PROIECT PROBABILITATI SI STATISTICI

1. 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 . După finalizarea etapei ***i***, A va trece ȋn etapa ***i+1*** cu probabilitatea sau va opri lucrul cu probabilitatea . Fie ***T*** timpul total petrecut de persoana A ȋn realizarea activității respective.
   1. Construiți un algoritm ȋn R care simulează valori pentru v.a. ***T*** și ȋn baza acestora aproximați ***E(T)***. Reprezentați grafic ȋntr-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. Ȋn baza simulărilor de la 1) aproximați probabilitatea ca persoana ***A*** să finalizeze activitatea.
4. Ȋn baza simulărilor de la 1) aproximați probabilitatea ca persoana ***A*** să finalizeze activitatea ȋntr-un timp mai mic sau egal cu .
5. Ȋn baza simulărilor de la 1) determinați timpul minim și respectiv timpul maxim ȋn 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. Ȋn baza simulărilor de la 1) aproximați probabilitatea ca persoana ***A*** să se oprească din lucru ȋnainte de etapa ***k***, unde . Reprezentați grafic probabilitățile obținute ȋntr-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 (probabilitatea de a trece la următoare etapă) și (vor reprezenta parametrii pentru fiecare etapa a activității). De asemena, am setat un seed pentru a genera același valori random de fiecare dată când rulăm programul.

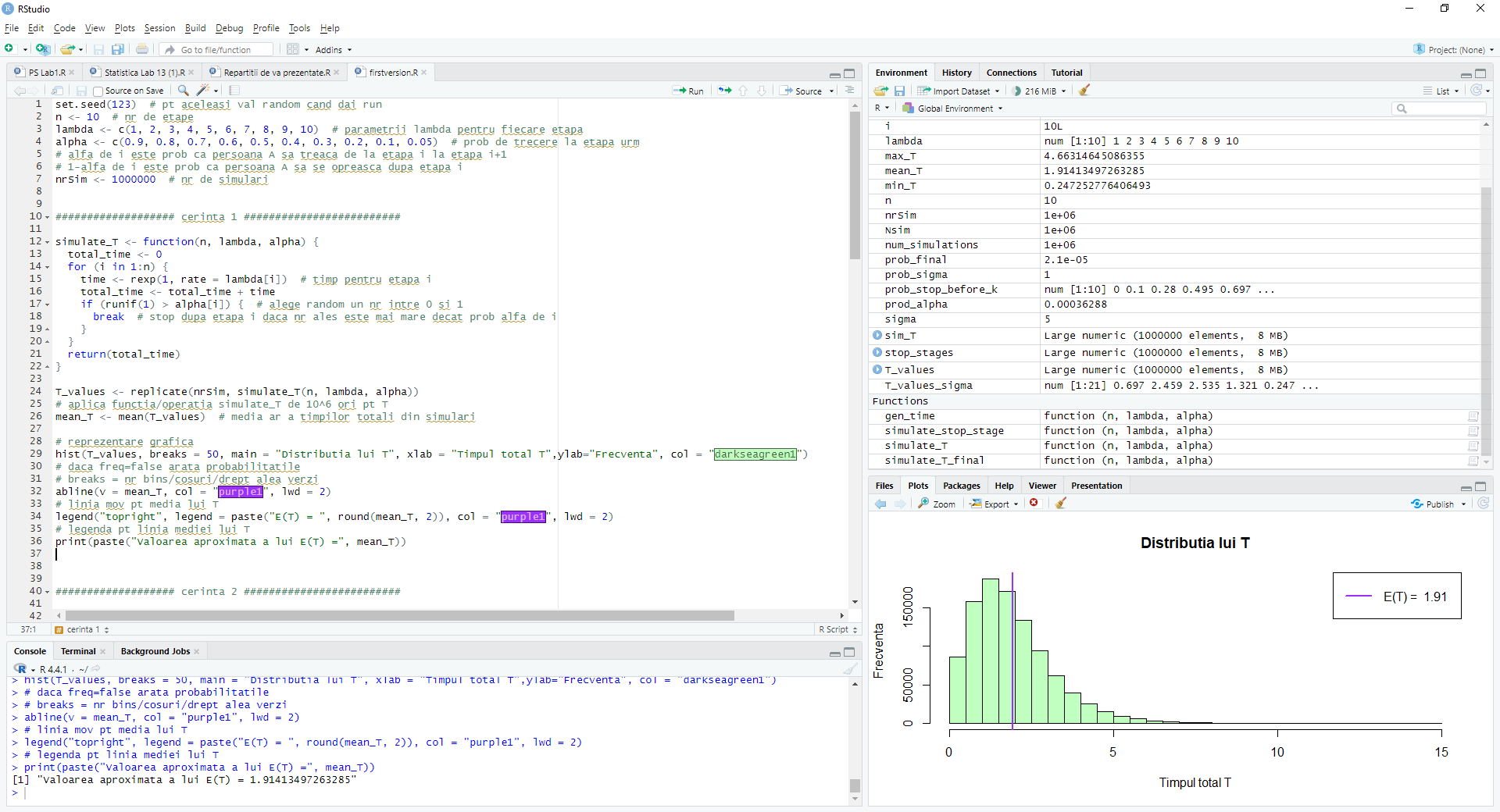
Funcția ***simulate\_T()*** va calcula timpul total ***T*** pentru activitate. Astfel, pentru o valoare a lui ***T,*** va trece prin toate cele ***n*** etape (dacă activitatea este finalizată), iar pentru fiecare etapă se va calcula timpul de finalizare a acelei etape cu ajutorul funcției ***rexp()***. Funcția ***rexp()*** va genera un număr random, dar care sa fie cât mai apropiat de (lambda corespunde numărului etapei, acesta fiind rata evenimentelor pentru distrbuția exponențială). Mai departe se verifică dacă numărul generat random între 0 și 1 de ***runif()*** este mai mare decât probabilitate 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.

