3d-surface-plot-with-contour-map-overlay-using-r>

1. Radyant - Marketing analysis using Shiny: [http://vnijs.rady.ucsd.edu:3838/marketing/#](http://vnijs.rady.ucsd.edu:3838/marketing/)

Code : <https://github.com/mostly-harmless/radyant/tree/master/inst/marketing>

1. Time Is on My Side - A Small Example for Text Analytics on a Stream:   
   <http://things-about-r.tumblr.com/>
2. dummyData = rep(c(1, 2, 2, 2), 25)

c(table(dummyData)) # get un-dimensioned integer vector, table will return table object

1. 2

25 75

1. Crime in Downtown Houston, combining ggplot2 and google Maps:   
   <https://github.com/hadley/ggplot2/wiki/Crime-in-Downtown-Houston,-Texas-:-Combining-ggplot2-and-Google-Maps>

Visualizing crime hotspots in England:  
<http://blenditbayes.blogspot.com/2013/06/visualising-crime-hotspots-in-england_25.html>

1. Predicting spatial locations using point processes: <http://rpubs.com/dahtah/6730>
2. How to remove boxplot outliers:   
   x[!x %in% boxplot.stats(x)$out]   
   <http://stackoverflow.com/questions/4787332/how-to-remove-outliers-from-a-dataset>
3. Rdocumentation, comprehensive R package and help information:   
   <http://www.rdocumentation.org/>
4. Stable marriage problem, using different language to write the algorithm except R: <http://rosettacode.org/wiki/Stable_marriage_problem#Python>
5. Running R scripts directly from Dropbox:   
   <http://www.econometricsbysimulation.com/2013/06/running-r-scripts-directly-from-dropbox.html>  
   rDrop: Dropbox interface via R: <https://github.com/karthikram/rDrop>

Opa, dropbox as a database: <http://blog.opalang.org/2012/10/dropbox-as-database.html>

1. Local R user Group Directory: <http://blog.revolutionanalytics.com/local-r-groups.html>
2. Convert IP address to geolocation, latitude and longitude etc:   
   <http://www.r-bloggers.com/convert-ip-addresses-to-geolocation-latitude-and-longitude-etc-etc/>
3. <<- and assign( ) can be used as ‘superassignment’, when the variable should be changed globally and permanently.
4. Using R to produce scalable vector Graphics for the Web:   
   <http://www.r-bloggers.com/using-r-to-produce-scalable-vector-graphics-for-the-web/>

Comparison of Raster Image and Scalar Vector Image:   
<http://statistical-research.com/wp-content/uploads/2013/06/eqImageCompare.html>

1. R不务正业之RCurl： <http://cos.name/cn/topic/17816>
2. Datamind website, study R interactively: <http://www.datamind.org/#/>
3. substitute returns the parse tree for the (unevaluated) expression expr, substituting any variables bound in env.   
    foo <- function(data) paste("T", substitute(data), sep = '.')

x = 1:10

foo(data = x)

[1] "T.x"

1. Shiny cool examples put together in slides: <http://ramnathv.github.io/shinyExamples/#18>
2. Jared Knowles providing rich resource for R: <http://jaredknowles.com/phd-research/>
3. use upper case Y if the data is four digit, lower case if two digit

as.Date('12-31-2002',format='%m-%d-%Y')

as.Date('12-31-02',format='%m-%d-%y')

1. Logging IP addresses of people who access the shiny-server?:

<https://groups.google.com/forum/#!searchin/shiny-discuss/get$20ip$20address/shiny-discuss/EGQhEyoEk3E/a-OPHYxGWlgJ>

<http://jsonip.com>

IP address to location: <http://www.geobytes.com/iplocator.htm>

1. Shiny locator: <https://groups.google.com/forum/#!searchin/shiny-discuss/grid$20locator/shiny-discuss/0B8VFC9XK4o/ejdES_2fCY0J>
2. Passing parameters to a Shiny app: <https://groups.google.com/forum/#!searchin/shiny-discuss/PHP/shiny-discuss/8ZpNyzPKaTg/GeV25pQq0d8J>

URL as Input/Output: <https://gist.github.com/alexbbrown/6e77383b48a044191771>

1. IP geolocation database: <http://dev.maxmind.com/geoip/legacy/geolite/>
2. Shiny app embedded in website: <http://stla.overblog.com/reactive-3d-surface>

<https://groups.google.com/forum/?fromgroups=#!topic/shiny-discuss/2VwEWXYJwYs>

1. Methods to get IP address in Chinese: <http://blog.csdn.net/zyw_anquan/article/details/7970869>
2. How to insert javascript in shiny: <https://groups.google.com/forum/#!searchin/shiny-discuss/javascript/shiny-discuss/LTvLLCRe0W8/GSlBz2A1pW8J>
3. Introduction to data science from Coursera (it introduces a lot of interesting topics about data science, for example, Mapreduce, NoSQL, visualization): <https://class.coursera.org/datasci-001/lecture/index>
4. Bike sharing in 100 cities, very beautiful shiny app: <http://www.r-bloggers.com/bike-sharing-in-100-cities/>
5. R-Node, a web front-end to the statistical analysis with R: <https://github.com/elijah/r-node>
6. Ggplot2 meet d3, how d3 zoom in/out ggplot2 graphs: <http://www.r-bloggers.com/ggplot2-meet-d3/>
7. Shiny matrixInput, editable table: <https://gist.github.com/jcheng5/3960092>
8. Stock analysis using R: <http://www.r-bloggers.com/stock-analysis-using-r/>
9. The Omega Project for Statistical Computing, integrate R with browsers: <http://www.omegahat.org>
10. How to automate multiple requests to a web search form using R: <http://stackoverflow.com/questions/5396461/how-to-automate-multiple-requests-to-a-web-search-form-using-r>
11. What if I want to web scrape with R for a page with parameters?: <http://stackoverflow.com/questions/16601520/what-if-i-want-to-web-scrape-with-r-for-a-page-with-parameters/16860430#16860430>
12. Scraping Twitter and Web Data Using R: <http://www.nyu.edu/projects/politicsdatalab/workshops/twitter.pdf>
13. Changing Tor Identity (IP) in R: <http://stackoverflow.com/questions/11089993/changing-tor-identity-in-r/11476662#11476662>
14. Proxy setting for R: <http://stackoverflow.com/questions/6467277/proxy-setting-for-r>
15. R, GeoJSON and GitHub: : <http://www.r-bloggers.com/r-geojson-and-github/>
16. HyetosR: used to downscale rainfall to finer scale <http://itia.ntua.gr/getfile/1200/1/documents/HyetosR_EGU_pres.pdf>
17. 用spTransform转换projection <http://gis.stackexchange.com/questions/31743/projecting-sp-objects-in-r>
18. Rgdal package installation in Linux: <http://stackoverflow.com/questions/15248815/rgdal-package-installation>
19. Epsg projection definition: <http://www.spatialreference.org/ref/epsg/26933/>
20. How to format plots for publication using ggplot2 (with some help from Inkscape): <http://www.r-bloggers.com/how-to-format-plots-for-publication-using-ggplot2-with-some-help-from-inkscape/>
21. Pyramid can used to draw population age pyramid plot.   
    <http://www.r-bloggers.com/my-first-web-application-with-shiny/>
22. for loop is much faster for matrix than data.frame: (avoid using data.frame in for loop)

data1 <- matrix(0, ncol = 1000, nrow = 1000)

data2 <- data.frame(data1)

