STAT 812: Computational Statistics

Introduction to R Programming

Longhai Li

2024-09-10

Contents

1 Vector

```r
a <- 1

a
```

```
## [1] 1
```

```r
b <- 2:10

b
```

```
## [1]  2  3  4  5  6  7  8  9 10
```

```r
#concatenate two vectors
c <- c(a,b)

c
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```r
#vector arithmetics
1/c
```

```
##  [1] 1.0000000 0.5000000 0.3333333 0.2500000 0.2000000 0.1666667 0.1428571
##  [8] 0.1250000 0.1111111 0.1000000
```

```
c^2
```

```
##  [1]   1   4   9  16  25  36  49  64  81 100
```

```
c^2 + 1
```

```
##  [1]   2   5  10  17  26  37  50  65  82 101
```

```
#apply a function to each element
log(c)
```

```
##  [1] 0.0000000 0.6931472 1.0986123 1.3862944 1.6094379 1.7917595 1.9459101
##  [8] 2.0794415 2.1972246 2.3025851
```

```
sapply(c,log)
```

```
##  [1] 0.0000000 0.6931472 1.0986123 1.3862944 1.6094379 1.7917595 1.9459101
##  [8] 2.0794415 2.1972246 2.3025851
```

```
#operation on two vectors
d <- (1:10)*10

d
```

```
##  [1]  10  20  30  40  50  60  70  80  90 100
```

```
c + d
```

```
##  [1]  11  22  33  44  55  66  77  88  99 110
```

```
c * d
```

```
##  [1]   10   40   90  160  250  360  490  640  810 1000
```

```
d ^ c
```

```
##  [1] 1.000000e+01 4.000000e+02 2.700000e+04 2.560000e+06 3.125000e+08
##  [6] 4.665600e+10 8.235430e+12 1.677722e+15 3.874205e+17 1.000000e+20
```

```
#more concrete example: computing variance of 'c'
sum((c - mean(c))^2)/(length(c)-1)
```

```
## [1] 9.166667
```

```
#of course, there is build-in function for computing variance:
var(c)
```

```
## [1] 9.166667
```

```
#subsetting vector
c
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
c[2]
```

```
## [1] 2
```

```
c[c(2,3)]
```

```
## [1] 2 3
```

```
c[c(3,2)]
```

```
## [1] 3 2
```

```
c[c > 5]
```

```
## [1]  6  7  8  9 10
```

```
#let's see what is "c > 5"
c > 5
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
```

```
c[c > 5 & c < 10]
```

```
## [1] 6 7 8 9
```

```
c[as.logical((c > 8) + (c < 3))]
```

```
## [1]  1  2  9 10
```

```
log(c)
```

```
##  [1] 0.0000000 0.6931472 1.0986123 1.3862944 1.6094379 1.7917595 1.9459101
##  [8] 2.0794415 2.1972246 2.3025851
```

```
c[log(c) < 2]
```

```
## [1] 1 2 3 4 5 6 7
```

```
#modifying subset of vector
c[log(c) < 2] <- 3

c
```

```
##  [1]  3  3  3  3  3  3  3  8  9 10
```

```
#introduce a function ``seq''
seq(0,10,by=1)
```

```
##  [1]  0  1  2  3  4  5  6  7  8  9 10
```

```
seq(0,10,length=20)
```

```
##  [1]  0.0000000  0.5263158  1.0526316  1.5789474  2.1052632  2.6315789
##  [7]  3.1578947  3.6842105  4.2105263  4.7368421  5.2631579  5.7894737
## [13]  6.3157895  6.8421053  7.3684211  7.8947368  8.4210526  8.9473684
## [19]  9.4736842 10.0000000
```

```
1:10
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
#seq is more reliable than ":"
n <- 0

1:n
```

```
## [1] 1 0
```

```
#seq(1,n,by=1)
#Error in seq.default(1, n, by = 1) : wrong sign in 'by' argument
#Execution halted
```

```
#function ``rep''
c<- 1:5

c
```

```
## [1] 1 2 3 4 5
```

```
rep(c,5)
```

```
##  [1] 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
```

```
rep(c,each=5)
```

```
##  [1] 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4 5 5 5 5 5
```

