

Esercizi vari

numeric

My_max

```
my_max <- function(x) {  
  if(!is.vector(x)) {stop("L'argomento deve essere un vettore")}  
  massimo <- ""  
  for(numero in x) {  
    if(numero > massimo) {massimo <- numero}  
  }  
  return(massimo)  
}  
o <- 1:10  
o
```

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

```
my_max(o)
```

```
## [1] 10
```

Scrivi uno script R che definisca una funzione personalizzata chiamata `somma_pari` che prenda in input un vettore di numeri interi e restituisca la somma dei numeri pari presenti nel vettore. Una volta definita la funzione, prova a testarla con un vettore di numeri interi a tua scelta.

```
a <- 1:10  
  
somma_pari <- function(x) {  
  if(!is.vector(x)) {stop("L'argomento inserito non è un vettore")}  
  contatore <- 0  
  i <- 1  
  while(i %in% 1:length(x)) {  
    if(x[i] %% 2 == 0){  
      contatore <- contatore + x[i]  
    } else {contatore <- contatore + 0}  
    i <- i + 1  
  }  
  return(contatore)  
}  
somma_pari(a)
```

```
## [1] 30
```

Scrivi uno script R che definisca una funzione personalizzata chiamata `media_valori_positivi` che prenda in input un vettore di numeri e restituisca la media dei valori positivi presenti nel vettore. Una volta definita la funzione, prova a testarla con un vettore di numeri a tua scelta.

```

b <- -50:50

media_valori_positivi <- function(x) {
  if(!is.vector(x)) {stop("L'argomento inserito non è un vettore")}
  f <- vector(mode = "numeric")
  for(i in x) {
    if(i >= 0) {
      f <- c(f,i)}
  }
  return(mean(f))
}
media_valori_positivi(b)

```

```
## [1] 25
```

my_unique: Crea una funzione che restituisca un vettore contenente solo gli elementi unici presenti nel vettore di input. Non usare funzione built-in unique(). Si può usare l'operatore %in%. Usare funzione vector() per creare un vettore vuoto.

```

my_unique <- function(x) {
  if(!is.vector(x)) {stop("L'argomento inserito non è un vettore")}
  unici <- c()
  i <- 1
  while(i <= length(x)) {
    if(!(x[i] %in% unici)) {
      unici <- c(unici, x[i])
    }
    i <- i + 1
  }
  return(unici)
}
x <- c(2,2,4,5,6)
my_unique(x)

```

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

my_median:

```

my_median <- function(x) {
  if(!is.vector(x)) {stop("L'argomento non è un vettore")}
  vettore_ordinato <- sort(x)
  if(length(x) %% 2 == 0) {
    mediana <- (x[(length(x)/2)] + x[(length(x)/2 + 1)])/2 } else {
    mediana <- x[(length(x) + 1)/2]
  }
  return(mediana)
}

my_median(o)

```

```
## [1] 5.5
```

```
median(o)
```

```
## [1] 5.5
```

my_matrix_transpose: Implementa una funzione che calcoli la trasposta di una matrice.

```
my_matrix_transpose <- function(m) {  
  nr <- nrow(m)  
  nc <- ncol(m)  
  dati <- c()  
  i <- 1  
  j <- 1  
  while (j <= nc){  
    while(i <= nr){  
      for(numero in m[i,]) {  
        dati <- c(dati, numero)  
      }  
      i <- i + 1  
    }  
    j <- j + 1  
  }  
  trasposta <- matrix(dati, nrow = nc, ncol = nr)  
  return(trasposta)  
}  
y <- matrix(1:8, nrow = 2, ncol = 4)  
y
```

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

```
my_matrix_transpose(y)
```

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

Scrivi uno script R che definisca una funzione personalizzata chiamata conta quanti sup che prenda in input un vettore e un numero k, e restituisca il conteggio di quante volte i numeri nel vettore sono superiori a k.

```
b <- -50:50
```

```
conta_quanti_sup <- function(x,k) {  
  if(!(is.vector(x))) {stop("L'argomento inserito non è un vettore")}  
  if(!(is.numeric(k))) {stop("L'argomento deve essere numerico")}  
  contatore <- 0  
  for(numero in x) {  
    if(numero > k) {  
      contatore <- contatore + 1  
    }  
  }  
}
```

```
    }  
  }  
  return(contatore)  
}  
gggg <- which(b > 0)  
length(gggg)
```

```
## [1] 50
```

```
conta_quanti_sup(b,0)
```

```
## [1] 50
```

Stringhe

Alcuni comandi base

```
k <- "ciao ciao ciao ciao"  
substr(k,1,10)
```

```
## [1] "ciao ciao "
```

```
strsplit(k, " ")
```

```
## [[1]]  
## [1] "ciao" "ciao" "ciao" "ciao"
```

```
gsub("a", "o", k)
```

```
## [1] "cioo cioo cioo cioo"
```

```
grepl("a", k)
```

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

```
typeof(substr(k,1,10))
```

```
## [1] "character"
```

```
typeof(strsplit(k, " "))
```

```
## [1] "list"
```

```
typeof(gsub("a", "o", k))
```

```
## [1] "character"
```

```
typeof(grepl("a", k))
```

```
## [1] "logical"
```

Scrivi uno script R che definisca una funzione personalizzata chiamata `conta_occorrenze_carattere` che prenda in input una stringa e un carattere e restituisca il conteggio di quante volte il carattere appare nella stringa. Una volta definita la funzione, prova a testarla con una stringa a tua scelta e un carattere arbitrario.

```
stringa <- "Zio perone"

conta_occorrenze_carattere <- function(x,c) {
  if(!is.character(x)) {stop("L'argomento inserito non è di tipo character")}
  if(!is.character(c)) {stop("L'argomento inserito non è di tipo character")}
  contatore <- 0
  for(i in strsplit(x,"") [[1]]) {
    if(i == c) {
      contatore <- contatore + 1
    }
  }
  return(contatore)
}

conta_occorrenze_carattere(stringa, "o")
```

