Løsningsforslag øving 8, ST1301

Oppgave 1 Hva gjør følgende funksjon? Hvilken fordeling har variabelen `n` som returneres som funksjonsverdi? Forklar hvorfor. Forutsett at to enkle positive tall blir oppgitt som argumenter.

```
xxx <- function(t,lambda) {
  tidspunkt <- 0
  n <- 0
  repeat {
     ventetid <- rexp(n=1,rate=lambda)
     tidspunkt <- tidspunkt + ventetid
     if (tidspunkt > t)
        break()
     else
        n <- n + 1
  }
  n
}
```

Inne i repeat-løkken simuleres eksponentielt fordelte ventetider. Disse legges forløpende til variabelen `tidspunkt`. Verdien av variabelen `tidspunkt` vil derfor kunne betraktes som tidspunkter til hendelser i en Poisson-prosess. Når summen av ventetidene (variabelen `tidspunkt`) blir større enn argumentet `t` avbrytes repeat-løkken. Antallet ventetider som inngår i summen (unntatt siste) (og dermed antall hendelser i intervallet) blir telt opp ved bruk av tellevariabelen `n`. Denne vil dermed blir Poissonfordelt med parameter lik produktet av argumentene `t` og `lambda`.

Oppgave 2 Anta at X er en triangulært fordelt stokastisk variabel med sannsynlighetstetthet

$$f_X(x) = 2x, \quad 0 \leq x \leq 1. \tag{1}$$

Utled kumulativ sannsynlighetstetthet til X, $F_X(x) = P(X \leq x)$. Programmer så en funksjon som simulerer n realisasjoner fra $f_X(x)$ ved bruk av inversjonsmetoden. Lag et histogram av 1000 realisasjoner fra fordelingen.

Estimer også forventningen til X basert på 1000 simulerte realisasjoner av X og kontroller resultatet ved å sammenligne med den teoretisk forventingen $E(X)$ (utled denne).

Den kumulative fordelingen blir

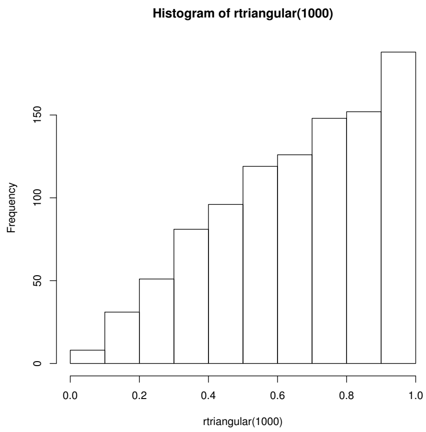
$$F_X(x) = \int_0^x f_X(u)du = \int_0^x 2udu = u^2\Big|_0^1 = x^2 \tag{2}$$

Inversjonsmetoden gir at

$$X = F_X^{-1}(U) = \sqrt{U}. \tag{3}$$

Funksjon som simulerer n realisasjoner fra (1) blir

```
rtriangular <- function(n) {
   u <- runif(n=n)
   return(sqrt(u))
}
> hist(rtriangular(1000))
```




```
> mean(rtriangular(n=1000))
[1] 0.6681152
```

Vi ser at estimatet av forventningen basert på simulerte realisasjoner blir tilnærmet lik den teoretiske forventningen

$$EX = \int_0^1 x f_X(x)dx = \int_0^1 2x^2 dx = \frac{2}{3}x^3\Big|_0^x = \frac{2}{3}. \tag{4}$$