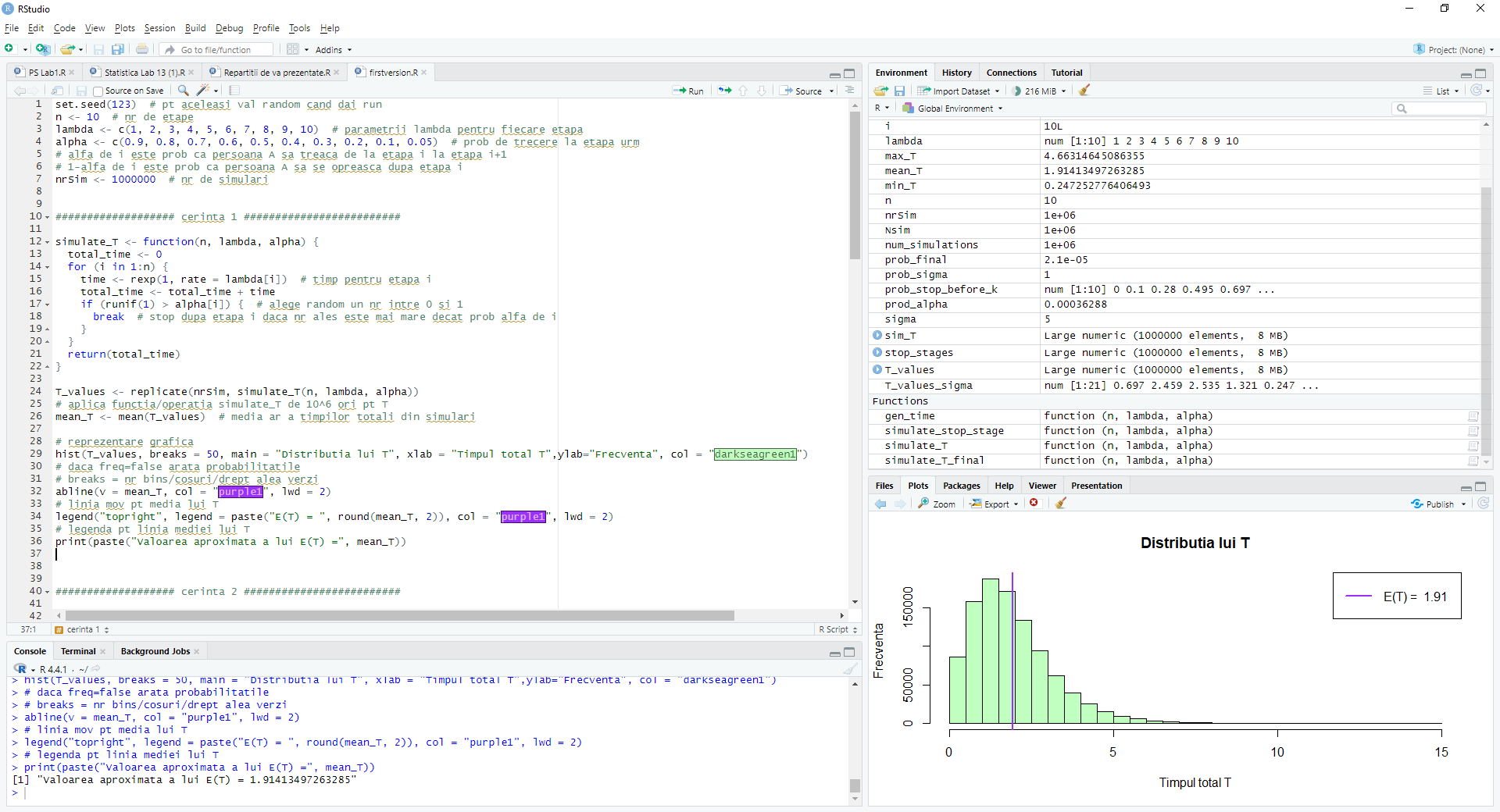
Pentru fiecare 1000000 valori a fost aplicată funcția ***simulate\_T()*** și creat un vector cu acești timpi totali. Ca să aproximăm ***E(T)*** am calculat media aritmetică a tuturor timpilor din ***T\_values.***

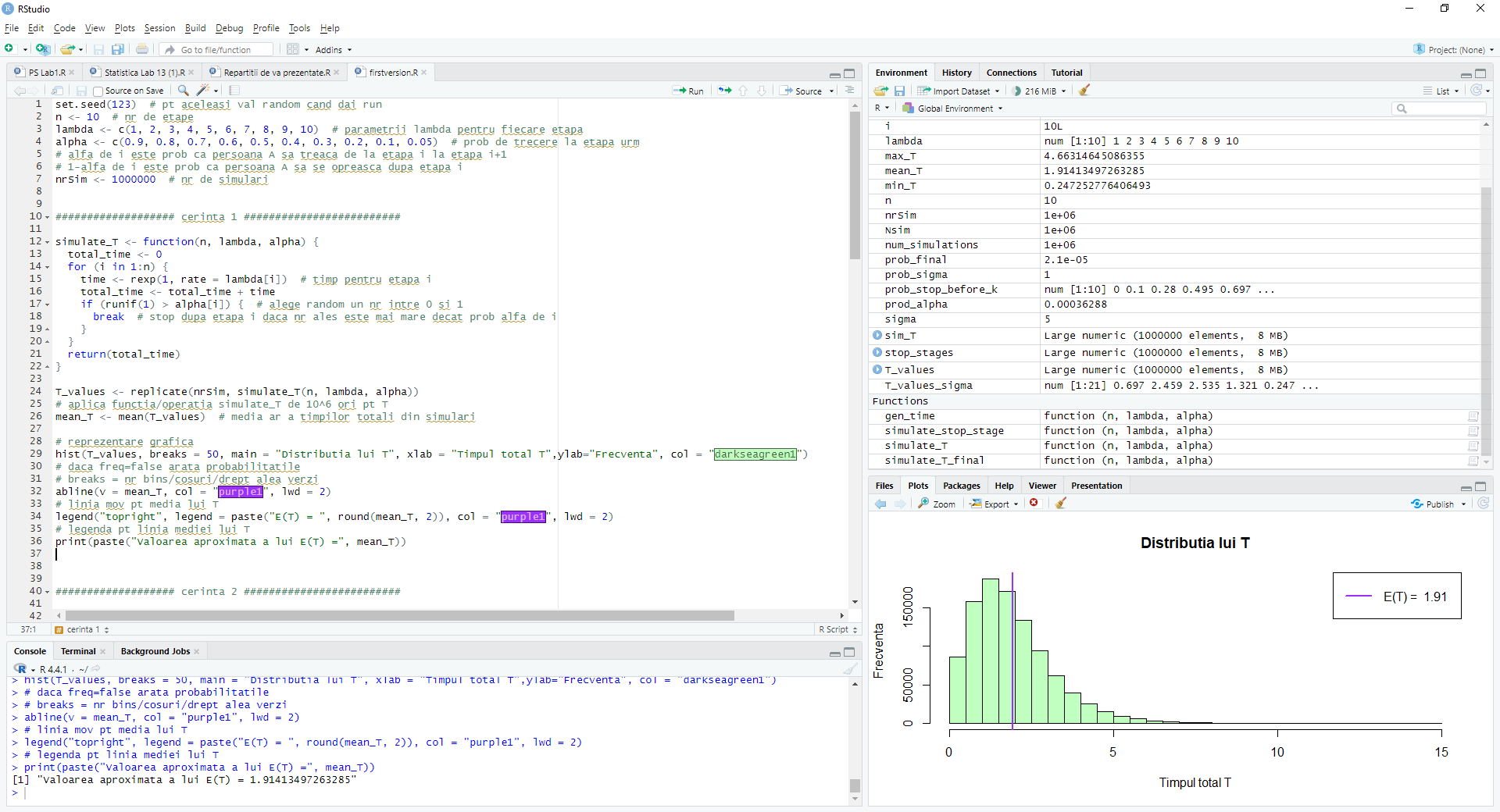
n = numărul de simulări

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

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







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 **,** iar suma acestor timpi este condiționată de probabilitățile **.**
  + 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 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, .
    - Dacă parcurge toate etapele, ***T*** este suma a ***n*** variabile exponențiale cu parametrii ,

Cerința 2:

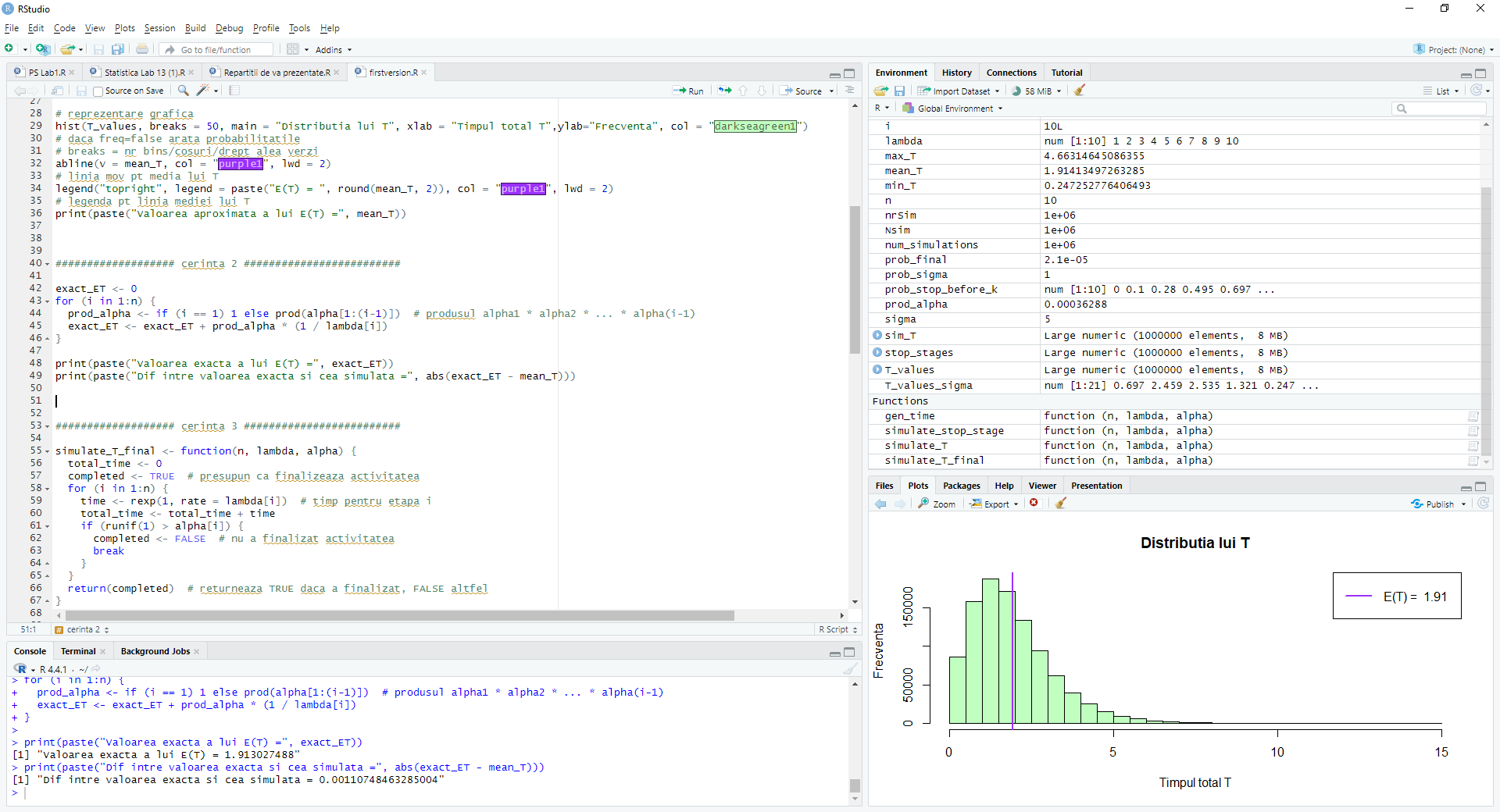
**Explicarea cerinței**: calculăm exact valoarea lui ***E(T)*** și o comparăm cu cea obținută prin simulare.

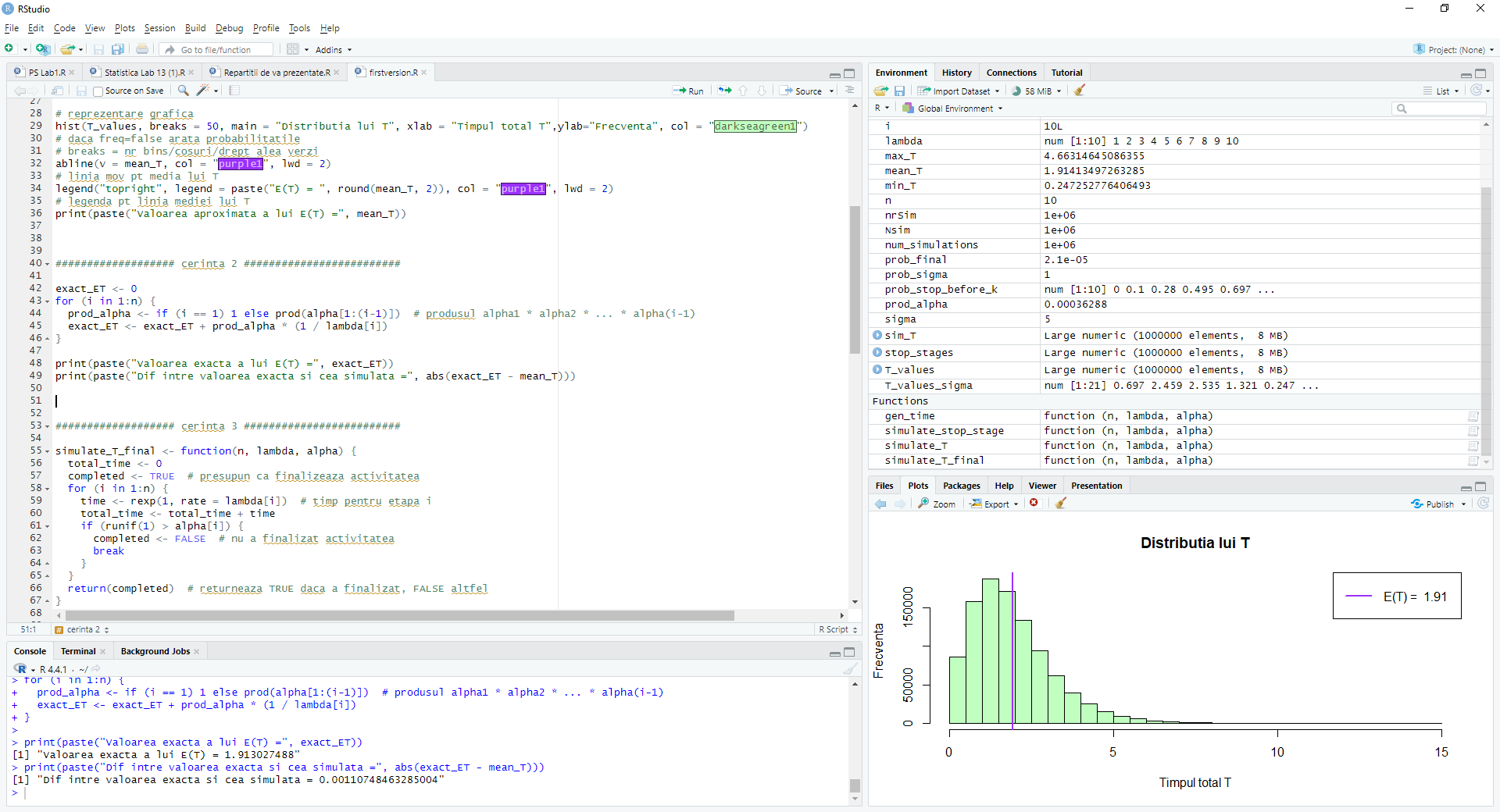
Ca să aflăm valoarea exactă a lui ***E(T)*** vom folosi formula:

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

* Valoarea așteptată a sumei de variabile aleatoare:
* Probabilitatea de a ajunge la etapa i:
* Contribuția fiecaruia la etapa i: ()
* Suma contribuțiilor: formula finală

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

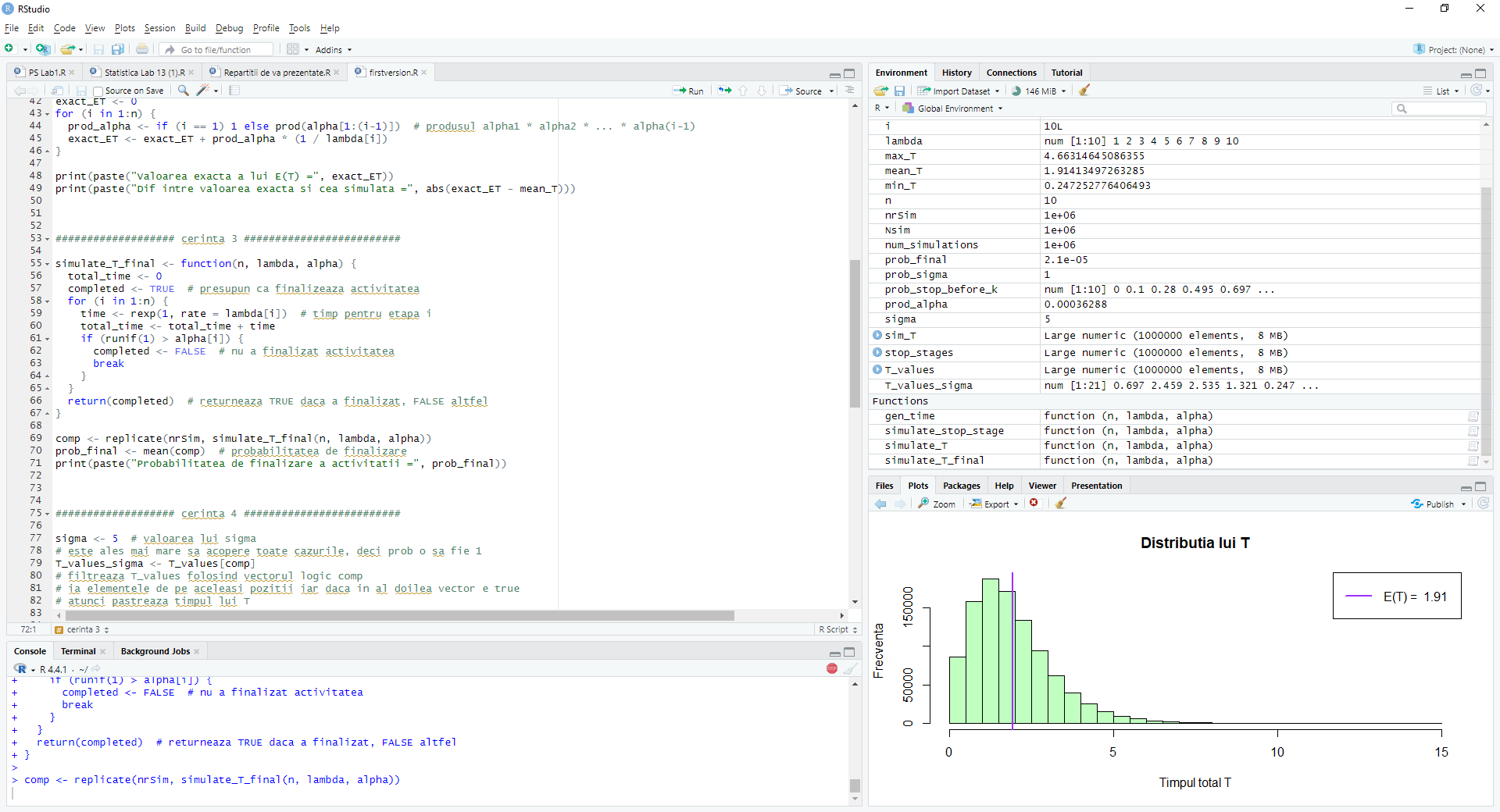


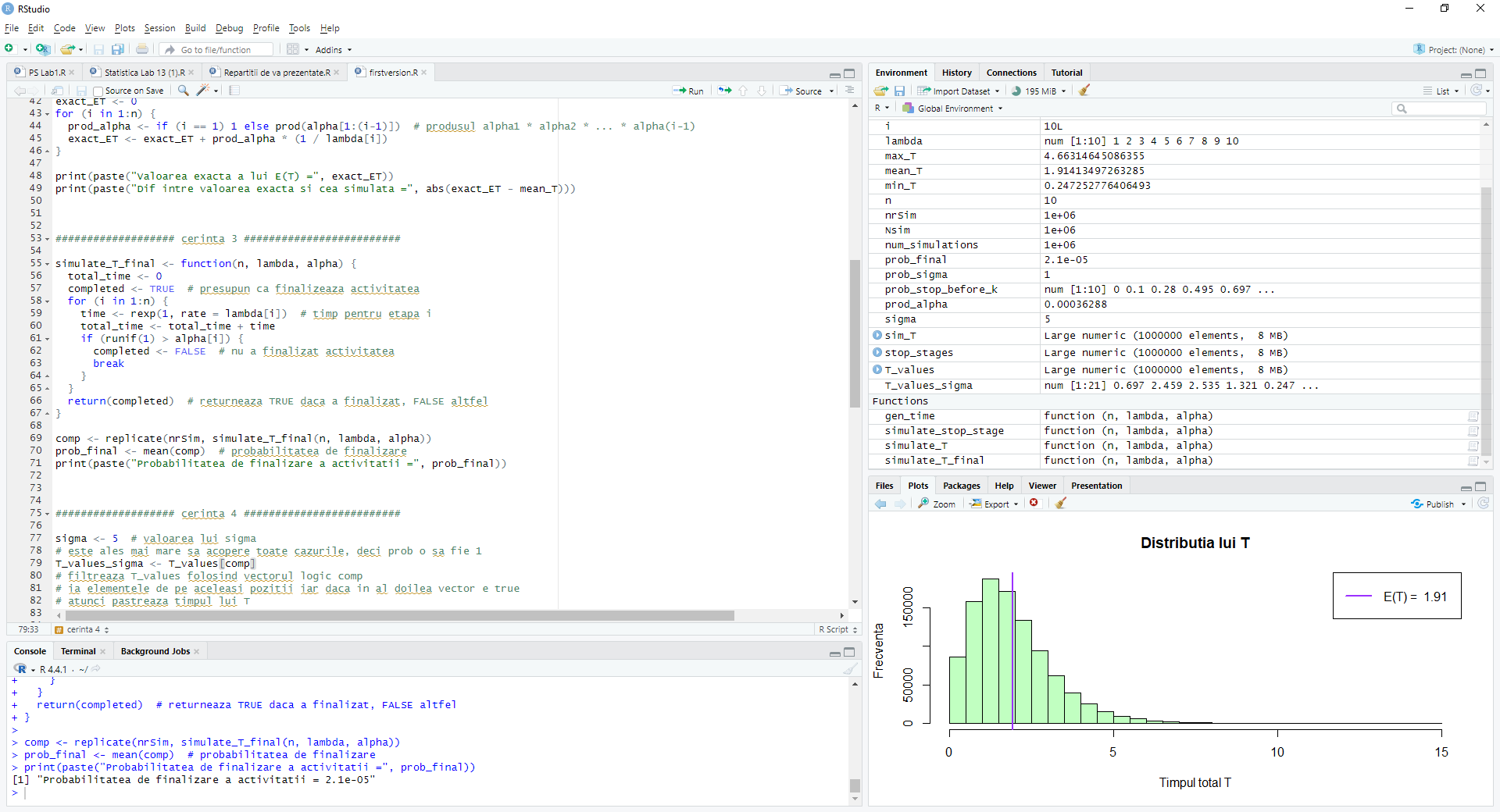


Cerința 3:

