Simulación ITAM
2 de noviembre de 2021 2021-II

Tarea 4.

La fecha de entrega es el **25 de octubre de 2021**.

Lecturas

- Introduction to simulation using R Capítulo 13
- Marc C. Bove, et. al Effect of El Niño on U.S. Landfalling Hurricanes, Revisited
- Dagpunar, Capítulo 7, Secciones 7.1-7.3

Problemas

1. Consideren este problema de Manufactura. La compañía A tiene una configuración de dos máquinas, cada una de las cuales produce un componente por hora. Cada componente puede ser probado de manera instantánea para identificar si es defectuoso o no defectuoso. Sea a_i la probabilidad de que un componente producido por la máquina i sea no defectuoso, $i = 1, 2$. Los componentes defectuosos son desechados y los no defectuosos producidos por cada máquina se almacenan en dos armarios separados. Cuando un componente está presente en cada armario, los dos se ensamblan instantáneamente y se envían. Cada armario puede mantener a lo más dos componentes. Cuando un armario está lleno, la máquina correspondiente se apaga. Se prende de nuevo cuando el armario tiene espacio para al menos un componente.

   - Modelar este proceso como una cadena de Markov.
   - Suponiendo que $a_1 = 0.3$ y $a_2 = 0.5$, generar 500 pasos de la cadena. Digan qué condiciones iniciales usaron para esa trayectoria. ¿Qué proporción del tiempo pasa cada una de las máquinas apagadas?

   Solución.

   - Este problema suena un poco complicado en el setup porque hay eventos anidados, por un lado es el producto que se genera de cada máquina, y por el otro, se tiene el almacenamiento. En realidad lo que define el proceso son los estados finales, es decir, el estado de los almacenes.

Definimos A_n el número de productos en el almacén de la máquina A y B_n el número de componentes de la máquina B, ambos al final de la n-ésima hora. Tenemos que $A_n, B_n \in \{0, 1, 2\}$. Notemos que los almacenes no pueden estar no vacíos simultáneamente, porque el ensamble es simultáneo. Así que $A_n > 0$ implica que $B_n = 0$ y simultáneamente. Sea X_n el estado del almacen en la hora n. Tenemos que X_n toma valores en $\{(0,0), (0,1), (0,2), (1,0), (2,0)\}$. Otra opción es definir X_n como la diferencia $A_n - B_n$, con espacio de estados $\{-2, -1, 0, 1, 2\}$. Cualquiera de las dos opciones funciona bien. Usando la última versión, tenemos que:

$$P = \left\{ \begin{array}{ccccc} 1-a_1 & a_1 & 0 & 0 & 0 \\ (1-a_1)a_2 & a_1a_2 + (1-a_1)(1-a_2) & a_1(1-a_2) & 0 & 0 \\ 0 & (1-a_1)a_2 & a_1a_2 + (1-a_1)(1-a_2) & a_1(1-a_2) & 0 \\ 0 & 0 & (1-a_1)a_2 & a_1a_2 + (1-a_1)(1-a_2) & a_1(1-a_2) \\ 0 & 0 & 0 & a_2 & 1-a_2 \end{array} \right\}$$

- Para los valores dados, generamos X, considerando $X_0 = 0$. Adicionalmente, los casos en donde las máquinas están apagadas son las rachas de -2 o 2.

```
a1 <- 0.3
a2 <- 0.5
a <- (1-a1)*a2
b <- a1*a2 + (1-a1)*(1-a2)
c <- a1*(1-a2)
P <- matrix(c(1-a2,a,0,0,0,
a1,b,a,0,0,
0,c,b,a,0,
0,0,c,b,a2,
0,0,0,c,1-a2),nrow= 5)

X <- 0  # estado inicial
S <- c(-2,-1,0,1,2)
for(i in 1:500){
ind <- which(X[i] == S) # fija el índice
X[i+1] <- sample(S,size = 1,prob = P[ind,])
}
X

  [1]  0  0 -1 -2 -2 -1 -1  0  1  2  2  1  1  1  1  1  0  0  0  0  0  0 -1 -1  0
 [26]  0 -1 -2 -1 -2 -2 -2 -2 -2 -1 -1 -2 -1 -1 -1 -2 -2 -2 -1 -2 -1  0 -1 -1 -2
 [51] -1 -1 -2 -1 -1 -1 -2 -1 -1 -1  0 -1 -2 -2 -2 -2 -1 -2 -2 -1 -2 -2 -2 -2 -1
 [76] -1 -1 -1 -2 -2 -2 -2 -1 -1 -2 -2 -2 -1 -1 -2 -1 -2 -2 -2 -1 -1  0  1  1  2
[101]  2  1  2  1  1  1  1  1  1  0 -1 -2 -1  0 -1 -1 -1 -1 -2 -1 -1 -1 -1 -2 -2
[126] -1 -2 -2 -2 -2 -1 -1 -2 -1 -1 -2 -2 -2 -1 -2 -1 -2 -2 -2 -2 -2 -2 -2 -2 -2
[151] -2 -2 -2 -2 -2 -1  0  1  1  1  1  1  1  1  0  0  0 -1  0 -1 -2 -2 -2 -2 -1
[176] -1 -1 -2 -1 -1 -1 -2 -2 -1 -1 -2 -1 -2 -2 -2 -1 -1 -2 -1 -2 -2 -1 -1  0  1
[201]  1  0  0  0 -1 -2 -2 -2 -1 -1 -1 -1 -2 -2 -2 -2 -1 -1 -1  0  1  1  1  2  1
[226]  1  1  0 -1 -2 -2 -2 -1  0  0  1  1  2  2  1  1  2  2  1  1  0  1  0 -1 -1
[251] -1  0 -1 -1 -2 -1 -1 -1 -1 -2 -2 -1 -1 -1 -1  0 -1  0  0  0 -1 -1 -1 -1 -1
[276] -2 -2 -2 -2 -2 -2 -2 -2 -1 -1 -2 -1 -1  0  0  0  1  0  0 -1 -1 -1 -1 -2 -2
[301] -2 -2 -1 -1 -1 -1 -1 -2 -1 -1 -1 -2 -2 -2 -2 -2 -2 -2 -2 -1  0 -1 -2 -2 -2
[326] -1 -1 -2 -2 -2 -2 -2 -2 -2 -2 -1 -2 -2 -2 -2 -1 -2 -1 -1  0 -1 -2 -2 -2 -2
[351] -2 -2 -2 -2 -1  0  0  0 -1 -1  0  0  1  1  2  1  0 -1 -2 -2 -1  0  1  0  0
[376]  0  0  1  1  2  1  1  2  1  2  2  1  0  1  1  1  1  0 -1 -2 -2 -2 -2 -2 -2
[401] -2 -2 -2 -2 -2 -1 -2 -1 -1 -1 -1  0 -1 -2 -1  0 -1 -1 -2 -1 -2 -1 -1 -1 -1
[426] -1 -2 -1 -2 -2 -2 -2 -2 -2 -2 -1 -1 -1 -1 -1  0  1  2  2  2  1  0 -1 -2 -1
[451] -2 -2 -1 -1  0  0  0  0  1  0  0  0 -1 -2 -2 -1 -1 -1 -2 -1 -1 -1 -2 -1 -2
[476] -2 -2 -2 -2 -2 -1 -1  0 -1 -1 -1 -2 -1  0  0  0 -1  0  0  1  1  0 -1 -1 -1
[501] -2

a <- table(X)/length(X)    # proporciones de valores observados.
sum(a[c(1,5)])  # suma los casos de -2 o 2

[1] 0.3892216
```

□

2. Para un proceso Poisson no homogéneo con función de intensidad dada por

$$\lambda(t) = \begin{cases} 5, t \in (1,2], (3,4], \cdots \\ 3, t \in (0,1], (2,3], \cdots \end{cases}$$

   a) Grafiquen una ejemplo del proceso considerando el intervalo de tiempo [0,100].

   b) Grafiquen el proceso hasta obtener 100 eventos

   c) Estimen la probabilidad de que el número de eventos observados en el periodo de tiempo (1.25,3] es mayor que 2.