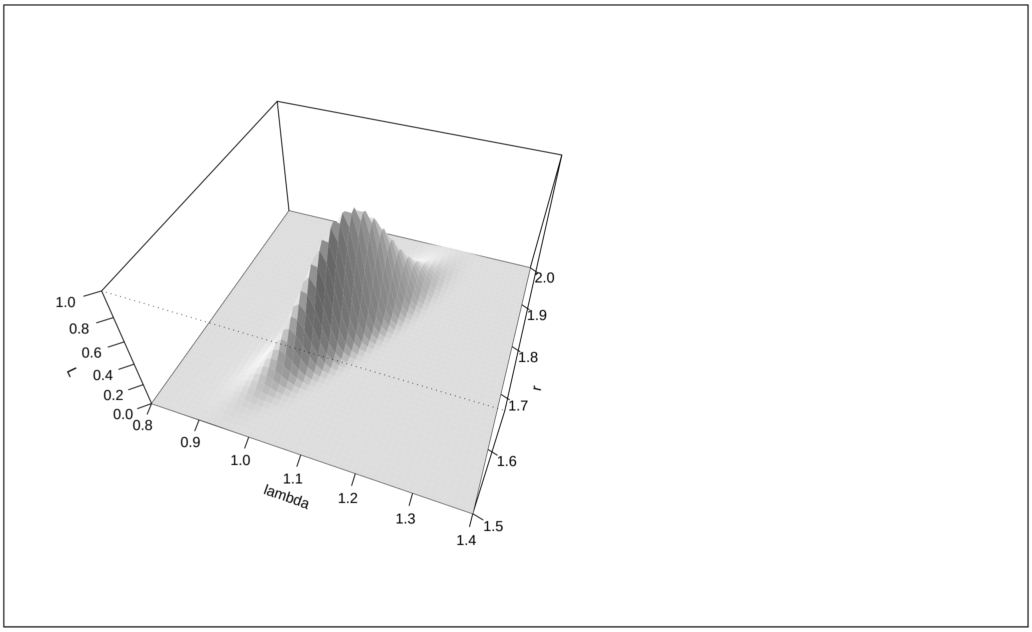
Oppgave 3 I øving 7 tilpasset vi gammefordeling til et sett observerte levetider og beregnet sannsynlighetsmaksimeringsestimatene av parameterne i modellen til $\hat{\lambda} = 1.07$ og $\hat{r} = 1.75$. Vi trenger å kontrollere at estimatene er riktige (d.v.s. lik parameterverdiene som maksimaliserer likelihoodfunksjonen). Lag et tredimensjonalt overflateplot av likelihoodfunksjonen (L) som funksjon av parameterne λ og r ved hjelp av funksjonen `persp`. Denne funksjonen trenger tre hovedargumenter; to vektorer `x` og `y` som spesifiserer punkter på et rutenett (i dette tilfelle sekvenser av λ og r verdier, og en matrise `z` som inneholder funksjonsverdier (i dette tilfelle $L(\lambda, r)$) i de ulike punktene på rutenettet.

Programmer en funksjon som lager et slikt overflateplot ved å bygge videre på funksjonen `lnL` fra øving 7. La max og min verdier for hver av de to parameterne være innargument. Bruk en nøstet for-løkke for å beregne matrisen av funksjonsverdier før kallet til `persp`.

En teknikalitet: Merk at likelihoodfunksjonen ($L = e^{\ln L}$) blir mindre enn minste tall vi kan jobbe med i R slik at `exp(lnL( ))`vil få verdien 0 i beregningene (numerisk underflow). Derfor er det bedre å ta vare på log-verdiene av L i nevnte matrise og så trekke fra maximum av alle verdiene før vi beregner L ved ekponentiering (dette vil i praksis bare innebære en reskalering av L som gjør at vi ungår underflow.)

Undersøk de forskjellige tilleggsargumentene til `persp` som `expand`, `theta`, og `phi`, `shade`, `border`, og `ticktype=”detailed”` (se hjelpesiden).

```
## Pluss log-likelihoodfunksjonen fra øving 7.
lnL <- function(p,t) {
   -sum(dgamma(t,rate=p[1],shape=p[2],log=T))
}
overflateplot <- function(min1,max1,min2,max2,gridsize=50,...) {
  x1 <- seq(min1,max1,length=gridsize)
  x2 <- seq(min2,max2,length=gridsize)
  lnLverdier <- matrix(NA,ncol=gridsize,nrow=gridsize)
  for (i in 1:gridsize) {
     for (j in 1:gridsize) {
        lnLverdier[i,j] <- -lnL(c(x1[i],x2[j]),...)
     }
  }
  persp(x1,x2,exp(lnLverdier-max(lnLverdier)),
    expand=.5,shade=.2,border=NA,ticktype="detailed",
    phi=45,theta=20,zlab="L",xlab="lambda",ylab="r")
}
overflateplot(.8,1.4, 1.5,2, t=t)
```



