[jyc@localhost compiled]$ df -h

Filesystem Size Used Avail Use% Mounted on

/dev/mapper/VolGroup-lv\_root

11G 6.8G 3.0G 70% /

tmpfs 250M 0 250M 0% /dev/shm

/dev/sda1 477M 62M 391M 14% /boot

[jyc@localhost ~]$ yum info R

Loaded plugins: fastestmirror

Determining fastest mirrors

epel/metalink | 11 kB 00:00

\* base: mirrors.cmich.edu

\* epel: ftp.osuosl.org

\* extras: distro.ibiblio.org

\* updates: ftp.usf.edu

base | 3.7 kB 00:00

epel | 4.3 kB 00:00

epel/primary\_db | 5.0 MB 00:11

extras | 3.3 kB 00:00

updates | 3.4 kB 00:00

Installed Packages

Name : R

Arch : i686

Version : 3.3.2

Release : 3.el6

Size : 0.0

Repo : installed

From repo : epel

Summary : A language for data analysis and graphics

URL : http://www.r-project.org

License : GPLv2+

Description : This is a metapackage that provides both core R userspace and

: all R development components.

:

: R is a language and environment for statistical computing and graphics.

: R is similar to the award-winning S system, which was developed at

: Bell Laboratories by John Chambers et al. It provides a wide

: variety of statistical and graphical techniques (linear and

: nonlinear modelling, statistical tests, time series analysis,

: classification, clustering, ...).

:

: R is designed as a true computer language with control-flow

: constructions for iteration and alternation, and it allows users to

: add additional functionality by defining new functions. For

: computationally intensive tasks, C, C++ and Fortran code can be linked

: and called at run time.

[jyc@localhost ~]$ df

Filesystem 1K-blocks Used Available Use% Mounted on

/dev/mapper/VolGroup-lv\_root

10714028 7100080 3063036 70% /

tmpfs 255368 0 255368 0% /dev/shm

/dev/sda1 487652 62480 399572 14% /boot

[jyc@localhost ~]$ df -h

Filesystem Size Used Avail Use% Mounted on

/dev/mapper/VolGroup-lv\_root

11G 6.8G 3.0G 70% /

tmpfs 250M 0 250M 0% /dev/shm

/dev/sda1 477M 62M 391M 14% /boot

[jyc@localhost ~]$ df -i

Filesystem Inodes IUsed IFree IUse% Mounted on

/dev/mapper/VolGroup-lv\_root

689520 177257 512263 26% /

tmpfs 63842 1 63841 1% /dev/shm

/dev/sda1 128016 51 127965 1% /boot

[jyc@localhost ~]$ sudo du -sh /\*

6.0M /bin

59M /boot

156K /dev

29M /etc

604M /home

238M /lib

16K /lost+found

8.0K /media

8.0K /mnt

4.0K /opt

du: cannot access `/proc/15667/task/15667/fd/4': No such file or directory

du: cannot access `/proc/15667/task/15667/fdinfo/4': No such file or directory

du: cannot access `/proc/15667/fd/4': No such file or directory

du: cannot access `/proc/15667/fdinfo/4': No such file or directory

0 /proc

116K /root

15M /sbin

0 /selinux

4.0K /srv

0 /sys

148K /tmp

[jyc@localhost ~]$ du . | sort -nr | head -n30

617732 .