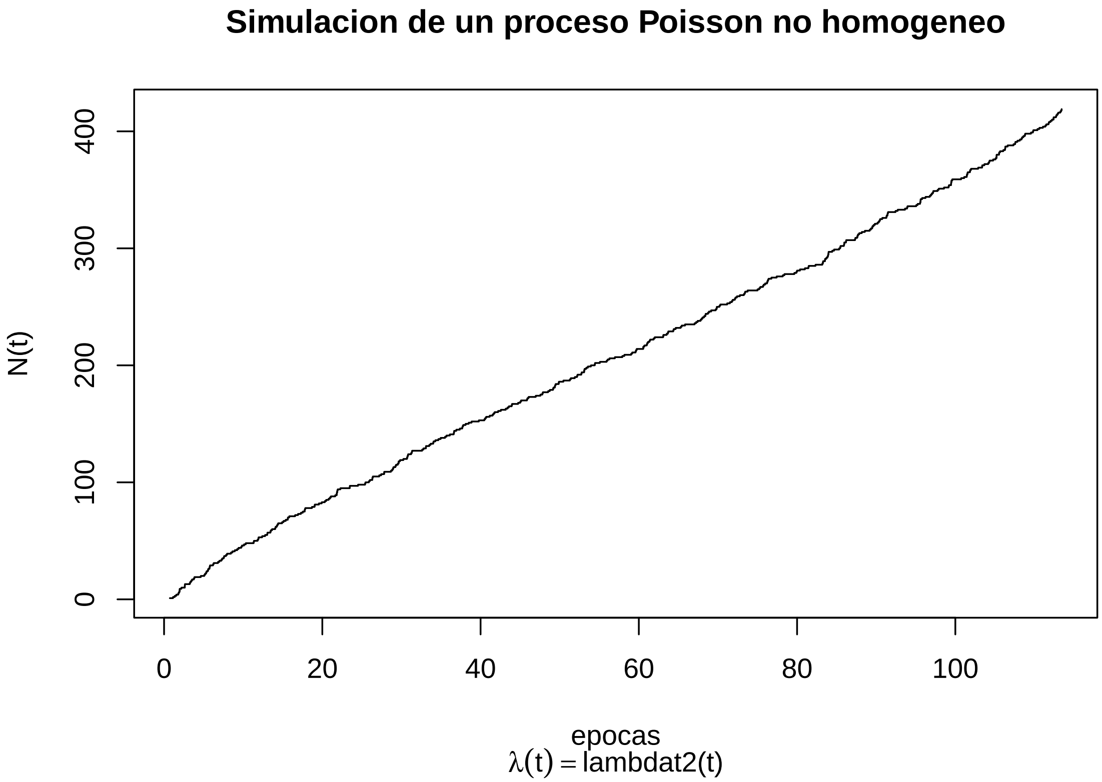
Solución.

Para el primer problema, tenemos que definir la función pulso. Debido a que en el proceso de aceptación-rechazo se pierden algunas observaciones, el numero de observaciones necesarios para llegar al tiempo 100 se obtiene por ensayo y error, o bien, cambiar la programación y primero generar todas las exponenciales hasta que se acumule el número que necesitamos. Yo hice la otra programación.

```
lambdat2 <- function(t){
                                x <- paste("","{",0,"<= t & t <",1,"}",sep="")
                                for(i in seq(2,100,2)){x <- paste(x,paste("","{",i,"<= t & t <",i+1,"}",sep=""),sep="|")}
                                return(ifelse(eval(parse(text=x)),3,5))
                        }

ppnh <- function(lambdat,n,pic=T){
                        lambda <- 5            # mayoriza la función lambdat
                        TT <- rexp(n,lambda)   # genera variables exponenciales para los tiempos.
                        s <- cumsum(TT)        # acumula los tiempos en el vector s
                        u <- runif(n)          # obten n uniformes
                        ss <- s[u <= lambdat(s)/lambda] # obten los tiempos que cumplen la condición de aceptación
                        Ns <- 1:length(ss)     # Conteo
                        if(pic==T){
                                plot(ss, Ns, type = "s", xlab = "epocas", ylab = "N(t)",
                    main = "Simulacion de un proceso Poisson no homogeneo",
                                sub = expression(lambda(t) == paste(lambdat2,"(t)")))
                        }
                        return(list(epocas = ss, cuenta=  Ns))
                }

ppnh(lambdat = lambdat2, n = 510)
```



