Primeros pasos en R

Los-Palmas

2023-03-29

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

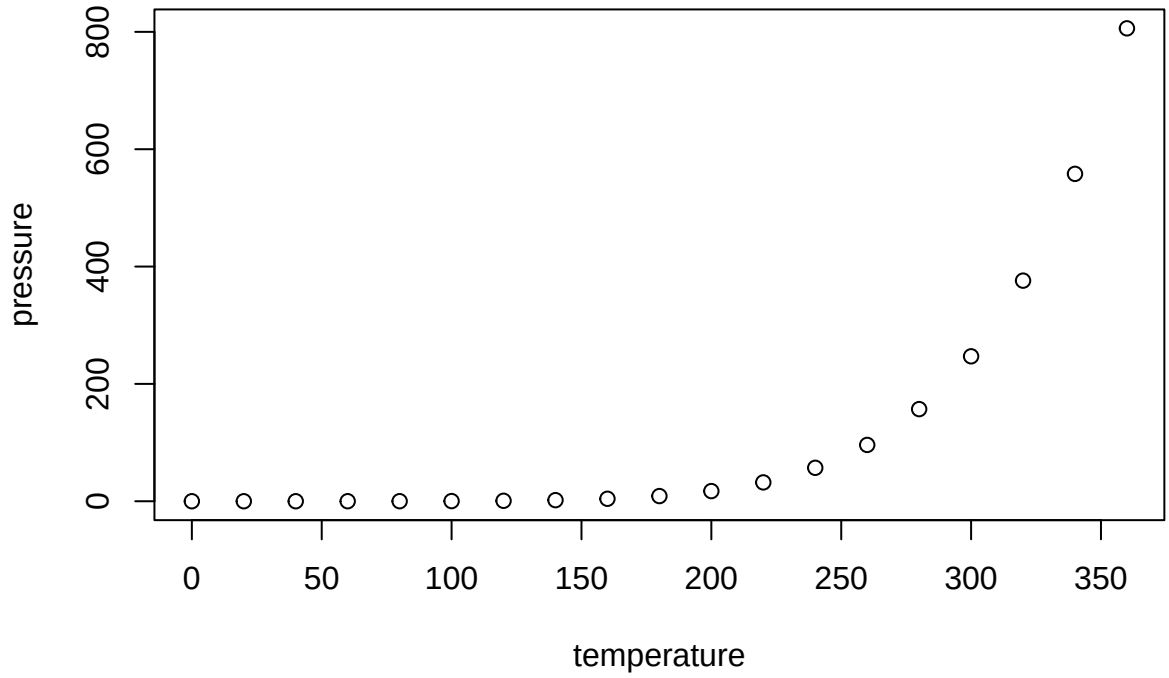
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Definición de variables numéricas

para definir variables numericas basta con darle un nombre a la variable y poner la flecha de asignación seguida del valor numérico que queremos asignar. en la ventana superior derecha aparecerá la variable con su valor