142676 ./compiled

142644 ./compiled/coreutils

91940 ./compiled/coreutils/coreutils-8.26

28160 ./gnutools

27668 ./compiled/coreutils/coreutils-8.26/po

26776 ./compiled/coreutils/coreutils-8.26/src

17652 ./gnutools/bin

13140 ./compiled/coreutils/coreutils-8.26/lib

10484 ./gnutools/share

9124 ./gnutools/share/locale

7032 ./downloads

4820 ./compiled/coreutils/coreutils-8.26/gnulib-tests

3640 ./compiled/coreutils/coreutils-8.26/tests

2408 ./compiled/coreutils/coreutils-8.26/m4

1636 ./compiled/coreutils/coreutils-8.26/src/.deps

1632 ./compiled/coreutils/coreutils-8.26/lib/.deps

1616 ./compiled/coreutils/coreutils-8.26/doc

1196 ./compiled/coreutils/coreutils-8.26/gnulib-tests/.deps

1136 ./compiled/coreutils/coreutils-8.26/old

1128 ./compiled/coreutils/coreutils-8.26/tests/misc

940 ./compiled/coreutils/coreutils-8.26/man

876 ./gnutools/share/info

740 ./compiled/coreutils/coreutils-8.26/build-aux

708 ./compiled/coreutils/coreutils-8.26/tests/pr

552 ./compiled/coreutils/coreutils-8.26/old/fileutils

508 ./tmp

480 ./gnutools/share/man

476 ./gnutools/share/man/man1

468 ./gnutools/share/locale/ru

timedatectl

timedatectl status

chronyc tracking

chronyc -a 'burst 4/4'

chronyc -a makestep

chronyc tracking

timedatectl

<http://stackoverflow.com/questions/37769985/how-to-install-r-on-centos>

wget https://cran.r-project.org/src/base/R-2/R-2.15.3.tar.gz

tar zxvf R-2.15.3.tar.gz; cd R-2.15.3/

./configure; make; sudo make install

yum groupinstall "Development Tools"

yum install ncurses-devel zlib-devel texinfo gtk+-devel gtk2-devel qt-devel tcl-devel tk-devel kernel-headers kernel-devel

./configure --with-x=no

make

make install

Thanks to @resscova's answer and some research on the net. Here's one way install R-2.X.tar.gz on Centos:

yum groupinstall "Development Tools"

yum install ncurses-devel zlib-devel texinfo gtk+-devel gtk2-devel qt-devel tcl-devel tk-devel kernel-headers kernel-devel

./configure --with-x=no

make

make install

sudo yum update

ysudo yum install epel-release

sudo yum install R

[jyc@q2-dev01 etc]$ yum list R-\\*

Loaded plugins: fastestmirror

Determining fastest mirrors

\* base: mirror.scalabledns.com

\* epel: mirrors.cat.pdx.edu

\* extras: mirror.supremebytes.com

\* updates: mirrors.unifiedlayer.com

Installed Packages

R.x86\_64 3.3.2-3.el7 @epel

R-core.x86\_64 3.3.2-3.el7 @epel

R-core-devel.x86\_64 3.3.2-3.el7 @epel

R-devel.x86\_64 3.3.2-3.el7 @epel

R-java.x86\_64 3.3.2-3.el7 @epel

R-java-devel.x86\_64 3.3.2-3.el7 @epel

Available Packages

R-RInside.x86\_64 0.2.13-3.el7 epel

R-RInside-devel.x86\_64 0.2.13-3.el7 epel

R-RInside-examples.x86\_64 0.2.13-3.el7 epel

R-RUnit.noarch 0.4.26-7.el7 epel

R-Rcpp.x86\_64 0.12.9-1.el7 epel

R-Rcpp-devel.x86\_64 0.12.9-1.el7 epel

R-Rcpp-examples.x86\_64 0.12.9-1.el7 epel

R-highlight.x86\_64 0.4.7-1.el7 epel

R-inline.noarch 0.3.14-1.el7 epel

R-littler.x86\_64 0.3.1-1.el7 epel

R-littler-examples.x86\_64 0.3.1-1.el7 epel

R-qtl.x86\_64 1.40.8-1.el7 epel

R-rlecuyer.x86\_64

mkdir $HOME/downloads ; cd $HOME/downloands

sudo yum install php gcc glibc glibc-common gd gd-devel make wget man

sudo wget <https://download2.rstudio.org/rstudio-server-rhel-1.0.44-x86_64.rpm> --no-check-certificate

cd

sudo yum install –nogpgcheck $HOME/downloands/rstudio-server-rhel-1.0.44-x86\_64.rpm

systemctl status rstudio-server.service

2017-02-26 16:56:47 (1.01 MB/s) - ‘rstudio-server-rhel-1.0.44-x86\_64.rpm’ saved [41364624/41364624]

