

PROIECT PROBABILITATI SI STATISTICI

I. Se consideră o activitate care presupune parcurgerea secvențială a n etape. Timpul necesar finalizării etapei i de către o persoană A este o variabilă aleatoare $T_i \sim \text{Exp}(\lambda_i)$. După finalizarea etapei i , A va trece în etapa $i+1$ cu probabilitatea α_i sau va opri lucrul cu probabilitatea $1 - \alpha_i$. Fie T timpul total petrecut de persoana A în realizarea activității respective.

- 1) Construiți un algoritm în R care simulează 10^6 valori pentru v.a. T și în baza acestora aproximați $E(T)$. Reprezentați grafic într-o manieră adecvată valorile obținute pentru T . Ce puteți spune despre repartiția lui T ?
- 2) Calculați valoarea exactă a lui $E(T)$ și comparați cu valoarea obținută prin simulare.
- 3) În baza simulărilor de la 1) aproximați probabilitatea ca persoana A să finalizeze activitatea.
- 4) În baza simulărilor de la 1) aproximați probabilitatea ca persoana A să finalizeze activitatea într-un timp mai mic sau egal cu σ .
- 5) În baza simulărilor de la 1) determinați timpul minim și respectiv timpul maxim în care persoana A finalizează activitatea și reprezentați grafic timpii de finalizare a activității din fiecare simulare. Ce puteți spune despre repartiția acestor timpi de finalizare a activității?
- 6) În baza simulărilor de la 1) aproximați probabilitatea ca persoana A să se oprească din lucru înainte de etapa k , unde $1 < k \leq n$. Reprezentați grafic probabilitățile obținute într-o manieră corespunzătoare. Ce puteți spune despre repartiția probabilităților obținute?

Cerința 1:

Explicarea cerinței: simulăm 1000000 valori pentru T , apoi calculăm media lor pentru a aproxima $E(T)$. Reprezentăm grafic valorile lui T .

Pentru a simula cele 1000000 valori ale lui T am ales un nr de etape $n = 10$ și valori diferite pentru α (probabilitatea de a trece la următoare etapă) și λ (vor reprezenta parametrii pentru fiecare

etapa a activității). De asemenea, am setat un seed pentru a genera același valori random de fiecare dată când rulăm programul.

Funcția *simulate_T()* va calcula timpul total *T* pentru activitate. Astfel, pentru o valoare a lui *T*, va trece prin toate cele *n* etape (dacă activitatea este finalizată), iar pentru fiecare etapă se va calcula timpul de finalizare a acelei etape cu ajutorul funcției *rexp()*. Funcția *rexp()* va genera un număr random, dar care să fie cât mai apropiat de $\frac{1}{\lambda_i}$ (lambda corespunde numărului etapei, acesta fiind rata evenimentelor pentru distribuția exponențială). Mai departe se verifică dacă numărul generat random între 0 și 1 de *runif()* este mai mare decât probabilitate α_i de a trece la următoarea etapă. În acest fel verificăm dacă activitatea se oprește la etapa *i* sau nu, la final returnând timpul total pentru îndeplinirea activității.

Pentru fiecare 1000000 valori a fost aplicată funcția *simulate_T()* și creat un vector cu acești timpi totali. Ca să aproximăm *E(T)* am calculat media aritmetică a tuturor timpilor din *T_values*.

