R Seminars დავალება II

დავალება I : იქმნება ვექტორი სახელად Vec. ის თავდაპირველად შეიცავს რიცხვებს 1-დან 8-მდე (წარმოქმნილი მიმდევრობით 1:8), რასაც მოჰყვება რიცხვები 200 და 300. ასე რომ, Vec არის ვექტორი მნიშვნელობებით 1, 2, 3, 4, 5, 6, 7, 8. , 200, 300. შემდეგ სტრიქონში, Vec-ის ვექტორს ემატება 10 დამატებითი შემთხვევითი რიცხვი, რომლებიც გენერირებულია ნორმალური განაწილებიდან.rnorm() არის ფუნქცია R-ში, რომელიც წარმოქმნის შემთხვევით რიცხვებს ნორმალური განაწილებიდან.n = 10 მიუთითებს, რომ გვინდა 10 შემთხვევითი რიცხვის გენერირება.mean = round(mean(Vec), 2) ითვლის Vec-ში არსებული მნიშვნელობების საშუალოს mean() ფუნქციის გამოყენებით და შემდეგ ამრგვალებს მას 2 ათობითი ადგილამდე round() ფუნქციის გამოყენებით. ეს საშუალო მნიშვნელობა გამოიყენება როგორც საშუალო პარამეტრი შემთხვევითი რიცხვების გენერირებისთვის.sd = round(var(Vec), 2) ითვლის Vec-ში არსებული მნიშვნელობების დისპერსიას var() ფუნქციის გამოყენებით და შემდეგ ამრგვალებს მას 2 ათობითი ადგილამდე round() ფუნქციის გამოყენებით.ამრიგად, მიღებული Vec ვექტორი შეიცავს საწყის მნიშვნელობებს 1, 2, 3, 4, 5, 6, 7, 8, 200, 300, რასაც მოჰყვება 10 დამატებითი შემთხვევითი რიცხვი, რომლებიც გენერირებულია ნორმალური განაწილებიდან.

Vec <- c(1:8,200,300)

Vec <- c(Vec,

         rnorm(n =10,mean = round(mean(Vec),2),

               sd = round(var(Vec),2)))

Vec

დავალება II: აქ Vec1 ვექტორის სიგრძე განისაზღვრება პუასონის განაწილებიდან შემთხვევითი რიცხვის გენერირებით 1000 პარამეტრით, rpois() ფუნქციის გამოყენებით. ეს შემთხვევითი რიცხვი შემდეგ გამოიყენება n არგუმენტის მნიშვნელობად runif() ფუნქციაში. runif() ფუნქცია წარმოქმნის შემთხვევითი რიცხვების ვექტორს ერთიანი განაწილებიდან ინტერვალში [1, 100]. მიღებული ვექტორი ინახება Vec1-ში.შემდეგ ლაინებზე დავითვალეთ ვექტორის სიგრძე(ელემენტების რაოდენობა) , ვიპოვეთ ამ უკანასკნელის დიაპაზონი, ვარიაციის კოეფიციენტი (სტანდარტული

გადახრისა და საშუალო მაჩვენებლის ფარდობა) , პირსონის მეორე ასიმეტრიის კოეფიციენტი

(საშუალოსა და მედიანას შორის სხვაობის გასამმაგების შეფარდება სტანდარტულ გადახრასთან) და ბოლოს კელის ასიმეტრიის კოეფიციენტი (პირველი და მეცხრე დეცილების ჯამისა და

გაორმაგებული მედიანის სხვაობის შეფარდება დეცილთაშორის დიაპაზონთან (ანუ მეცხრე და

პირველ დეცილებს შორის სხვაობასთან)).

number <- rpois(1, lambda = 1000)

Vec1 <- runif(n = rpois(n = 1, lambda = 1000), min = 1, max = 100)

length(Vec1)

max(Vec1) - min(Vec1)

sd(Vec1) / mean(Vec1)

pearson <- (3 \* (mean(Vec1) - median(Vec1))) / sd(Vec1)

first\_decile <- quantile(Vec1, probs = 0.1)

ninth\_decile <- quantile(Vec1, probs = 0.9)

median\_value <- median(Vec1)

range <- ninth\_decile - first\_decile

kelly <- (first\_decile + ninth\_decile) / (2 \* median\_value - range)

დავალება III:

ეს ხაზი ქმნის ვექტორს l1, რომელიც შეიცავს 100 შემთხვევით ლათინურ ასოს. replace() ფუნქცია გამოიყენება ასოების ვექტორიდან შემთხვევითი შერჩევისთვის, რაც ნიშნავს, რომ თითოეული ასო შეიძლება შეირჩეს არაერთხელ. შემდეგი ხაზი ქმნის სხვა ვექტორს l2 იგივე დანიშნულებით, რაც l1. ის ასევე ირჩევს 100 შემთხვევით ლათინურ ასოს, მაგრამ ამ შემთხვევაში, replace() ფუნქციის პირველ არგუმენტად იყენებს length(letters). ეს მიდგომა აღწევს იმავე შედეგს, როგორც მე-2 მეთოდით 3 ხაზი ამოწმებს, არის თუ არა ასოების ვექტორის ყველა ასო l1 ვექტორში. %in% ოპერატორი გამოიყენება შედარების შესასრულებლად და all() ფუნქცია აბრუნებს TRUE-ს, თუ ყველა ელემენტი არის TRUE (რაც ნიშნავს ყველა ასოს არსებობას) ან FALSE სხვა შემთხვევაში. 4 ხაზი წარმოქმნის შემთხვევით კოდს 1-დან 10-მდე სიგრძით. ის იყენებს sample() ფუნქციას ორჯერ კოდის სიგრძის შემთხვევით განსაზღვრა. ჯერ (1:10, 1) ირჩევს შემთხვევით რიცხვს 1-დან 10-მდე. შემდეგ, შემთხვევით ირჩევს ასოებს ასოების ვექტორიდან ჩანაცვლებით. საბოლოოდ, paste0() ფუნქცია აერთიანებს არჩეულ ასოებს ერთ სტრიქონში. 5 ხაზი წარმოქმნის შემთხვევით კოდს 10 სიგრძით. ის იყენებს sample() ფუნქციას შემთხვევითად აირჩევს ასოებს (როგორც მცირე, ასევე დიდი) და რიცხვები (0-დან 9-მდე) კომბინირებული ვექტორიდან c(letters, LETTERS, 0:9 არგუმენტი collapse = "" გამოიყენება paste0()-თან ერთად არჩეული ელემენტების ერთ სტრიქონში შესაერთებლად. შემდეგი ხაზი წინას მსგავსია, მაგრამ ის უზრუნველყოფს რომ კოდი უნდა შეიცავდეს სიმბოლოს "1". ის ამატებს "1"-ს შესაძლო ელემენტების ვექტორს და sample() ფუნქცია ირჩევს ამ გაფართოებული ვექტორიდან. ბოლო ხაზი კი წარმოქმნის მანქანის შემთხვევით ნომერს სამი დიდი ასოს ფორმატით

l1 <- sample(letters, 100, replace = TRUE)

l2 <- letters[sample(length(letters), 100, replace = TRUE)]

el <- all(letters %in% l1)

random\_code <- paste0(sample(letters, sample(1:10, 1), replace = TRUE), collapse = "")

random\_code\_symbols <- paste0(sample(c(letters, LETTERS, 0:9), 10, replace = TRUE), collapse = "")

random\_code\_symbols\_with\_1 <- paste0(sample(c(letters, LETTERS, 0:9, "1"), 10, replace = TRUE), collapse = "")

random\_car\_number <- paste0(sample(LETTERS, 3, replace = TRUE),

                            sample(0:9, 3, replace = TRUE),

                            sample(LETTERS, 2, replace = TRUE))

დავალება IV :შევქმენით ათი ათასი შემთხვევითი რიცხვი ნორმალური განაწილებიდან საშუალოთი 100 და სტანდარტული გადახრით 20, მოვახდინეთ მონაცემების სტანდარტიზაცია საშუალოს გამოკლებით და სტანდარტული გადახრით გაყოფითგამოვთვალეთ სტანდარტიზებული მონაცემების საშუალო და სტანდარტული გადახრა.

randomn <- rnorm(10000, mean = 100, sd = 20)

standard  <- (randomn - mean(randomn)) / sd(randomn)

mean\_value <- mean(standard)

sd\_value <- sd(standard)

mean\_value

sd\_value

დავალება V:

nums <- rnorm(n = 2, mean = c(1, 100), sd = c(0.01, 30)): ეს ხაზი წარმოქმნის ორ შემთხვევით რიცხვს ნორმალური განაწილებიდან. პირველ რიცხვს აქვს საშუალო 1 და სტანდარტული გადახრა 0.01, ხოლო მეორე რიცხვს აქვს საშუალო 100 და სტანდარტული გადახრა 30. მიღებული რიცხვები ინახება ვექტორულ რიცხვებში.

Vec5 <- rpois(n = 10000, lambda = runif(n = 10000, min = 0, max = 10)): ეს ხაზი წარმოქმნის Vec5 ვექტორს, რომელიც შეიცავს 10000 შემთხვევით რიცხვს პუასონის განაწილების შემდეგ. თითოეული შემთხვევითი რიცხვისთვის ლამბდა პარამეტრი გენერირებულია runif(n = 10000, min = 0, max = 10) გამოყენებით, რომელიც წარმოქმნის 10000 შემთხვევით რიცხვს 0-დან 10-მდე.