[jyc@q2-dev01 ~]$ sudo yum install –nogpgcheck $HOME/downloands/rstudio-server-rhel-1.0.44-x86\_64.rpm

Loaded plugins: fastestmirror

Loading mirror speeds from cached hostfile

\* base: mirror.scalabledns.com

\* epel: mirrors.cat.pdx.edu

\* extras: mirror.supremebytes.com

\* updates: mirrors.lga7.us.voxel.net

https://mirrors.cat.pdx.edu/epel/7/x86\_64/repodata/678cb3658a8ba4348162453f3edf2023851804b6cd97b719e45d92aff31d3534-filelists.sqlite.xz: [Errno 14] HTTPS Error 404 - Not Found

Trying other mirror.

To address this issue please refer to the below knowledge base article

https://access.redhat.com/articles/1320623

If above article doesn't help to resolve this issue please create a bug on https://bugs.centos.org/

https://mirrors.xmission.com/fedora-epel/7/x86\_64/repodata/678cb3658a8ba4348162453f3edf2023851804b6cd97b719e45d92aff31d3534-filelists.sqlite.xz: [Errno 14] HTTPS Error 404 - Not Found

Trying other mirror.

epel/x86\_64/filelists\_db | 7.6 MB 00:00:07

extras/7/x86\_64/filelists\_db | 423 kB 00:00:00

updates/7/x86\_64/filelists\_db | 1.7 MB 00:00:02

No package /home/jyc/downloands/rstudio-server-rhel-1.0.44-x86\_64.rpm available.