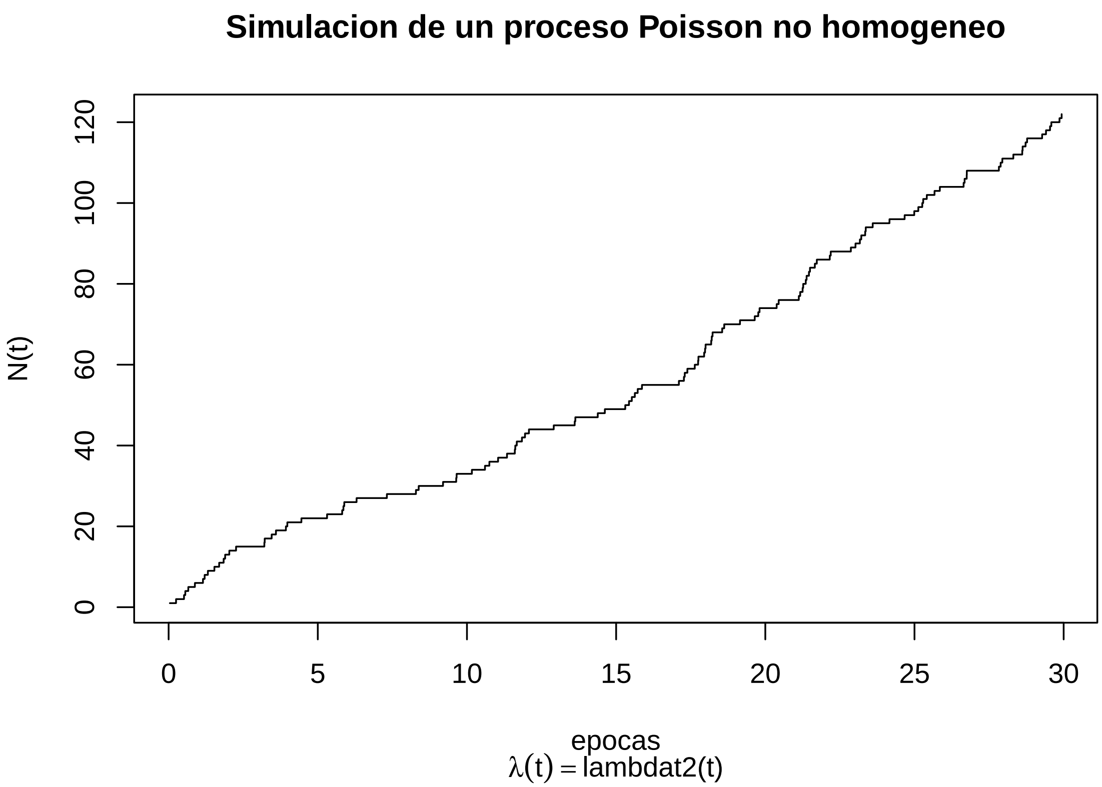

```
$epocas
  [1]   0.7273535   1.1338008   1.3552472   1.5469604   1.7917484   1.8586496
  [7]   1.9387997   1.9673649   1.9741642   2.1908091   2.6207073   2.6236586
 [13]   2.6330046   3.2525420   3.3158688   3.4334635   3.5402580   3.7688992
 [19]   3.8384862   4.6394271   5.0858394   5.1887024   5.3124164   5.3714443
 [25]   5.5394629   5.5803193   5.7337350   5.7607577   5.8031695   6.1889744
 [31]   6.2693518   6.8147633   6.9900900   7.2493361   7.3549049   7.5414199
 [37]   7.5936208   7.8296747   7.9571245   8.4497609   8.6361922   8.9391282
 [43]   9.2392034   9.4091465   9.7495532   9.8434041  10.1063639  10.3245953
 [49]  11.3202017  11.3536483  11.8205068  11.9155830  11.9416496  12.3941958
 [55]  12.7684581  13.0431094  13.0629136  13.4462693  13.4843982  13.6436229
 [61]  14.0450121  14.0925341  14.2463440  14.3243700  14.4181247  14.9181117
 [67]  15.0975058  15.3700316  15.6381201  15.6463827  15.8204983  16.5615922
 [73]  16.9260204  17.2945940  17.5281280  17.7755107  17.7847989  17.8448141
 [79]  18.6870383  19.0211184  19.0440896  19.5691781  19.9205975  20.3156347
 [85]  20.4870939  20.7934705  20.9552478  21.0773011  21.6277807  21.8217409
 [91]  21.8239780  21.8730442  21.8919720  21.9418455  22.2840036  23.4720915
 [97]  23.4752403  24.5115071  25.3989402  25.4491098  25.8892038  26.0076481
[103]  26.3219599  26.3281609  26.3839011  27.1900769  27.4378156  27.8147747
[109]  27.8207874  28.6684819  28.8317401  28.9342260  28.9846293  29.2387442
[115]  29.3083967  29.5323066  29.6447735  29.6481687  29.8176106  30.2426176
[121]  30.7011055  30.7454994  30.7937562  30.8417129  31.2248349  31.3084350
[127]  31.3440529  32.6135376  32.7776891  33.0916930  33.0989957  33.5226132
[133]  33.6766503  34.0180673  34.0523241  34.2876682  34.6648736  34.9505334
[139]  35.5122092  35.6913163  36.1153665  36.6078178  36.6378108  36.6621682
[145]  36.9350670  37.3768803  37.6833465  37.7237535  37.8066341  38.1055837
[151]  38.4971614  38.8591864  39.7788447  40.4823632  40.5911093  40.7049497
[157]  41.1542778  41.5206517  41.6484244  41.7762769  42.2162619  42.5629989
[163]  43.1771297  43.4110118  43.5747921  43.9313959  43.9747378  44.7425470
[169]  45.0155948  45.1113615  45.8644247  45.9273589  46.0672420  46.9707690
[175]  47.5619032  47.8119583  47.8690824  48.5353052  48.7486323  49.1548129
[181]  49.2051229  49.3892360  49.4165069  49.4850327  49.8756650  49.8924494
[187]  50.4651693  51.2817923  51.3981107  51.9067635  52.1907779  52.2464224
[193]  52.7282716  52.7688026  53.0879191  53.0960012  53.1460518  53.3590002
[199]  53.5632005  53.9452399  54.4333029  54.4613283  55.0807249  55.9313770
[205]  56.0565203  56.2869749  56.9731566  57.9193479  58.1728275  59.0344076
[211]  59.1627575  59.5924753  59.6443139  59.7232180  60.5595843  60.5645086
[217]  60.7034270  61.0069995  61.0571172  61.1490030  61.3399563  61.4231372
[223]  61.8643597  62.0081260  63.0758238  63.0981780  63.5186951  63.6688987
[229]  63.7202409  64.3526218  64.4073313  64.6519161  65.2926738  65.4189049
[235]  65.8392633  67.0452981  67.2382194  67.4318599  67.7997264  67.9307289
[241]  68.0556411  68.2512601  68.4094513  68.4407055  68.7331961  68.8649878
```

```
[247]  69.1396618  69.7168906  69.8341127  69.8521323  70.1886434  70.2920934
[253]  71.1638953  71.5137203  71.7535805  71.8584489  72.1345316  72.2095194
[259]  72.3985560  72.7671913  73.2779864  73.3857405  73.4382792  73.7458136
[265]  75.0038445  75.2236345  75.3476847  75.6878215  75.7766490  75.9596430
[271]  76.2055628  76.2512934  76.3192955  76.3880338  76.7548409  77.4162306
[277]  78.1917689  78.3991957  79.6563693  79.9158596  79.9925320  80.3648888
[283]  80.9842333  81.4345778  81.4633398  82.3256598  83.1909296  83.2622338
[289]  83.2850392  83.5277234  83.5341853  83.6983583  83.8389134  83.9054786
[295]  83.9354277  83.9692231  83.9805892  84.4445654  84.6743640  85.3136458
[301]  85.4501956  85.4863637  85.9143311  85.9574491  85.9620968  86.1845758
[307]  86.2052994  87.3290688  87.3680089  87.6043988  87.6402261  87.7089351
[313]  87.8591309  88.1779908  88.5622743  89.1796012  89.3270288  89.4958634
[319]  89.5286618  89.6696823  89.8249070  90.1659207  90.3146763  90.4246748
[325]  90.4910635  90.7660317  91.2894780  91.3162132  91.4249657  91.4424916
[331]  91.4936230  92.4454103  92.7130238  93.6381506  93.9067281  93.9515450
[337]  95.0595397  95.2357893  95.5413642  95.5849048  95.6059042  95.6106720
[343]  95.8002589  96.2378695  96.7728502  96.9177850  97.0326046  97.1693794
[349]  97.2151813  97.7616178  97.8971839  98.5690271  99.1534576  99.1873865
[355]  99.4614831  99.4620068  99.5011058  99.5074935  99.6119207 100.7333294
[361] 101.1189023 101.4443589 101.4547555 101.5265310 101.5539204 101.8455666
[367] 101.8758377 101.9811766 102.8835619 103.3866524 103.4013592 103.6745246
[373] 104.1571065 104.2654400 104.3244028 104.7799636 105.0812331 105.2024795
[379] 105.2098205 105.2156497 105.5248935 105.5364964 105.6430823 106.0577667
[385] 106.2766484 106.3084718 106.3166256 106.6129007 107.3500631 107.5614022
[391] 107.5930692 107.8483937 108.1501134 108.3644955 108.4473698 108.6051796
[397] 108.7912095 108.8266174 109.5585691 109.8048012 109.8568226 110.3798668
[403] 110.6225358 111.0668179 111.3753811 111.4955393 111.7806925 111.8323500
[409] 112.0295372 112.2075713 112.4143631 112.4174161 112.7236258 112.7796756
[415] 112.9012131 113.0033867 113.2675867 113.3736240 113.4364990

$cuenta
  [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
 [19]  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
 [37]  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
 [55]  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
 [73]  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
 [91]  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
[109] 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
[127] 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
[145] 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162
[163] 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
[181] 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198
[199] 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216
[217] 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234
[235] 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252
[253] 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270
[271] 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288
[289] 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306
[307] 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324
[325] 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342
[343] 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360
[361] 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378
[379] 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396
[397] 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414
[415] 415 416 417 418 419
```

Para obtener 100 eventos es similar, con un tamaño de muestra mucho menor:

```r
ppnh(lambdat2,140)
```




```
$epocas
  [1]  0.04039113  0.24935863  0.51599309  0.55848366  0.65933261  0.88046690
  [7]  1.15154011  1.20753218  1.31547205  1.53444666  1.69366642  1.84370409
 [13]  1.89500143  2.03437802  2.26081417  3.20845370  3.22012695  3.45423840
 [19]  3.59726034  3.93118377  3.97844706  4.45047340  5.31090233  5.81735468
 [25]  5.85828583  5.88702871  6.29981204  7.31614669  8.28936309  8.38249804
 [31]  9.19739366  9.63954746  9.65288273 10.16520545 10.60398328 10.75092080
 [37] 11.04279787 11.34200183 11.60457762 11.61883600 11.66966291 11.84250622
 [43] 11.94570465 12.07708084 12.90849278 13.61128674 13.63221226 14.38410399
 [49] 14.62152599 15.30570955 15.43152207 15.52518385 15.62638568 15.72245934
 [55] 15.86545641 17.10338734 17.27165516 17.29981642 17.38725535 17.63570106
 [61] 17.74785355 17.75816091 17.94986639 17.98100713 17.99754043 18.18625454
 [67] 18.20390612 18.23235409 18.55404022 18.62207310 19.15168047 19.64607340
 [73] 19.76414017 19.80664220 20.38101868 20.44866766 21.12340188 21.16891551
 [79] 21.25035737 21.26964963 21.35613678 21.38409462 21.46292756 21.49768124
 [85] 21.66018703 21.72651005 22.16111979 22.19293020 22.86818213 23.02312898
 [91] 23.16984549 23.21647198 23.34774400 23.36585282 23.59986403 24.16125895
 [97] 24.66976494 24.99224042 25.12875085 25.26055104 25.29161946 25.41235056
[103] 25.67167890 25.85011150 26.64663804 26.68012655 26.75038610 26.75423746
[109] 27.82954415 27.88771601 27.94400168 28.31294276 28.61195933 28.62444473
[115] 28.72345757 28.77479452 29.27816136 29.40847380 29.54418031 29.58673558
[121] 29.86083569 29.93309720

$cuenta
  [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
 [19]  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
 [37]  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
 [55]  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
 [73]  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
 [91]  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
[109] 109 110 111 112 113 114 115 116 117 118 119 120 121 122
```