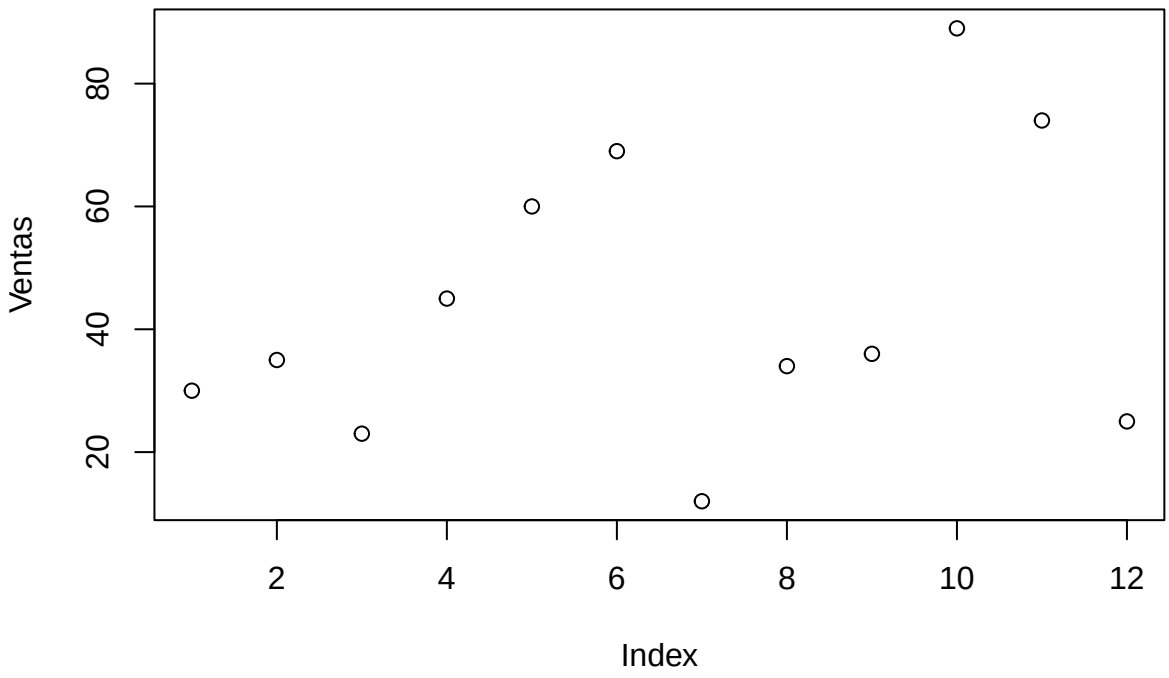
```
ca<-16
ca
```

```
## [1] 16
```

VECTORES

Los vectores son otra estructura de datos de R que a diferencia de las variables que vimos antes, alojan una colección de valores. Desde el punto de vista de objetos una variable simple es un vector que solo tiene un elemento.

```
Ventas <- c (30,35,23,45,60,69,12,34,36,89,74,25)
plot(Ventas)
```



Quisiera saber cuál es el nivel promedio de ventas anuales y el desvío estandar.

```
mean (Ventas)
```

```
## [1] 44.33333
```

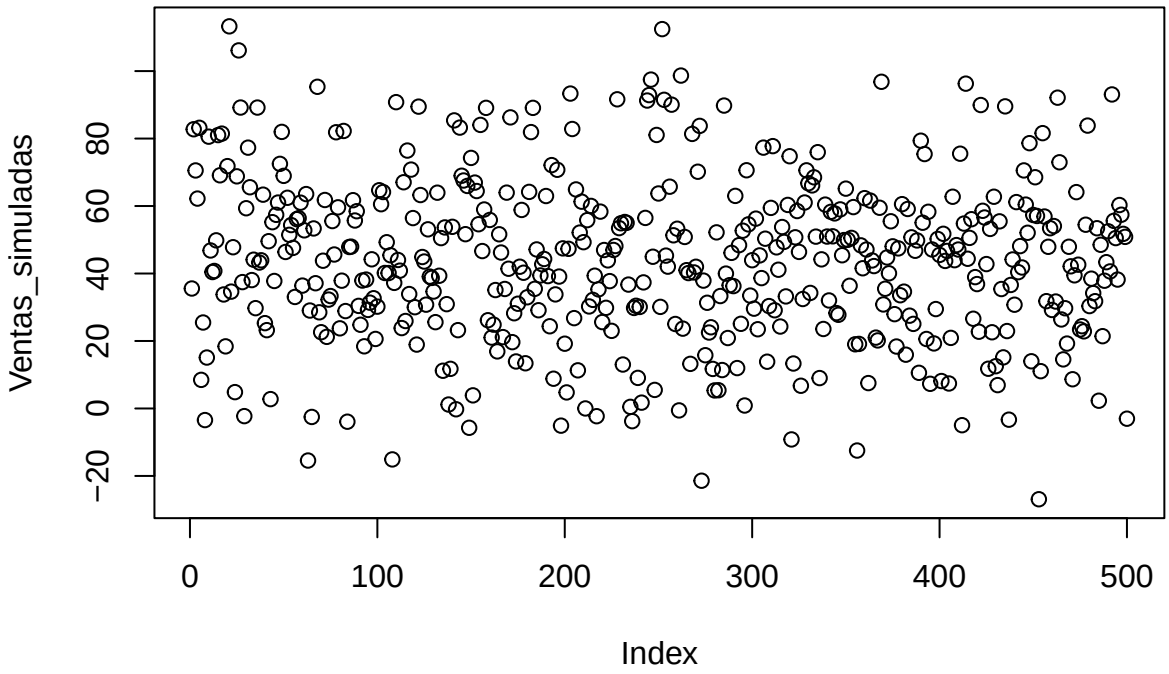
```
sd (Ventas)
```

```
## [1] 23.49597
```