Error: Nothing to do

[jyc@q2-dev01 usr]$ cd /usr/lib64/R

[jyc@q2-dev01 R]$ ls -l

total 32

drwxr-xr-x. 3 root root 4096 Feb 26 16:12 bin

-rw-r--r--. 1 root root 18011 Dec 15 06:59 COPYING

drwxr-xr-x. 2 root root 89 Feb 26 16:13 etc

lrwxrwxrwx. 1 root root 15 Feb 26 16:13 include -> ../../include/R

drwxr-xr-x. 2 root root 83 Feb 26 16:12 lib

drwxr-xr-x. 32 root root 4096 Feb 26 16:12 library

drwxr-xr-x. 2 root root 73 Feb 26 16:12 modules

-rw-r--r--. 1 root root 46 Dec 15 06:59 SVN-REVISION

[jyc@q2-dev01 bin]$ pwd

/usr/lib64/R/bin

[jyc@q2-dev01 bin]$ ls -l

total 480

-rwxr-xr-x. 1 root root 1828 Dec 15 06:59 BATCH

-rwxr-xr-x. 1 root root 317 Dec 15 06:59 build

-rwxr-xr-x. 1 root root 317 Dec 15 06:59 check

-rwxr-xr-x. 1 root root 2156 Dec 15 06:59 COMPILE

-rwxr-xr-x. 1 root root 9712 Dec 15 06:59 config

drwxr-xr-x. 2 root root 15 Feb 26 16:12 exec

-rwxr-xr-x. 1 root root 3607 Dec 15 06:59 f77\_f2c

-rwxr-xr-x. 1 root root 823 Dec 15 06:59 INSTALL

-rwxr-xr-x. 1 root root 14603 Dec 15 06:59 javareconf

-rwxr-xr-x. 1 root root 355433 Dec 15 06:59 libtool

-rwxr-xr-x. 1 root root 1433 Dec 15 06:59 LINK

-rwxr-xr-x. 1 root root 3494 Dec 15 06:59 mkinstalldirs

-rwxr-xr-x. 1 root root 481 Dec 15 06:59 pager

-rwxr-xr-x. 1 root root 8712 Dec 15 06:59 R

-rwxr-xr-x. 1 root root 1373 Dec 15 06:59 Rcmd

-rwxr-xr-x. 1 root root 312 Dec 15 06:59 Rd2pdf

-rwxr-xr-x. 1 root root 359 Dec 15 06:59 Rdconv

-rwxr-xr-x. 1 root root 258 Dec 15 06:59 Rdiff

-rwxr-xr-x. 1 root root 158 Dec 15 06:59 REMOVE

-rwxr-xr-x. 1 root root 310 Dec 15 06:59 Rprof

-rwxr-xr-x. 1 root root 11376 Dec 15 07:00 Rscript

-rwxr-xr-x. 1 root root 4121 Dec 15 06:59 rtags

-rwxr-xr-x. 1 root root 155 Dec 15 06:59 SHLIB

-rwxr-xr-x. 1 root root 318 Dec 15 06:59 Stangle

-rwxr-xr-x. 1 root root 316 Dec 15 06:59 Sweave

[jyc@q2-dev01 exec]$ locate R-FAQ.html

/usr/share/doc/R-3.3.2/manual/R-FAQ.html

|  |  |
| --- | --- |
| CentOS6 | CentOS7 |
| /usr/lib/python2.6/site-package | /usr/lib/python2.7/site-package |
| chkconfig --list ntpd  ntpstat  --if time is unsynchronized --do the following:  sudo ntpdate -u <time server from your config>  sudo ntpdate -u 1.centos.pool.ntp.org  sudo service ntpd restart  date | timedatectl  timedatectl status  --if time is unsynchronized --do the following:  chronyc tracking  chronyc -a 'burst 4/4'  chronyc -a makestep  chronyc tracking  timedatectl |
| sudo yum update  ysudo yum install epel-release  sudo yum install R | sudo yum update  ysudo yum install epel-release  sudo yum install R |
| sudo –i R  > install.packages(“quantmod”)  > library('quantmod')  Loading required package: xts  Loading required package: zoo  Attaching package: ‘zoo’  The following objects are masked from ‘package:base’:  as.Date, as.Date.numeric  Loading required package: TTR  Version 0.4-0 included new data defaults. See ?getSymbols.  > data <-new.env()  > getSymbols('RHT',data)  As of 0.4-0, ‘getSymbols’ uses env=parent.frame() and  auto.assign=TRUE by default.  This behavior will be phased out in 0.5-0 when the call will  default to use auto.assign=FALSE. getOption("getSymbols.env") and  getOptions("getSymbols.auto.assign") are now checked for alternate defaults  This message is shown once per session and may be disabled by setting  options("getSymbols.warning4.0"=FALSE). See ?getSymbols for more details.  [1] "RHT"  > plot(data$RHT)  Warning message:  In plot.xts(data$RHT) : only the univariate series will be plotted  > | sudo yum groupinstall “Development Tools”  R  demo() = for some demos,  R –help or help() = for on-line help, or  help.start() = for an HTML browser interface to help.  q() = to quit R.  sudo –i R  install.packages(“quantmod”)    \*\* building package indices  \*\* testing if installed package can be loaded  \* DONE (quantmod)  Making 'packages.html' ... done  The downloaded source packages are in  ‘/tmp/RtmpdJWZUO/downloaded\_packages’  Updating HTML index of packages in '.Library'  Making 'packages.html' ... done |
| [jyc@localhost ~]$ locate packages.html  /usr/lib/R/library/base/html/zpackages.html  /usr/lib/R/library/utils/html/available.packages.html  /usr/lib/R/library/utils/html/download.packages.html  /usr/lib/R/library/utils/html/install.packages.html  /usr/lib/R/library/utils/html/installed.packages.html  /usr/lib/R/library/utils/html/make.packages.html.html  /usr/lib/R/library/utils/html/remove.packages.html  /usr/lib/R/library/utils/html/update.packages.html  /usr/share/doc/R-3.3.2/html/packages.html  /usr/share/doc/gettext-devel-0.17/javadoc2/packages.html  /usr/share/doc/rsyslog-5.8.10/rsyslog\_packages.html | [jyc@q2-dev01 ~]$ locate packages.html  /usr/lib64/R/library/base/html/zpackages.html  /usr/lib64/R/library/utils/html/available.packages.html  /usr/lib64/R/library/utils/html/download.packages.html  /usr/lib64/R/library/utils/html/install.packages.html  /usr/lib64/R/library/utils/html/installed.packages.html  /usr/lib64/R/library/utils/html/make.packages.html.html  /usr/lib64/R/library/utils/html/remove.packages.html  /usr/lib64/R/library/utils/html/update.packages.html  /usr/share/doc/R-3.3.2/html/packages.html |