**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 aritemtică).



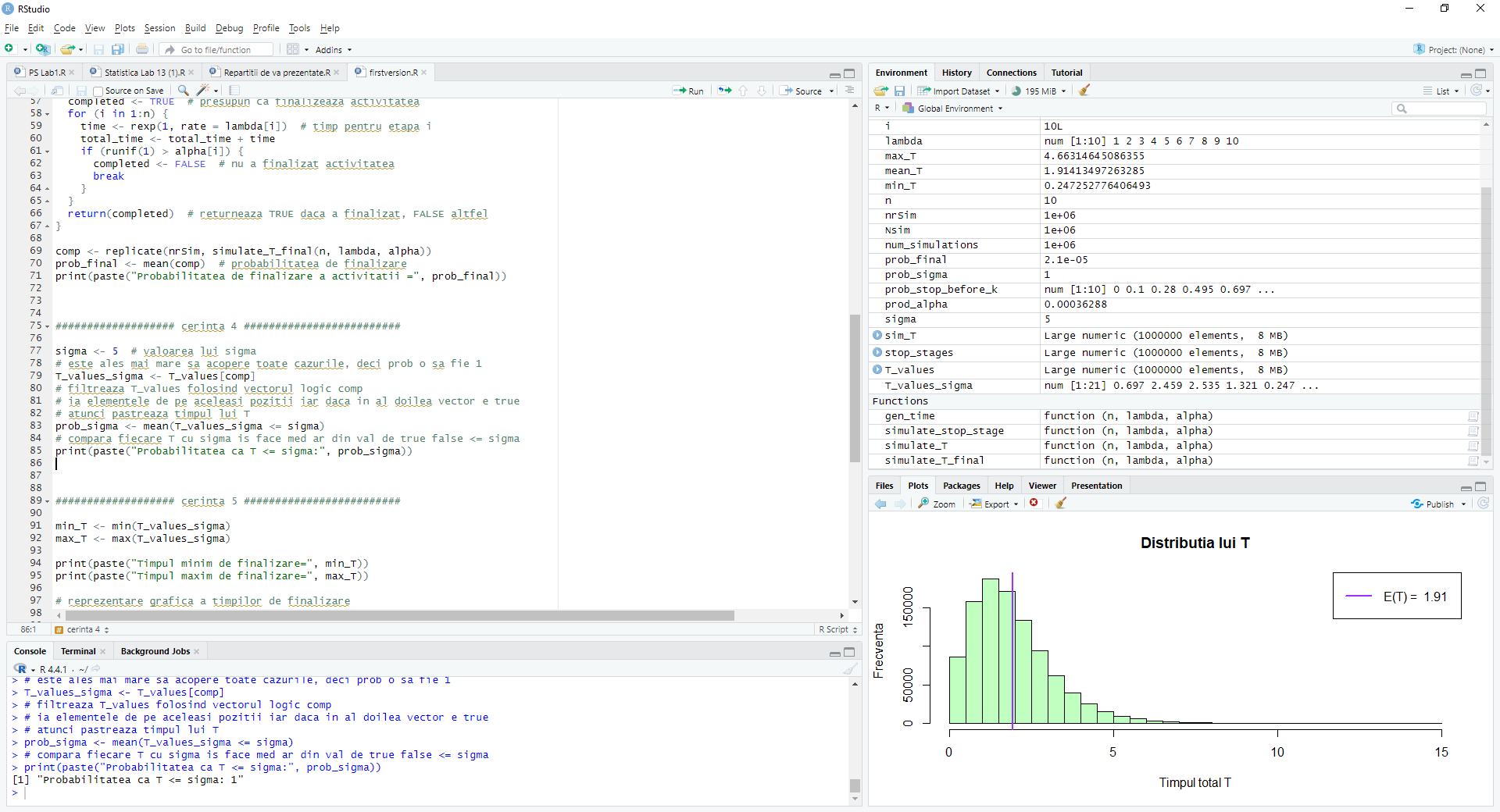


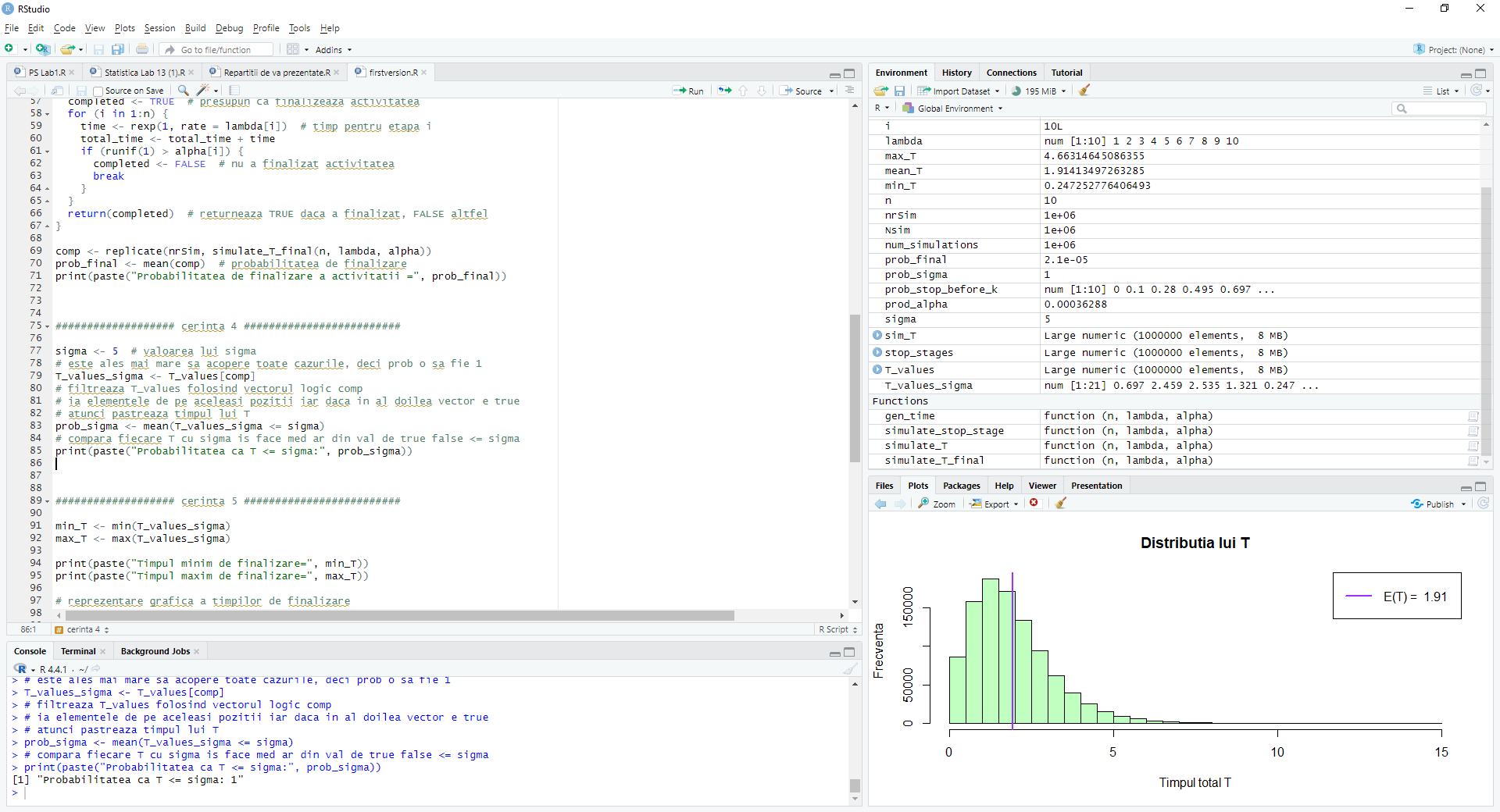
Cerința 4:

**Explicarea cerinței**: calculăm probabilitatea ca T să fie ≤ σ (unde σ este un prag dat).

Alegem o valoare pentru (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 probabillitate cerută vom compara fiecare timp din ***T\_values\_sigma*** cu valoarea lui , 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 destul de mare pentru a include toate cazurile.

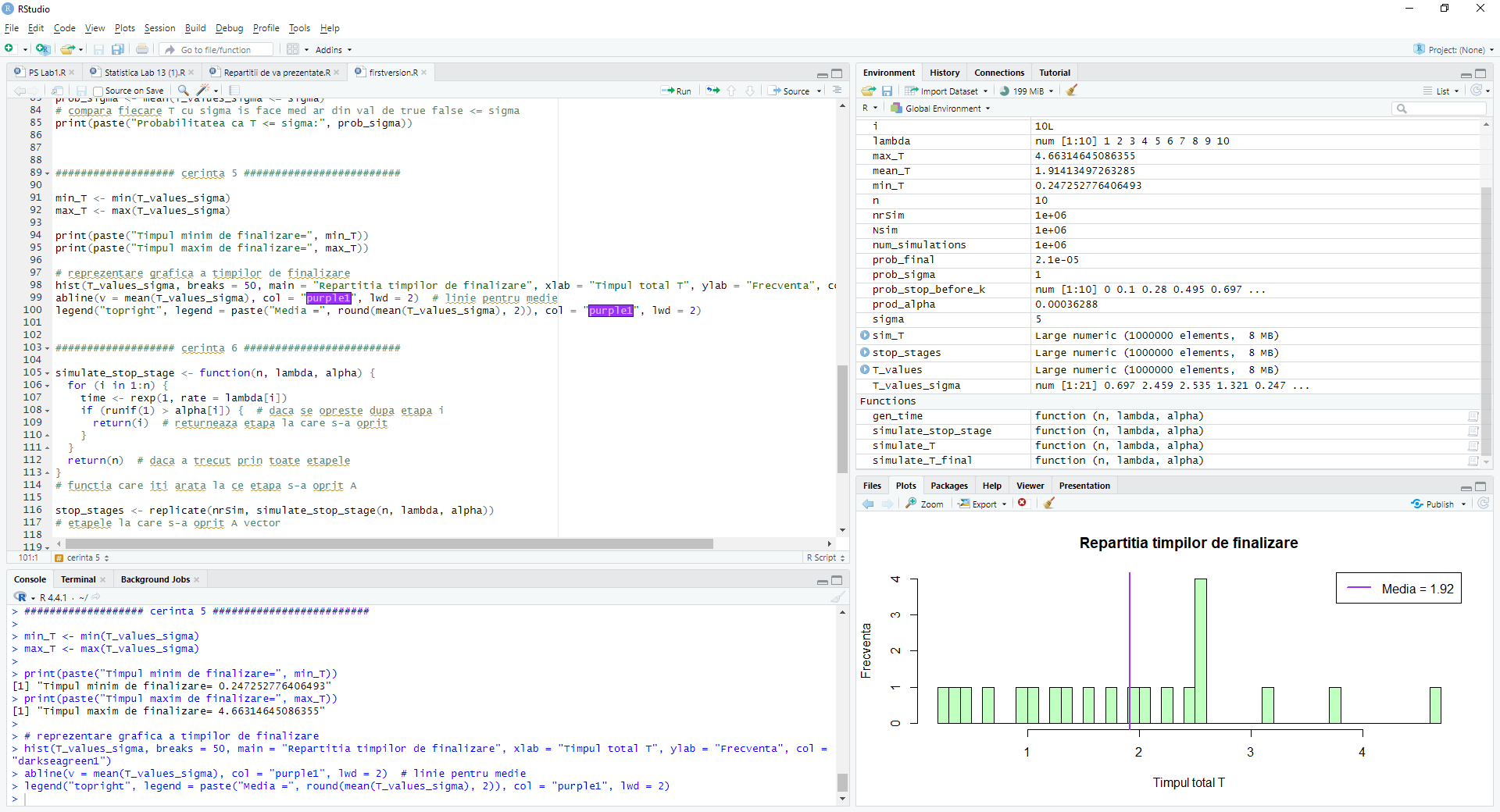
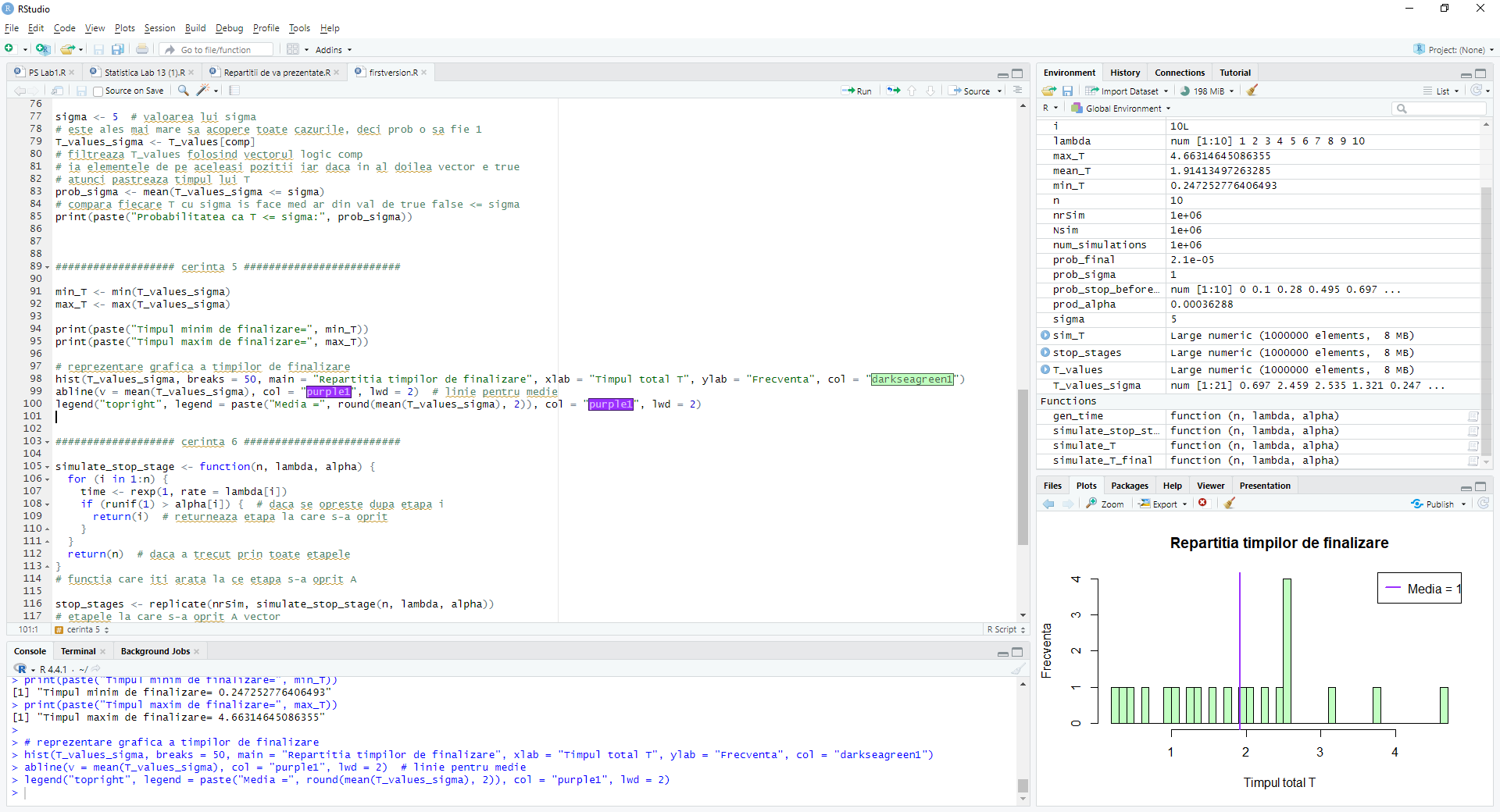