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

Scrivi uno script R che definisca una funzione personalizzata chiamata `inverte_stringa` che prenda in input una stringa e restituisca la stringa invertita. Una volta definita la funzione, prova a testarla con una stringa a tua scelta.

```
inverte_stringa <- function(x) {
  if(!is.character(x)) {stop("L'argomento inserito non è di tipo character")}
  k <- nchar(x)
  f <- ""
  while(k %in% nchar(x):1) {
    f <- paste(f, substr(x, k, k), sep = "")
    k <- k - 1
  }
  return(f)
}

inverte_stringa(stringa)
```

```
## [1] "enorep oiZ"
```

Scrivi uno script R che definisca una funzione personalizzata chiamata `rimuovi_vocali` che prenda in input una stringa e restituisca la stringa con tutte le vocali rimosse. Una volta definita la funzione, prova a testarla con alcune stringhe a tua scelta.

```
a <- "Ciao"
rimuovi_vocali <- function(x) {
  if(!is.character(x)) {stop("L'argomento non è una stringa")}

```

```

vocali <- c("a", "e", "i", "o", "u")
i <- 1
no_vocali <- ""
parola <- strsplit(x, split = "")
for(lettera in parola [[1]]) {
  if(!(lettera %in% vocali)) {no_vocali <- paste(no_vocali, lettera)}
}
return(no_vocali)
}
rimuovi_vocali(a)

```

```
## [1] " C"
```

```

k <- "Zio Pera"
rimuovi_vocali(k)

```

```
## [1] " Z   P r"
```

Scrivi uno script R che definisca una funzione personalizzata chiamata `conta_parole_contenenti` che prenda in input una stringa e una sottostringa, e restituisca il numero di parole nella stringa che contengono la sottostringa. Una volta definita la funzione, prova a testarla con alcune stringhe e sottostringhe a tua scelta.

```

conta_parole_contenenti <- function(x) {
  if(!(is.character(x))) {stop("L'argomento non è una stringa")}
  conteggio <- 0
  parola <- unlist(strsplit(x, split = " "))
  for(parole in parola) {
    conteggio <- conteggio + 1
  }
  return(conteggio)
}
k <- "ciao ciao ciao ciao"
conta_parole_contenenti(k)

```

```
## [1] 4
```

Array

Scrivi uno script R che definisca una funzione personalizzata chiamata `calcola_varianza` che prenda in input un array di numeri e restituisca la varianza dei numeri presenti nell'array. Una volta definita la funzione, prova a testarla con un array di numeri a tua scelta.

```

a <- matrix(c(1:9), nrow = 3, ncol = 3)
b <- matrix(c(10:18), nrow = 3, ncol = 3)
c <- matrix(c(19:27), nrow = 3, ncol = 3)
m <- array(data = c(a,b,c), dim = c(3,3,3))

calcola_varianza <- function(x) {
  if(!(is.array(x))) {stop("L'argomento non è un array")}
  varianza <- c()

```

```

i <- 1
while(i <= dim(x)[3]) {
  varianza_matrix <- mean((x[, , i])^2) - (mean(x[, , i]))^2
  varianza <- c(varianza, varianza_matrix)
  i <- i + 1
}
return(varianza)
}
calcola_varianza(m)

```

```
## [1] 6.666667 6.666667 6.666667
```

my_array_sum: Scrivi una funzione che calcoli la somma degli elementi di un array tridimensionale.

```

m <- 1:9
n <- 10:18
o <- 19:27
array_ex_data <- c(m,n,o
)
array_ex <- array(array_ex_data, dim = c(2,2,3))
dim(array_ex)

```

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

```

my_array_sum <- function(x){
  if(!is.array(x)) {stop("L'argomento non è un array")}
  contatore <- 0
  for(numero in x) {
    contatore <- contatore + numero
  }
  return(contatore)
}
my_array_sum(array_ex)

```

```
## [1] 78
```

```
sum(array_ex)
```

```
## [1] 78
```

my_array_mean: Implementa una funzione che calcoli la media delle matrici di un array multidimensionale.

```

my_array_mean <- function(x) {
  if(!is.array(x)) {stop("L'argomento non è un array")}
  z <- 1
  vettore_medie <- c()
  #n_elem <- dim(x)[2] * dim(x)[3]
  #contatore <- 0
  while(z <= dim(x)[3]) {
    vettore_medie <- c(vettore_medie, mean(x[, , z]))
  }
}

```

```

    z <- z + 1
  }
  return(vettore_medie)
}

```

```
my_array_mean(array_ex)
```

```
## [1] 2.5 6.5 10.5
```

```
mean(array_ex [, , 1])
```

```
## [1] 2.5
```

```
mean(array_ex [, , 2])
```

```
## [1] 6.5
```

```
mean(array_ex [, , 3])
```

```
## [1] 10.5
```

```

#funzione anonima quadrato degli elementi
quadrato <- function(x) x^2
quadrato(my_array_mean(array_ex))

```

```
## [1] 6.25 42.25 110.25
```

my_array_sup: Implementa una funzione che restituisca TRUE se la media degli elementi di un array multidimensionale supera un valore di soglia s dato in input.

```

my_array_sup <- function(x,s) {
  if(!(is.array(x))) {stop("L'argomento non è un array")}
  if(!(is.numeric(s))) {stop("L'argomento non è di tipo numerico")}
  n_elem <- dim(x)[1]*dim(x)[2]*dim(x)[3]
  contatore <- 0
  for(numero in x) {
    contatore <- contatore + numero
  }
  media <- contatore/n_elem
  return(media > s)
}
mean(array_ex)

```

```
## [1] 6.5
```

```
my_array_sup(array_ex, 7)
```

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


Liste

my_list_length: Scrivi una funzione che restituisca il numero di elementi presenti in una lista.

```
lista_ex <- list(a = c("a", "b", "c"), b = 1:10)
my_list_length <- function(x) {
  if(!is.list(x)) {stop("L'argomento inserito non è una lista")}
  contatore <- 0
  i <- 1
  while(i <= length(x)) {
    for(elemento in x[[i]]) {
      contatore <- contatore + 1
    }
    i <- i + 1
  }
  return(contatore)
}
my_list_length(lista_ex)
```

```
## [1] 13
```

my_list_reverse: Implementa una funzione che inverte l'ordine degli elementi in una lista.

```
my_list_reverse <- function(x) {
  if(!is.list(x)) {stop("L'argomento inserito non è una lista")}
  j <- 1
  while(j <= length(x)) {
    x[[j]] <- rev(x[[j]])
    j <- j + 1
  }
  return(x)
}

my_list_reverse(lista_ex)
```

```
## $a
## [1] "c" "b" "a"
##
## $b
## [1] 10  9  8  7  6  5  4  3  2  1
```

