

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>.

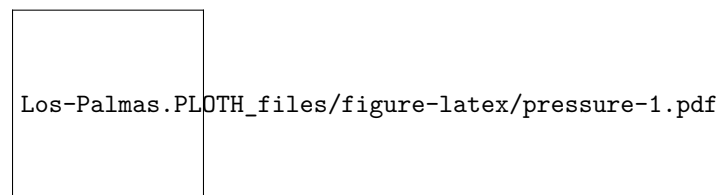
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
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)
```

Los-Palmas.PLOT_files/figure-latex/unnamed-chunk-2-1.pdf

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

```
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)
```

Los-Palmas.PLOT_files/figure-latex/unnamed-chunk-4-1.pdf

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 puedes 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] 1 5 6 10 11 12 13 15 19 21 22 24 25 28 31 35 36 38
## [19] 39 40 42 43 50 52 54 56 57 60 62 63 64 65 70 71 72 73
## [37] 74 75 77 81 82 83 84 85 86 87 88 91 94 95 96 97 98 100
## [55] 101 103 105 108 111 114 115 117 118 119 121 122 123 126 127 129 130 131
## [73] 132 133 134 135 140 144 146 147 148 152 153 154 155 156 158 159 160 161
## [91] 162 164 168 169 170 171 172 173 174 176 178 179 183 187 189 190 192 194
## [109] 195 196 197 198 199 200 202 204 205 207 208 213 214 220 221 222 223 224
## [127] 227 228 229 232 234 235 237 239 242 243 248 250 251 252 253 254 256 257
## [145] 259 260 262 263 265 266 267 269 271 272 275 276 278 279 280 281 282 283
## [163] 284 285 286 288 289 293 294 298 301 302 304 305 311 313 314 316 318 320
## [181] 321 323 325 326 329 330 331 333 335 337 338 342 343 345 347 348 350 351
## [199] 353 354 356 357 359 360 362 364 365 366 367 368 369 371 372 374 377 380
## [217] 381 382 383 384 385 386 387 389 390 391 393 395 396 397 399 400 401 402
## [235] 403 404 405 406 407 409 410 411 412 413 414 415 418 420 421 422 423 424
## [253] 428 430 431 432 433 434 439 442 445 446 449 452 454 455 457 459 461 463
## [271] 464 465 466 469 470 471 472 473 474 475 479 480 481 482 484 486 490 492
## [289] 494 495 497 498 499
```