2 Strings

```
A <- c("a","b","c")

A
```

```
## [1] "a" "b" "c"
```

```
paste("a","b",sep="")
```

```
## [1] "ab"
```

```
paste0("a","b",sep="")
```

```
## [1] "ab"
```

```
paste(A,c("d","e"))
```

```
## [1] "a d" "b e" "c d"
```

```
paste(A,10)
```

```
## [1] "a 10" "b 10" "c 10"
```

```
paste(A,10,sep="")
```

```
## [1] "a10" "b10" "c10"
```

```
paste0(A,10)
```

```
## [1] "a10" "b10" "c10"
```

```
paste0(A,1:10)
```

```
##  [1] "a1"  "b2"  "c3"  "a4"  "b5"  "c6"  "a7"  "b8"  "c9"  "a10"
```

```
sprintf("unit%g.pdf", 1:10)
```

```
##  [1] "unit1.pdf"  "unit2.pdf"  "unit3.pdf"  "unit4.pdf"  "unit5.pdf"
##  [6] "unit6.pdf"  "unit7.pdf"  "unit8.pdf"  "unit9.pdf"  "unit10.pdf"
```

```
sprintf("unit%g.%s", 1:10, "html")
```

```
##  [1] "unit1.html"  "unit2.html"  "unit3.html"  "unit4.html"  "unit5.html"
##  [6] "unit6.html"  "unit7.html"  "unit8.html"  "unit9.html"  "unit10.html"
```

```
filelist <- list.files(); filelist
```

```
##  [1] "afig.pdf"                         "alist.RDS"
##  [3] "cauchyNR.R"                       "css"
##  [5] "images"                           "js"
##  [7] "mixmodel.bug"                     "mixmodel.jags"
##  [9] "mtcars.csv"                       "mtcars.RData"
## [11] "mtcars2.csv"                      "normmodel.bug"
## [13] "nr-cauchy1.html"                  "nr-cauchy2.html"
## [15] "nr-cauchy3.html"                  "numbers.txt"
## [17] "oldRcode"                         "opsnr.stt"
## [19] "rdemo.Rproj"                      "regmodel.bug"
## [21] "s348.fld"                         "unit01_introduction_longer.html"
## [23] "unit01_introduction_longer.pdf"   "unit01_introduction_longer.Rmd"
## [25] "unit02_comparith.docx"            "unit02_comparith.html"
## [27] "unit02_comparith.pdf"             "unit02_comparith.Rmd"
## [29] "unit03_sampling_basics.html"      "unit03_sampling_basics.Rmd"
## [31] "unit04_simulation_cache"          "unit04_simulation_files"
## [33] "unit04_simulation.html"           "unit04_simulation.Rmd"
## [35] "unit05_MLE1_cache"                "unit05_MLE1_files"
## [37] "unit05_MLE1.html"                 "unit05_MLE1.Rmd"
## [39] "unit06_MLEm_cache"                "unit06_MLEm_files"
## [41] "unit06_MLEm.html"                 "unit06_MLEm.Rmd"
## [43] "unit07_em_cache"                  "unit07_em_files"
## [45] "unit07_em.html"                   "unit07_em.Rmd"
## [47] "unit08_integral.html"             "unit08_integral.pdf"
## [49] "unit08_integral.Rmd"              "unit09_laplace_cache"
## [51] "unit09_laplace.html"              "unit09_laplace.pdf"
## [53] "unit09_laplace.Rmd"               "unit10_rejection sampling.Rmd"
## [55] "unit10_rejection_sampling_cache"  "unit10_rejection_sampling_files"
## [57] "unit10_rejection_sampling.html"   "unit11_importance_sampling_cache"
## [59] "unit11_importance_sampling_files" "unit11_importance_sampling.html"
## [61] "unit11_importance_sampling.Rmd"   "unit11_mcmc_intro.html"
## [63] "unit11_mcmc_intro.Rmd"            "unit12_gibbs_cache"
## [65] "unit12_gibbs_files"               "unit12_gibbs.html"
## [67] "unit12_gibbs.Rmd"                 "unit14_MHsampling.html"
## [69] "unit14_MHsampling.Rmd"            "unit15_jags_cache"
## [71] "unit15_jags_files"                "unit15_jags.html"
## [73] "unit15_jags.Rmd"                  "unit16_stan_cache"
## [75] "unit16_stan_files"                "unit16_stan.html"
## [77] "unit16_stan.Rmd"                  "y.txt"
```

```
# selecting strings
filelist[grep( "*.pdf",filelist)]
```