$$E(T) \approx \frac{1}{n} \sum_{i=1}^n T_i$$

n = numărul de simulări

T_i = valoarea lui *T* obținută la simularea *i*

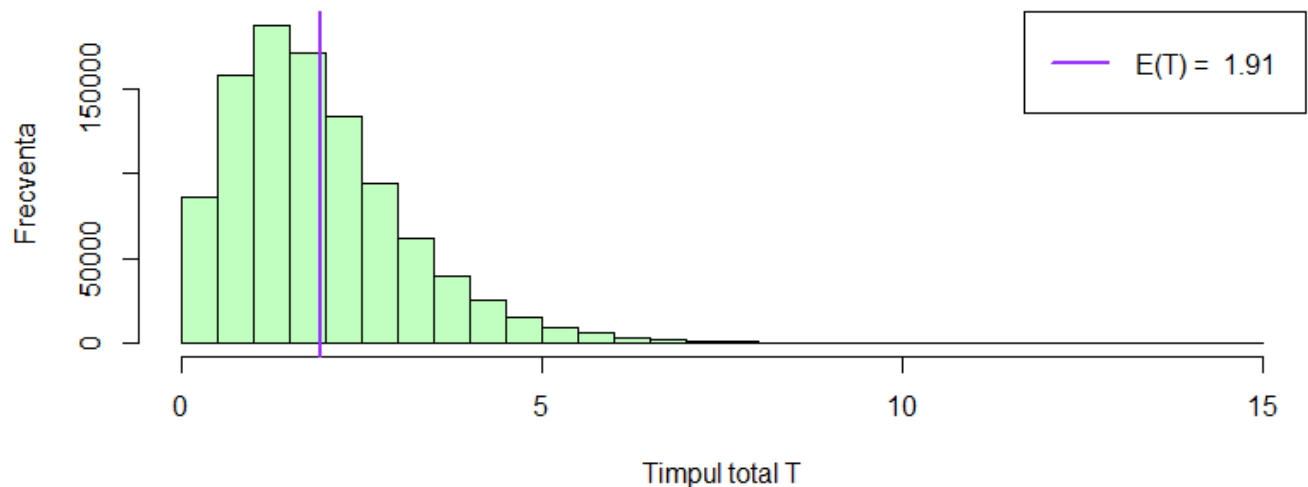
În continuare am reprezentat grafic valorile obținute printr-o histogramă cu ajutorul funcțiilor *hist()*, *abline()*, *legend()*.

```

1 set.seed(123) # pt aceleasi val random cand dai run
2 n <- 10 # nr de etape
3 lambda <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10) # parametrii lambda pentru fiecare etapa
4 alpha <- c(0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0.05) # prob de trecere la etapa urm
5 # alfa de i este prob ca persoana A sa treaca de la etapa i la etapa i+1
6 # 1-alfa de i este prob ca persoana A sa se opreasca dupa etapa i
7 nrsim <- 1000000 # nr de simulari
8
9
10 ##### cerinta 1 #####
11
12 simulate_T <- function(n, lambda, alpha) {
13   total_time <- 0
14   for (i in 1:n) {
15     time <- rexp(1, rate = lambda[i]) # timp pentru etapa i
16     total_time <- total_time + time
17     if (runif(1) > alpha[i]) { # alege random un nr intre 0 si 1
18       break # stop dupa etapa i daca nr ales este mai mare decat prob alfa de i
19     }
20   }
21   return(total_time)
22 }
23
24 T_values <- replicate(nrsim, simulate_T(n, lambda, alpha))
25 # aplica functia/operatia simulate_T de 10^6 ori pt T
26 mean_T <- mean(T_values) # media ar a timpilor totali din simulari
27
28 # reprezentare grafica
29 hist(T_values, breaks = 50, main = "Distributia lui T", xlab = "Timpul total T", ylab = "Frecventa", col = "darkseagreen1")
30 # daca freq=false arata probabilitatile
31 # breaks = nr bins/cosuri/drept alea verzi
32 abline(v = mean_T, col = "purple1", lwd = 2)
33 # linia mov pt media lui T
34 legend("topright", legend = paste("E(T) = ", round(mean_T, 2)), col = "purple1", lwd = 2)
35 # legenda pt linia mediei lui T
36 print(paste("Valoarea aproximata a lui E(T) =", mean_T))
37
38

```

Distributia lui T



```

> # legenda pt linia mediei lui T
> print(paste("Valoarea aproximata a lui E(T) =", mean_T))
[1] "Valoarea aproximata a lui E(T) = 1.91413497263285"
>

```

Ce putem spune despre repartiția lui T ?