Por último, para estimar la probabilidad en el intervalo dado, lo que podemos hacer es obtener N simulaciones, y calcular la proporción de esas simulaciones que dan un valor de conteo mayor a 2 en ese intervalo.

```r
fr <- NULL
N <- 10000
for(i in 1:N){
                x <- ppnh(lambdat2,10,pic=F);
                fr <- c(fr,ifelse(1.25 < x$epocas[x$cuenta==2] & x$epocas[x$cuenta==2] <3,1,0))}
sum(fr)/N
```

```
[1] 0.0781
```

□

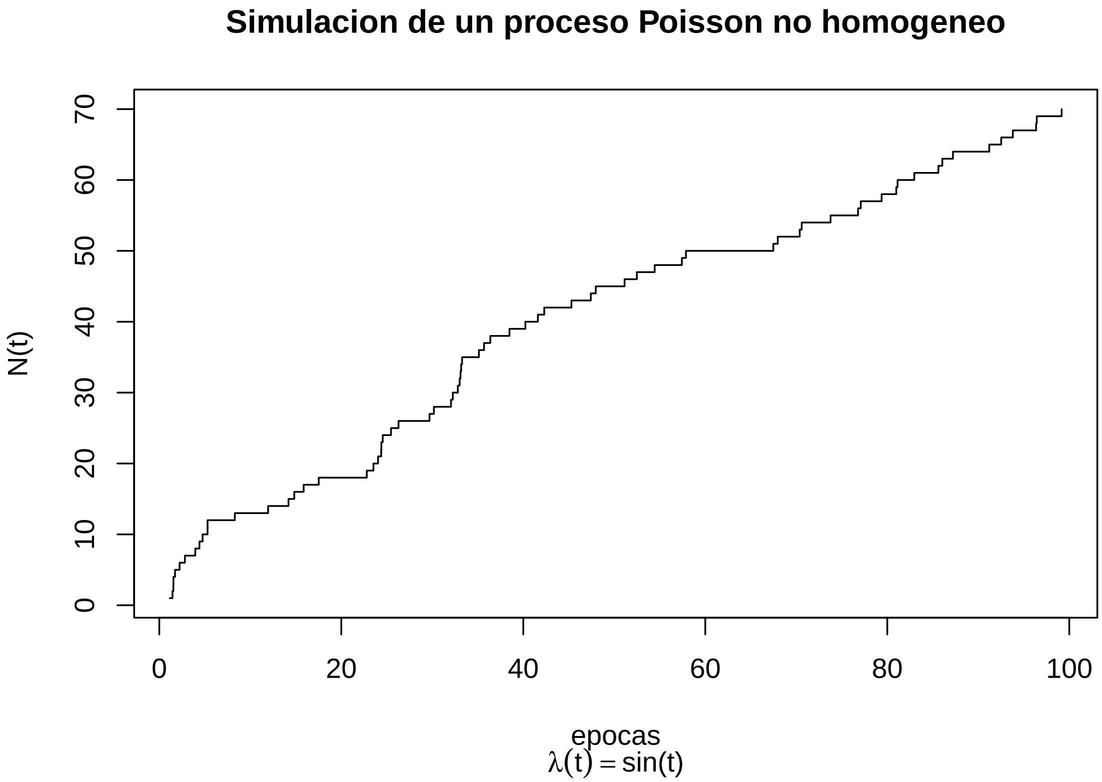
3. Simular un proceso Poisson no homogéneo con función de intensidad dada por $\lambda(t) = |sen(t)|$.

   Solución.

```r
lambdat <- function(t){abs(sin(t))}

ppnh<- function(lambdat,n){
                                                 lambda <- 1 #mayoriza la función lambdat
                                                 TT <- rexp(n,lambda) #genera variables exponenciales para los tiempos.
                                                 s <- cumsum(TT) #acumula los tiempos en el vector s
                                                 u <- runif(n) #obten n uniformes
                                                 ss <- s[u <= lambdat(s)/lambda] #obten los tiempos que cumplen la condición de aceptación
                                                 Ns <- 1:length(ss) # Conteo
                                         plot(ss, Ns, type = "s", xlab = "epocas", ylab = "N(t)",
                                                          main = "Simulacion de un proceso Poisson no homogeneo",
                                                          sub = expression(lambda(t) == paste("sin","(t)")))
                                                 return(list(epocas = ss, cuenta=  Ns))
                                          }

x <- ppnh(lambdat,100)
```




```
x


$epocas
 [1]  1.156444  1.447589  1.545649  1.552236  1.727585  2.229193  2.816492
 [8]  3.953588  4.409725  4.761165  5.300725  5.304903  8.299511 11.958019
[15] 14.199565 14.827901 15.863703 17.521399 22.803352 23.532837 24.038860
[22] 24.385218 24.404631 24.552686 25.464197 26.289372 29.691469 30.174817
[29] 32.057273 32.252766 32.806239 33.006611 33.103138 33.168834 33.271402
[36] 35.121519 35.685809 36.374653 38.481711 40.229621 41.603202 42.301236
[43] 45.292080 47.421639 47.964861 51.129163 52.482674 54.439863 57.427575
[50] 57.873739 67.474235 67.965336 70.379400 70.593735 73.767571 76.787640
[57] 77.085360 79.379497 80.993811 81.132659 82.963705 85.615073 86.050630
[64] 87.219433 91.208226 92.519156 93.798390 96.362216 96.430616 99.163415

$cuenta
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
[26] 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
[51] 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70
```

□

4. Una máquina es sujeta a shocks que llegan de dos fuentes independientes. Los shocks de la fuente 1 llegan de acuerdo a un proceso Poisson con tasa de 3 por día y los de la fuente 2 a una tasa de 4 por día. ¿Cuáles son la media y varianza del número total de shocks que llegan de ambas fuentes en un turno de 8 hrs?

   - ¿Cuál es la probabilidad de que la máquina se encuentre sujeta a un total de exactamente dos shocks en una hora?

- Supongan que un shock de la fuente i puede causar que la máquina falle con probabilidad p_i, independiente de todo lo demás. Supongan que $p_1 = 0.011$ y $p_2 = 0.005$ La máquina descompuesta se reemplaza de manera inmediadta. Sea $N(t)$ el número de reemplazos de la máquina sobre el intervalo $(0, t]$. ¿Es $\{N(t), t \geq 0\}$ un proceso Poisson, y si lo es, ¿cuál es el parámetro?

Solución.