```
## [1] "afig.pdf"                       "unit01_introduction_longer.pdf"
## [3] "unit02_comparith.pdf"           "unit08_integral.pdf"
## [5] "unit09_laplace.pdf"
```

```
filelist[grep( "*.Rmd",filelist)]
```

```
##  [1] "unit01_introduction_longer.Rmd" "unit02_comparith.Rmd"
##  [3] "unit03_sampling_basics.Rmd"     "unit04_simulation.Rmd"
##  [5] "unit05_MLE1.Rmd"                "unit06_MLEm.Rmd"
##  [7] "unit07_em.Rmd"                  "unit08_integral.Rmd"
##  [9] "unit09_laplace.Rmd"             "unit10_rejection sampling.Rmd"
## [11] "unit11_importance_sampling.Rmd" "unit11_mcmc_intro.Rmd"
## [13] "unit12_gibbs.Rmd"               "unit14_MHsampling.Rmd"
## [15] "unit15_jags.Rmd"                "unit16_stan.Rmd"
```

```
unit.files <- filelist[grep( "^unit",filelist)]; unit.files
```

```
##  [1] "unit01_introduction_longer.html"  "unit01_introduction_longer.pdf"
##  [3] "unit01_introduction_longer.Rmd"   "unit02_comparith.docx"
##  [5] "unit02_comparith.html"            "unit02_comparith.pdf"
##  [7] "unit02_comparith.Rmd"             "unit03_sampling_basics.html"
##  [9] "unit03_sampling_basics.Rmd"       "unit04_simulation_cache"
## [11] "unit04_simulation_files"          "unit04_simulation.html"
## [13] "unit04_simulation.Rmd"            "unit05_MLE1_cache"
## [15] "unit05_MLE1_files"                "unit05_MLE1.html"
## [17] "unit05_MLE1.Rmd"                  "unit06_MLEm_cache"
## [19] "unit06_MLEm_files"                "unit06_MLEm.html"
## [21] "unit06_MLEm.Rmd"                  "unit07_em_cache"
## [23] "unit07_em_files"                  "unit07_em.html"
## [25] "unit07_em.Rmd"                    "unit08_integral.html"
## [27] "unit08_integral.pdf"              "unit08_integral.Rmd"
## [29] "unit09_laplace_cache"             "unit09_laplace.html"
## [31] "unit09_laplace.pdf"               "unit09_laplace.Rmd"
## [33] "unit10_rejection sampling.Rmd"    "unit10_rejection_sampling_cache"
## [35] "unit10_rejection_sampling_files"  "unit10_rejection_sampling.html"
## [37] "unit11_importance_sampling_cache" "unit11_importance_sampling_files"
## [39] "unit11_importance_sampling.html"  "unit11_importance_sampling.Rmd"
## [41] "unit11_mcmc_intro.html"           "unit11_mcmc_intro.Rmd"
## [43] "unit12_gibbs_cache"               "unit12_gibbs_files"
## [45] "unit12_gibbs.html"                "unit12_gibbs.Rmd"
## [47] "unit14_MHsampling.html"           "unit14_MHsampling.Rmd"
## [49] "unit15_jags_cache"                "unit15_jags_files"
## [51] "unit15_jags.html"                 "unit15_jags.Rmd"
## [53] "unit16_stan_cache"                "unit16_stan_files"
## [55] "unit16_stan.html"                 "unit16_stan.Rmd"
```

```
unit.pdf.files <- unit.files[grep("*.pdf", unit.files)]; unit.pdf.files
```

```
## [1] "unit01_introduction_longer.pdf" "unit02_comparith.pdf"
## [3] "unit08_integral.pdf"            "unit09_laplace.pdf"
```

```
unit.html.files <- unit.files[grep("*.html", unit.files)]; unit.html.files
```

```
##  [1] "unit01_introduction_longer.html" "unit02_comparith.html"
##  [3] "unit03_sampling_basics.html"     "unit04_simulation.html"
##  [5] "unit05_MLE1.html"                "unit06_MLEm.html"
##  [7] "unit07_em.html"                  "unit08_integral.html"
##  [9] "unit09_laplace.html"             "unit10_rejection_sampling.html"
## [11] "unit11_importance_sampling.html" "unit11_mcmc_intro.html"
## [13] "unit12_gibbs.html"               "unit14_MHsampling.html"
## [15] "unit15_jags.html"                "unit16_stan.html"
```