- T este o sumă de variabile aleatoare exponențiale cu probabilități de trecere între etape. Fiecare etapă i are un timp $T_i \sim \text{Exp}(\lambda_i)$, iar suma acestor timpi este condiționată de probabilitățile α_i .
- Repartiția lui T este asimetrică, deoarece timpul total poate fi foarte mare dacă persoana A parcurge multe etape, dar nu poate fi negativ.
- Are o coadă lungă la dreapta, deoarece există o probabilitate mică (dar nenulă) ca persoana A să parcurgă toate etapele.
- Media $E(T)$ este finită și poate fi calculată exact (așa cum am făcut la punctul 2) și ar trebui să fie apropiată de valoarea teoretică (valoarea exactă).
- Dacă valorile α_i sunt mici (probabilități mici de trecere la etapa următoare), persoana A se va opri rapid, iar T va avea valori mici.
- Repartiția lui T este condiționată de numărul de etape parcurse. De exemplu:
 - Dacă persoana A se oprește după prima etapă, $T \sim \text{Exp}(\lambda)$.
 - Dacă parcurge toate etapele, T este suma a n variabile exponențiale cu parametrii $\lambda_1, \lambda_2, \dots, \lambda_n$

Cerința 2:

Explicarea cerinței: calculăm exact valoarea lui $E(T)$ și o comparăm cu cea obținută prin simulare.

Ca să aflăm valoarea exactă a lui $E(T)$ vom folosi formula:

$$E(T) = \sum_{i=1}^n \left(\prod_{j=1}^{i-1} \alpha_j \right) \cdot \frac{1}{\lambda_i}$$

$$\prod_{j=1}^{i-1} \alpha_j = \text{produsul probabilităților de trecere de la etapa } i \text{ la } i+1$$

$$\frac{1}{\lambda_i} = \text{valoarea așteptată a timpului pentru etapa } i$$

Această formulă este dedusă din mai multe (alte formule) și anume:

- Valoarea așteptată a sumei de variabile aleatoare: $E(T) = E(T_1) + E(T_2) + \dots + E(T_k)$
- Probabilitatea de a ajunge la etapa i : $\prod_{j=1}^{i-1} \alpha_j$
- Contribuția fiecăruia la etapa i : $E(T_i) = \prod_{j=1}^{i-1} \alpha_j \cdot \frac{1}{\lambda_i}$
- Suma contribuțiilor: formula finală

Observăm ca cele doua rezultate obținute (valoarea aproximată și cea exactă) sunt foarte apropiate, diferența dintre ele fiind foarte mică.

```

39
40 ##### cerința 2 #####
41
42 exact_ET <- 0
43 for (i in 1:n) {
44   prod_alpha <- if (i == 1) 1 else prod(alpha[1:(i-1)]) # produsul alpha1 * alpha2 * ... * alpha(i-1)
45   exact_ET <- exact_ET + prod_alpha * (1 / lambda[i])
46 }
47
48 print(paste("valoarea exacta a lui E(T) =", exact_ET))
49 print(paste("Dif între valoarea exacta și cea simulata =", abs(exact_ET - mean_T)))
50
51 |
52
53 ##### cerința 3 #####
54
55 >
56 > print(paste("valoarea exacta a lui E(T) =", exact_ET))
57 [1] "valoarea exacta a lui E(T) = 1.913027488"
58 > print(paste("Dif între valoarea exacta și cea simulata =", abs(exact_ET - mean_T)))
59 [1] "Dif între valoarea exacta și cea simulata = 0.00110748463285004"
60 >

```

Cerința 3:

Explicarea cerinței: aproximăm probabilitatea ca activitatea să fie finalizată (adică să ajungem la etapa n).