###DATOS SIMULADOS

Queremos generar unos datos para meter en un simulador que representen a nuestra empresa. Vamos a generar 500 datos con el mismo promedio y desvío estándar.

```
Ventas_simuladas<-rnorm(500,44.33333,23.49597)
plot(Ventas_simuladas)
```



Qué probabilidad tengo de que las ventas sean menores a 30

```
pnorm(30,44.33,23.49)
```

```
## [1] 0.2709154
```

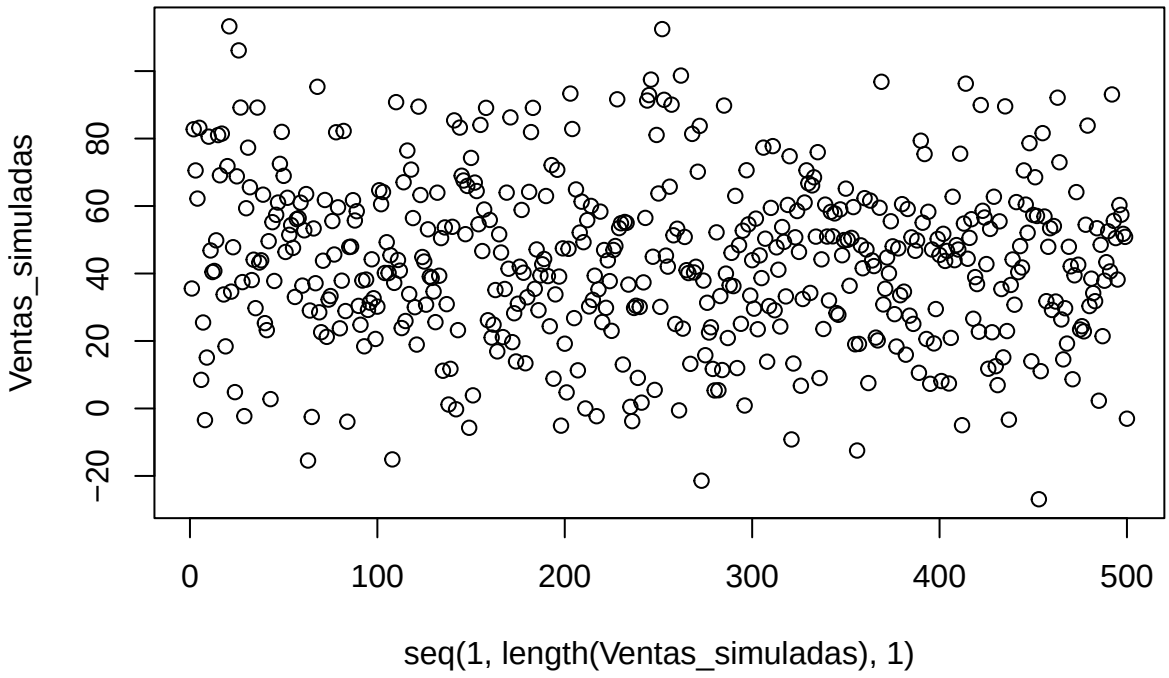
VARIABLES DE TIPO LISTA

Las variables de tipo lista son vectores pero que se generan con el comando "scan()". Como scan es un comando interactivo no podemos colocarlo dentro de un documento Rmarkdown. Sólo podemos ejecutar en la ventana de consola el comando.

```
compras <-scan()
```

##Comando length y seq length: funciona para evidenciar la longitud de datos con la que cuenta una variable definida anteriormente. seq: se utiliza para generar una secuencia de números con un valor inicial, un valor final y el rango entre valores que se van dando. Mediante estas funciones se puede generar una gráfica de un cierto valor sin necesidad de contar con dos datos de la siguiente manera:

```
plot(seq(1,length(Ventas_simuladas),1),Ventas_simuladas)
```



MÉTODOS ALTERNATIVOS DE GENERAR SECUENCIAS

A los programadores les encanta usar los métodos de programación que llevan comandos como “for”, “until”, “while” ; pero estos métodos son caros computacionalmente hablando. Como R trabaja con comando matricial podes hacerlo solo en una linea de comando.

```
impuestos<-0
impuestos[1] <- 0
for (i in 2:24) {
impuestos[i] <- impuestos[i-1]+2*i
impuestos
}
impuestos
```

```
##  [1]   0   4  10  18  28  40  54  70  88 108 130 154 180 208 238 270 304 340 378
## [20] 418 460 504 550 598
```