Apply

```
apply(a, 2, mean)
```

```
## [1] 2 5 8
```

```
prezzi <- 1:20
lista <- list(a = 1:9, b = 1:10)
lapply(lista, mean)
```

```
## $a
## [1] 5
##
## $b
## [1] 5.5
```

```
vapply(lista, mean, FUN.VALUE = numeric(1))
```

```
## a b
## 5.0 5.5
```

```
sapply(lista, mean, simplify = "vector")
```

```
## a b
## 5.0 5.5
```

```
Età = c(18:90)
mapply(sum, prezzi, Età)
```

```
## Warning in mapply(sum, prezzi, Età): l'argomento più lungo non è un multiplo
## della lunghezza del più piccolo
```

```
## [1] 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55
## [20] 57 39 41 43 45 47 49 51 53 55 57 59 61 63 65 67 69 71 73
## [39] 75 77 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91
## [58] 93 95 97 79 81 83 85 87 89 91 93 95 97 99 101 103
```

Definisci una funzione che prenda in input una lista di parole e restituisca la parola più lunga.

```
Prodotti <- c("latte", "uova", "formaggio", "pasta", "burro")
parola_più_lunga <- function(x) {
  grande <- ""
  i <- 1
  while(i < length(x)) {
    if(is.character(x[[i]])) {
      for(parola in x[[i]]) {
        if(nchar(substr(parola, 1, nchar(parola))) > nchar(grande)) {
          grande <- parola
        }
      }
    }
    i <- i + 1
  }
  return(grande)
}

lista_parole <- list(a = Prodotti, b = 1:3, c = Prodotti )
parola_più_lunga(lista_parole)
```

```
## [1] "formaggio"
```

```
which.max(nchar(Prodotti))
```

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

```
Prodotti[3]
```

```
## [1] "formaggio"
```

Dataframe

my_df_mean: Scrivi una funzione che calcoli la media delle colonne numeriche in un dataframe.

```
Prodotti <- c("latte", "uova", "formaggio", "pasta", "burro")
prezzi <- 1:20
disponibilità <- c("si", "no")
supermercato <- data.frame(prodotti = sample(Prodotti, 1000, replace = T), prezzi = sample(prezzi, 1000, replace = T),
                           disponibilità = sample(disponibilità, 1000, replace = T))

my_df_mean <- function(x) {
  if(!is.data.frame(x)) {stop("L'argomento inserito non è un dataframe")}
  i <- 1
  colonne <- colnames(x)
  medie <- c()
  while(i <= length(colonne)) {
    if(is.numeric(x[,i])) {
      medie <- c(medie, mean(x[,i], na.rm = T))
    }
    i <- i + 1
  }
  return(medie)
}
my_df_mean(supermercato)
```

```
## [1] 10.426 50.439
```

```
mean(supermercato$prezzi, na.rm = T)
```

```
## [1] 10.426
```

```
mean(supermercato$quantità, na.rm = T)
```

```
## [1] 50.439
```

Dato un dataframe chiamato dati_prodotti con le seguenti colonne: Prodotto, Prezzo, Disponibilità. L'esercizio consiste nel filtrare il dataframe per selezionare solo i prodotti disponibili e con un prezzo inferiore a 10€.

```
Prodotti <- c("latte", "uova", "formaggio", "pasta", "burro")
prezzi <- 1:20
disponibilità <- c("si", "no")
supermercato <- data.frame(prodotti = sample(Prodotti, 1000, replace = T), prezzi = sample(prezzi, 1000))
library(dplyr)
```

```
## Warning: il pacchetto 'dplyr' è stato creato con R versione 4.3.3
```

```
##
```

```
## Caricamento pacchetto: 'dplyr'
```

```
## I seguenti oggetti sono mascherati da 'package:stats':
```

```
##
```

```
##      filter, lag
```

```
## I seguenti oggetti sono mascherati da 'package:base':
```

```
##
```

```
##      intersect, setdiff, setequal, union
```

```
expensive <- supermercato %>%
  filter(prezzi > 10)
print(expensive)
```

```
##      prodotti prezzi disponibilità quantità
## 1  formaggio    17             no        77
## 2  formaggio    12             si         46
## 3    burro     17             no         14
## 4    latte     17             no         12
## 5  formaggio    17             no          1
## 6  formaggio    18             no         51
## 7    burro     19             si         81
## 8  formaggio    19             si         44
## 9     uova     20             si         67
## 10    uova     13             si         92
## 11    burro     12             no         15
## 12    burro     15             si         22
## 13    pasta     12             si         86
## 14    uova     13             no          7
## 15    burro     14             si         48
## 16    latte     16             no         88
## 17    burro     16             no         61
## 18    uova     15             si         85
## 19 formaggio    14             si         91
## 20    pasta     13             si         54
## 21    latte     11             si         75
## 22    uova     13             si         51
## 23 formaggio    12             no         77
## 24 formaggio    17             no         67
## 25    burro     11             si         33
## 26    pasta     15             si          6
## 27 formaggio    16             si         75
```