pscp.exe [jyc@ulc-178.ulcert.uw.edu:/usr/share/doc/R-3.3.2/html/packages.html](mailto:jyc@ulc-178.ulcert.uw.edu:/usr/share/doc/R-3.3.2/html/packages.html) C:\\_UWclass\Unix\unix105\Week 7>

/tmp/RtmpdJWZUO/

The assignment operator in R is <- as in

e <- m\*c^2.

It is also possible, though uncommon, to reverse the arrow and put the receiving variable on the right, as in

m\*c^2 -> e.

**Variable name gotchas**

Because the underscore was not allowed as a variable character, the convention arose to use dot as a name separator.

Unlike its use in many object oriented languages, the dot character in R has no special significance, with two exceptions. First, the ls() function in R lists active variables

R has several one-letter reserved words: c, q, s, t, C, D, F, I, and T.

(Actually, these are not reserved, but it's best to think of them as reserved. For example, c is a built-in function for creating vectors, though you could also create a variable named c. Worse, T and F are not synonyms for TRUE and FALSE but variables that have the expected values by default. So someone could include the code T <- FALSE; F <- TRUE and reverse their meanings!)

## Vectors

The primary data type in R is the vector.

x\*y

log(x)

p <- c(2,3,5,7)

Elements of a vector can be accessed using []. So in the above example, p[3] is 5.

> x <- c(1,2,3)  
> names(x) <- c('a','b','c')  
> x['c']  
c  
3

x[x>3] is the subset of x consisting of elements larger than 3, i.e. x[x>3] equals (4, 5, 9).

When a vector with a Boolean subscript appears in an assignment, the assignment applies to the elements that would have been extracted if there had been no assignment. For example, x[x > 3] <- 7 turns (3, 1, 4, 1, 5, 9) into (3, 1, 7, 1, 7, 7). Also, x[x > 3] <- c(10, 11, 12) would produce (3, 1, 10, 1, 11, 12)

In R, a negative is an instruction to remove an element from a vector. So y = x[-2] would set y equal to the vector (3, 4, 1, 5, 9), i.e. the vector x with the element x[2] removed.

a[0] It is silently ignored

> a

[1] 10 20 30 40 50

> a[0]

numeric(0)

> a[c(4,2)]

[1] 40 20

> a[c(4,0,2,0)]

[1] 40 20

> a[0] <- 7

[1] 10 20 30 40 50

> a[c(4,0,2,0)] <- 7

[1] 10 7 30 7 50

> a

[1] 10 20 30 40 50

> a[7]

[1] NA

> a[c(4,7,2)]

[1] 40 NA 20

> a[7] <- 7

[1] 10 20 30 40 50 NA 7

a<-c(NA,3)  
a[a==3]  
# the first element might well be 3, were it not missing!!!!  
# but if you want just 3’s, no NA’s:  
# and it’s not about subsetting, it’s about comparing with `==`

a[a %in% 3]

#or  
b <- na.omit(a)  
b[b==3]

# or even [very funny!!!]  
a[sapply(a, identical, 3)]

I almost always use %in% instead of == in functions

## Sequences

The expression seq(a, b, n) creates a closed interval from a to b in steps of size n. For example, seq(1, 10, 3) returns the vector containing 1, 4, 7, and 10. This is similar to range(a, b, n) in Python, except Python uses open intervals and so the 10 would not be included in this example. The step size argument n defaults to 1 in both R and Python.