- Para la primera parte de la pregunta, sólo hay que notar que el total de shocks N_t corresponde a una superposición de procesos Poisson $N_t^{(1)}$ y $N_t^{(2)}$, $N_t = N_t^{(1)} + N_t^{(2)}$, por lo que el parámetro de la superposición es $\lambda_1 + \lambda_2 = 3 + 4 = 7$. en un periodo de 8 horas, se tiene que $E(N_t) = 7 \times 8/24 = 56/24$ y la varianza en el proceso Poisson es la misma que la media, $\text{Var}(N_t) = 56/24$.
- La probabilidad de dos shocks en una hora se obtiene considerando que $N_1 \sim Poi(7)$. Por lo tanto es $P(N_1 = 2)$

```
dpois(2,lambda= 7)

[1] 0.02234111
```

- Este problema puede ser tratado como un adelgazamiento de cada uno de los procesos $N_t^{(1)}$ y $N_t^{(2)}$ y luego a una superposición de los procesos ya adelgazados. Cada proceso $N^{(i)}$ se puede escribir como la suma de los shocks que descomponen la máquina y los que no lo descomponen. Sea $R^{(i)}$ la parte del proceso que descompone la máquina. Estos procesos son procesos adelazados que tienen parámetro $p_i * \lambda_i$. EL proceso $R_t^{(1)} + R_t^{(2)}$ que corresponde al número de reemplazo de máquinas es un proceso Poisson con parámetro $p_1 * \lambda_1 + p_2 * \lambda_2 = 0.011 \times 3 + 0.005 \times 4 = 0.053$.

$\square$

5. Las ocurrencias de huracanes que tocan tierra durante el fenómeno meteorológico "el Niño" se modelan como un proceso Poisson (ver Bove et al (1998)). Los autores aseguran que "Durante un año de 'El Niño', la probabilidad de dos o más huracanes haciendo contacto con tierra en los estados Unidos es 0.28". Encontrar la tasa del proceso Poisson.

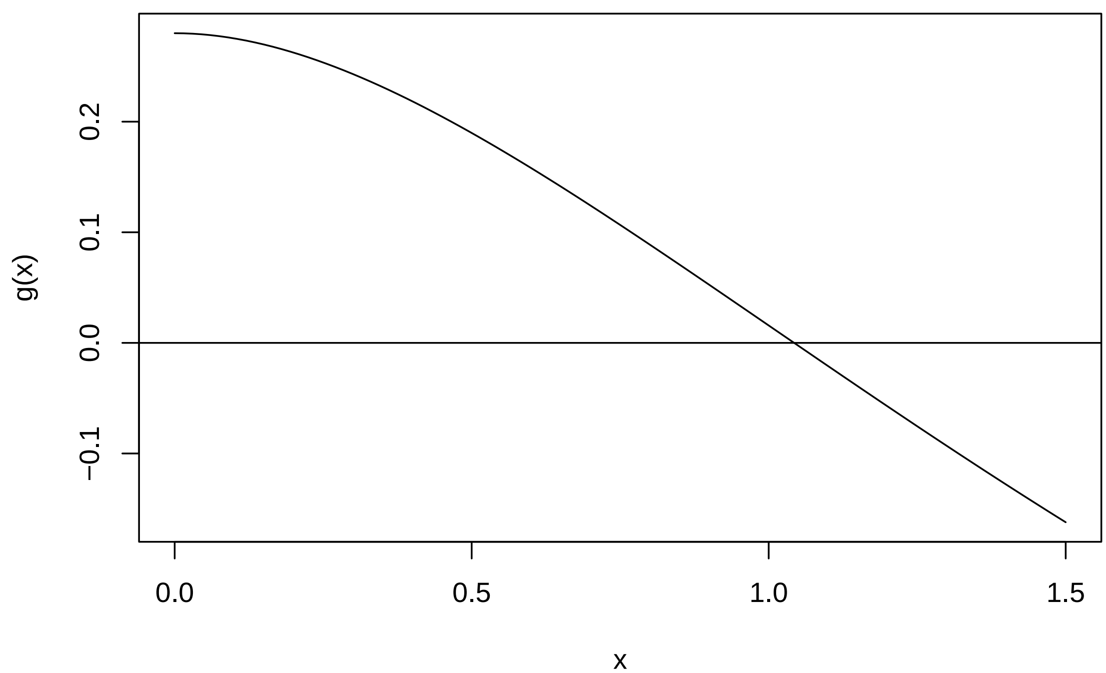
   Solución.

   Este problema dice que un proceso Poisson con parámetro λ cumple la siguiente condición: $P(N_2 \geq 2) = 0.28$. Entonces tenemos que encontrar λ.

$$
\begin{aligned}
P(N_2 \geq 2) &= 1 - \sum_{i=0}^{1} \frac{e^{-\lambda}\lambda^i}{i!} \\
&= 1 - e^{-\lambda}(1+\lambda) \\
&= 0.28
\end{aligned}
$$

Tenemos que resolver la ecuación: $e^{-\lambda}(1+\lambda) = 0.78$

```
g <- function(x) exp(-x)*(1+x)-0.72
curve(g,0,1.5)
abline(h=0)
```




```
a <- uniroot(g,interval=c(1,1.5))
a


$root
[1] 1.042851

$f.root
[1] -3.173615e-07

$iter
[1] 3

$init.it
[1] NA

$estim.prec
[1] 6.103516e-05
```

Entonces $\lambda = 1.0428507824926$

$\square$

6. Comenzando a mediodía, los comensales llegan a un restaurante de acuerdo a un proceso Poisson a una tasa de 5 clientes por minuto. El tiempo que cada cliente pasa comiendo en el restaurante tiene una distribución exponencial con media de 40 minutos, independiente de otros clientes e independiente de los tiempos de arribo. Encuentra la distribución así como la media y varianza, del número de comensales en el restaurante a las 2:00pm. Simular el restaurante para verificar los resultados obtenidos.

Solución.

Sea X_t el número de clientes que llegan en el tiempo t. De acuerdo al problema, X_t sigue un proceso Poisson con media 5 cada hora. Sea t_i el tiempo de servicio del cliente i. De acuerdo al problema, $t_i \sim \exp(1/40)$

Nos piden determinar el número de clientes al tiempo t. Sea N_t este número. La distribución de N_t se puede obtener condicional al número de clientes que han llegado al restaurante y los que ya están ahí. Entonces:

$$P(N_t = k) = \sum_{n=k}^{\infty} P(N_t = k | X_t = n) P(X_t = n) = \sum_{n=k}^{\infty} P(N_t = k | X_t = n) \frac{e^{-\lambda t}(\lambda t)^n}{n!}$$

Si los n clientes en el restaurante llegaron en $s_1, \ldots, s_n$, entonces los clientes dejan el restaurante en los tiempos $s_i + t_i$. Ahora, noten que hay k clientes en el restaurante si y sólo si hay k índices tales que $s_i + t_i > t$, por lo que, recordando que los tiempos de arribo se distribuyen como las estadísticas de orden, se tiene

$$\begin{aligned} P(N_t = k | X_t = n) &= P(k \text{ de los } n \text{ valores } s_i + t_i > t | X_t = n) \\ &= P(k \text{ de los } n \text{ valores } u_{(i)} + t_i > t | X_t = n) \\ &= P(k \text{ de los } n \text{ valores } u_i + t_i > t | X_t = n) \\ &= \binom{n}{k} p^k (1-p)^{n-k} \end{aligned}$$

La probabilidad de éxito en este caso es $P(U_1 + t_1 > t) = \frac{1}{t}\int_0^t P(t_1 > t - x)dx = \frac{1}{t}\int_0^t (1 - F(t-x))dx = \frac{1}{t}\int_0^t (1 - F(x))dx$.

$$
\begin{aligned}
P(N_t = k) &= \sum_{n=k}^{\infty} \binom{n}{k} p^k (1-p)^{n-k} \frac{e^{-\lambda t}(\lambda t)^n}{n!} \\
&= \frac{p^k(\lambda t)^k}{k!} \sum_{n=k}^{\infty} (1-p)^{n-k} \frac{e^{-\lambda t}(\lambda t)^{n-k}}{(n-k)!} \\
&= \frac{p^k(\lambda t)^k}{k!} e^{\lambda(1-p)t} e^{-\lambda t} \\
&= \frac{p^k(\lambda t)^k}{k!} e^{-\lambda p t}
\end{aligned}
$$