# 0.42 seconds

for (i in 1:1000){

for (j in 1:1000){

temp = data1[i,j]

}

}

# 33.2 seconds

for (i in 1:1000){

for (j in 1:1000){

temp = data2[i,j]

}

}

But the difference for apply function is small. Whether the data type is matrix or data.frame, the time consumption is similar.

rs3 = apply(data1, c(1,2), function(x) x^2)

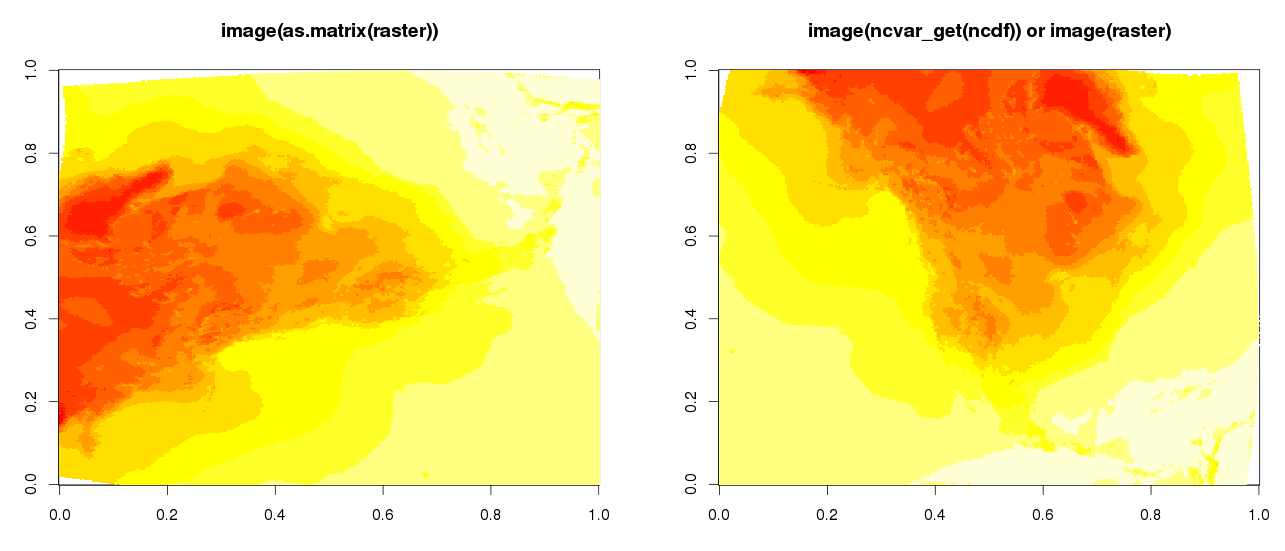
rs4 = apply(data2, c(1,2), function(x) x^2)

1. Quickly reading very large tables as dataframes in R

<http://stackoverflow.com/questions/1727772/quickly-reading-very-large-tables-as-dataframes-in-r>

1. The ‘fread’ function in data.table package can read csv file into R much faster than read.csv
2. Exchanging data between R and MS Windows apps (Excel, etc): <http://rwiki.sciviews.org/doku.php?id=tips%3adata-io%3ams_windows#exchanging_data_between_r_and_ms_windows_apps_excel_etc>

1. raster extract values by row, while matrix extract values by column. The dimension of ncdf and raster are reverse, but their image will be the same. Since they extract data reversely too.



1. avoid using the spDists function, because it will generate huge matrix. Use the spDistsN1 instead, and then use the lapply function for parallel computation, returning only the values which minimize the distance. The following codes play the same role, but spDistsN1 is faster, and save some space

system.time(test1 <- mclapply(1:nrow(smp), function(x) {

dist <- spDistsN1(coordinates(T8.state), coordinates(smp)[x,], longlat= F)

NN <- which.min(dist)

return(NN)

gc()

}, mc.cores = 5))

system.time(test2 <- spDists(coordinates(T8.state), coordinates(smp), longlat = F))

1. Three parallel computation methods:

## multicore is the simplest and fast, but it will copy everything in the environment directly to each core

## Thus it will take up huge memory if there are a lot of variables in the environment.

## So, try to run this parallel computation first, before loading other variables into the environment.

system.time(test2 <- mclapply(1:nrow(smp), function(x) {

dist <- spDistsN1(T8.coord, smp.coord[x,], longlat= F)

NN <- which.min(dist)

return(NN)

gc()

}, mc.cores = 6))

## snow will not use the variables and libraries in the environment, unless you export it to each core

## but it will take a very long time to export large variables to each core. When there are a lot of variables ## in the environment, but you only need few of them, it is good to use the ‘snow’ package, because you ## can use more cores.

cl <- makeCluster(30, type = 'SOCK')

clusterExport(cl = cl, varlist= c('T8.coord', 'smp.coord'), envir=environment())

clusterEvalQ(cl, {library(sp)})

system.time(test4 <- parLapply(cl, 1:nrow(smp), function(x) {

dist <- spDistsN1(T8.coord, smp.coord[x,], longlat= F)

NN <- which.min(dist)

return(NN)

gc()

}))

## snowfall is the improved version of snow, the syntax is simpler, and the speed a little faster.

sfInit(parallel=T, cpus = 30)

sfLibrary(sp)

sfExport(list=c('T8.coord', 'smp.coord'))

system.time(test3 <- sfLapply(1:nrow(smp), function(x) {

dist <- spDistsN1(T8.coord, smp.coord[x,], longlat= F)

NN <- which.min(dist)

return(NN)

gc()

}))

sfStop()

1. No need for SPSS – beautiful output in R #rstats:   
   <http://strengejacke.wordpress.com/2014/02/20/no-need-for-spss-beautiful-output-in-r-rstats/>

<http://strengejacke.wordpress.com/2014/03/04/beautiful-table-outputs-in-r-part-2-rstats-sjplot/#comment-772>