## 28	burro	18	no	98
## 29	uova	18	si	55
## 30	uova	19	si	69
## 31	burro	14	si	14
## 32	formaggio	13	si	47
## 33	latte	13	si	42
## 34	burro	18	no	66
## 35	burro	16	no	28
## 36	burro	19	si	57
## 37	burro	12	si	34
## 38	uova	20	si	62
## 39	formaggio	17	no	3
## 40	formaggio	15	si	90
## 41	latte	12	no	88
## 42	burro	18	no	59
## 43	pasta	15	si	63
## 44	pasta	20	no	82
## 45	latte	15	si	37
## 46	uova	19	si	77
## 47	burro	14	si	62
## 48	latte	15	no	81
## 49	pasta	17	si	66
## 50	uova	15	si	11
## 51	formaggio	14	no	29
## 52	formaggio	19	no	70
## 53	formaggio	18	no	37
## 54	burro	13	no	96
## 55	formaggio	12	no	22
## 56	uova	12	no	88
## 57	burro	12	no	9
## 58	formaggio	19	si	75
## 59	latte	12	no	28
## 60	pasta	20	no	45
## 61	latte	20	si	47
## 62	uova	12	si	5
## 63	latte	18	si	64
## 64	formaggio	11	si	40
## 65	pasta	13	si	3
## 66	burro	20	no	97
## 67	formaggio	19	si	17
## 68	formaggio	17	no	48
## 69	latte	18	no	11
## 70	burro	17	no	98
## 71	uova	19	no	48
## 72	pasta	15	si	85
## 73	burro	11	no	79
## 74	uova	15	si	66
## 75	formaggio	15	no	84
## 76	burro	20	no	79
## 77	burro	18	no	66
## 78	burro	11	si	4
## 79	pasta	17	si	85
## 80	uova	16	si	72
## 81	uova	13	no	36

## 82	burro	20	no	57
## 83	pasta	20	si	14
## 84	formaggio	18	no	37
## 85	burro	11	no	88
## 86	latte	20	no	3
## 87	pasta	12	no	94
## 88	pasta	17	si	35
## 89	uova	11	no	83
## 90	latte	16	no	33
## 91	formaggio	12	no	5
## 92	pasta	11	no	12
## 93	pasta	18	no	100
## 94	uova	11	si	87
## 95	pasta	13	no	16
## 96	formaggio	18	no	50
## 97	uova	19	no	30
## 98	pasta	19	no	82
## 99	pasta	18	no	29
## 100	uova	12	si	22
## 101	pasta	15	no	35
## 102	pasta	18	si	31
## 103	burro	11	no	24
## 104	latte	18	si	89
## 105	formaggio	18	no	18
## 106	burro	14	no	28
## 107	formaggio	17	si	91
## 108	pasta	18	si	6
## 109	uova	15	si	25
## 110	uova	16	si	80
## 111	burro	14	si	97
## 112	burro	18	no	9
## 113	uova	14	no	20
## 114	pasta	20	no	40
## 115	pasta	13	no	55
## 116	pasta	18	si	86
## 117	formaggio	14	no	27
## 118	burro	12	no	55
## 119	pasta	14	no	74
## 120	uova	19	no	22
## 121	burro	14	si	21
## 122	burro	15	no	83
## 123	pasta	17	si	81
## 124	formaggio	17	no	29
## 125	burro	16	si	97
## 126	uova	20	no	2
## 127	formaggio	19	si	49
## 128	uova	13	no	68
## 129	latte	14	no	79
## 130	pasta	15	si	71
## 131	uova	12	si	14
## 132	formaggio	15	si	18
## 133	latte	18	si	9
## 134	uova	16	no	60
## 135	latte	15	si	80

## 136	pasta	18	no	67
## 137	pasta	11	si	49
## 138	uova	15	si	57
## 139	uova	18	si	78
## 140	formaggio	18	no	63
## 141	latte	19	si	89
## 142	pasta	16	si	39
## 143	uova	18	no	41
## 144	burro	17	si	79
## 145	latte	20	si	41
## 146	uova	17	no	41
## 147	uova	15	si	22
## 148	formaggio	13	si	71
## 149	burro	17	no	37
## 150	pasta	12	si	42
## 151	formaggio	17	si	16
## 152	uova	12	no	75
## 153	formaggio	11	no	78
## 154	uova	12	no	89
## 155	latte	12	no	94
## 156	uova	17	no	41
## 157	pasta	20	si	96
## 158	burro	14	si	92
## 159	formaggio	16	no	95
## 160	uova	13	si	10
## 161	pasta	13	no	13
## 162	uova	18	si	90
## 163	uova	13	si	96
## 164	uova	20	no	76
## 165	burro	18	si	95
## 166	burro	14	si	83
## 167	burro	13	no	98
## 168	uova	19	no	20
## 169	latte	19	si	89
## 170	pasta	17	no	36
## 171	burro	18	no	41
## 172	pasta	16	si	29
## 173	burro	13	si	67
## 174	burro	11	si	70
## 175	latte	20	no	99
## 176	burro	14	si	54
## 177	burro	16	no	55
## 178	formaggio	14	no	78
## 179	formaggio	13	si	62
## 180	formaggio	19	no	3
## 181	burro	11	no	33
## 182	uova	20	si	91
## 183	burro	16	si	87
## 184	uova	19	no	47
## 185	burro	15	no	27
## 186	pasta	12	si	56
## 187	pasta	20	no	16
## 188	burro	17	si	80
## 189	burro	19	no	52

## 190	pasta	19	no	24
## 191	burro	14	si	5
## 192	uova	20	no	11
## 193	pasta	17	si	24
## 194	pasta	15	no	47
## 195	latte	16	si	49
## 196	burro	15	si	97
## 197	uova	11	no	90
## 198	pasta	17	si	37
## 199	burro	14	si	65
## 200	burro	15	no	61
## 201	pasta	16	no	46
## 202	uova	17	no	15
## 203	latte	17	no	35
## 204	pasta	12	si	23
## 205	uova	13	si	38
## 206	formaggio	12	si	51
## 207	pasta	18	si	21
## 208	burro	17	si	58
## 209	latte	16	no	46
## 210	formaggio	18	no	85
## 211	formaggio	20	no	50
## 212	uova	13	si	93
## 213	latte	11	no	25
## 214	uova	18	si	49
## 215	burro	15	si	36
## 216	uova	12	si	54
## 217	pasta	11	si	51
## 218	pasta	19	si	80
## 219	burro	13	si	2
## 220	burro	17	no	77
## 221	latte	16	si	49
## 222	pasta	19	no	6
## 223	uova	11	si	75
## 224	formaggio	16	si	22
## 225	burro	20	no	29
## 226	pasta	16	no	40
## 227	pasta	15	si	74
## 228	uova	17	si	60
## 229	burro	14	si	43
## 230	burro	13	no	87
## 231	formaggio	18	no	52
## 232	formaggio	20	no	88
## 233	latte	19	no	16
## 234	formaggio	17	si	1
## 235	uova	20	no	84
## 236	latte	18	si	11
## 237	pasta	12	si	56
## 238	formaggio	19	no	54
## 239	burro	16	no	80
## 240	burro	20	si	28
## 241	pasta	14	si	98
## 242	burro	13	no	37
## 243	pasta	19	si	6