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

```
## [1] 1 5 6 10 11 12 13 15 19 21 22 24 25 28 31 35 36 38
## [19] 39 40 42 43 50 52 54 56 57 60 62 63 64 65 70 71 72 73
## [37] 74 75 77 81 82 83 84 85 86 87 88 91 94 95 96 97 98 100
## [55] 101 103 105 108 111 114 115 117 118 119 121 122 123 126 127 129 130 131
## [73] 132 133 134 135 140 144 146 147 148 152 153 154 155 156 158 159 160 161
## [91] 162 164 168 169 170 171 172 173 174 176 178 179 183 187 189 190 192 194
## [109] 195 196 197 198 199 200 202 204 205 207 208 213 214 220 221 222 223 224
```

```
## [127] 227 228 229 232 234 235 237 239 242 243 248 250 251 252 253 254 256 257
## [145] 259 260 262 263 265 266 267 269 271 272 275 276 278 279 280 281 282 283
## [163] 284 285 286 288 289 293 294 298 301 302 304 305 311 313 314 316 318 320
## [181] 321 323 325 326 329 330 331 333 335 337 338 342 343 345 347 348 350 351
## [199] 353 354 356 357 359 360 362 364 365 366 367 368 369 371 372 374 377 380
## [217] 381 382 383 384 385 386 387 389 390 391 393 395 396 397 399 400 401 402
## [235] 403 404 405 406 407 409 410 411 412 413 414 415 418 420 421 422 423 424
## [253] 428 430 431 432 433 434 439 442 445 446 449 452 454 455 457 459 461 463
## [271] 464 465 466 469 470 471 472 473 474 475 479 480 481 482 484 486 490 492
## [289] 494 495 497 498 499
```

```
Ventas_simuladas[indice]
```

```
## [1] 46.83590 65.06948 62.57818 78.08376 63.44202 51.04693 69.50536
## [8] 80.17992 56.65650 43.41466 86.33916 55.87132 67.63577 80.77591
## [15] 57.66507 48.26090 43.07597 43.80674 82.95567 92.93352 42.96170
## [22] 73.06230 54.46559 55.45667 50.49488 61.18249 58.27293 79.01149
## [29] 47.17709 60.68620 68.81592 66.31432 103.61212 66.30513 48.14868
## [36] 46.17706 72.69778 72.32891 52.97242 52.06689 47.06352 65.54953
## [43] 69.08880 71.40270 93.75076 65.22550 45.08963 69.37027 100.73970
## [50] 65.71235 69.17884 45.17688 55.24433 47.87294 65.06033 48.25283
## [57] 60.20354 68.04090 76.57312 51.39925 46.25547 47.47019 52.81494
## [64] 44.28321 46.66421 45.32959 51.19392 47.41955 50.56035 48.95895
## [71] 44.09387 59.82841 65.81790 40.96309 41.69537 87.93686 60.13538
## [78] 56.20144 45.14951 40.02510 72.56484 78.52203 68.76971 52.12912
## [85] 80.13125 66.52998 65.03406 75.77379 73.44528 68.23126 43.10722
## [92] 45.57066 41.49521 55.29596 57.31101 73.96757 49.72153 72.29063
## [99] 73.78732 55.46618 48.62803 67.11449 67.40882 70.21603 43.65238
## [106] 82.48330 53.35530 63.06894 44.19358 44.87700 69.30868 73.48288
## [113] 54.61914 55.98856 43.72287 43.63435 87.32974 64.59781 50.23357
## [120] 44.65155 103.42677 43.65234 70.66736 68.76149 46.03182 56.51420
## [127] 65.86297 51.08347 48.17442 89.96903 45.32177 81.43243 63.13641
## [134] 61.39159 40.62683 64.98608 63.67618 67.94737 47.04349 57.67598
## [141] 69.40909 60.79579 67.43536 58.09524 41.73270 80.19839 71.75655
## [148] 104.88335 58.45636 61.31048 47.20133 64.23371 47.65124 73.21156
## [155] 58.66257 103.72873 63.39367 57.47385 42.65484 45.35237 57.48588
## [162] 95.94052 54.30842 56.61086 100.50435 78.05939 95.43012 54.60479
## [169] 49.35941 64.76063 51.78323 63.02639 65.98076 70.26215 91.89876
## [176] 48.57516 47.00610 49.49892 52.02966 50.78687 69.69635 47.31255
## [183] 49.18656 62.44071 46.90104 45.71765 62.13423 40.09063 70.97419
## [190] 49.33610 81.39472 40.46337 71.76063 46.96214 43.29234 56.74785
## [197] 54.35525 56.85113 60.86633 59.70632 55.69210 56.22333 69.22331
## [204] 85.06353 73.10240 53.39555 100.95110 42.81126 46.91479 58.01282
## [211] 65.79041 42.63375 44.36882 52.87135 40.98439 47.12779 54.15367
## [218] 49.44845 82.64871 60.47237 46.83504 74.53677 65.17254 49.12791
## [225] 43.21780 46.61284 45.67518 54.26274 65.68117 50.72620 75.24049
## [232] 52.80365 72.36384 69.37702 51.64563 41.68487 73.94420 47.96121
## [239] 73.08975 66.37168 61.48163 52.92111 64.84065 48.73291 72.70966
## [246] 52.94036 58.19006 86.10732 59.03566 53.51746 53.63011 58.78376
## [253] 50.03728 64.52788 40.01586 52.65996 47.09590 49.48303 58.27739
## [260] 60.10112 74.26223 54.91876 53.08556 54.20004 62.41656 48.90475
## [267] 52.35011 73.39410 47.13783 68.65180 108.39637 56.87751 69.62121
## [274] 70.05964 56.47492 87.05177 65.32172 48.76354 50.00732 50.34906
## [281] 59.35363 56.85577 81.36812 89.62171 44.32806 45.50121 66.17820
## [288] 84.12112 75.27379 44.04976 51.17513 50.05501 88.33963
```

```
length(indice)
```

```
## [1] 293
```

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

```
## [1] 17763.75
```

```
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
```