The notation a:b is an abbreviation for seq(a, b, 1).

The notation seq(a, b, length=n) is a variation that will set the step size to (b-a)/(n-1) so that the sequence has n points.

## Types

The type of a vector is the type of the elements it contains and must be one of the following: logical, integer, double, complex, character, or raw. All elements of a vector must have the same underlying type. This restriction does not apply to lists.

Type conversion functions have the naming convention as.xxx for the function converts its argument to type xxx. For example, as.integer(3.2) returns the integer 3, and as.character(3.2) returns the string "3.2".

## Boolean operators

You can input T or TRUE for true values and F or FALSE for false values.

The operators & and | apply element-wise on vectors. The operators && and || are often used in conditional statements and use lazy evaluation as in C: the operators will not evaluate their second argument if the return value is determined by the first argument.

## Lists

Lists are like vectors, except elements need not all have the same type. For example, the first element of a list could be an integer and the second element be a string or a vector of Boolean values.

Lists are created using the list function. Elements can be access by position using [[]]. Named elements may be accessed either by position or by name.

Named elements of lists act like C structs, except a dollar sign rather than a dot is used to access elements. For example, consider,

a <- list(name="Joe", 4, foo=c(3,8,9))

Now a[[1]] and a$name both equal the string "Joe".

If you attempt to access a non-existent element of a list, say a[[4]] above, you will get an error. However, you can assign to a non-existent element of a list, thus extending the list. If the index you assign to is more than one past the end of the list, intermediate elements are created and assigned NULL values. You can also assign to non-existent named fields, such as saying a$baz = TRUE.

## Matrices

In a sense, R does not support matrices, only vectors. But you can change the dimension of a vector, essentially making it a matrix.

For example, m <- array( c(1,2,3,4,5,6), dim=c(2,3) ) creates a matrix m. However, it may come as a surprise that the first row of m has elements 1, 3, and 5. This is because by default, R fills matrices by column, like FORTRAN. To fill m by row, add the argument by.row = TRUE to the call to the array function.

## Missing values and NaNs

As in other programming languages, the result of an operation on numbers may return NaN, the symbol for "not a number." For example, an operation might overflow the finite range of a machine number, or a program might request an undefined operation, such as dividing by zero.

R also has a different type of non-number, NA for "not applicable." NA is used to indicate missing data, and is unfortunately fairly common in data sets. NA in R is similar to NULL in SQL or nullable types in C#. However, one must be more careful about NA values in R than about nulls in SQL or C#. The designer of database or the author of a piece of C# code specifies which values are nullable and can avoid the issue by simply not allowing such values. The author of an R function, however, has no control over the data his function will receive because NA is a legal value inside an R vector. There is no way to specify that a function takes only vectors with non-null components. You must handle NA values, even if you handle them by returning an error.

The function is.nan will return TRUE for those components of its argument that are NaN. The function is.na will return true for those components that are NA or NaN.

## Comments

Comments begin with # and continue to the end of the line, as in Python or Perl.

## Functions

The function definition syntax of R is similar to that of JavaScript. For example:

f <- function(a, b)

{

return (a+b)

}

The function function returns a function, which is usually assigned to a variable, f in this case, but need not be. You may use the function statement to create an anonymous function (lambda expression).

Note that return is a function; its argument must be contained in parentheses, unlike C where parentheses are optional. The use of return is optional; otherwise the value of the last line executed in a function is its return value.

Default values are defined similarly to C++. In the following example, b is set to 10 by default.

f <- function(a, b=10)

{

return (a+b)

}

So f(5, 1) would return 6, and f(5) would return 15. R allows more sophisticated default values than does C++. A default value in R need not be a static type but could, for example, be a function of other arguments.