Así que el número de restaurantes es un proceso adelgazado (podemos separarlo en los clientes que siguen y en los clientes que ya se fueron) y la probabilidad depende de la distribución de los tiempos de arribo. Para nuestro problema, $t = 120$ (dos horas después del mediodía y con unidad de tiempo el minuto) $p_t = \frac{40(1-e^{120/40}}{120} = 0.3167$, y por lo tanto el número esperado de clientes en el cliente a las 2 de la tarde es $5 \times p \times 120 = 190.0426$

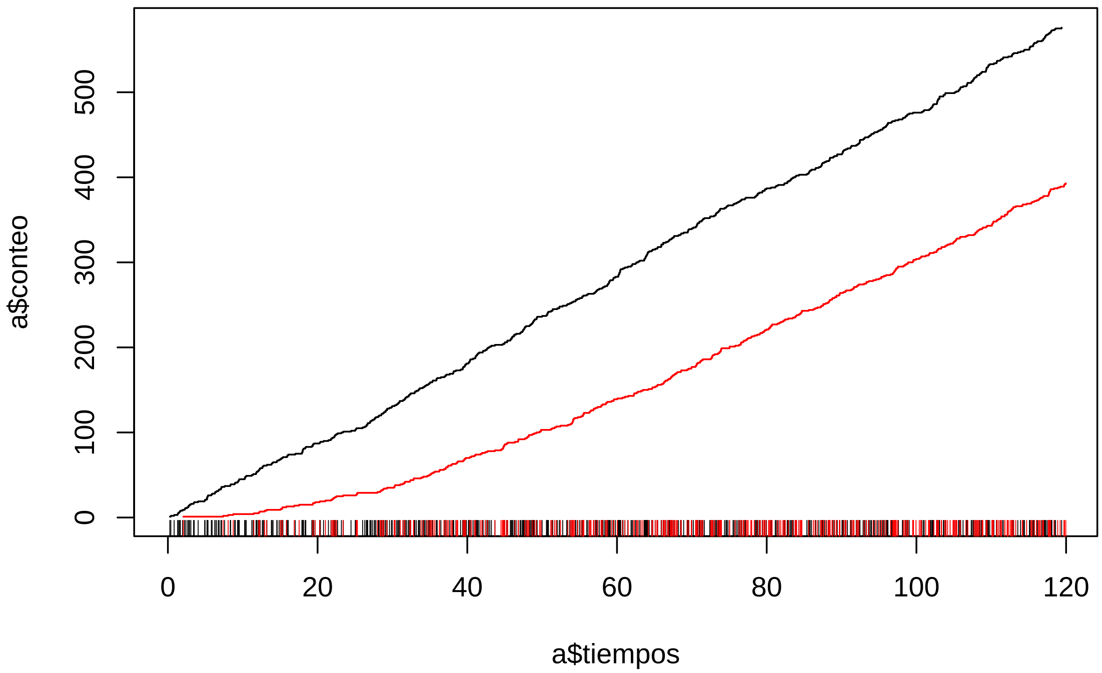
Simulando el sistema:

```
# Simula un proceso Poisson

pp <- function(t,lambda){
N <- rpois(1,lambda*t)
u <- runif(N,0,t)
return(list(tiempos=sort(u),conteo = 1:N, total = N))
}
a <- pp(120,5)  # proceso Poisson(5) evluado n t = 120
plot(a$tiempos,a$conteo,type = "s")

# Sobre los tiempos, sumamos los tiempos que corresponden al tiempo que tardan en comer
# cada uno de los clientes
tc <- rexp(a$total,rate = 1/40)

rug(a$tiempos)
nt <- a$tiempos + tc
rug(nt[nt<=120],col = "red")
# Conteo de los que se van antes de las 2 hrs.
lines(sort(nt[nt<120]),1:length(nt[nt<120]),col = "red", type ="s")
```




```
# Simula para calcular la media de los que no se han ido a las 2 hrs

res <- NULL
for(i in 1:10000){
a <- pp(120,5)
nt <- a$tiempos + rexp(a$total,rate = 1/40)
res[i] <- length(nt[nt>120])
}

mean(res)


[1] 190.3767


sd(res)


[1] 13.82229
```

□

7. Sea X_t que satisface la ecuación diferencial estocástica $dX_t = -\frac{1}{3}\,dt + \frac{1}{2}\,dZ_t$, donde $X_0 = 0$ y Z_t es un proceso de Wiener estándar. Definan $S_t = e^{X_t}$ así que $S_0 = 1$.

   - Encontrar la ecuación diferencial estocástica que sigue S_t
   - Simular 10 trayectorias de S_t para $t = 1, \ldots, 30$. Llamen a esas trayectorias S_t^i, $i = 1, \ldots, 10$ y grafíquenlas en la misma gráfica.
   - ¿Qué se puede concluir sobre S_t para t grande?

- Con $n = 10$. evaluar $\bar{S}_{30} = \frac{1}{n}\sum_{i=1}^{n} S_{30}^{i}$.
- Simular 100 trayectorias independientes y evaluar $\bar{S}_{100}$ con la fórmula dada arriba con $n = 100$. ¿Qué se puede concluir sobre $\bar{S}_{1000}$ cuando $n \to \infty$?

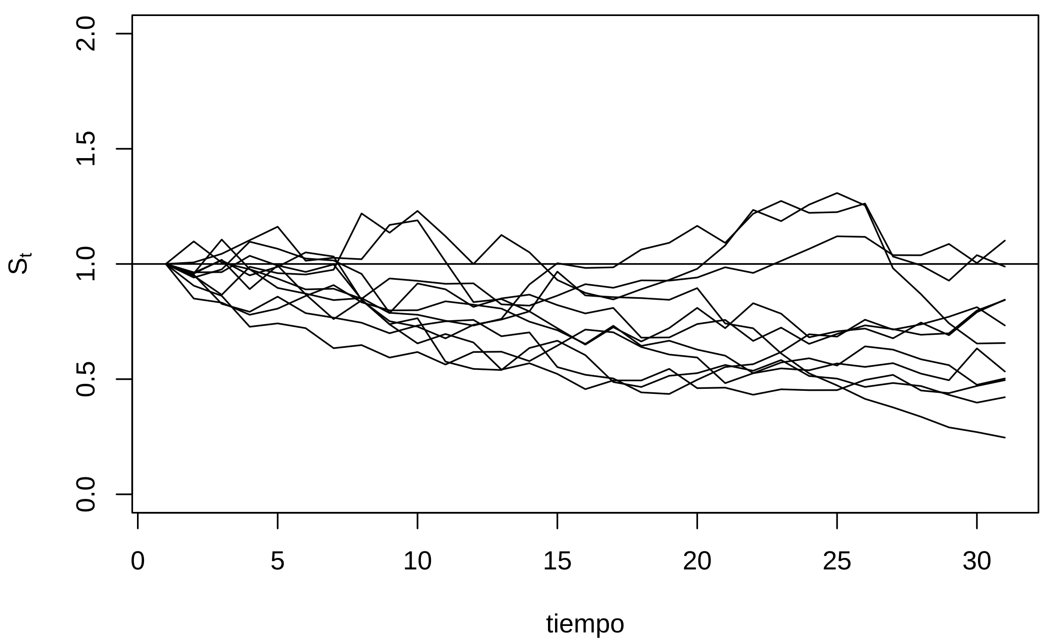
Solución.