## 244	latte	14	si	81
## 245	latte	12	si	45
## 246	uova	12	no	14
## 247	pasta	16	no	43
## 248	latte	13	si	74
## 249	latte	18	si	35
## 250	formaggio	19	no	35
## 251	formaggio	11	si	85
## 252	latte	13	si	25
## 253	burro	17	si	54
## 254	latte	14	si	66
## 255	uova	13	si	44
## 256	formaggio	20	si	2
## 257	latte	19	no	56
## 258	formaggio	11	si	42
## 259	pasta	12	si	49
## 260	formaggio	12	no	42
## 261	uova	20	no	23
## 262	latte	17	si	9
## 263	latte	20	si	8
## 264	pasta	13	si	64
## 265	pasta	17	si	5
## 266	uova	18	no	42
## 267	formaggio	12	no	31
## 268	latte	11	no	20
## 269	burro	18	no	89
## 270	pasta	13	no	100
## 271	uova	14	si	58
## 272	latte	18	si	28
## 273	burro	17	si	72
## 274	burro	20	no	42
## 275	uova	19	si	62
## 276	burro	16	si	16
## 277	latte	16	no	69
## 278	pasta	14	si	59
## 279	latte	18	no	25
## 280	formaggio	14	no	14
## 281	formaggio	14	no	82
## 282	formaggio	20	no	53
## 283	burro	19	no	87
## 284	pasta	19	si	5
## 285	formaggio	19	si	16
## 286	burro	16	si	85
## 287	pasta	14	si	70
## 288	burro	16	no	3
## 289	pasta	16	si	67
## 290	pasta	12	si	69
## 291	burro	15	si	11
## 292	formaggio	17	si	18
## 293	uova	11	si	84
## 294	latte	15	si	22
## 295	burro	14	si	70
## 296	burro	15	si	30
## 297	uova	13	no	12

## 298	latte	12	si	1
## 299	pasta	16	si	96
## 300	uova	19	si	69
## 301	uova	11	no	92
## 302	burro	11	no	28
## 303	uova	18	no	99
## 304	latte	11	no	94
## 305	uova	14	si	73
## 306	burro	18	no	23
## 307	formaggio	18	no	51
## 308	uova	12	no	3
## 309	formaggio	19	no	32
## 310	latte	14	si	66
## 311	burro	17	si	10
## 312	formaggio	16	no	64
## 313	burro	13	si	27
## 314	latte	15	no	50
## 315	pasta	15	si	74
## 316	pasta	17	no	60
## 317	latte	14	si	57
## 318	burro	12	no	14
## 319	latte	19	si	28
## 320	burro	15	si	9
## 321	burro	18	no	57
## 322	uova	19	si	23
## 323	pasta	18	si	71
## 324	pasta	17	si	17
## 325	burro	17	si	13
## 326	formaggio	12	no	50
## 327	uova	12	si	76
## 328	formaggio	19	si	84
## 329	burro	11	si	77
## 330	formaggio	20	si	62
## 331	burro	13	no	34
## 332	pasta	19	no	14
## 333	burro	16	no	61
## 334	uova	14	si	74
## 335	formaggio	16	si	70
## 336	formaggio	19	no	65
## 337	uova	17	no	20
## 338	formaggio	14	no	14
## 339	burro	14	si	36
## 340	pasta	11	si	45
## 341	pasta	17	si	62
## 342	burro	15	no	76
## 343	burro	12	no	48
## 344	pasta	13	no	36
## 345	uova	18	no	62
## 346	formaggio	15	no	69
## 347	formaggio	14	no	53
## 348	formaggio	12	no	8
## 349	latte	11	no	81
## 350	formaggio	16	si	75
## 351	latte	11	no	67

## 352	latte	11	no	80
## 353	latte	16	no	86
## 354	latte	16	si	77
## 355	pasta	13	no	97
## 356	pasta	11	no	55
## 357	latte	19	no	89
## 358	formaggio	11	si	88
## 359	burro	16	si	91
## 360	burro	11	no	21
## 361	uova	18	no	22
## 362	burro	15	no	15
## 363	burro	17	si	51
## 364	burro	19	si	21
## 365	burro	17	no	75
## 366	burro	16	no	74
## 367	formaggio	17	si	23
## 368	formaggio	18	no	57
## 369	burro	15	si	13
## 370	burro	13	no	7
## 371	pasta	15	no	15
## 372	pasta	15	no	79
## 373	latte	13	no	49
## 374	latte	16	no	49
## 375	latte	19	no	22
## 376	latte	18	no	71
## 377	pasta	17	no	6
## 378	burro	15	si	47
## 379	formaggio	15	si	70
## 380	formaggio	14	no	74
## 381	pasta	13	no	87
## 382	uova	20	no	62
## 383	uova	16	no	15
## 384	uova	12	si	7
## 385	pasta	17	no	46
## 386	uova	20	no	23
## 387	formaggio	19	no	35
## 388	formaggio	19	si	68
## 389	uova	15	si	100
## 390	burro	16	si	43
## 391	formaggio	20	si	68
## 392	pasta	11	no	75
## 393	latte	11	no	27
## 394	latte	17	si	52
## 395	pasta	13	si	33
## 396	burro	16	si	65
## 397	uova	12	no	31
## 398	formaggio	16	no	56
## 399	formaggio	17	si	29
## 400	uova	17	si	82
## 401	uova	18	si	70
## 402	pasta	12	no	76
## 403	burro	11	no	99
## 404	latte	16	si	61
## 405	latte	19	no	29