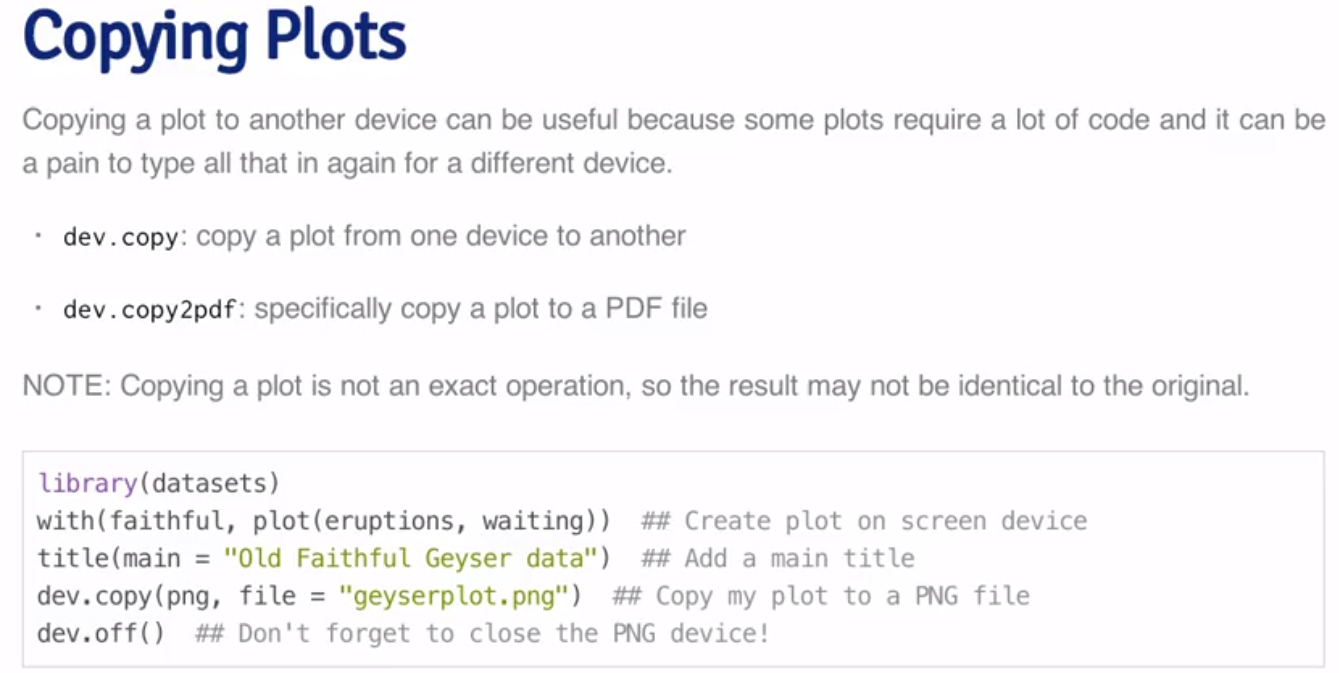
1. R gsub Function, regular expression: <http://www.endmemo.com/program/R/gsub.php>
2. Beautiful interactive map using R and QGIS:   
   <http://www.r-bloggers.com/poverty-and-crime-in-mexico-interactive-map/>
3. Time series modeling with independent variables: <https://www.otexts.org/fpp/9/1/>  
   <http://stats.stackexchange.com/questions/52461/time-series-modeling-with-independent-variables>
4. Spatial.ly Visualisation, analysis, and resources:   
   <http://spatial.ly/category/r-spatial-data-hints/>
5. How to map connections with great circles:   
   <http://flowingdata.com/2011/05/11/how-to-map-connections-with-great-circles/>

<http://blog.ouseful.info/2014/03/24/experimenting-with-r-point-to-point-mapping-with-great-circles/>

1. A simple Introduction to the graphing philosophy of ggplot2:   
   <http://www.r-bloggers.com/a-simple-introduction-to-the-graphing-philosophy-of-ggplot2/>
2. Bayesian Inference Using OpenBUGS (R2OpenBUGS):   
   <http://www.r-tutor.com/bayesian-statistics/openbugs>
3. Circular Migration Flow Plots in R:   
   <http://gjabel.wordpress.com/2014/03/28/circular-migration-flow-plots-in-r/>  
   <http://global-migration.info/>
4. How to map connection with great circles:   
   <http://flowingdata.com/2011/05/11/how-to-map-connections-with-great-circles/>  
   Visualizing Facebook Friends: Eye Candy in R: <http://paulbutler.org/archives/visualizing-facebook-friends/>
5. Make your ggplots shareable, collaborative, and with D3:   
   <http://www.r-bloggers.com/make-your-ggplots-shareable-collaborative-and-with-d3/>

<https://gist.github.com/sckott/10991885>

1. Play 2048… using R! <http://www.r-bloggers.com/play-2048-using-r/>
2. R Scripts (K-means clustering) for multiple variables using principle components analysis (princomp):   
   <https://onlinecourses.science.psu.edu/stat857/node/125>  
   determine the optimal number of cluster :   
   <http://stackoverflow.com/questions/15376075/cluster-analysis-in-r-determine-the-optimal-number-of-clusters>
3. R activity around the world:   
   <http://blog.rapporter.net/2014/04/r-activity-around-world.html?utm_content=buffercc326&utm_medium=social&utm_source=plus.google.com&utm_campaign=buffer>
4. Who will win the senate: (visualization with R)   
   <http://www.nytimes.com/newsgraphics/2014/senate-model/?hp#spinners>
5. Regression Diagnostics: <http://www.statmethods.net/stats/rdiagnostics.html>
6. Converting shapefiles to rasters in R <http://www.r-bloggers.com/converting-shapefiles-to-rasters-in-r/>
7. Use example(points) to check the details of the plot arguments.
8. Copying plots:

dev.copy() or dev.copy2pdf  
  


1. Show me Shiny: gallery of R web apps with codes: <http://www.showmeshiny.com/>
2. loess局部多项式回归如何取得极值点<http://cos.name/cn/topic/140675>
3. Breakpoint analysis, segmented regression:   
   <http://rstudio-pubs-static.s3.amazonaws.com/12164_ce6e9b8542e8494b8d225b61b632efd9.html>  
   Robustness of Segmented Regression Contributed by Muggeo:   
   <https://stat.ethz.ch/pipermail/r-help/2005-June/073284.html>

Predicting from segmented regression:   
<https://stat.ethz.ch/pipermail/r-help/2009-September/210454.html>

1. Rseek website cover the broad search of R function and packages: <http://www.rseek.org/>
2. RStudio Server Amazon Machine Image (AMI) <http://www.louisaslett.com/RStudio_AMI/>  
   <http://www.surefoss.org/workflow/running-r-on-32-cores-for-2usdh/>
3. loess method to find the breaking points, the results look not very good

# lss <- loess(y ~ x)

# j <- order(x)

# newX=seq(min(x), max(x), 0.1)

# newY=predict(lss, newdata= newX)

# bkp <- newX[which(diff(diff(newY)>0)!=0L)+1L]

# abline(v = bkp, lty = 2)

# lines(x[j],lss$fitted[j],col="red",lwd=3)

1. To project coordinates (degree) to meter or other unit, the project( ) function needs some parameters. In the ncdf file, they are list in the “int” variable:

[1] " int Lambert\_Conformal[] "

[1] " false\_easting: 5632642.22547"

[1] " false\_northing: 4612545.65137"

[1] " grid\_mapping\_name: lambert\_conformal\_conic"

[1] " latitude\_of\_projection\_origin: 50"

[1] " longitude\_of\_central\_meridian: -107"

[1] " standard\_parallel: 50" " standard\_parallel: 50"

Some the parameters can be generated with the int variables as:

“+x\_0=5632642.22547 +y\_0=4612545.65137 +proj=lcc +lat\_0=50 +lon\_0=-107 +lat\_1=50 +lat\_2=50”

The definition of each parameter is listed here: <http://trac.osgeo.org/proj/wiki/GenParms>

We can also get eh projection directly using the raster function to open the ncdf file. But sometimes the raster function can’t extract the projection correctly. So, have to double check with the results using nc\_open.

