

```
RGui - [R Console]
File Edit View Misc Packages Windows Help

> q<-3; t<-5
> if(q<t) w<-q+t else w<-q-t
> w
[1] 8
> q1<-3; t1<--5
> if(q1<t1)
+ {
+ w1<-q1+t1
+ } else
+ {
+ w1<-q1-t1
+ }
> w1
[1] 8
> a<-2; b<-3; c<-4
> if(a<b & b<=c) x<-a+b+c
> x
[1] 9
> if(a==b | a>=c) y<-a*b*c
> y
Error: object 'y' not found
> u<-c(2,3,4,1)
> if(u!=3) v<-u
Warning message:
In if (u != 3) v <- u :
  the condition has length > 1 and only the first element will be used
> v
[1] 2 3 4 1
> d<-c(6,12,4,8)
> e<-1:4
> f<-ifelse(d*e>10,d,e)
> f
[1] 1 12 4 8
> |
```

```
RGui - [R Console]
File Edit View Misc Packages Windows Help

> y<-vector(mode="numeric") #f-ja vector() kreira vektor tipa 'mode' i duzine 'length'
> for(i in 1:10)
+ y[i]<-i+1
> y
 [1] 2 3 4 5 6 7 8 9 10 11
> ?iris
> mydf<-iris
> myve<-NULL #kreira se prazan objekat za "skladistenje"
> for(i in seq(along=mydf[,1])) {
+ myve <- c(myve, mean(as.numeric(mydf[i, 1:3])))
+ }
> str(myve)
 num [1:150] 3.33 3.1 3.07 3.07 3.33 ...
> x <- 1:10
> z <- NULL
> for(i in seq(along=x)) {
+ if(x[i] < 5) {
+   z <- c(z, x[i] - 1)
+ } else {
+   z <- c(z, x[i] / x[i])
+ }
+ }
> z
 [1] 0 1 2 3 1 1 1 1 1 1
> x <- 1:10
> z <- NULL
> for(i in seq(along=x)) {
+   if (x[i]<5) {
+     z <- c(z,x[i]-1)
+   } else {
+     stop("values need to be <5")
+   }
+ }
Error: values need to be <5
> z
 [1] 0 1 2 3
> |
```

Edgar Anderson's Iris Data

Description

This famous (Fisher's or Anderson's) iris data set gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are *Iris setosa*, *versicolor*, and *virginica*.

Usage

```
iris  
iris3
```

Format

`iris` is a data frame with 150 cases (rows) and 5 variables (columns) named `Sepal.Length`, `Sepal.Width`, `Petal.Length`, `Petal.Width`, and `Species`.

`iris3` gives the same data arranged as a 3-dimensional array of size 50 by 4 by 3, as represented by S-PLUS. The first dimension gives the case number within the species subsample, the second the measurements with names `Sepal L.`, `Sepal W.`, `Petal L.`, and `Petal W.`, and the third the species.

Source

Fisher, R. A. (1936) The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7, Part II, 179–188.

The data were collected by Anderson, Edgar (1935). The irises of the Gaspé Peninsula, *Bulletin of the American Iris Society*, 59, 2–5.

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole. (has `iris3` as `iris`.)

See Also

[matplot](#) some examples of which use `iris`.