De la lista ventas_simuladas ¿Cuántos valores son superiores a 40?

```
indice <- which(Ventas_simuladas > 40)
indice
```

```
##   [1]   2   3   4   5  10  11  12  13  14  15  16  17  20  21  23  25  26  27
##  [19]  30  31  32  34  36  37  38  39  42  44  46  47  48  49  50  51  52  53
```

```
##  [37]  54  55  57  58  59  61  62  66  68  71  72  76  77  78  79  82  85  86
##  [55]  87  88  89  97 101 102 103 104 105 106 107 110 111 112 114 116 118 119
##  [73] 122 123 124 125 127 132 134 136 140 141 144 145 146 147 148 150 152 153
##  [91] 154 155 156 157 158 160 165 166 169 170 171 176 177 178 181 182 183 185
## [109] 188 189 190 193 196 199 202 203 204 206 208 209 210 212 214 219 221 223
## [127] 226 227 228 229 230 232 233 243 244 245 246 247 249 250 252 253 254 255
## [145] 256 257 258 260 262 264 265 266 268 269 270 271 272 281 285 289 291 293
## [163] 295 297 298 300 302 304 306 307 310 311 313 314 316 317 319 320 323 324
## [181] 325 328 329 330 332 333 334 335 337 339 340 342 343 344 347 348 349 350
## [199] 351 353 354 358 359 360 361 363 364 365 368 369 372 373 374 375 378 380
## [217] 383 385 387 388 390 391 392 394 396 399 400 402 403 404 407 408 409 410
## [235] 411 413 414 415 416 417 422 423 424 425 427 429 432 435 439 441 442 443
## [253] 444 445 446 447 448 450 451 452 455 456 458 459 461 463 464 469 470 473
## [271] 474 478 479 484 486 489 490 491 492 493 494 496 497 498 499
```

```
indice <-  which(Ventas_simuladas > 40)
indice
```

```
##   [1]   2   3   4   5  10  11  12  13  14  15  16  17  20  21  23  25  26  27
##  [19]  30  31  32  34  36  37  38  39  42  44  46  47  48  49  50  51  52  53
##  [37]  54  55  57  58  59  61  62  66  68  71  72  76  77  78  79  82  85  86
##  [55]  87  88  89  97 101 102 103 104 105 106 107 110 111 112 114 116 118 119
##  [73] 122 123 124 125 127 132 134 136 140 141 144 145 146 147 148 150 152 153
##  [91] 154 155 156 157 158 160 165 166 169 170 171 176 177 178 181 182 183 185
## [109] 188 189 190 193 196 199 202 203 204 206 208 209 210 212 214 219 221 223
## [127] 226 227 228 229 230 232 233 243 244 245 246 247 249 250 252 253 254 255
## [145] 256 257 258 260 262 264 265 266 268 269 270 271 272 281 285 289 291 293
## [163] 295 297 298 300 302 304 306 307 310 311 313 314 316 317 319 320 323 324
## [181] 325 328 329 330 332 333 334 335 337 339 340 342 343 344 347 348 349 350
## [199] 351 353 354 358 359 360 361 363 364 365 368 369 372 373 374 375 378 380
## [217] 383 385 387 388 390 391 392 394 396 399 400 402 403 404 407 408 409 410
## [235] 411 413 414 415 416 417 422 423 424 425 427 429 432 435 439 441 442 443
## [253] 444 445 446 447 448 450 451 452 455 456 458 459 461 463 464 469 470 473
## [271] 474 478 479 484 486 489 490 491 492 493 494 496 497 498 499
```

```
Ventas_simuladas[indice]
```

```