1. Use row.names to extract the row names of spatial objects:

all.equal(row.names(cmip.cor.cc), rownames([cmip.cor.cc@data)](mailto:cmip.cor.cc@data)))

1. Aggregate all columns with data.table:   
   <http://stackoverflow.com/questions/18091146/aggregate-all-columns-with-data-table>
2. Don’t use aggregate to sum by columns, it is very slow! It is not vectorized! Use rowsum instead

**rowsum(x, group)**

1. Integer format can only contains number less than 10^9. So, it is better to avoid using as.integer when the number is big.
2. SPEED UP YOUR R CODE USING A JUST-IN-TIME (JIT) COMPILER:   
   <http://www.r-statistics.com/2012/04/speed-up-your-r-code-using-a-just-in-time-jit-compiler/>
3. How to get help from Shiny: <http://shiny.rstudio.com/articles/help.html>
4. legend.box = 'horizontal' or legend.box = vertical in theme to change the position of multiple legends in ggplot
5. ggplot2: Adding Regression Line Equation and R2 on graph:  
   <http://stackoverflow.com/questions/7549694/ggplot2-adding-regression-line-equation-and-r2-on-graph>
6. A simple workflow for using R with Microsoft Office products:   
   <http://beckmw.wordpress.com/2014/06/05/a-simple-workflow-for-using-r-with-microsoft-office-products/>
7. ReporteRs, an R package to generate Mircosoft Word, PPT, and HTML reports:   
   <http://davidgohel.github.io/ReporteRs/index.html>
8. ggsave to save plot, and file.show to show it:  
   ggsave('/NAS/jhuang/Projects/Set.Point.Temperature.NARR/outputs/figures/ggsave.png')

file.show('/NAS/jhuang/Projects/Set.Point.Temperature.NARR/outputs/figures/ggsave.png')

1. If an object is saved within a function, which include a lot of other objects, the saved object will be huge. That is related to the R environment.
2. FAQ: Why are R2 and F so large for models without a constant?  
   <http://www.ats.ucla.edu/stat/mult_pkg/faq/general/noconstant.htm>
3. Don’t assign an empty matrix with NA, it is logical mode. When you input values into the matrix, all logical format NA will be coerced to numeric or integer format, which will waste a lot of time. Instead, you can just assign 0, or 0L to the empty matrix:   
   <http://www.r-bloggers.com/pitfall-did-you-really-mean-to-use-matrixnrow-ncol/>
4. R markdown v2: Interactive Documents (embedded Shiny app):  
   <http://rmarkdown.rstudio.com/authoring_shiny.html>

<http://www.r-bloggers.com/interactive-documents-an-incredibly-easy-way-to-use-shiny/>

1. Generate random numbers with fixed mean and sd:   
   <http://stackoverflow.com/questions/18919091/r-generate-random-numbers-with-fixed-mean-and-sd>

rnorm2 <- function(n,mean,sd) { mean+sd\*scale(rnorm(n)) }

r <- rnorm2(100,4,1)

1. Tools for making a paper <http://conjugateprior.org/2013/03/tools-for-making-a-paper/>

<https://github.com/cpfaff/Open-Science-Paper>

1. Great circles lines on a world map with rworldmap and ggplot2 packages:  
   <http://www.r-bloggers.com/great-circles-lines-on-a-world-map-with-rworldmap-and-ggplot2-packages/>
2. Remove everything inside parentheses:

gsub("\\([^)]\*\\)", "", myvector)

<http://r.789695.n4.nabble.com/Regular-expression-to-define-contents-between-parentheses-td848782.html>

1. The Shiny Cheat Sheet: <http://shiny.rstudio.com/articles/cheatsheet.html>
2. LeafLet-shiny example (US population): <https://github.com/jcheng5/leaflet-shiny>
3. Scheduling R scripts to run on a regular basis (intermittently):   
   <http://tgmstat.wordpress.com/2013/09/11/schedule-rscript-with-cron/>
4. How to buy a used car with R:   
   <http://web.stanford.edu/~knoepfle/cgi-bin/flatpress/?x=entry:entry101219-223914>
5. Google’s R style guide: <https://google-styleguide.googlecode.com/svn/trunk/Rguide.xml>
6. R抓取机票数据问题（利用javascript工具PhantomJS直接抓取网页所呈现内容，而不是source page。有些网页的内容被js封装，不会直接出现在source page里面，但是会出现在网页界面，利用这个办法可以直接抓取界面）：<http://cos.name/cn/topic/157918>
7. Making Your Code Citable <http://www.r-bloggers.com/making-your-code-citable/>
8. Print matrix without dimnames (colnames and rownames):  
   <http://stackoverflow.com/questions/9366596/matrix-display-without-row-and-column-names>

print.matrix <- function(m){

write.table(format(m, justify="right"),

row.names=F, col.names=F, quote=F)

}

1. zoom into your googleVis chart: <http://www.r-bloggers.com/zoom-zoom-googlevis/>
2. Number of days in each month: <https://stat.ethz.ch/pipermail/r-help/2007-August/138116.html>

*diff(seq(as.Date("2007-01-01"), as.Date("2021-01-01"), by = "month"))*

1. Return argument names: deparse(substitute(tas))  
   <http://stackoverflow.com/questions/5754367/using-substitute-to-get-argument-name-with>
2. The output of cbind(x, y) is matrix, and data.frame(x, y) is data.frame.  
   so, only one data type is allowed for cbind. Or else, the data type will be coerced to one single type.
3. Track how visitors use your Shiny app with Google Analytics:  
   <http://www.r-bloggers.com/track-how-visitors-use-your-shiny-app-with-google-analytics/>  
   <http://shiny.rstudio.com/articles/google-analytics.html>
4. When the path is long than 80 characters, use the paste0 function to separate it into several lines:  
   hourDir <- paste0('C:/study/Projects/Energy\_CO2/EnergyPlus/',

'ResidentialBuilding2012.V7.2.Hour')

1. Transparent R plot (Transparent hurricane paths in R)  
    <http://www.r-bloggers.com/transparent-hurricane-paths-in-r/>  
   **color=”red”** replaced by: **color=rgb(255,0,0,18,maxColorValue=255)**.
2. Using GitHub with R and Rstudio (setup Rstudio to connect to github):

<http://www.molecularecologist.com/2013/11/using-github-with-r-and-rstudio/>

RSA key is needed before this: Rstudio -> Tools -> Global options -> Git/SVN -> create RSA Key

Then copy the key to github: github -> Settings -> SSH keys -> Add SSH key

1. glob2rx: Change wildcard aka globbing patterns into the corresponding regular expressions ([regexp](http://ec2-54-172-96-200.compute-1.amazonaws.com:8787/help/library/utils/help/regexp)).

glob2rx('\*USA\*.idf')

[1] "^.\*USA.\*\\.idf$"

1. SVG + a little extra (d3.js) in RStudio Browser | No Pipes This Time: (To change plot size)   
   <http://timelyportfolio.blogspot.com/2014/10/svg-little-extra-d3js-in-rstudio.html>
2. What are some good hacks at using R?: <http://www.quora.com/What-are-some-good-hacks-at-using-R>

Higher Order Functions in R:   
<http://www.johnmyleswhite.com/notebook/2010/09/23/higher-order-functions-in-r/>

1. swirl courses, Learn R in R: <https://github.com/swirldev/swirl_courses#swirl-courses>   
   <http://swirlstats.com/>
2. Spatial Cheatsheet: <http://www.maths.lancs.ac.uk/~rowlings/Teaching/UseR2012/cheatsheet.html>
3. Improving R Data Visualisations Through Design:  
   <http://www.r-bloggers.com/improving-r-data-visualisations-through-design/>
4. When the column names include “.” (e.g., station.loc), it can be called directly using var$station when it will not cause ambiguity (Only one column name include station)
5. Interactive 3D Plots with ggplot2 and Plotly:   
   <http://www.r-bloggers.com/3d-plots-with-ggplot2-and-plotly/>  
   Plotly R Library: <https://plot.ly/r/>
6. Statistical Analysis with Open-Source R and RStudio on Amazon EMR:   
   <http://blogs.aws.amazon.com/bigdata/post/Tx37RSKRFDQNTSL/Statistical-Analysis-with-Open-Source-R-and-RStudio-on-Amazon-EMR>
7. The Symbol used for Date formatting:  
   <http://www.statmethods.net/input/dates.html>

|  |  |  |
| --- | --- | --- |
| **Symbol** | **Meaning** | **Example** |
| **%d** | day as a number (0-31) | 01-31 |
| **%a %A** | abbreviated weekday  unabbreviated weekday | Mon Monday |
| **%m** | month (00-12) | 00-12 |
| **%b %B** | abbreviated month unabbreviated month | Jan January |
| **%y %Y** | 2-digit year  4-digit year | 07 2007 |

1. failwith: Modify a function so that it returns a default value when there is an error.