Am creat o funcție aproape identică cu cea de la cerința 1), dar în loc să returneze timpul total, aceasta va returna dacă activitatea a fost finalizată pentru toate cele 1000000 simulări ale lui T . Astfel, am creat un vector logic **comp** care înregistrează toate valorile de TRUE/FALSE (finalizat/nefinalizat). Probabilitatea ca se calculează folosind funcția **mean()** (media aritmetică).

```

53 ##### cerinta 3 #####
54
55 simulate_T_final <- function(n, lambda, alpha) {
56   total_time <- 0
57   completed <- TRUE # presupun ca finalizeaza activitatea
58   for (i in 1:n) {
59     time <- rexp(1, rate = lambda[i]) # timp pentru etapa i
60     total_time <- total_time + time
61     if (runif(1) > alpha[i]) {
62       completed <- FALSE # nu a finalizat activitatea
63       break
64     }
65   }
66   return(completed) # returneaza TRUE daca a finalizat, FALSE altfel
67 }
68
69 comp <- replicate(nrsim, simulate_T_final(n, lambda, alpha))
70 prob_final <- mean(comp) # probabilitatea de finalizare
71 print(paste("Probabilitatea de finalizare a activitatii =", prob_final))
72

```

```

> comp <- replicate(nrsim, simulate_T_final(n, lambda, alpha))
> prob_final <- mean(comp) # probabilitatea de finalizare
> print(paste("Probabilitatea de finalizare a activitatii =", prob_final))
[1] "Probabilitatea de finalizare a activitatii = 2.1e-05"
> |

```

Cerința 4:

Explicarea cerinței: calculăm probabilitatea ca T să fie $\leq \sigma$ (unde σ este un prag dat).

Alegem o valoare pentru σ (în cazul nostru 5 pentru a cuprinde cât mai multe cazuri). Vectorul ***T_values_sigma*** va conține doar valorile timpilor a căror activitate a fost finalizată. Ca să calculăm probabilitatea cerută vom compara fiecare timp din ***T_values_sigma*** cu valoarea lui σ , returnând TRUE sau FALSE pentru fiecare comparație/valoare, iar cu ajutorul funcției ***mean()*** aceste valori de TRUE sunt numărate și mai apoi împărțite la numărul total de valori din vectorul ***T_values_sigma***.

Probabilitatea/rezultatul nostru a fost 1 deoarece am ales o valoare pentru σ destul de mare pentru a include toate cazurile.

```

74
75 - ##### cerinta 4 #####
76
77 sigma <- 5 # valoarea lui sigma
78 # este ales mai mare sa acopere toate cazurile, deci prob o sa fie 1
79 T_values_sigma <- T_values[comp]
80 # filtreaza T_values folosind vectorul logic comp
81 # ia elementele de pe aceleasi pozitii iar daca in al doilea vector e true
82 # atunci pastreaza timpul lui T
83 prob_sigma <- mean(T_values_sigma <= sigma)
84 # compara fiecare T cu sigma si face med ar din val de true false <= sigma
85 print(paste("Probabilitatea ca T <= sigma:", prob_sigma))
86
87
88

```

```

> print(paste("Probabilitatea ca T <= sigma
[1] "Probabilitatea ca T <= sigma: 1"
>

```

Cerința 5:

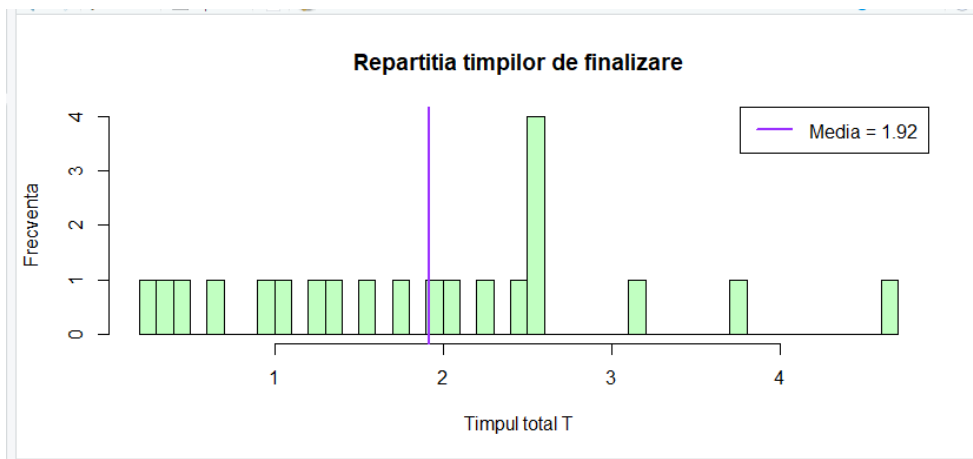
Explicarea cerinței: determinăm timpul minim și maxim în care se finalizează activitatea și reprezentăm grafic toți timpii obținuți.