- Aplicando la fórmula de Ito, que hicimos en clase, a $G(X,t) = e^{X_t}$, obtenemos que $dS_t = S_t dt + S_t dZ_t$
- Para simular las trayectorias:

```
BGeo <- function(n, TT, a, b, S0 = 1){
  # Función para generar un proceso Browniano Geométrico
  # n es el número de puntos de partición del intervalo [0,TT]
  # a es el drift y b la volatilidad

dt <- TT/n #incremento de los intervalos para cubrir [0,TT]
S <- S0 #valor inicial
for(i in 2:(n+1)){
 S <- append(S, S[i-1]*exp((a-b^2/2)*dt + b*sqrt(dt)*rnorm(1)))
}
return(S)
}

plot(BGeo(n = 30,TT = 1,a = -0.3, b = 0.5, S0 = 1), type = "l",
     ylim = c(0,2), xlab = "tiempo", ylab = expression(S[t]))
abline(h = 1)
for(i in 1:10) lines(BGeo(30,1,-0.3,0.5,1))
```



- Podemos ver que la varianza crece cuando t crece como ya sabemos, pero la media parece tender a ser negativa.

- ```r
  S30 <- NULL
  for(i in 1:10)  S30[i] <- BGeo(30,1,-0.3,0.5,1)[30]
  S30
  ```

  ```
   [1] 0.5557008 0.6357064 0.5186123 0.3473275 0.5402613 0.5262218 1.6966094
   [8] 0.4293865 0.6659437 0.4962864
  ```

  ```r
  mean(S30)
  ```

  ```
  [1] 0.6412056
  ```

  ```r
  sd(S30)
  ```

  ```
  [1] 0.3818819
  ```

- Este es como el ejercicio de arriba pero con $n = 100$:

  ```r
  S100 <- NULL
  for(i in 1:100)  S100[i] <- BGeo(100,1,-0.3,0.5,1)[100]
  S100
  ```

  ```
    [1] 0.3512556 0.8686633 0.7049163 0.3212150 1.6971439 0.5596593 0.6422189
    [8] 0.4282768 1.7164081 0.7671254 0.6137362 0.2190055 0.9951213 0.8871927
   [15] 1.9903307 0.4131432 0.9338997 0.3826419 0.4786844 0.6100982 0.6155368
   [22] 0.8543113 0.5460138 0.6055372 0.8016313 0.9110656 0.8308149 0.3201241
   [29] 0.7926637 1.0749062 1.7716437 0.8792847 0.4451177 1.1642340 0.6037655
   [36] 1.1904333 0.4320932 0.4637130 0.4555791 0.5390291 1.3195169 0.7366638
   [43] 0.4360978 1.0191959 0.9353211 0.8419030 0.8251485 0.6100356 0.3695865
   [50] 1.4624807 0.3885751 1.0022101 1.4955577 0.2972225 0.6932528 0.4998177
   [57] 0.3434639 0.2846758 1.0397888 0.8600553 0.7152472 1.0482260 1.1346993
   [64] 0.6905379 0.2786404 0.7119424 0.8884609 0.6209028 0.8986302 0.4303360
   [71] 1.2412902 0.9868548 1.2476295 0.4411096 1.2329850 1.1875648 0.5701531
   [78] 0.3927023 0.6974068 0.3669915 2.2625719 0.7486518 1.0622894 0.9549669
   [85] 0.3905040 0.3748865 1.4023212 1.5182162 0.3404229 0.7250680 0.4327181
   [92] 0.6205835 1.4254467 0.4114172 0.4232662 0.6877344 0.4098998 0.7464361
   [99] 1.2900240 0.3551852
  ```

  ```r
  mean(S100)
  ```

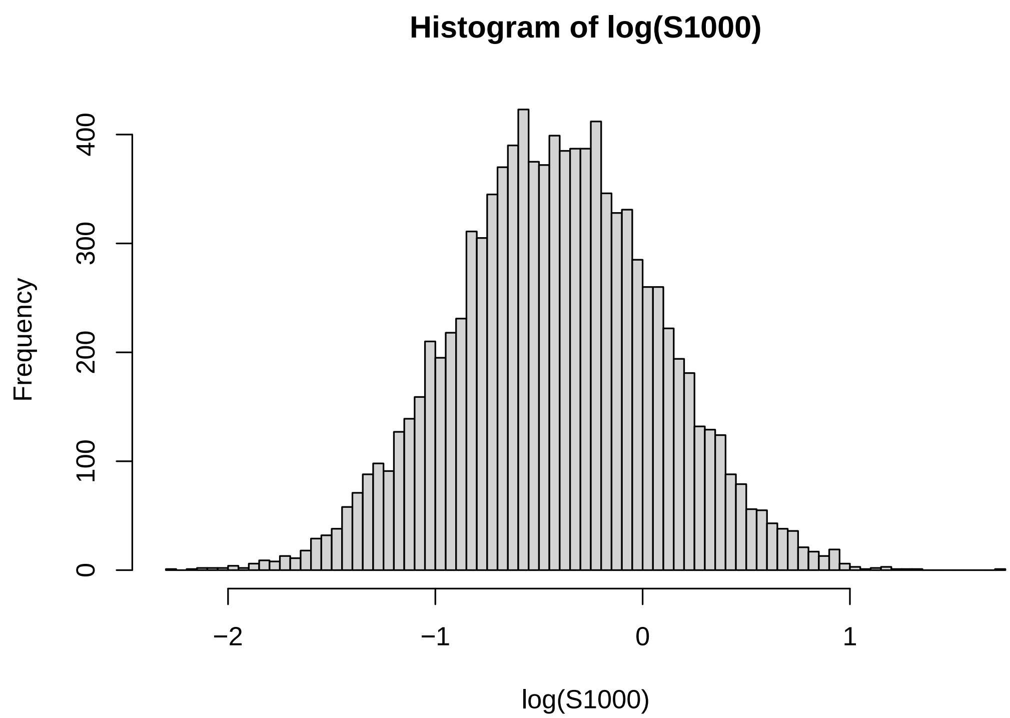
  ```
  [1] 0.7870372
  ```

  ```r
  sd(S100)
  ```

  ```
  [1] 0.411077
  ```

  Para el punto 1000:

  ```r
  S1000 <- NULL
  for(i in 1:10000)  S1000[i] <- BGeo(100,1,-0.3,0.5,1)[100]
  hist(log(S1000),breaks = 100)
  ```
  ```




```
summary(log(S1000))