It is useful when some of the data may cause error. When they cause error, you can skip them, instead of stopping the whole process. Similar to tryCatch.

failwith(NULL, lm)

1. clusterApply : apply the functions to a cluster  
   <http://matloff.wordpress.com/2014/11/26/how-about-a-snowdoop-package/>
2. Object-oriented (OO) system in R: <http://adv-r.had.co.nz/OO-essentials.html>

The Most Basic Elements of Object-Oriented Programming in R (create your own methods):   
<http://www.johnmyleswhite.com/notebook/2009/12/13/the-most-basic-elements-of-object-oriented-programming-in-r/>

1. The OO system types:

* + S3: no formal definition of class, generic function decides which method to call
  + S4: has formal class definition, multiple dispatch, generic functions can pick methods based on class of any number of arguments.
  + RC: methods belong to object, not function. $ is used to separate object and method
  + Base type: the internal C-level types that underline the other OO systems.

2. The typeof() a function is ‘closure’ and the type of a primitive function is ‘builtin’

typeof(mean)

[1] "closure"

typeof(sum)

[1] "builtin"

3. Define two methods 'PrintInteger' and 'PrintCharcter' under the id generic

user1 <- user2 <- list(id = 1, password = '\*\*', email = '\*\*@gmail.com')

class(user1) <- 'PrintInteger' # define class

class(user2) <- 'PrintCharcter' # define class

id.PrintInteger <- function(PrintInteger.object){

+ return(PrintInteger.object[['id']])

+ } # define method for 'PrintInteger' class

id.PrintCharcter <- function(PrintCharcter.object){

+ return(as.character(PrintCharcter.object[['id']]))

+ } # define method for 'user' class

id <- function(.){

+ UseMethod('id', .)

+ }

id(user1)

[1] 1

id(user2)

[1] "1"

4. use methods() to list all methods for a generic.fun or class:

a. methods(generic.function = 'ts') # list all methods for ‘ts’ generic

b. methods(class = 'ts') # list all methods for class ‘ts’

1. Practical Regression and Anova using R (A free R book)   
   <http://cran.r-project.org/doc/contrib/Faraway-PRA.pdf>
2. aov function fits an analysis of variance model by a call to lm for each stratum, and anova function compute analysis of variance (or deviance) tables for one or more fitted model objects  
   Thus, the result of aov(y ~ x) is equal to anova(lm(y ~ x)
3. ordered function can be used to reorder factors: d$f <- ordered(d$f, c( "grapes", "apples", "oranges"))
4. plot(value ~ factors, data) will plot the boxplot grouped by factors automatically. It is equal to convert the data from long format to wide format, then boxplot() the data.
5. file() is used to open a file connection, then you can read or write to it.

Tfile <- file()

cat("abc\ndef\n", file = Tfile)

readLines(Tfile)

close(Tfile)

1. In-depth introduction to machine learning in 15 hours of expert videos (ESL class with hw solutions):

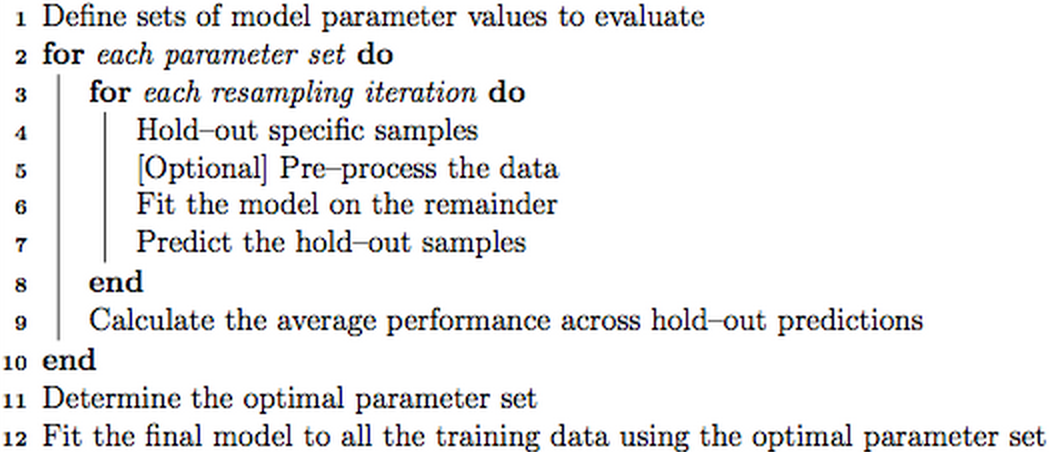
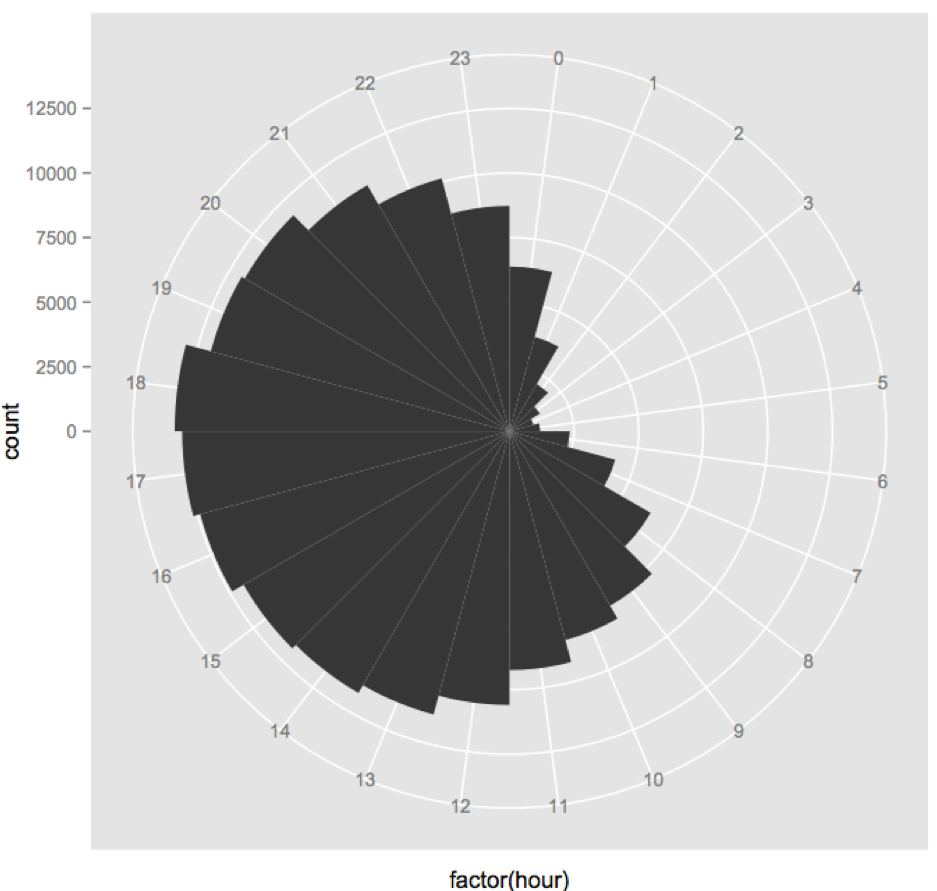
ISL (Introduction to Statistical Learning) is the textbook used for the course, it is based on ESL (Elements of Statistical Learning), which is more advanced and include more theory.   
<http://www.r-bloggers.com/in-depth-introduction-to-machine-learning-in-15-hours-of-expert-videos/s>