Minimul/maximul este luat din vectorul creat la cerința anterioară și pus în variabila *min_T*/*max_T*. Aceste valori sunt reprezentate cu ajutorul unei histograme.

```

88
89 - ##### cerinta 5 #####
90
91 min_T <- min(T_values_sigma)
92 max_T <- max(T_values_sigma)
93
94 print(paste("Timpul minim de finalizare=", min_T))
95 print(paste("Timpul maxim de finalizare=", max_T))
96
97 # reprezentare grafica a timpilor de finalizare
98 hist(T_values_sigma, breaks = 50, main = "Repartitia timpilor de finalizare", xlab = "Timpul total T", ylab = "Frecventa", col = "darkseagreen1")
99 abline(v = mean(T_values_sigma), col = "purple", lwd = 2) # linie pentru medie
100 legend("topright", legend = paste("Media =", round(mean(T_values_sigma), 2)), col = "purple", lwd = 2)
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```



Ce putem spune despre repartiție?

- Distribuția este asimetrică, cu o coadă lungă la dreapta. Acest lucru înseamnă că există cazuri în care timpul de finalizare este mult mai mare decât media, dar acestea sunt mai puțin probabile.
- Vârful distribuției este în jurul valorii medii (1.92), ceea ce indică faptul că majoritatea timpilor de finalizare sunt concentrați în jurul acestei valori.
- Timpul minim de finalizare este aproape de 0, deoarece persoana A se poate opri foarte devreme (după prima etapă).
- Timpul maxim de finalizare este mai mare, reflectând cazurile în care persoana A parcurge toate etapele.

Cerința 6:

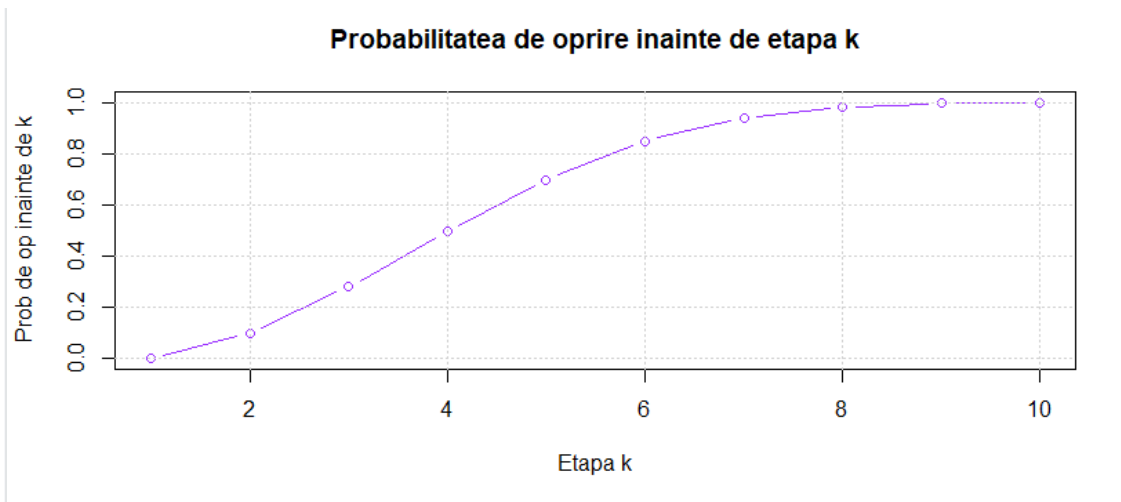
Explicarea cerinței: aproximăm probabilitatea ca activitatea să se oprească înainte de etapa k și reprezentăm aceste probabilități într-un grafic.

Am creat o funcție similară ca cea de la cerința 1), dar în loc să returneze timpul total, ea va returna etapa la care s-a oprit activitatea pentru fiecare T . Aceste etape sunt reținute într-un vector `stop_stages`. Ca să calculăm probabilitatea, am decis să iau pentru fiecare k . Astfel, vectorul ***prob_stop_before_k*** conține toate probabilitățile ca activitatea să se oprească înainte k pentru fiecare k de la 1 la n . Reprezentarea grafică este dată de o linie, în loc de o histogramă.