3 Special Values

```
a <- 0/0

a
```

```
## [1] NaN
```

```
is.nan(a)
```

```
## [1] TRUE
```

```
b <- log(0)

b
```

```
## [1] -Inf
```

```
is.finite(b)
```

```
## [1] FALSE
```

```
c <- c(0:4,NA)

c
```

```
## [1]  0  1  2  3  4 NA
```

```
is.na(c)
```

```
## [1] FALSE FALSE FALSE FALSE FALSE  TRUE
```

4 Matrices

```
A <- matrix(0,4,5)

A
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    0    0    0    0
## [2,]    0    0    0    0    0
## [3,]    0    0    0    0    0
## [4,]    0    0    0    0    0
```

```
A <- matrix(1:20,4,5)

B <- matrix(1:20,4,5,byrow = T)



#subsectioning and modifying subsection

D <- A[c(1,4),c(2,3)]

A[c(1,4),c(2,3)] <- 1

A[c(1,4),c(2,3)] <- 101:104
```

```r
A[c(1,4),c(2,3)] <- matrix (1001:1004, 2,2)

a<- A[4,]

b<- A[3:4,]

A[1,1]
```

```
## [1] 1
```

```r
a2<- A[4,, drop = FALSE]

#combining two matrices

#create another matrix using another way
A2 <- array(1:20,dim=c(4,5))

A2
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    5    9   13   17
## [2,]    2    6   10   14   18
## [3,]    3    7   11   15   19
## [4,]    4    8   12   16   20
```

```r
cbind(A,A2)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]    1 1001 1003   13   17    1    5    9   13    17
## [2,]    2    6   10   14   18    2    6   10   14    18
## [3,]    3    7   11   15   19    3    7   11   15    19
## [4,]    4 1002 1004   16   20    4    8   12   16    20
```

```r
rbind(A,A2)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1 1001 1003   13   17
## [2,]    2    6   10   14   18
## [3,]    3    7   11   15   19
## [4,]    4 1002 1004   16   20
## [5,]    1    5    9   13   17
## [6,]    2    6   10   14   18
## [7,]    3    7   11   15   19
## [8,]    4    8   12   16   20
```

```r
#operating matrice

#transpose matrix
t(A)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3    4
## [2,] 1001    6    7 1002
## [3,] 1003   10   11 1004
## [4,]   13   14   15   16
## [5,]   17   18   19   20
```

```
A
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1 1001 1003   13   17
## [2,]    2    6   10   14   18
## [3,]    3    7   11   15   19
## [4,]    4 1002 1004   16   20
```

```
A + 1
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    2 1002 1004   14   18
## [2,]    3    7   11   15   19
## [3,]    4    8   12   16   20
## [4,]    5 1003 1005   17   21
```

```
x <- 1:5

A*x
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1 5005 4012   39   34
## [2,]    4    6   50   56   54
## [3,]    9   14   11   75   76
## [4,]   16 3006 2008   16  100
```

```
#the logical here is coercing the matrix "A" into a vector by joining the column