<http://beautifuldata.net/2015/01/2014-highlight-statistical-learning/>

<http://www-bcf.usc.edu/~gareth/ISL/> (ILS)

1. Introducing practical and robust anomaly detection in a time series: (AnomalyDetection package)

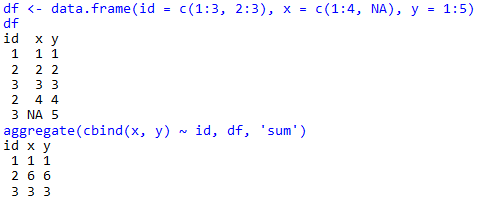
<https://blog.twitter.com/2015/introducing-practical-and-robust-anomaly-detection-in-a-time-series>

1. Fully reproducible parallel models using caret:   
   <http://stackoverflow.com/questions/13403427/fully-reproducible-parallel-models-using-caret>
2. PML project reference:   
   <http://rstudio-pubs-static.s3.amazonaws.com/20380_01d51a675de747c0a60ab5967cdedade.html>
3. Gbm model usually takes much more time than rf
4. Model Training and Parameter Tuning: (very detailed about using caret gbm model to train data and build model)   
   <http://topepo.github.io/caret/training.html> 
5. Use the ave function to cumsum by group, aggregate function doesn’t work as expected, because it will collapse all cumsum values into one cell, separated by comma.
   * within(mydf, { product <- ave(product, level1, level2, FUN = cumsum)})
   * aggregate(product ~ level1 + level2, mydf, cumsum)
6. get(‘variable’) will return the content of variable, it is the same as eval(parse(text = ‘variable’))
7. count.field can be used to return the number of column for each row in the file.
8. In ggplot2, the aes() contains the information that varies, e.g., x, y, size. The value that is constant should be outside of aes(), e.g., shape = 1 for all data points:  
   geom\_point(data = tmy3\_pop, aes(x = long, y = lat, size = pop2010), shape = 1)
9. It is nice to use the polar coordinates chart to plot the diurnal change. It makes you think of the clock:  
   <http://www.r-bloggers.com/nyc-is-a-city-that-does-sleep-a-bit/>  
   
10. Use lapply to change several columns in the same time:   
    mtcars[, c("cyl", "am", "gear")] <- lapply(mtcars[, c("cyl", "am", "gear")], as.factor)
11. Add Text Annotations to ggplot2 Faceted Plot:   
    <https://trinkerrstuff.wordpress.com/2012/09/01/add-text-annotations-to-ggplot2-faceted-plot/>
12. Xor indicates elementwise exclusive OR:

xor(c(TRUE, TRUE, FALSE, FALSE), c(TRUE, FALSE, TRUE, FALSE))

[1] FALSE TRUE TRUE FALSE

1. Aggregate will exclude the whole rows with NA. If one of the column contains NA value, it will affect other columns. For example, the whole fifth row is removed: (setting na.rm = F) doesn’t help



To overcome this, you must aggregate x and y separately, and then merge the results

It is much easier to use the dplyr::summarise function

1. How to install rgdal in Linux (Redhat):
   * Create /etc/ld.so.conf.d/gdal.conf file, and put ‘/usr/local/lib’ into the gdal.conf

With the command: sudo vim /etc/ld.so.conf.d/gdal.conf

* + Check where is the gdal-config file with the command:

Locate gdal-config

It will shows ‘/usr/local/bin/gdal-config’ if gdal is installed

If not, install gdal using ‘yum install gdal’ firstly

* + Install the ‘rgdal’ package with the following command in R:

install.packages('rgdal', lib = '/NAS/jhuang/R/x86\_64-redhat-linux-gnu-library/3.1',

configure.args = '--with-gdal-config=/usr/local/bin/gdal-config')

1. Apply can be used to a matrix and return a matrix:  
   apply(matrix(1:10, 2), 2, function(x) x/1:2)

[,1] [,2] [,3] [,4] [,5]

[1,] 1 3 5 7 9

[2,] 1 2 3 4 5

1. str\_replace\_all() gains a convenient syntax for applying multiple pairs of pattern and replacement to the same vector:

x <- c("abc", "def")

str\_replace\_all(x, c("[ad]" = "!", "[cf][/cf]" = "?"))

[1] "!b?" "!e?"

1. data.table::fread is used to read data file at a lightning speed! it can replace the widely used read.table and read.csv functions. But sometimes the fread function can't detect the data correctly, and thus not as stable as the other two functions.
2. data.table::rbindlist is used to rbind the data.frame inside a list. it is equivalent to do.call(rbind, list), but much faster. Other similar functions in *dplyr* package are rbind\_all and rbind\_list.
3. rowsum can be used to sum data by ID. It is similar to aggregate. But when the data has a lot of columns, the rowsum function is much faster.
4. match can be used to match the IDs between two datasets. When we want to combine two datasets by IDs, we usually use the merge function directly. But when the datasets are huge, it takes very long time to merge them. Instead, we can use the match function to match two datasets firstly, and then apply the values of one dataset to another by the index return by match.
5. dim function can turn an array into any dimension you want. For example, 10*8760 array (matrix) can be turn into 3 dimensional (10*365\*24) array. This is very helpful when you need to calculate the marginal sum/mean, e.g. the daily (24 hour) sum.
6. add 'options(stringsAsFactors = FALSE)' to the beginning of your R code! It will save you tons of jobs when you dealing with some character variables.
7. use ave function to cunsum values by ID. It will return a vector of the same length as ID. We can use aggregate for similiar purpose, but all values are put together for each ID. For example, ave(1:3, c(1,1,2), FUN = mean) will calculate the mean for the first two and the last one respectively, and return c(1.5, 1.5, 3)
8. use scan to read data fast! It will return a vector or list, and it needs to be convert to data.frame or matrix mannually.
9. 'pmin/pmax' returns the parallel minimum (maximum) of the input values.

min(5:1, pi) #-> one number

pmin(5:1, pi) #-> 5 numbers

1. *years, months, days, hours,…..*in the “lubridate” package can be easily used to add years, months, days, and hours…… to a time, e.g.,

head(dd.cmip$Date)

[1] "1950-01-01" "1950-01-02" "1950-01-03" "1950-01-04" "1950-01-05" "1950-01-06"

head(dd.cmip$Date) + years(72)

[1] "2022-01-01" "2022-01-02" "2022-01-03" "2022-01-04" "2022-01-05" "2022-01-06"

head(dd.cmip$Date) + days(3)

[1] "1950-01-04" "1950-01-05" "1950-01-06" "1950-01-07" "1950-01-08" "1950-01-09"