mean\_value <- mean(Vec5): ეს ითვლის Vec5 ვექტორში ელემენტების საშუალოს და ანიჭებს მას mean\_value.ცვლადს.variance <- var(Vec5): ეს ითვლის Vec5 ვექტორში ელემენტების დისპერსიას და ანიჭებს მას ცვლად variance.mean\_1 <- mean(sample(Vec5, size = 100, replace = T)): ეს ხაზი იღებს 100 ზომის შემთხვევით ნიმუშს Vec5 ვექტორიდან, ჩანაცვლებით sample() ფუნქციის გამოყენებით. შემდეგ ის ითვლის შერჩეული მნიშვნელობების საშუალოს და ანიჭებს მას mean\_1 ცვლადს.variance\_1 <- var(sample(Vec5, size = 100, replace = T)): ანალოგიურად, ეს ხაზი იღებს 100 ზომის შემთხვევით ნიმუშს Vec5-დან ჩანაცვლებით და ითვლის შერჩეული მნიშვნელობების დისპერსიას, რომელიც ინახება ცვლადში variance\_1 .მოკლედ, ეს კოდი აგენერირებს შემთხვევით რიცხვებს ნორმალური და პუასონის განაწილებიდან, ითვლის ვექტორის საშუალოსა და დისპერსიას და ასევე ითვლის შემთხვევითი ნიმუშის საშუალოსა და ვარიანსს ამ ვექტორიდან. მიღებული მნიშვნელობები ინახება ცვლადებში და იბეჭდება კონსოლში.

nums <- rnorm(n = 2, mean = c(1, 100), sd = c(0.01, 30))

Vec5 <- rpois(n = 10000, lambda = runif(n = 10000, min = 0, max = 10))

mean\_value <- mean(Vec5)

variance <- var(Vec5)

mean\_1 <- mean(sample(Vec5, size = 100, replace = T))

variance\_1 <- var(sample(Vec5, size = 100, replace = T))

mean\_1

variance\_1

დავალება VI:

შევქმენით ცარიელი ვექტორი შემდეგ, ჩვენ ვიყენებთ ციკლს პუასონის განაწილებიდან შემთხვევითი რიცხვების გენერირებისთვის 1-დან 9-მდე საშუალებებით. თითოეული i-სთვის 1-დან 9-მდე, ჩვენ ვქმნით შემთხვევით რიცხვს giorgi-ს გამოყენებით rpois(1, lambda = i), სადაც ლამბდა არის შემთხვევითი რიცხვი, განაწილებული პუასონის კანონით, საშუალოთი iგენერირებული რიცხვები ინახება ვექტორის ვექტრის შესაბამის პოზიციებზე. დაბოლოს, ვაერთებთ რიცხვებს vec-ში, თითოეული რიცხვის გაყოფით 10-ის შესაბამის ხარისხზე და შევაჯამებთ საბოლოო შედეგს.

options(digits = 9)

vec <- numeric(9)

for (i in 1:9) {

  giorgi <- rpois(1, lambda = i)

  vec[i] <- ai

}

o <- sum(vec / 10^(1:9))

o

დავალება VII: კოდი ითვლის ალბათობებს P(Π2 = 1), P(Π2 = 2) და P(Π2 = 3) პუასონის განაწილების ფორმულის გამოყენებით პარამეტრით λ = 2. გამოსათვლელად გამოიყენება exp() ფუნქცია. factorial() ფუნქცია ითვლის ფაქტორიალს.

lambda <- 2

pro1 <- (exp(-lambda) \* lambda^1) / factorial(1)

pro2 <- (exp(-lambda) \* lambda^2) / factorial(2)

pro3 <- (exp(-lambda) \* lambda^3) / factorial(3)

pro1

pro2

pro3

დავალება VIII: მოცემულ კოდში Vec8 არის ერთი მილიონი შემთხვევითი რიცხვის ვექტორი, რომელიც გენერირებულია ნორმალური განაწილებიდან, საშუალოდ 150 და სტანდარტული გადახრით 20. abs(Vec8 - 150) ითვლის აბსოლუტურ სხვაობას Vec8-ის თითოეულ ელემენტსა და საშუალო მნიშვნელობას შორის 150. ეს წარმოადგენს თითოეული ელემენტის გადახრას საშუალოდან. პირობა abs(Vec8 - 150) > 60 ამოწმებს, არის თუ არა თითოეული ელემენტის გადახრა 60-ზე მეტი. ეს პირობა ქმნის ლოგიკურ ვექტორს, სადაც TRUE მიუთითებს, რომ შესაბამისი ელემენტის გადახრა მეტია 60-ზე, ხოლო FALSE მიუთითებს, რომ არ არის. sum() ფუნქციის გამოყენებით, გამოითვლება ელემენტების ჯამური რაოდენობა Vec8-ში, სადაც გადახრა 60-ზე მეტია. ამ რაოდენობის გაყოფით Vec8-ში ელემენტების საერთო რაოდენობაზე (რაც ამ შემთხვევაში არის 10^6), მივიღებთ იმ ელემენტების პროპორციას, რომელთა გადახრა 60-ზე მეტია.დაბოლოს, 100-ზე გამრავლება გვაძლევს Vec8-ში ელემენტების პროცენტს, რომლებსაც აქვთ 60-ზე მეტი გადახრა

Vec8 <- rnorm(n = 10^6, mean = 150,sd = 20)sum(abs(Vec8-150)>60)/10^6\*100