Examples

```
RGui - [R Console]
File Edit View Misc Packages Windows Help

> z <- 0
> while(z < 5) {
+   z <- z + 2
+   print(z)
+ }
[1] 2
[1] 4
[1] 6
> z <- 0
> repeat {
+   z <- z + 1
+   if(z > 100) break()
+ }
> z
[1] 101
> apply(iris[,1:3], 1, mean)
 [1] 3.333333 3.100000 3.066667 3.066667 3.333333 3.666667 3.133333 3.300000 2.900000 3.166667 3.533333 3.266667 3.066667 2.800000 3.666667 3.866667 3.533333 3.333333
 [19] 3.733333 3.466667 3.500000 3.433333 3.066667 3.366667 3.366667 3.200000 3.333333 3.400000 3.333333 3.166667 3.166667 3.433333 3.600000 3.700000 3.166667 3.133333
 [37] 3.433333 3.300000 2.900000 3.333333 3.266667 2.700000 2.966667 3.366667 3.600000 3.066667 3.500000 3.066667 3.500000 3.233333 4.966667 4.700000 4.966667 3.933333
 [55] 4.633333 4.333333 4.766667 3.533333 4.700000 3.933333 3.500000 4.366667 4.066667 4.566667 4.033333 4.733333 4.366667 4.200000 4.300000 4.000000 4.633333 4.300000
 [73] 4.566667 4.533333 4.533333 4.666667 4.800000 4.900000 4.466667 3.933333 3.900000 3.866667 4.133333 4.600000 4.300000 4.633333 4.833333 4.333333 4.233333 4.000000
 [91] 4.166667 4.566667 4.133333 3.533333 4.166667 4.300000 4.266667 4.466667 3.533333 4.200000 5.200000 4.533333 5.333333 4.933333 5.100000 5.733333 3.966667 5.500000
 [109] 5.000000 5.633333 4.933333 4.800000 5.100000 4.400000 4.566667 4.966667 5.000000 6.066667 5.733333 4.400000 5.266667 4.433333 5.733333 4.633333 5.233333 5.466667
 [127] 4.600000 4.666667 4.933333 5.333333 5.433333 6.033333 4.933333 4.733333 4.766667 5.600000 5.100000 5.000000 4.600000 5.133333 5.133333 5.033333 4.533333 5.300000
 [145] 5.233333 4.966667 4.600000 4.900000 5.000000 4.666667
> x <- rnorm(10, -5, 0.1)
> y <- rnorm(10, 5, 2)
> X <- cbind(x, y)
> apply(X, 2, mean)
      x      y
-4.996094  5.399609
> apply(X, 2, sd)
      x      y
0.07596778 1.78813897
> data(airquality)
> apply(airquality,2,mean) #svi podaci MORAJU biti tipa 'numeric', da bi mogla da se izvrsi f-ja 'mean'
  Ozone  Solar.R  Wind  Temp  Month  Day
    NA      NA  9.957516 77.882353  6.993464 15.803922
```

airquality {datasets} R Documentation

New York Air Quality Measurements

Description

Daily air quality measurements in New York, May to September 1973.

Usage

```
airquality
```

Format

A data frame with 154 observations on 6 variables.

```
[,1] Ozone    numeric Ozone (ppb)
[,2] Solar.R  numeric Solar R (lang)
[,3] Wind     numeric Wind (mph)
[,4] Temp    numeric Temperature (degrees F)
[,5] Month   numeric Month (1--12)
[,6] Day     numeric Day of month (1--31)
```

Details

Daily readings of the following air quality values for May 1, 1973 (a Tuesday) to September 30, 1973.