1. **The following function enables the use of string to group: group\_by\_**

**group\_by\_(**dt.D\_RD, .dots = setdiff(colnames(dt.info), 'POP')) %>%

summarise\_each(funs(sum))

1. How to install and configure java, the dependent of ‘rJava’ package:

sudo apt-get install openjdk-7-\*

sudo R CMD javareconf

install.packages("rJava")

1. Search for the largest palindrome, which is a product of three digit numbers:

# Method 1

system.time({

mat <- expand.grid(x1 = 100:999, x2 = 100:999)

prod = mat$x1 \* mat$x2 # all posibile values, product of x1 and x2

prod = sort(unique(prod), decreasing = TRUE) # sort from biggest to smallest

reverse.str <- function(s) {

paste0(rev(unlist(strsplit(as.character(s), NULL))), collapse = '')

}

for (n in prod) {

if (n == reverse.str(n)) {

print(n)

break

}

}

})

# Method 2

system.time({

x1 = 1000

x2 = 9999

reverse.str <- function(s) {

paste0(rev(unlist(strsplit(as.character(s), NULL))), collapse = '')

}

values = (x1\*x1):(x2\*x2)

for (n in rev(values)) {

low = max(ceiling(n/x2), x1)

high = min(x2, floor(n/x1))

if (any(n %% low:high == 0)) {

if (n == reverse.str(n)) {

print(n)

break

}

}

}

})

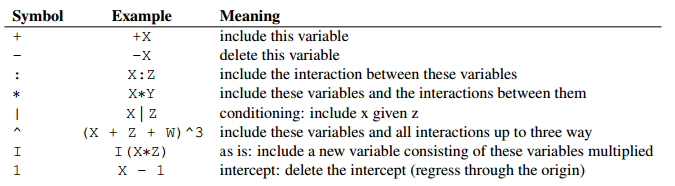
1. Use SSL for HTTPS website:

<https://s3.amazonaws.com/rstudio-server/rstudio-server-pro-0.98.490-admin-guide.pdf>

Here is the instruction to get a free SSL certificate from startSSL

<https://konklone.com/post/switch-to-https-now-for-free>

1. Regression formula symbols:



1. Stepwise regression in R – Critical p-value (alpha to enter and alpha to leave)  
   <http://stats.stackexchange.com/questions/97257/stepwise-regression-in-r-critical-p-value>

To control the p-value, change the k parameter in step from 2 to something else.

You want 10% instead? Make it 2.7:

qchisq(0.10,1,lower.tail=FALSE)

[1] 2.705543

You want 2.5%? Set k=5:

qchisq(0.025,1,lower.tail=FALSE)

[1] 5.023886

1. This website contains some useful details about various modeling in R:

<http://www.uni-kiel.de/psychologie/rexrepos/posts/survivalCoxPH.html>

1. CRAN Task View: Survival Analysis (comprehensive list of survival analysis packages):

<https://cran.r-project.org/web/views/Survival.html>

1. Detailed example of the survival modeling, with ROC/AUC functions:

<https://rstudio-pubs-static.s3.amazonaws.com/3506_36a9509e9d544386bd3e69de30bca608.html>

1. as.data.frame.matrix(tb) can convert a table to matrix as it is:

0 1

[3,3.5] 9582 3846

(3.5,3.75] 7130 1785

(3.75,3.92] 7002 1121

(3.92,4] 3928 159

1. This website gives a good explanation about ROC and AUC: <http://thestatsgeek.com/2014/05/05/area-under-the-roc-curve-assessing-discrimination-in-logistic-regression/>
2. Information value and WOE packages:

* Information: <https://cran.r-project.org/web/packages/Information/vignettes/Information-vignette.html> (This link provide very useful information and illustration)
* woe: <https://cran.r-project.org/web/packages/woe/woe.pdf>
* woe (another woe package, developed in Github): <https://github.com/tomasgreif/woe>
* klaR: <https://cran.r-project.org/web/packages/klaR/klaR.pdf>
* infotheo: <https://cran.r-project.org/web/packages/infotheo/infotheo.pdf>
* smbinning: <https://cran.r-project.org/web/packages/smbinning/smbinning.pdf>

1. select(data, one\_of(‘a’, ‘b’, ‘c’)) can be used to select the ‘a’, ‘b’, and ‘c’ columns from data. This is very useful to select the variables based on the quoted variable names
2. How to develop R package: <http://r-pkgs.had.co.nz/>
3. rpivotTable can be used to generate the pivot table interactively.
4. Connect R to SQL Server 2012: <https://andersspur.wordpress.com/2013/11/26/connect-r-to-sql-server-2012-and-14/>
5. Use saveRDS to write a single R object, so that it can be renamed:

x <- 5

saveRDS(x, "x.rds")

y <- readRDS("x.rds")

all.equal(x, y) # return TRUE

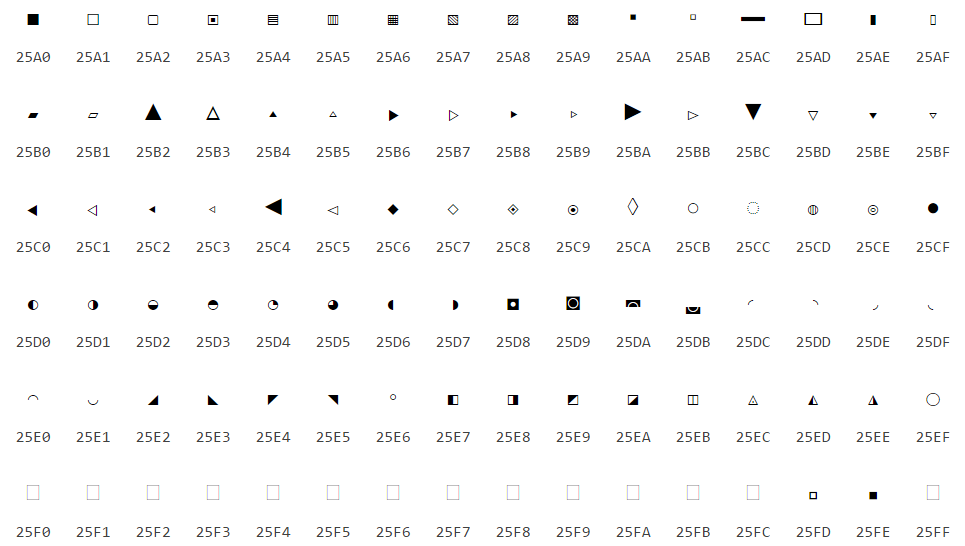
1. Use show.legend = F to turn off the legend for some geom shape:

*geom\_point(data = pr0\_7.max, size = 4, alpha = .9, pch = 16, show.legend = F)*

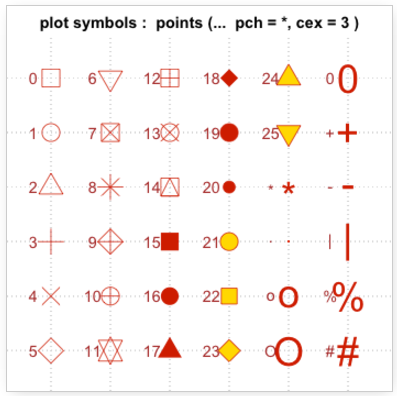
1. Unicode used for ggplot2 point shape (pch): <http://jrgraphix.net/r/Unicode/25A0-25FF>

Add ‘\u’ to the beginning when using these codes (e.g.):