```

103 ##### cerinta 6 #####
104
105 simulate_stop_stage <- function(n, lambda, alpha) {
106   for (i in 1:n) {
107     time <- rexp(1, rate = lambda[i])
108     if (runif(1) > alpha[i]) { # daca se opreste dupa etapa i
109       return(i) # returneaza etapa la care s-a oprit
110     }
111   }
112   return(n) # daca a trecut prin toate etapele
113 }
114 # functia care iti arata la ce etapa s-a oprit A
115
116 stop_stages <- replicate(nrsim, simulate_stop_stage(n, lambda, alpha))
117 # etapele la care s-a oprit A vector
118
119 prob_stop_before_k <- sapply(1:n, function(k) {
120   mean(stop_stages < k) # prob ca A sa se opreasca inainte de k
121 })
122 # se aplica pt toate k-urile de la 1 la n
123
124 # reprezentare grafica
125 plot(1:n, prob_stop_before_k, type = "b", col = "purple",
126      xlab = "Etapa k", ylab = "Prob de op inainte de k",
127      main = "Probabilitatea de oprire inainte de etapa k")
128 grid()
129 |
130
131

```



Ce puteți spune despre repartiția probabilităților obținute?

- Aceste probabilități sunt crescătoare în raport cu k , deoarece persoana A are mai multe șanse de a se opri pe măsură ce parcurge mai multe etape.
- Probabilitățile formează o curbă crescătoare.

- La $k = 1$, probabilitatea este 0, deoarece persoana A nu poate să se oprească înainte de prima etapă.
- La $k = n$, probabilitatea este maximă, deoarece persoana A poate să se oprească în orice etapă până la n .
- Graficul arată o curbă monoton crescătoare, care reflectă faptul că probabilitatea de oprire crește odată cu creșterea lui k .
- Panta curbei depinde de probabilitățile α_i . Dacă valorile alfa sunt mici, panta este mai abruptă.

Concluzie:

În concluzie, am analizat un proces format din mai multe etape, unde fiecare etapă are un timp de execuție aleator și o probabilitate de a continua spre următoarea. Proiectul arată cum putem folosi simulările pentru a înțelege mai bine procesele aleatorii și comportamentul lor în diverse situații.

Dificultățile în realizarea cerințelor:

Începutul a fost foarte greu neștiind de unde să încep, iar înțelegerea cerinței a durat mai mult timp decât mă așteptam. Găsirea soluțiilor cerințelor nu a fost una tocmai ușoară, fiind nevoie de documentație specială care nu a fost tocmai ușor de înțeles. În ciuda acestor dificultăți, am reușit finalizarea proiectului.

Sursele:

- Cursurile domnului profesor Niculescu Cristian
- https://alexamarioarei.quarto.pub/curs-ps-fmi/Introducere_R/Chapter_4/Elemente_grafica_R.html
- <https://cran.r-project.org/doc/manuals/r-release/R-intro.html#Index-vectors>

- https://www.math.uaic.ro/~maticiuc/didactic/Probability_Theory_Course_7_8_9_10.pdf
- https://en.wikipedia.org/wiki/Exponential_distribution
- https://en.wikipedia.org/wiki/Expected_value
- https://en.wikipedia.org/wiki/Markov_chain