## 406	formaggio	12	si	34
## 407	uova	16	no	60
## 408	pasta	15	si	9
## 409	pasta	14	no	37
## 410	uova	19	si	68
## 411	uova	13	si	46
## 412	burro	14	si	83
## 413	formaggio	18	no	5
## 414	latte	13	no	41
## 415	uova	19	no	58
## 416	pasta	12	no	93
## 417	uova	17	no	19
## 418	uova	13	no	11
## 419	formaggio	13	no	92
## 420	formaggio	15	no	37
## 421	latte	16	si	70
## 422	uova	15	no	78
## 423	burro	13	si	50
## 424	uova	14	si	33
## 425	burro	16	no	76
## 426	pasta	16	no	94
## 427	burro	18	no	43
## 428	latte	20	no	5
## 429	pasta	12	no	47
## 430	formaggio	17	no	7
## 431	formaggio	19	si	7
## 432	uova	16	si	49
## 433	burro	17	no	7
## 434	latte	13	no	51
## 435	burro	20	no	50
## 436	uova	18	si	62
## 437	uova	12	no	96
## 438	pasta	12	no	32
## 439	latte	19	si	89
## 440	pasta	14	si	100
## 441	formaggio	11	no	87
## 442	latte	14	no	24
## 443	burro	16	no	64
## 444	latte	15	si	9
## 445	pasta	17	no	71
## 446	formaggio	17	no	29
## 447	burro	16	si	96
## 448	pasta	13	no	79
## 449	formaggio	16	si	74
## 450	burro	12	si	20
## 451	burro	18	si	99
## 452	latte	19	no	82
## 453	latte	14	si	37
## 454	pasta	12	no	63
## 455	latte	20	si	78
## 456	burro	19	no	4
## 457	latte	19	si	20
## 458	uova	17	si	9
## 459	latte	17	si	2

## 460	uova	14	si	94
## 461	pasta	18	no	61
## 462	pasta	19	no	82
## 463	burro	16	no	46
## 464	formaggio	18	si	21
## 465	pasta	11	si	30
## 466	burro	19	no	95
## 467	burro	19	no	34
## 468	pasta	14	no	53
## 469	pasta	14	si	58
## 470	burro	17	no	49
## 471	latte	13	no	7
## 472	formaggio	14	si	76
## 473	uova	11	si	54
## 474	pasta	13	si	33
## 475	burro	16	si	37
## 476	formaggio	14	si	81
## 477	uova	13	si	2
## 478	uova	14	no	22
## 479	latte	12	no	70
## 480	latte	12	si	30
## 481	pasta	15	si	31
## 482	uova	13	no	19
## 483	burro	18	si	28
## 484	burro	12	si	69
## 485	burro	19	no	45
## 486	pasta	19	si	39
## 487	burro	13	si	71
## 488	burro	20	si	71
## 489	pasta	20	si	69
## 490	formaggio	14	no	45
## 491	pasta	14	no	38
## 492	formaggio	20	si	11
## 493	latte	17	si	7
## 494	latte	14	si	30
## 495	latte	19	si	25
## 496	pasta	14	si	15
## 497	uova	13	no	37
## 498	latte	12	si	12
## 499	formaggio	19	si	50
## 500	pasta	17	no	78
## 501	burro	12	no	71
## 502	latte	14	si	49
## 503	uova	13	si	16
## 504	uova	16	si	25
## 505	latte	15	si	11
## 506	pasta	12	si	87
## 507	formaggio	13	si	96
## 508	pasta	19	no	27
## 509	burro	14	si	81
## 510	uova	18	no	2
## 511	pasta	19	si	60
## 512	latte	13	no	87
## 513	uova	18	no	87

```
## 514      uova      16      si      21
## 515      pasta     17      no      61
## 516      burro     11      si       7
## 517      latte     13      si     64
## 518      pasta     16      si     87
## 519      uova      15      si     46
## 520      burro     18      si     78
## 521      burro     17      no     35
## 522 formaggio     15      no     78
## 523      latte     11      no     40
## 524      latte     17      no     36
## 525      burro     11      si     11
## 526      latte     13      no     28
```