#and repeat the shorter vector,x, as many times as making it have the same
#length as the vector coerced from "A"

#see another example

x <- 1:3

A*x
```

```
## Warning in A * x: longer object length is not a multiple of shorter object
## length
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1 2002 3009   13   34
## [2,]    4   18   10   28   54
## [3,]    9    7   22   45   19
## [4,]    4 2004 3012   16   40
```

```
A^2
```

```
##      [,1]    [,2]    [,3] [,4] [,5]
## [1,]    1 1002001 1006009  169  289
## [2,]    4      36     100  196  324
## [3,]    9      49     121  225  361
## [4,]   16 1004004 1008016  256  400
```

```
A <- matrix(sample(1:20),4,5)

A
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1   19   13    6    7
## [2,]   18    5   11    2    4
## [3,]   10   12    9   16    3
## [4,]   15   20   14    8   17
```

```
B <- matrix(sample(1:20),5,4)

B
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    9   12    8   15
## [2,]   13    2   17   14
## [3,]   18    1    7   16
## [4,]    5    4    3   20
## [5,]   19    6   11   10
```

```
C <- A %*% B

C
```

```
##      [,1] [,2] [,3] [,4]
## [1,]  653  129  517  679
## [2,]  511  269  356  596
## [3,]  545  235  428  812
## [4,] 1010  368  769 1059
```

```
solve(C)
```

```
##              [,1]         [,2]         [,3]         [,4]
## [1,]  0.024307461  0.032853292 -0.012007448 -0.024868064
## [2,] -0.014870298 -0.008235801  0.001710190  0.012858163
## [3,] -0.030162383 -0.044228242  0.010357550  0.036288914
## [4,]  0.003887258  0.003645390  0.003336371 -0.006157917
```

```
#solving linear equation

x <- 1:4

d <- C %*% x

solve(C,d)
```

```
##      [,1]
## [1,]    1
## [2,]    2
## [3,]    3
## [4,]    4
```

```
#altenative way (but not recommended)
solve(C) %*% d
```

```
##      [,1]
## [1,]    1
## [2,]    2
## [3,]    3
## [4,]    4
```

```
#SVD (C = UDV') and determinant

svd.C <- svd(C)
svd.C
```

```
## $d
## [1] 2450.55274  176.42977  116.11196   11.82537
##
## $u
##            [,1]       [,2]       [,3]       [,4]
## [1,] -0.4392016  0.5475870 -0.5181754 -0.4886151
## [2,] -0.3666462 -0.3572732  0.5530395 -0.6573231
## [3,] -0.4424775 -0.6991172 -0.5324095  0.1788542
## [4,] -0.6905694  0.2893778  0.3770698  0.5451539
##
## $v
##            [,1]       [,2]       [,3]        [,4]
## [1,] -0.5765149  0.4889217  0.3006748 -0.58153311
## [2,] -0.2095024 -0.4719685  0.8230715  0.23644817
## [3,] -0.4399096  0.4490316 -0.0768158  0.77391832
## [4,] -0.6559107 -0.5801482 -0.4756546 -0.08343818
```

```
#calculating determinant of C

prod(svd.C$d)
```

```
## [1] 593646208
```