Programmer en funksjon som simulerer 1000 tilfeldig utvalg fra den tilpassede modellen (bruk `rgamma` og anta at parameterverdiene er lik estimatene beregnet ovenfor) og returner tilhørende estimat i form av en 1000×2 matrise. Estimer standardfeilen til estimatene på grunnlag av de simulerte verdiene. Lag også et spredningsplot av de simulerte estimatene. Er det noen likhet mellom spredningsplottet og overflateplottet av likelihoodfunksjonen?

Simuleringsalgoritmen blir som tidligere bortsett fra at vi må ta vare på simulerte estimat i en matrise. Legg også merke til bruken av radvis tilordning:

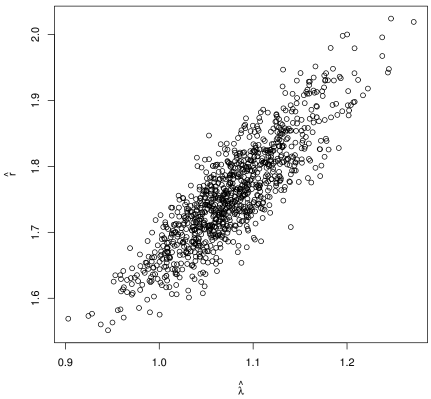
```
gammasim <- function(par,n=870,nsim=1000) {
   parmatrise <- matrix(NA,ncol=2,nrow=nsim)
   for (i in 1:nsim) {
      t <- rgamma(n=n,rate=par[1],shape=par[2])
      parmatrise[i,] <- mlegamma(t)
   }
   return(parmatrise)
}
> gammaboot<-gammasim(gammahat)
> sd(gammaboot)
[1] 0.05334645 0.07885344
> apply(gammaboot,2,mean)
[1] 1.077368 1.757135
```

```
> apply(gammaboot,2,mean)-gammahat
[1] 0.004026239 0.005243380
```

Punktestimatene av parameterne med standardfeil blir altså $\hat{\lambda} = 1.07 \pm 0.05$ og $\hat{r} = 1.75 \pm 0.07$. Vi ser også at forventningsfeilen blir svært liten; dette er ikke uventet når utvalgstørrelsen er såpass stor. Det kan vises at sannsynlighetsmaksimeringsestimater under svært generelle betingelser vil være såkalt konsistente; dette betyr at sannsynligheten for at estimatene vil ligge vilkårlig nærme de sanne parameterverdiene vil gå mot 1 når utvalgsstørrelsen går mot uendelig.

Spredningsplot kan lages på følgende måte.

```
> plot(gammaboot,xlab=expression(hat(lambda)),
                 ylab=expression(hat(r)))
```



Skriv `help(plotmath)` for informasjon om hvordan vi kan skrive matematiske uttrykk i plot.

Vi ser at simultantettheten til estimatorene $\hat{\lambda}$ og $\hat{r}$ langt på vei faller sammen med (er proporsjonal med) likelihoodfunksjonen plottet i del b).

Følgende kan og bemerkes. I følge såkalt asymptotisk teori (se f.eks. kurset Statistisk Inferens) vises det at fordelingen til sannsynlighetsmaksimeringsestimater vil konvergere mot en multivariat normalfordeling når $n \to \infty$.

I følge Bernstein-von Mises teoremet vil også formen på likelihoodfunksjonen bli Gaussisk og proposjonal med fordeling til SME'ene når når utvalgsstørrelsen går mot uendelig.