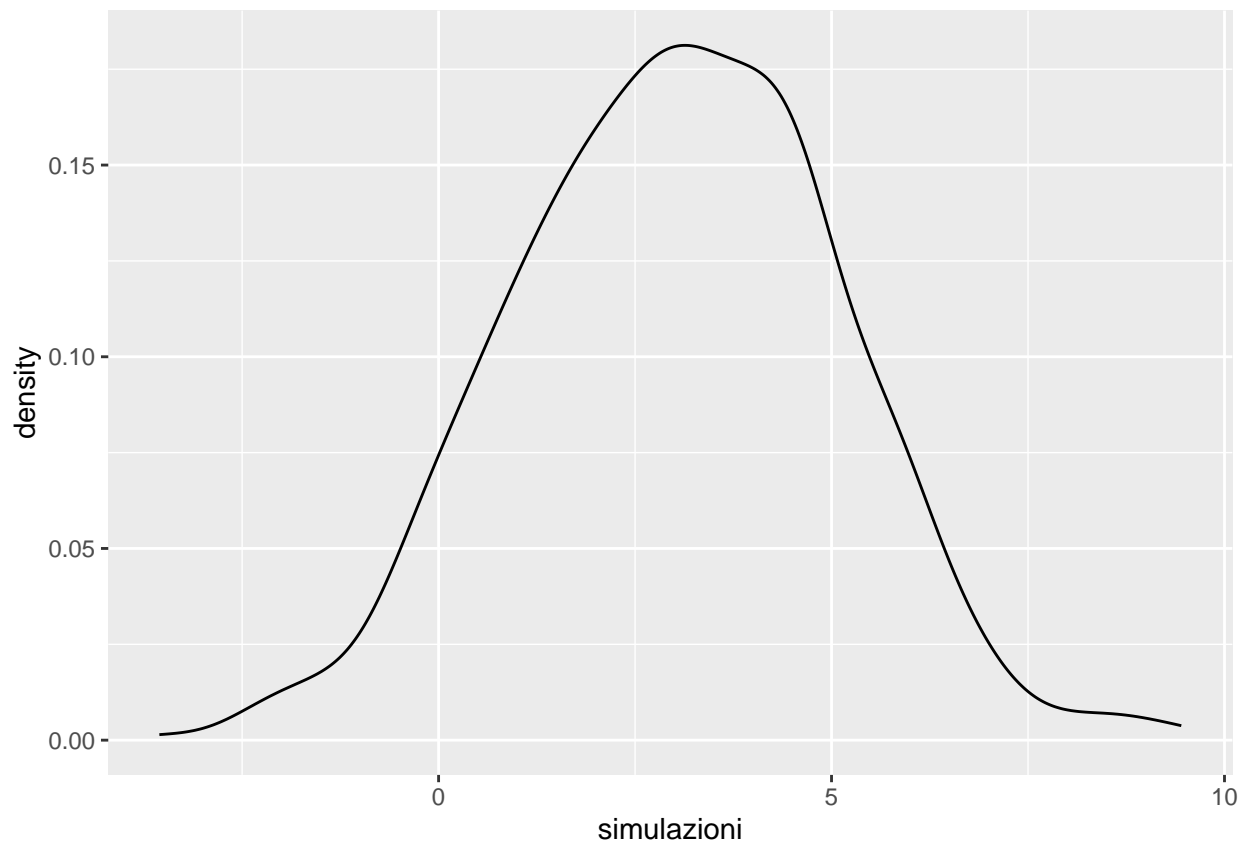
```
conteggio_prodotti <- supermercato %>%
  group_by(prodotti) %>%
  summarise("numero di prodotti" = length(prodotti))
media_prezzi <- supermercato %>%
  group_by(prodotti) %>%
  summarise(media = mean(prezzi))
print(media_prezzi)
```

```
## # A tibble: 5 x 2
##   prodotti media
##   <chr>      <dbl>
## 1 burro      11.5
## 2 formaggio  10.3
## 3 latte      10.1
## 4 pasta       11
## 5 uova      10.9
```

```
library(ggplot2)
```

```
## Warning: il pacchetto 'ggplot2' è stato creato con R versione 4.3.3
```

```
x <- runif(500, 0, 1)
y <- rnorm(700, mean = 3, sd = 2)
norm <- data.frame(simulazioni = y)
ggplot(norm, aes(x = simulazioni)) + geom_density()
```



my_df_filter: Implementa una funzione che filtri le righe di un dataframe in base a una condizione specificata.

```
my_df_filter <- function(x, condizione, colonna) {
  if(!is.data.frame(x)) {stop("L'argomento inserito non è un dataframe")}
  if(!(is.character(condizione) & (!is.numeric(condizione)))) {
    stop("La condizione deve essere un carattere oppure numerica")
  }
  nome_colonne <- names(x)
  col <- which(nome_colonne == colonna)
  #colonne <- ncol(x)
  df <- data.frame(x[which(x[,col] == condizione), ])
  colnames(df) <- nome_colonne
  return(df)
}
supermercato[which(supermercato[,2] ==17),]
```

```
##      prodotti prezzi disponibilità quantità
## 2   formaggio    17             no         77
## 5     burro     17             no         14
## 7      latte     17             no         12
## 12  formaggio    17             no          1
## 44  formaggio    17             no         67
## 67  formaggio    17             no          3
## 85     pasta     17             si         66
## 131 formaggio    17             no         48
## 140     burro     17             no         98
```

## 164	pasta	17	si	85
## 178	pasta	17	si	35
## 207	formaggio	17	si	91
## 236	pasta	17	si	81
## 237	formaggio	17	no	29
## 267	burro	17	si	79
## 271	uova	17	no	41
## 275	burro	17	no	37
## 281	formaggio	17	si	16
## 290	uova	17	no	41
## 319	pasta	17	no	36
## 358	burro	17	si	80
## 365	pasta	17	si	24
## 378	pasta	17	si	37
## 390	uova	17	no	15
## 397	latte	17	no	35
## 404	burro	17	si	58
## 425	burro	17	no	77
## 435	uova	17	si	60
## 446	formaggio	17	si	1
## 477	burro	17	si	54
## 493	latte	17	si	9
## 496	pasta	17	si	5
## 514	burro	17	si	72
## 555	formaggio	17	si	18
## 588	burro	17	si	10
## 595	pasta	17	no	60
## 615	pasta	17	si	17
## 616	burro	17	si	13
## 637	uova	17	no	20
## 642	pasta	17	si	62
## 673	burro	17	si	51
## 677	burro	17	no	75
## 682	formaggio	17	si	23
## 706	pasta	17	no	6
## 725	pasta	17	no	46
## 749	latte	17	si	52
## 756	formaggio	17	si	29
## 757	uova	17	si	82
## 783	uova	17	no	19
## 810	formaggio	17	no	7
## 815	burro	17	no	7
## 839	pasta	17	no	71
## 841	formaggio	17	no	29
## 869	uova	17	si	9
## 873	latte	17	si	2
## 897	burro	17	no	49
## 930	latte	17	si	7
## 948	pasta	17	no	78
## 980	pasta	17	no	61
## 987	burro	17	no	35
## 996	latte	17	no	36


```
my_df_filter(supermercato, "uova", "prodotti")
```