C++ requires that if an argument has a default value then so do all values to the right. This is not the case in R, though it is still a good idea. The function definition

f <- function(a=10, b)

{

return (a+b)

}

is legal, but calling f(5) would cause an error. The argument a would be assigned 5, but no value would be assigned to b. The reason such a function definition is not illegal is that one could still call the function with one named argument. For example, f(b=2) would return 12.

Function arguments are passed by value. The most common mechanism for passing variables by reference is to use non-local variables. (Not necessarily global variables, but variables in the calling routine's scope.) A safer alternative is to explicitly pass in all needed values and return a list as output.

## Scope

R uses lexical scoping while S-PLUS uses static scope. The difference can be subtle, particularly when using closures.

Since variables cannot be declared — they pop into existence on first assignment — it is not always easy to determine the scope of a variable. You cannot tell just by looking at the source code of a function whether a variable is local to that function.

## Misc.

Here are a few miscellaneous facts about R that may be useful.

* help(fctn) displays help on any function fctn, as in Python.
* To invoke complex arithmetic, add 0i to a number. For example, sqrt(-1) returns NaN, but sqrt(-1 + 0i) returns 0 + 1i.
* sessionInfo() prints the R version, OS, packages loaded, etc.
* ls() shows which objects are defined.
* rm(list=ls()) clears all defined objects.
* dev.new() opens a new plotting window without overwriting the previous one.
* The function sort() does not change its argument.
* Distribution function prefixes d, p, q, r stand for density (PDF), probability (CDF), quantile (CDF-1), and random sample. For example, dnorm is the density function of a normal random variable and rnorm generates a sample from a normal random variable. The corresponding functions for a uniform random variable are dunif and runif.

whose arguments are a matrix X and a vector s of column indexes, and which returns a vector containing the Euclidean norm of each row of X, looking only at columns in s:

subset.norms <- function (X, s)

{ sqrt(apply(X[,s]^2,1,sum))

}

Here’s an example of its use:

> M

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

[1,] 1 4 7 10

[2,] 2 5 8 11

[3,] 3 6 9 12

> subset.norms(M,c(1,3))

[1] 7.071068 8.246211 9.486833

Perhaps it would be interesting to see how the norms get smaller as we drop leading dimensions:

> for (k in 1:4) print(subset.norms(M,k:4))

[1] 12.88410 14.62874 16.43168

[1] 12.84523 14.49138 16.15549

[1] 12.20656 13.60147 15.00000

Error in apply(X[, s]^2, 1, sum) : dim(X) must have a positive length

Oops… When the loops gets to where k has the value 4, the subset k:4 contains just one dimension. When R comes to evaluating X[,s], it doesn’t return a matrix with one column, but rather a vector. The apply function doesn’t work on vectors, so we get an error message rather than the answer.

There’s a fix. The “[” operator takes a “drop”argument, which defaults to TRUE, giving the behaviour above, but which can be explicitly set to FALSE to disable conversion from a matrix to a vector in these circumstances. If we modify the function as follows:

subset.norms2 <- function (X, s)

{ sqrt(apply(X[,s,drop=FALSE]^2,1,sum))

}

We get the right answers:

> for (k in 1:4) print(subset.norms2(M,k:4))

[1] 12.88410 14.62874 16.43168

[1] 12.84523 14.49138 16.15549

[1] 12.20656 13.60147 15.00000

[1] 10 11 12

There are two problems with this fix. One is that in some programs, one needs to put drop=FALSE in numerous places, making the code rather hard to read. The more serious problem, and what makes this a real design flaw, is that writing code without drop=FALSE is so much easier that it’s often left out even when it is needed. Indeed, since the errors typically occur only for extreme cases, it’s easy for the programmer to not realize there’s a problem. (As an aside, this is another context where the [reversing sequences problem](https://radfordneal.wordpress.com/2008/08/06/design-flaws-in-r-1-reversing-sequences) arises — if n is 0, subset.norms2(M,1:n) should produce all zeros, but doesn’t, not because of a bug in subset.norms2 but because 1:0 doesn’t produce the empty sequence.)