5 Data frame

```
name <- c("john","peter","jennifer")

gender <- factor(c("m","m","f"))

hw1 <- c(60,60,80)

hw2 <- c(40,50,30)

grades <- data.frame(name,gender,hw1,hw2)


grades[,"gender"]
```

```
## [1] m m f
## Levels: f m
```

```
grades[,2]
```

```
## [1] m m f
## Levels: f m
```

```
#subsectioning a data frame

grades[1,2]
```

```
## [1] m
## Levels: f m
```

```r
grades[,"name"]
```

```
## [1] "john"     "peter"    "jennifer"
```

```r
grades$name
```

```
## [1] "john"     "peter"    "jennifer"
```

```r
grades[grades$gender=="m",]
```

```
##    name gender hw1 hw2
## 1  john      m  60  40
## 2 peter      m  60  50
```

```r
subset (grades, hw1 >60)
```

```
##       name gender hw1 hw2
## 3 jennifer      f  80  30
```

```r
grades[,"hw1"]
```

```
## [1] 60 60 80
```

```r
#divide the subjects by "gender", and calculating means in each group
tapply(grades[,"hw1"], grades[,"gender"],mean)
```

```
##  f  m
## 80 60
```

6 List

```r
a <- 1:10

b <- matrix(1:10,2,5)

c <- c("name1","name2")

alst <- list(aa=a,b=b,c=c)

names (alst)
```

```
## [1] "aa" "b"  "c"
```

```r
str(alst)
```

```
## List of 3
##  $ aa: int [1:10] 1 2 3 4 5 6 7 8 9 10
##  $ b : int [1:2, 1:5] 1 2 3 4 5 6 7 8 9 10
##  $ c : chr [1:2] "name1" "name2"
```

```r
#refering to component of a list

alst$aa
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```r
alst[[2]]
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    3    5    7    9
## [2,]    2    4    6    8   10
```

```
blst <- list(d=2:10*10)

#concatenating list
ablst <- c(alst,blst)

ablst
```

```
## $aa
##  [1]  1  2  3  4  5  6  7  8  9 10
##
## $b
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    3    5    7    9
## [2,]    2    4    6    8   10
##
## $c
## [1] "name1" "name2"
##
## $d
## [1]  20  30  40  50  60  70  80  90 100
```

7 Reading and saving data to harddrive

```
a<- scan (text = "3 4 5.3 3")
numbers <- scan (file="numbers.txt")
mtcars <- read.csv ("mtcars.csv")

## save objects
save (mtcars, numbers, file = "mtcars.RData")
load ("mtcars.RData")
## note that load will override the objects with the same in .RData file

## output numbers to a text file
cat (numbers, file = "numbers.txt")

## save data frame as csv or other types of file
write.csv(mtcars, file = "mtcars2.csv")

## save an object into an RDS file

alist <- list (A = rnorm (100), B = letters[1:10])
saveRDS(alist, file = "alist.RDS")
blist <- readRDS("alist.RDS")
## note that a list is not erased by readRDS
identical(alist, blist)
```

```
## [1] TRUE
```

8 Function

```
#looking for the maximum value of a numeric vector x
find.max <- function(x)
{
    n <- length(x)

    x.m <- x[1]
    ix.m <- 1

    if(n > 1)
    {
        for( i in seq(2,n,by=1) )
        {
            if(x[i] > x.m)
            {
                x.m <- x[i]
                ix.m <- i
            }
        }
    }

    #return the maximum value and the index
    list(max=x.m,index.max=ix.m)
}

# To use this function

a <- rnorm (5); a
```

```
## [1] -0.2957290  1.0360246  0.4198477  0.8454541 -0.2753126
```

```
find.max(a)
```

```
## $max
## [1] 1.036025
##
## $index.max
## [1] 2
```

```
# Some relevant R built-in functions
max(a)
```

```
## [1] 1.036025
```

```
which.max(a)
```

```
## [1] 2
```

```
order (a)
```

```
## [1] 1 5 3 4 2
```

```
sort (a)
```

```
## [1] -0.2957290 -0.2753126  0.4198477  0.8454541  1.0360246
```