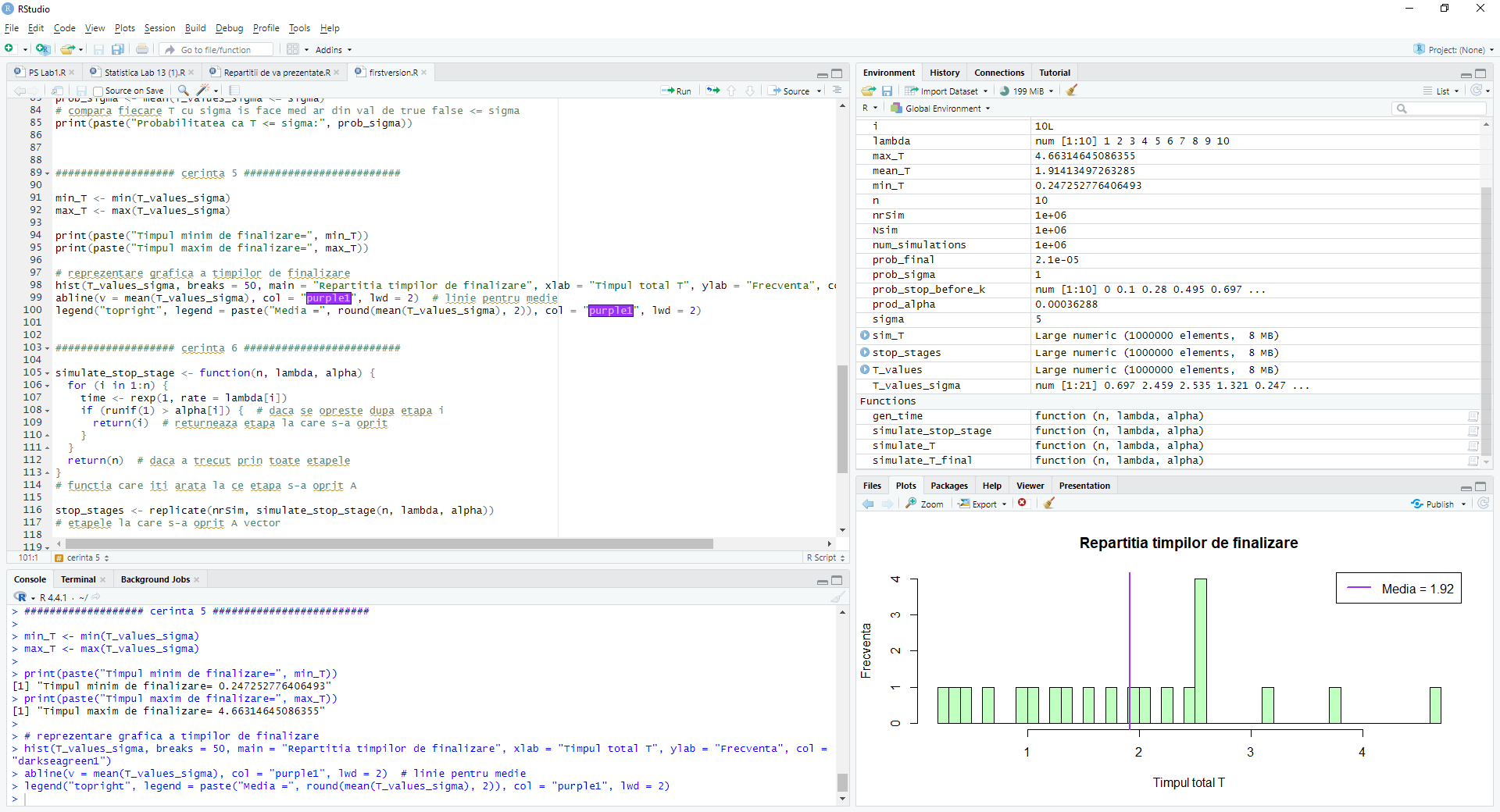


Cerința 5:

**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 in variabila ***min\_T/max\_T*** . Aceste valori sunt reprezentate cu ajutorul unei histograme.