Can we solve the dimenion dropping problem by just changing the default for drop from TRUE to FALSE? Of course not, since this would break too many existing programs. But even if backward compatibility weren’t a problem, this wouldn’t be a good solution, since the times when we want drop to be TRUE are even more numerous than when we want it to be FALSE. Look at the following, for instance:

> M[2,3,drop=FALSE]

[,1]

[1,] 8

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

and a vector

v <- c(0,0,1,0)

example:

> x <- c(1,2,3)

> x

[1] 1 2 3

> x\*7

[1] 7 14 21

> x\*x

[1] 1 4 9

> x[x>2]

[1] 3

> y<-c(2,3,4)

> x\*y

[1] 2 6 12

> p <-x\*y

> p

[1] 2 6 12

> log(p)

[1] 0.6931472 1.7917595 2.4849066

> x <-c('a','b','c')

> x

[1] "a" "b" "c"

> seq(1,8,2)

> m<-array(c(1,2,3,4,5,6),dim=c(2,3))

> m

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

[1,] 1 3 5

[2,] 2 4 6

> m<-array(c(seq(1,6,1)),dim=c(2,3))

> m

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

[1,] 1 3 5

[2,] 2 4 6

R is case sensitive:

Load the data from package = use library() command

Load it in

library(lattice)

= takes all the function

library(dateset)

= load pre pack datesets

PlantGrowth dataset

str(PlantGrowth) = to see the data structure of PlantGrowth

weight and group vectors

groups has 2 groups=treatment and control group

use attach() to bring

attach(PlantGrown) to see the data frame

weight -> get the vector

summary(weight) – give you basic statistics about weight

summary(group) --

numerical vertor correspond to wright

weight~group = weight is modeling on group=> linear function

summary(weight~group)

treatment is any different

how the treatment

plot() – scatter plot

plot((weight~group) -- box plot

boxplot(weight~group)

plot(1:5,1:5) –create 2 vectors and return scatter plot

install.package(“lattice”) 🡪 sending text string to this

download the package

to run it type library(lattice) again

histogram(~weight|group) -> trying to predict weight

install.package(“ggplot2”)

library(ggplot2)

qplot()

qplot(weight,facets=.~group)

> getwd()

[1] "/home/jyc"

> setwd("/home/jyc/midterm")

> getwd()

[1] "/home/jyc/midterm"

>

> library(lattice)

> library(datasets)

> PlantGrowth

weight group

1 4.17 ctrl

2 5.58 ctrl

3 5.18 ctrl

4 6.11 ctrl

5 4.50 ctrl

6 4.61 ctrl

7 5.17 ctrl

8 4.53 ctrl

9 5.33 ctrl

10 5.14 ctrl

11 4.81 trt1

12 4.17 trt1

13 4.41 trt1

14 3.59 trt1

15 5.87 trt1

16 3.83 trt1

17 6.03 trt1

18 4.89 trt1

19 4.32 trt1

20 4.69 trt1

21 6.31 trt2

22 5.12 trt2

23 5.54 trt2

24 5.50 trt2

25 5.37 trt2

26 5.29 trt2

27 4.92 trt2

28 6.15 trt2

29 5.80 trt2

30 5.26 trt2

> str()

Error in str.default() : argument "object" is missing, with no default

> str(PlantGrowth)

'data.frame': 30 obs. of 2 variables:

$ weight: num 4.17 5.58 5.18 6.11 4.5 4.61 5.17 4.53 5.33 5.14 ...

$ group : Factor w/ 3 levels "ctrl","trt1",..: 1 1 1 1 1 1 1 1 1 1 ...

> attach(PlantGrowth)

> weight

[1] 4.17 5.58 5.18 6.11 4.50 4.61 5.17 4.53 5.33 5.14 4.81 4.17 4.41 3.59 5.87

[16] 3.83 6.03 4.89 4.32 4.69 6.31 5.12 5.54 5.50 5.37 5.29 4.92 6.15 5.80 5.26