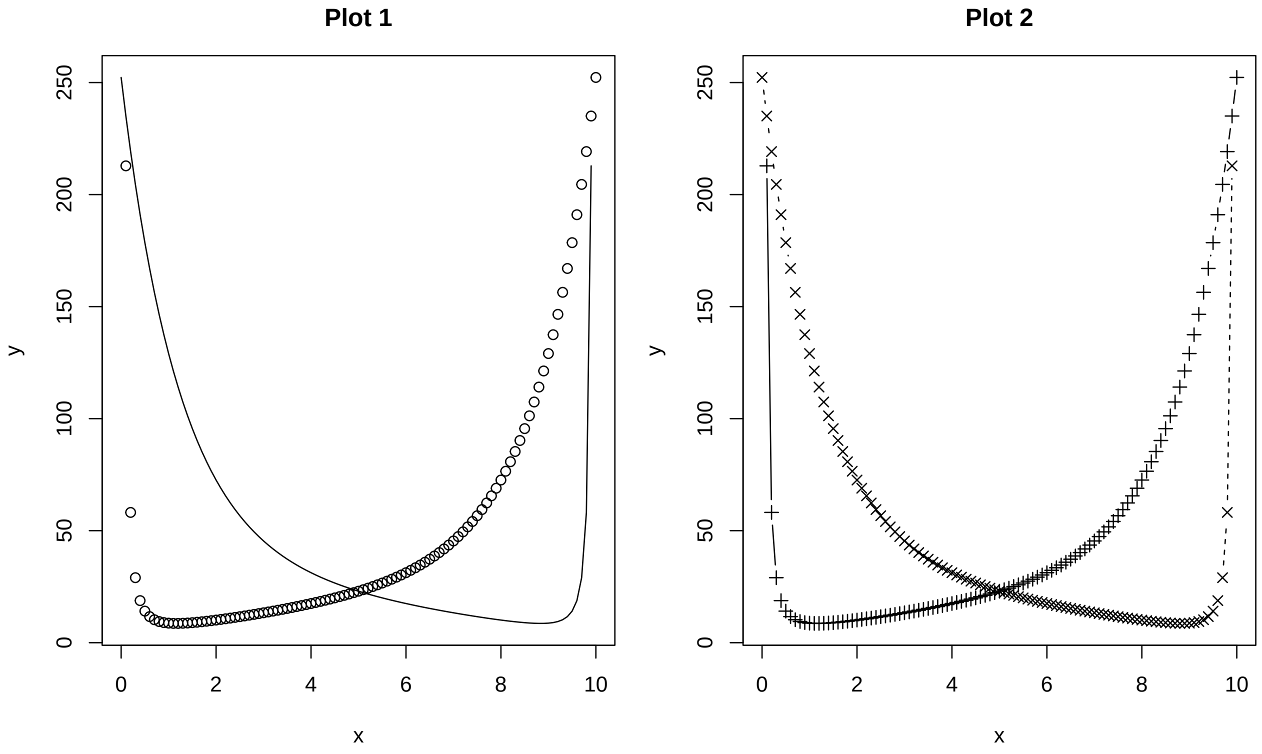
```
sort(a, index.return=TRUE)
```

```
## $x
## [1] -0.2957290 -0.2753126  0.4198477  0.8454541  1.0360246
##
## $ix
## [1] 1 5 3 4 2
```

9 Graphics

```
demofun1 <- function(x)
{
    ( 1 + 2*x^2 + 3*x^3 + exp(x) ) / x^2
}
demofun2 <- function(x)
{
    ( 1 + 2*(10-x)^2 + 3*(10-x)^3 + exp(10-x) ) / (10-x)^2
}

# plot in R windows (for quick look)
#specify plotting parameters
par(mfrow=c(1,2), mar = c(4,4,3,1))
x <- seq(0,10,by=0.1)
#make "Plot 1"
plot(x, demofun1(x), type="p", pch = 1, ylab="y", main="Plot 1")
#add another line to "Plot 1"
points(x, demofun2(x), type="l", lty = 1)
#make "plot 2"
plot(x, demofun1(x), type="b", pch = 3, lty=1, ylab="y", main="Plot 2")
#add another line to "Plot 2"
points(x, demofun2(x), type="b", pch = 4, lty = 2)
```



```r
# save plot in a file (for publication)
pdf ("afig.pdf", height=4.8, width=10)
#specify plotting parameters
par(mfrow=c(1,2), mar = c(4,4,3,1))
x <- seq(0,10,by=0.1)
#make "Plot 1"
plot(x, demofun1(x), type="p", pch = 1, ylab="y", main="Plot 1")
#add another line to "Plot 1"
points(x, demofun2(x), type="l", lty = 1)
#make "plot 2"
plot(x, demofun1(x), type="b", pch = 3, lty=1, ylab="y", main="Plot 2")
#add another line to "Plot 2"
points(x, demofun2(x), type="b", pch = 4, lty = 2)

dev.off()
```

```
## pdf
##   2
```