##   [1]  82.75636  70.54907  62.20499  83.18504  80.55881  46.82477  40.43363
##   [8]  40.76003  49.83444  81.02762  69.11179  81.46140  71.82738 113.25615
##  [15]  47.79018  68.80240 106.13166  89.19128  59.34781  77.30335  65.56560
##  [22]  44.09719  89.18774  43.25720  43.91739  63.36801  49.57804  55.21476
##  [29]  57.39702  61.03370  72.49850  81.97948  68.94383  46.38850  62.44350
##  [36]  51.46776  54.36196  47.59295  56.10728  56.40431  61.00247  52.79145
##  [43]  63.54719  53.35389  95.34361  43.79098  61.78923  55.54415  45.63354
##  [50]  81.86805  59.61494  82.24940  47.85401  47.96018  61.73475  55.71565
##  [57]  58.46592  44.20955  64.68630  60.53730  64.15047  40.15911  49.26739
##  [64]  40.12961  45.40930  90.78557  44.01392  40.84359  67.04703  76.44375
##  [71]  70.79981  56.43301  89.45729  63.26225  44.89162  43.54909  53.09412
##  [78]  63.95775  50.49321  53.65989  53.83773  85.38084  83.24627  68.98306
##  [85]  67.56560  51.65787  65.95496  74.26249  66.94954  64.44174  54.60376
##  [92]  84.04325  46.65020  58.97940  89.10035  55.87539  51.62606  46.21951
##  [99]  63.97219  41.41203  86.27716  42.01541  58.84074  40.22459  64.19139
## [106]  81.90904  89.09557  47.13292  42.73359  44.30832  62.96479  72.12455
```

```
## [113]  70.74653  47.45205  47.31897  93.33152  82.82812  64.91803  52.13225
## [120]  61.29391  49.28708  55.82397  60.03444  58.34654  46.96515  43.89503
## [127]  46.98145  48.14481  91.63078  53.45854  54.89390  55.27266  55.01826
## [134]  56.47453  91.26861  92.87938  97.47776  44.97197  81.06993  63.72009
## [141] 112.43698  91.48276  45.30437  42.03766  65.73443  90.04633  51.32641
## [148]  53.26793  98.66958  50.83465  40.99058  40.13389  81.37608  40.31529
## [155]  41.99813  70.16805  83.75551  52.19181  89.77696  46.12408  63.00818
## [162]  48.39929  52.69637  70.60344  54.56182  43.94148  56.26704  45.36224
## [169]  77.31534  50.31964  59.44640  77.75189  47.75418  41.12413  53.77468
## [176]  49.36486  60.32376  74.75644  50.73399  58.51771  46.39995  61.04298
## [183]  70.57683  66.74409  66.11217  68.51061  50.92712  75.95818  44.19695
## [190]  60.42633  50.94992  58.28028  51.02352  57.81142  58.99844  45.41047
## [197]  49.90893  65.15498  49.98872  50.55388  59.68780  48.33822  41.55977
## [204]  62.33842  47.10518  61.61443  43.87442  42.20167  59.48808  96.82327
## [211]  44.75995  40.06068  55.48321  48.05557  47.44081  60.55482  59.12660
## [218]  50.72155  46.73857  49.80288  79.36449  55.05756  75.40410  58.29773
## [225]  47.11709  50.24629  45.51087  51.77122  43.72860  46.68901  62.75736
## [232]  43.94812  48.42232  47.10732  75.50772  54.80551  96.27582  44.37116
## [239]  50.55697  56.08761  89.93173  58.58811  56.59425  42.79988  53.21279
## [246]  62.75305  55.45542  89.53620  44.16474  61.16650  40.40217  48.12214
## [253]  41.78221  70.55209  60.42388  52.07871  78.63618  57.33720  68.53230
## [260]  57.05425  81.60687  56.88980  47.91756  53.42326  54.07170  92.11557
## [267]  72.96842  47.92499  42.13587  64.12312  42.61849  54.46686  83.84199
## [274]  53.42085  48.51364  43.35616  52.50625  40.61629  93.07234  55.61217
## [281]  50.50679  60.29792  57.43041  51.75593  50.85709
```

```
length(indice)
```

```
## [1] 285
```

```
sum(Ventas_simuladas[indice])
```

```
## [1] 17125.87
```

```
A <- 0
 A[1] <- 0
 A[2] <- 1
 A[3] <- A[1]+A[2]
 for (i in 3:31) {A[i] <- A[i-1]+A[i-2]}
A
```

```
##  [1]      0      1      1      2      3      5      8     13     21     34
## [11]     55     89    144    233    377    610    987   1597   2584   4181
## [21]   6765  10946  17711  28657  46368  75025 121393 196418 317811 514229
## [31] 832040
```