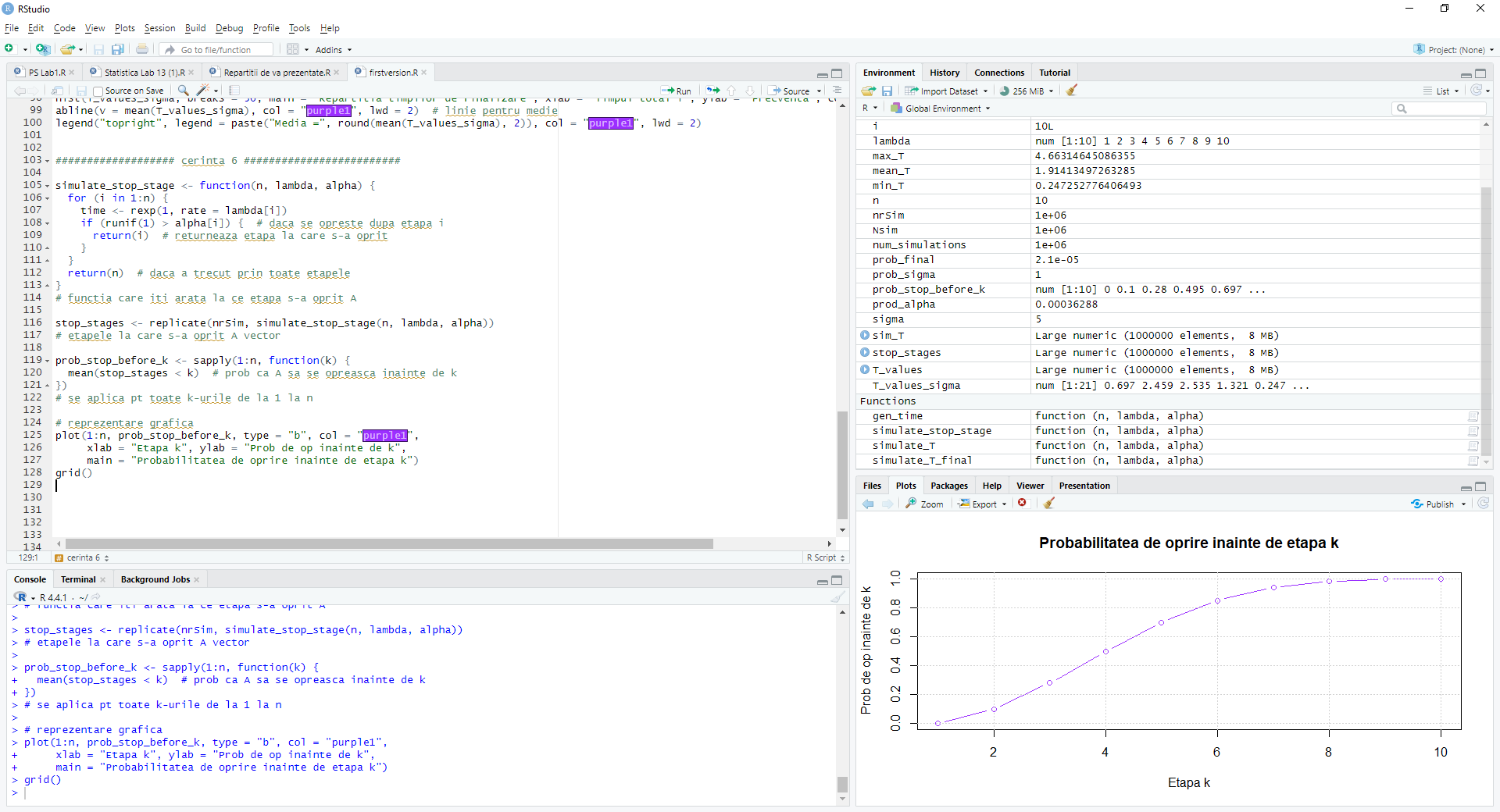
Ce putem spune despre repartiție?

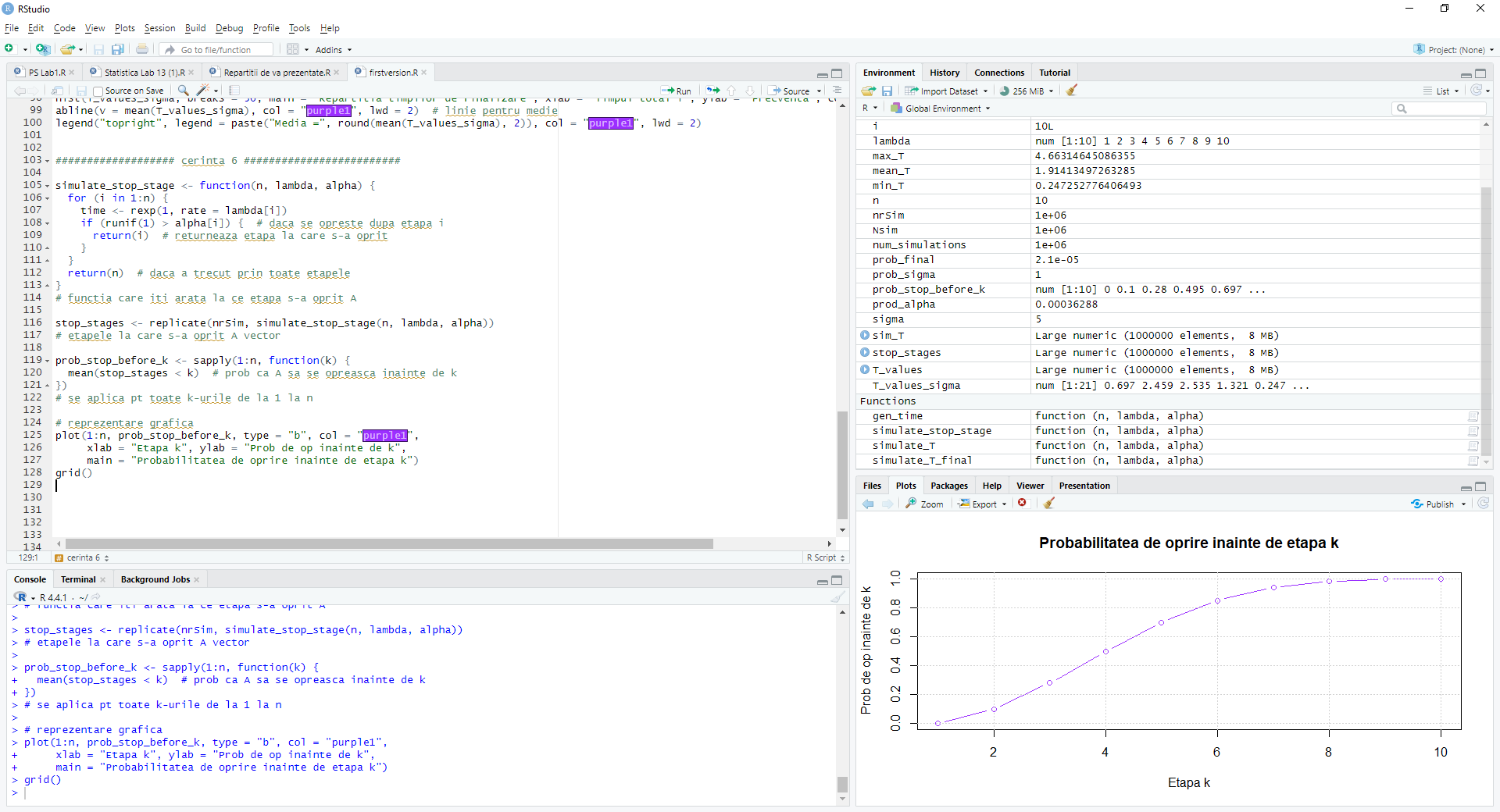
* + Distribuția este asimetrică, cu o coadă lungă la dreapta. Acest lucru înseamnă că există cazuri în care timpul de finalizare este mult mai mare decât media, dar acestea sunt mai puțin probabile.
  + Vârful distribuției este în jurul valorii medii (1.92), ceea ce indică faptul că majoritatea timpilor 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 probablitatea, am decis să iau pentru fiecare ***k***. Astfel, vectorul ***prob\_stop\_before\_k*** conține toate probablitățile 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.





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 ​. 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>