- `Ozone`: Mean ozone in parts per billion from 1300 to 1500 hours at Roosevelt Island
- `Solar.R`: Solar radiation in Langleys in the frequency band 4000–7700 Angstroms from 0800 to 1200 hours at Central Park
- `Wind`: Average wind speed in miles per hour at 0700 and 1000 hours at LaGuardia Airport
- `Temp`: Maximum daily temperature in degrees Fahrenheit at La Guardia Airport.

```
RGui - [R Console]
File Edit View Misc Packages Windows Help

> ?airquality
> apply(airquality,2,mean,na.rm=TRUE)
  Ozone   Solar.R   Wind   Temp   Month   Day
42.129310 185.931507  9.957516  77.882353  6.993464 15.803922
> apply(airquality,1,mean) #daje aritmetičke sredine podataka za svaku vrstu; za ovakve podatke to nije korisna informacija
 [1] 51.90000 40.16667 42.60000 68.91667      NA      NA 67.93333 33.96667 20.35000      NA      NA 61.28333 65.70000 64.31667 29.03333 74.08333 73.50000 30.40000
[19] 75.91667 25.28333 17.28333 74.60000 21.28333 37.66667      NA      NA 24.66667 71.15000 76.28333 72.56667      NA      NA      NA      NA      NA      NA
[37]      NA 43.45000      NA 80.13333 79.41667      NA      NA 46.66667      NA      NA 54.31667 72.78333 25.86667 40.25000 43.71667      NA      NA      NA
[55]      NA      NA      NA      NA      NA      NA      NA 83.35000 66.70000 61.36667      NA 56.43333 76.81667 76.68333 79.55000 80.45000 62.23333      NA
[73] 63.38333 52.98333      NA 28.55000 69.81667 70.88333 76.88333 64.01667 67.75000 21.98333      NA      NA 83.26667 76.00000 37.43333 44.33333 70.90000 75.73333
[91] 74.06667 73.53333 36.48333 22.96667 32.23333      NA      NA      NA 80.83333 72.38333 72.00000      NA      NA 58.91667 69.25000 55.61667      NA 34.05000
[109] 36.71667 41.23333 65.15000 58.38333 66.91667 26.88333      NA 62.95000 87.23333 69.33333      NA 70.28333 79.38333 76.88333 68.71667 61.81667 63.85000 60.63333
[127] 65.10000 41.73333 39.75000 63.15000 58.05000 59.15000 64.11667 65.98333 65.41667 61.88333 22.98333 38.58333 65.48333 58.13333 25.55000 61.38333 56.00000 59.60000
[145] 24.70000 49.71667 28.05000 24.60000 55.81667      NA 55.21667 45.16667 60.25000
>
> tapply(as.vector(iris[,4]), factor(iris[,5]), mean) #dobijaju se uzoracke sredine obelezja 'Petal.Width' po vrstama cvetova (po nivoima faktora 'Species')
  setosa versicolor virginica
  0.246   1.326   2.026
> aggregate(iris[,1:4], list(iris$Species), mean)
  Group.1 Sepal.Length Sepal.Width Petal.Length Petal.Width
1  setosa      5.006      3.428      1.462      0.246
2 versicolor      5.936      2.770      4.260      1.326
3 virginica      6.588      2.974      5.552      2.026
>
> (mylist <- as.list(iris[1:3,1:3])) #kreira se jednostavna lista
$Sepal.Length
[1] 5.1 4.9 4.7

$Sepal.Width
[1] 3.5 3.0 3.2

$Petal.Length
[1] 1.4 1.4 1.3

> |
```

```
RGui - [R Console]
File Edit View Misc Packages Windows Help

> lapply(mylist, sum) #izracunava se zbir svake komponente liste i kao rezultat se dobija lista
$Sepal.Length
[1] 14.7

$Sepal.Width
[1] 9.7

$Petal.Length
[1] 4.1

> sapply(mylist, sum) #izracunava se zbir svake komponente liste i kao rezultat se dobija vektor
Sepal.Length Sepal.Width Petal.Length
14.7          9.7          4.1

> sapply(iris,is.factor) #odredjuje koje su kolone u bazi podataka - faktori
Sepal.Length Sepal.Width Petal.Length Petal.Width Species
FALSE        FALSE        FALSE        FALSE        TRUE

> sapply(iris, function(x) if(!is.factor(x))return(0) else length(levels(x))) #odredjuje koliko nivoa ima faktor
Sepal.Length Sepal.Width Petal.Length Petal.Width Species
0            0            0            0            3

> #spajanje dve baze
> authors <- data.frame(surname = I(c("Tukey", "Venables", "Tierney", "Ripley", "McNeil")),nationality = c("US", "Australia", "US", "UK", "Australia"),
+ deceased = c("yes", rep("no", 4)))
> books <- data.frame(name = I(c("Tukey", "Venables", "Tierney", "Ripley", "Ripley", "McNeil", "R Core")),
+ title = c("Exploratory Data Analysis", "Modern Applied Statistics ...", "LISP-STAT", "Spatial Statistics", "Stochastic Simulation", "Interactive Data Analysis",
+ "An Introduction to R"), other.author = c(NA, "Ripley", NA, NA, NA, NA, "Venables & Smith"))
> (m1 <- merge(authors, books, by.x = "surname", by.y = "name"))
  surname nationality deceased          title other.author
1  McNeil  Australia      no  Interactive Data Analysis      <NA>
2  Ripley      UK        no      Spatial Statistics      <NA>
3  Ripley      UK        no      Stochastic Simulation      <NA>
4 Tierney      US        no          LISP-STAT      <NA>
5  Tukey      US        yes  Exploratory Data Analysis      <NA>
6 Venables  Australia      no  Modern Applied Statistics ...      Ripley

> # 'R Core' iz 'books' nedostaje u novoj bazi m1, jer nema odgovarajucih podataka za to u 'authors'
> |
```