    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-2.29130 -0.74638 -0.41794 -0.41880 -0.08344  1.70516
```

Como se espera, S_{1000} tiene distribución lognormal.

$\square$

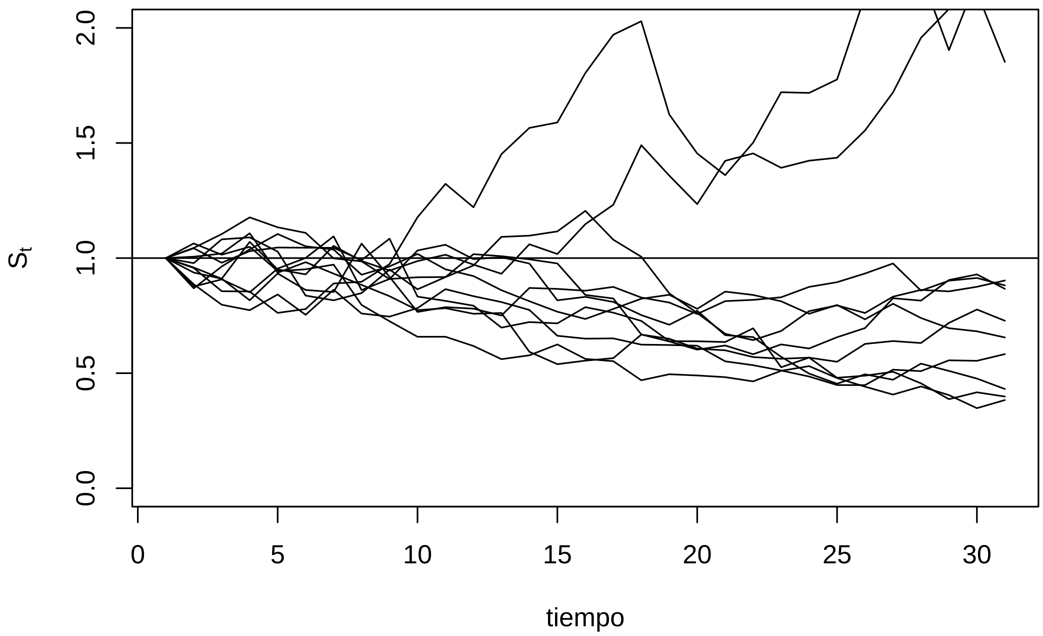
8. Considerar el precio de una acción que sigue el proceso de Wiener geométrico $\frac{dS_t}{S_t} = 0.10\,dt + 0.30\,dZ_t$ donde dZ_t es un proceso de Wiener.

   - Usando $\Delta t = 1/12$ y $S_0 = 1$ simular 5,000 años del proceso $\log S_t$ y evaluar $\frac{1}{t}\log S_t$ como función de t. Noten que esta ecuación tiende a un límite p. ¿Cuál es el valor teórico de p ¿La simulación lo reproduce?
   - Evaluar $\frac{1}{t}\left(\log S_t - pt\right)^2$ como una función de t. ¿tiende a un límite?

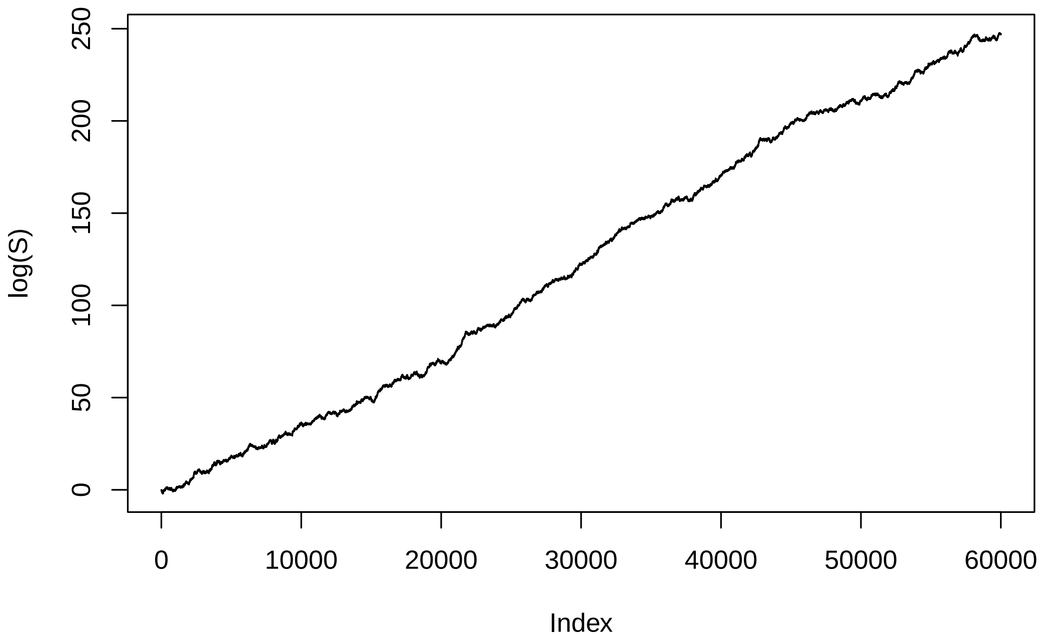
Solución.

- Para esta parte del ejercicio, tenemos:

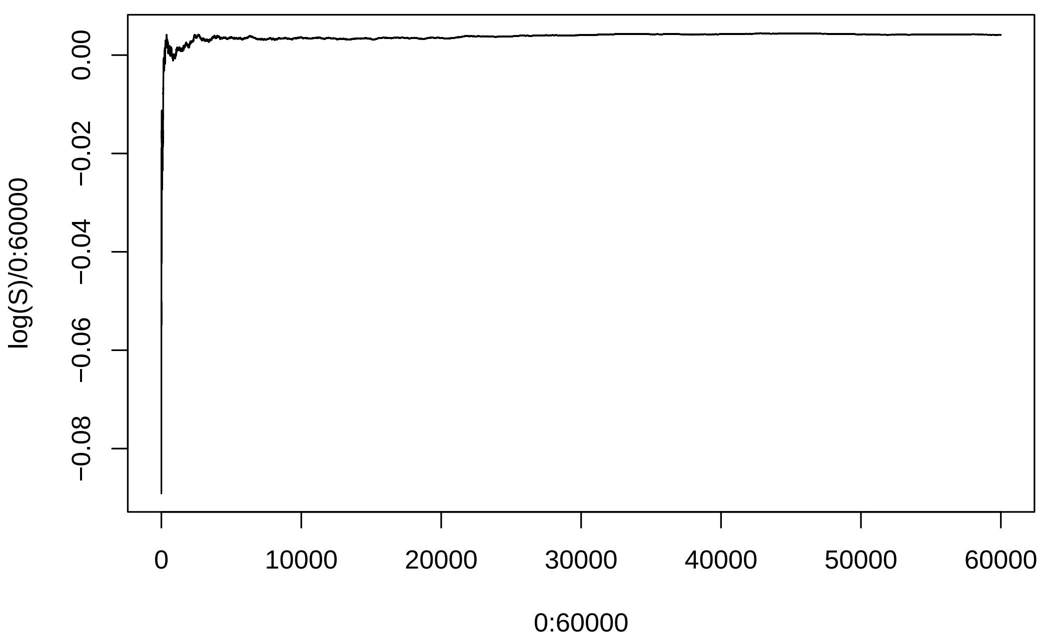
```
plot(BGeo(n = 30,TT = 1,a = -0.3, b = 0.5, S0 = 1), type = "l", ylim = c(0,2),
xlab = "tiempo", ylab = expression(S[t]))
abline(h = 1)
for(i in 1:10) lines(BGeo(30,1,-0.3,0.5,1))
```




```
S <- BGeo(n=5000*12,TT=5000,a=0.10,b=0.30,1)
plot(log(S), type ="l")
```




```r
plot(0:60000,log(S)/0:60000, type ="l")
```



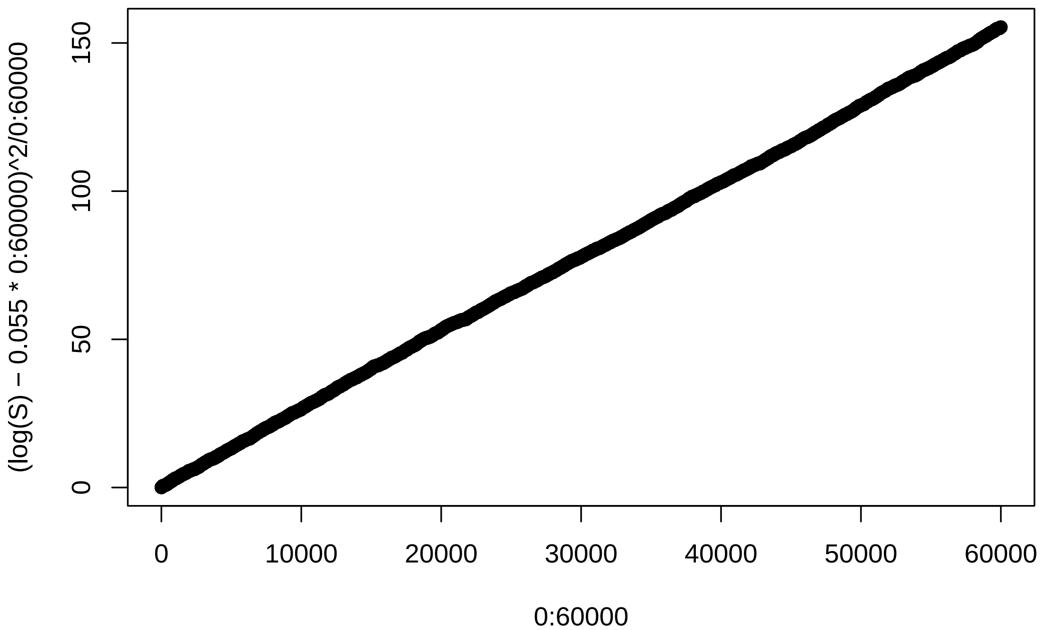
```r
# valor de p
log(S[60000])/60000

[1] 0.004115358
```

El valor teórico de p es $\mu - \sigma^2/2 = 0.1 - 0.3^2/2 = 0.055$. Se repoduce el número. Esto es porque $\log S_t \sim \mathcal{N}(\mu - \sigma^2/2, \sigma^2 t)$.

- La varianza no converge, pues depende de t.

```r
plot(0:60000,(log(S)-0.055*0:60000)^2/0:60000)
```



$\square$