##	prodotti	prezzi	disponibilità	quantità
## 4	uova	10	si	16
## 6	uova	2	si	91
## 10	uova	9	si	95
## 13	uova	9	si	55
## 18	uova	20	si	67
## 19	uova	13	si	92
## 24	uova	13	no	7
## 26	uova	8	si	86
## 30	uova	6	no	33
## 34	uova	15	si	85
## 42	uova	13	si	51
## 49	uova	18	si	55
## 50	uova	19	si	69
## 55	uova	5	no	30
## 63	uova	20	si	62
## 69	uova	8	no	90
## 76	uova	1	si	87
## 77	uova	9	no	97
## 78	uova	3	si	65
## 82	uova	19	si	77
## 87	uova	15	si	11
## 93	uova	2	no	70
## 96	uova	12	no	88
## 107	uova	1	no	41
## 117	uova	10	si	54
## 118	uova	12	si	5
## 138	uova	9	si	28
## 139	uova	4	no	96
## 142	uova	3	si	80
## 145	uova	19	no	48
## 146	uova	5	no	85
## 147	uova	8	si	72
## 151	uova	15	si	66
## 163	uova	10	si	52
## 166	uova	16	si	72
## 169	uova	13	no	36
## 177	uova	6	si	33
## 179	uova	11	no	83
## 185	uova	5	si	94
## 187	uova	11	si	87
## 191	uova	19	no	30
## 194	uova	12	si	22
## 206	uova	4	no	47
## 211	uova	10	si	40
## 212	uova	15	si	25
## 214	uova	16	si	80
## 217	uova	14	no	20
## 232	uova	19	no	22
## 239	uova	7	no	13
## 240	uova	20	no	2

## 243	uova	13	no	68
## 246	uova	12	si	14
## 250	uova	16	no	60
## 258	uova	15	si	57
## 259	uova	18	si	78
## 265	uova	18	no	41
## 268	uova	8	no	82
## 271	uova	17	no	41
## 273	uova	15	si	22
## 277	uova	1	si	71
## 282	uova	12	no	75
## 285	uova	12	no	89
## 289	uova	4	si	68
## 290	uova	17	no	41
## 295	uova	13	si	10
## 298	uova	18	si	90
## 300	uova	13	si	96
## 307	uova	20	no	76
## 312	uova	19	no	20
## 333	uova	6	no	79
## 340	uova	3	si	18
## 345	uova	10	no	2
## 346	uova	20	si	91
## 349	uova	5	si	30
## 350	uova	19	no	47
## 363	uova	20	no	11
## 369	uova	7	si	26
## 374	uova	7	si	99
## 376	uova	11	no	90
## 382	uova	4	si	16
## 390	uova	17	no	15
## 396	uova	3	no	8
## 399	uova	13	si	38
## 411	uova	13	si	93
## 414	uova	18	si	49
## 415	uova	10	no	54
## 417	uova	1	si	92
## 418	uova	12	si	54
## 429	uova	11	si	75
## 435	uova	17	si	60
## 447	uova	20	no	84
## 448	uova	6	no	93
## 452	uova	10	si	27
## 466	uova	12	no	14
## 474	uova	9	no	34
## 480	uova	6	si	75
## 481	uova	13	si	44
## 487	uova	3	si	31
## 491	uova	20	no	23
## 497	uova	2	no	5
## 499	uova	18	no	42
## 500	uova	7	si	17
## 507	uova	14	si	58
## 509	uova	5	si	23

## 511	uova	8	no	46
## 512	uova	10	no	5
## 516	uova	19	si	62
## 526	uova	10	si	2
## 535	uova	9	si	59
## 548	uova	6	no	27
## 556	uova	11	si	84
## 561	uova	13	no	12
## 566	uova	19	si	69
## 567	uova	11	no	92
## 570	uova	18	no	99
## 576	uova	14	si	73
## 580	uova	7	no	21
## 584	uova	12	no	3
## 599	uova	10	si	82
## 600	uova	6	si	35
## 607	uova	9	no	83
## 611	uova	19	si	23
## 623	uova	12	si	76
## 625	uova	5	si	55
## 627	uova	7	no	70
## 633	uova	14	si	74
## 637	uova	17	no	20
## 647	uova	18	no	62
## 667	uova	10	no	67
## 671	uova	18	no	22
## 679	uova	10	si	53
## 680	uova	3	si	28
## 688	uova	3	si	63
## 703	uova	7	no	3
## 704	uova	6	no	60
## 705	uova	4	si	80
## 716	uova	3	no	81
## 719	uova	20	no	62
## 720	uova	16	no	15
## 721	uova	10	no	52
## 722	uova	12	si	7
## 726	uova	20	no	23
## 737	uova	7	no	69
## 738	uova	7	si	15
## 740	uova	4	si	26
## 741	uova	15	si	100
## 742	uova	2	si	86
## 744	uova	7	no	76
## 745	uova	3	si	39
## 753	uova	12	no	31
## 754	uova	3	si	32
## 757	uova	17	si	82
## 758	uova	18	si	70
## 760	uova	1	si	12
## 762	uova	1	no	49
## 767	uova	16	no	60
## 770	uova	19	si	68
## 771	uova	13	si	46

## 779	uova	8	no	58
## 781	uova	19	no	58
## 783	uova	17	no	19
## 784	uova	13	no	11
## 792	uova	7	si	57
## 793	uova	1	si	88
## 795	uova	15	no	78
## 797	uova	6	si	45
## 798	uova	8	no	17
## 799	uova	14	si	33
## 813	uova	3	no	55
## 814	uova	16	si	49
## 822	uova	18	si	62
## 823	uova	12	no	96
## 833	uova	9	no	17
## 835	uova	3	no	45
## 838	uova	4	no	35
## 847	uova	4	si	1
## 853	uova	4	si	27
## 855	uova	7	no	56
## 869	uova	17	si	9
## 875	uova	14	si	94
## 881	uova	9	si	38
## 888	uova	9	si	32
## 889	uova	5	no	93
## 901	uova	11	si	54
## 906	uova	13	si	2
## 907	uova	14	no	22
## 912	uova	13	no	19
## 921	uova	8	si	61
## 937	uova	6	no	83
## 941	uova	2	si	30
## 942	uova	13	no	37
## 953	uova	13	si	16
## 954	uova	8	no	87
## 955	uova	16	si	25
## 959	uova	5	si	89
## 971	uova	18	no	2
## 973	uova	7	si	61
## 975	uova	4	si	76
## 977	uova	18	no	87
## 979	uova	16	si	21
## 984	uova	15	si	46
## 989	uova	5	si	88
## 990	uova	10	si	76

```
names(supermercato)[names(supermercato) == "prezzi"] <- "k"
```