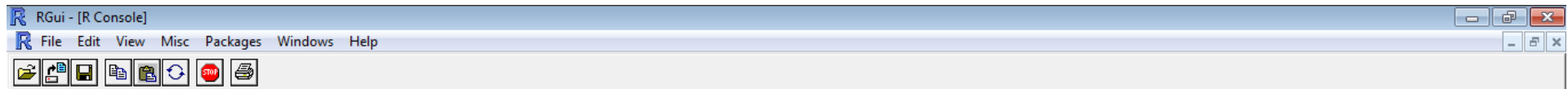
```
RGui - [R Console]
File Edit View Misc Packages Windows Help

> #osnove pisanja f-ja
> myfct <- function(x1, x2=5) {
+   z1 <- x1/x1
+   z2 <- x2*x2
+   myvec <- c(z1, z2)
+   return(myvec)
+ }
> myfct #ispisuje definiciju f-je
function(x1, x2=5) {
  z1 <- x1/x1
  z2 <- x2*x2
  myvec <- c(z1, z2)
  return(myvec)
}
> myfct(x1=2, x2=5) #f-ja se primenjuje na vrednosti argumenata 2, 5, redom
[1] 1 25
> myfct(2, 5) #imena argumenata nisu obavezna, ali ako se izostave mora se postovati redosled argumenata i on mora biti isti onakav kakav stoji u def. f-je
[1] 1 25
> myfct(x1=2) #dobija se isti izlaz kao i iznad, ali se koristi default vrednost '5' za drugi argument
[1] 1 25
> #primer f-je sa opcionalnim argumentom
> myfct2 <- function(x1=5, opt_arg) {
+   if(missing(opt_arg)) { #f-ja 'missing()' se koristi da se proveriti da li je neka vrednost odredjena kao argument
+     z1 <- 1:10
+   } else {
+     z1 <- opt_arg
+   }
+   cat("my function returns:", "\n")
+   return(z1/x1)
+ }
> myfct2(x1=5)
my function returns:
[1] 0.2 0.4 0.6 0.8 1.0 1.2 1.4 1.6 1.8 2.0
> myfct2(x1=5, opt_arg=30:20)
my function returns:
[1] 6.0 5.8 5.6 5.4 5.2 5.0 4.8 4.6 4.4 4.2 4.0
> |
```



```
RGui - [R Console]
File Edit View Misc Packages Windows Help

> myfct <- function(x1) {
+   if (x1>=0) print(x1) else stop("This function did not finish, because x1 < 0")
+   warning("Value needs to be > 0")
+ }
> myfct(x1=2)
[1] 2
Warning message:
In myfct(x1 = 2) : Value needs to be > 0
> myfct(x1=-2)
Error in myfct(x1 = -2) : This function did not finish, because x1 < 0
> #koriscenje f-je unutar druge f-je
> x <- 1:10
> test <- function(x) { #ovim se definise f-ja 'test'
+   if(x < 5) {
+     x-1
+   } else {
+     x / x
+   }
+ }
> apply(as.matrix(x), 1, test)
[1] 0 1 2 3 1 1 1 1 1 1
> apply(as.matrix(x), 1, function(x) { if (x<5) { x-1 } else { x/x } }) #dobija se isti izlaz kao u prethodnom pozivu, samo u jednom redu :)
[1] 0 1 2 3 1 1 1 1 1 1
> f1 <- function(x, ...) {
+ y <- x+2
+ return(y)
+ #druge naredbe
+ }
> f2 <- function( ... , x) {
+ y <- x+2
+ return(y)
+ #druge naredbe
+ }
> f1(3)
[1] 5
> f1(x=3)
[1] 5
> f2(3)
Error in x + 2 : 'x' is missing
> f2(x=3)
[1] 5
> |
```



```
> myStats <- function(x, ...)  
+ {  
+ m <- sum(x, ...)/length(x)  
+ s <- sd(x)  
+ return(list(mean=m, sd=s))  
+ }  
> y <- myStats(0:10)  
> y  
$mean  
[1] 5  
  
$sd  
[1] 3.316625  
  
> y1 <- myStats(0:10, c(1,2))  
> y1  
$mean  
[1] 5.272727  
  
$sd  
[1] 3.316625  
  
> |  
> f1<-function(x,...)  
+ {  
+ y<-sum(x,1,...)  
+ return(y)  
+ }  
> f1(1:10)  
[1] 56  
> f1(1:10,c(1,2,3))  
[1] 62  
> |
```

Вежбања:

1. Са сајта <http://cran.r-project.org/> преузети пакет `PASWR`, и инсталирати га. Учитати базу података `Cows` и испитати какве податке садржи. Одредити средњу вредност масти у путеру (обележје `butterfat`) за сваки од фактора. Направити табелу фактора.
2. Написати функцију која генерише узорак од 100 независних реализација случајне величине, која је униформно расподељена на интервалу $[3.7, 5.8]$, а затим испишује узорачку средину, дисперзију и стандардно одступање (за њихово рачунање НЕ користити готове ф-је `mean`, `var`, `sd`) и пореди те резултате са теоријским вредностима ових величина. Модификовати ф-ју тако да описани поступак може да се извршава за било који интервал.
3. Написати ф-ју која симулира ситуацију: студент полаже испит који садржи 40 питања, за свако питање постоје два одговора: Тачно, Нетачно, а студент погађањем (тј. на случајан начин) бира САМО ЈЕДАН од одговора на свако питање. Ф-ја, затим, израчунава средњу вредност и дисперзију броја тачних одговора студента на тесту и врши поређење са теоријским вредностима.
4. Написати ф-ју која израчунава средину \bar{v}_h вектора v , која је дефинисана са:

$$\bar{v}_h := \left(\prod_{i=1}^n v_i \right)^{1/n}$$

где је n дужина вектора v , а v_i је i -та компонента вектора v . Ф-ја треба да проверава своје аргументе и испишује одговарајуће поруке у случајевима када није могуће израчунати средину.