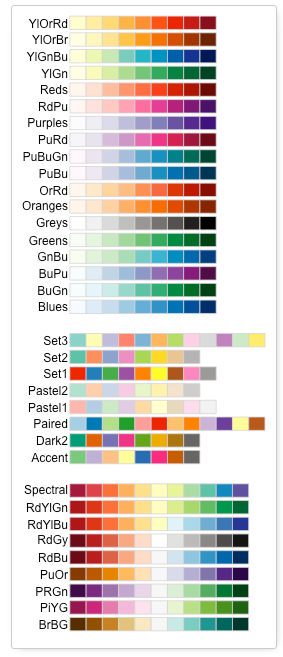
geom\_point(data = pr0\_7.max, size = 5, alpha = .9, pch = '\u25B2', show.legend = F)



The default pch symbols:



1. Color brewer palette:



1. Calculated the number of months between two date:

LOS <- as.period(interval(ASD, Graduation\_Date), 'month')@month + 1

1. predict.glm type argument:
   * link: the linear prediction for logit: log(p/(1 – p))

Two methods can be used to calculate the link:

* predict(mod, type = ‘link’)
* mod$linear.predictors
  + response: predicted probability: p

Two methods can be used to extract response:

* predict(mod, type = ‘response’)
* mod$fitted.values
  + terms: the linear prediction for each term, which is the coefficients times the centered variable value: <http://stats.stackexchange.com/questions/35682/contribution-of-each-covariate-to-a-single-prediction-in-a-logistic-regression-m>
* predict(mod, type = ‘terms’)

The output of this should be equal to (dt$var - mean(dt$var)) \* coef(dt)['var']

1. Two ways to build the training model:
   * mod.train <- coxph(model$formula, data = train)
   * mod.train <- update(model, data = train)

The second method is preferred, because it doesn’t need to call the modeling method (coxph) again. With the update function, we can update any model without calling the method explicitly.

1. Extract variables from formula:
   * Extract all variables: all.vars(model$formula)
   * Extract only x variables: labels(terms(model))
   * The str(x) function is useful to check the available information that can be extracted directly
2. Add one year/month to a date object:
   * as.Date('2016-05-31') + years(1) # return 2017-05-31
   * as.Date('2016-05-31') %m+% months(1) # return 2016-06-30
3. This link shows very detailed introduction about ggplot2: <https://www.ling.upenn.edu/~joseff/rstudy/summer2010_ggplot2_intro.html>

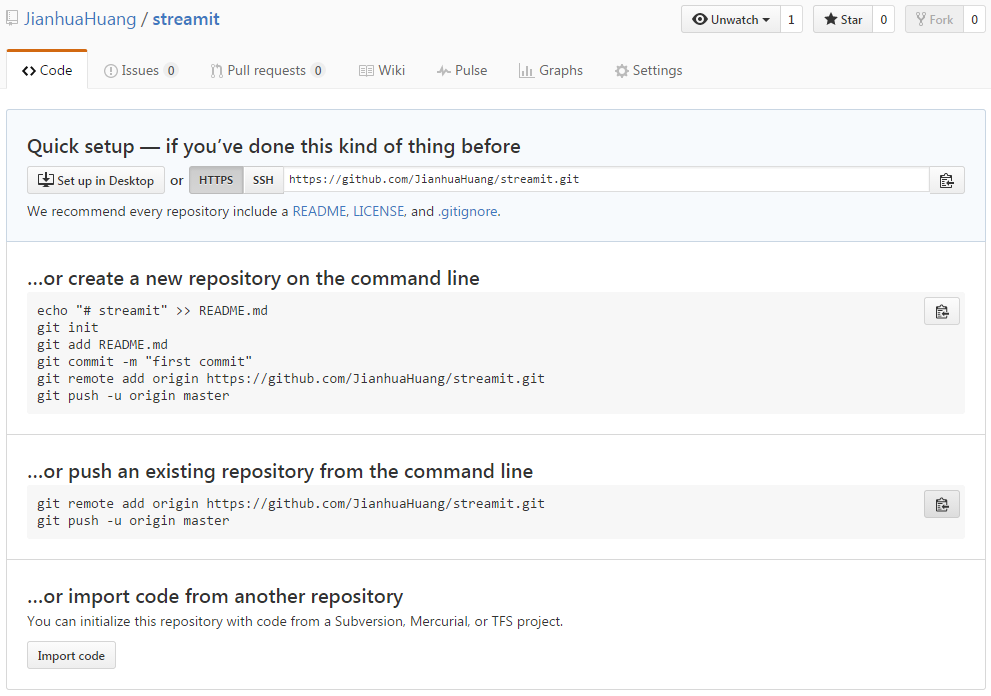
Geoms also have default statistics associated with them. For geom\_point(), the default statistic isstat\_identity(), but we could also change that.

## These should produce equivalent plots.  
p + geom\_point(stat = "smooth")  
p + stat\_smooth(geom = "point")

1. How to connect Rstudio with Git and Github for package development: <http://r-pkgs.had.co.nz/git.html>
2. How to setup remote Github repo with Rstudio: <http://r-pkgs.had.co.nz/git.html>
   * Generate a SSH key from Rstudio 🡪 Global Options 🡪 Git/SVN 🡪 Create RSA Key. Click ‘View public key’, and copy & paste it to

<https://github.com/settings/ssh>.

* + Setup a new repo on Github, then it will show this page



* + Initiate a new project (R package) with the same name as the Github repo. In the Tool 🡪 Shell, config the user.name and user.email for your Github account:
* git config --global user.name “JianhuaHuang”
* git config --global user.email [jh.eco.cas@gmail.com](mailto:jh.eco.cas@gmail.com)
  + Copy and run the following information from the Github page into the Shell:
* echo "# streamit" >> README.md
* git init
* git add README.md
* git commit -m "first commit"
* git remote add origin https://github.com/JianhuaHuang/streamit.git
* git push -u origin master
  + Then, when you work on another computer, you can create an R project with the Version Control option, to pull the repo from Github.
  + After this, you can use the commit and push/pull for Github, but it may requires user.name and password every time. To overcome this, change the http url to ssh url instead. reference to: <http://stackoverflow.com/questions/8588768/git-push-username-password-how-to-avoid>
* git remote set-url origin git+ssh://git@github.com/username/reponame.git

1. R package metadata (e.g., dependencies): <http://r-pkgs.had.co.nz/description.html>
2. Difference between suggests, imports, and depends to load/attach dependent packages in the DESCRIPTION file: <http://r-pkgs.had.co.nz/namespace.html#namespace>

* Suggests: the packages suggests to installed
* Imports: the packages must installed
* Depends: the packages must be installed and attached to the environment (equivalent to using the library function)
* Using imports and call the package name explicitly before the function, is equivalent to using depends in the DESCRIPTION file

1. Use NAMESPACE to import functions/methods from other packages, or export functions/methods, so that other package can use them. <http://r-pkgs.had.co.nz/namespace.html#namespace>
2. Two different ways to adjust the text position:

* **nudge\_x/nudge\_y** : Horizontal and vertical adjustment to nudge labels by. Useful for offsetting text from points, particularly on discrete scales. (This function can extend the plot boarder automatically to reserve enough space for the text)
* **hjust/vjust**: can be used to adjust text alignment (The text may be cut off, if it is out of the border)

1. Some of the website used to hold R package are blocked by the UOP, resulting in the failure of installing R packages. Choose other mirrors that are still usable with the following function, then install.package
   * chooseCRANmirror()
   * install.packages()