STK1110 – Statistiske metoder og dataanalyse
Obligatorisk innlevering 1

Rohullah Akbari[1]
[1] Institutt for matematikk, Universitetet i Oslo (UiO)

October 1, 2020

Contents

Oppgave 1

Vi antar at $Y_1, ..., Y_n \overset{uif}{\sim} N(\mu, \sigma^2)$, og at $X_i = e^{Y_i}$ der $i = 1, ..., n$. Da er $X_1 ... X_n \overset{uif}{\sim} log - N(\mu, \sigma^2)$.

a) Vi har den momentgenerende funksjon lik:

$$M_y(t) = e^{\mu t + \frac{1}{2}\sigma^2 t^2}$$

Fra STK1100 har vi at $E(e^{tx}) = M_x(t)$. I vårt tilfellet blir dette lik:

$$E(e^{ty}) = M_y(t)$$

Så for $E(X_i)$ og $E(X_i^2)$ er henholdsvis $t = 1$ og $t = 2$. Starter med å sette verdien for X_i:

$$E(X_i) = E(e^{Y_i}) = M(1) = e^{\mu + \frac{1}{2}\sigma^2}$$

Og verdien for X_i^2:

$$E(X_i^2) = E(e^{2Y_i}) = M(2) = e^{2\mu + 2\sigma^2}$$

Dermed har vist uttrykkene for $E(X_i)$ og $E(X_i^2)$.

b) Før vi finner momentestimatorene så er det viktig å ha ett uttrykk for variansen. Vi har at:

$$Var(X_i) = E(X_i^2) - E(X_i)^2 = e^{2\mu + 2\sigma^2} - e^{2\mu + \sigma^2}$$

$$= e^{2\mu + \sigma^2}(e^{\sigma^2} - 1) = E(X)^2(e^{\sigma^2} - 1).$$

Momemtestimatoren for μ:

$$E(X_i) = E(e^{Y_i}) = M(1) = e^{\mu + \frac{1}{2}\sigma^2}$$

$$\ln[E(X_i)] = \mu + \frac{1}{2}\sigma^2$$

Dette gir:

$$\hat{\mu} = \ln[E(X_i)] - \frac{1}{2}\sigma^2$$

$$\hat{\mu} = \ln[\bar{X}] - \frac{1}{2}\sigma^2$$

Momemtestimatoren for σ^2:

$$Var(X_i) = \bar{X}^2(e^{\sigma^2} - 1)$$

$$\implies e^{\sigma^2} = \ln\left(\frac{V(x)}{\bar{X}^2} + 1\right)$$

Dette gir:

$$\hat{\sigma^2} = \ln\left(\frac{V(x)}{\bar{X}^2} + 1\right)$$

Der $E(X_i) = \bar{X}$ og $V(x) = S^2$ er de empiriske forventningsverdien og variansen. Bruker R til å beregne de tilsvarende estimatene for bilforsikringskravene med følgende kode:

```
data = read.table("https://www.uio.no/studier/emner/
    matnat/math/STK1110/data/forsikringskrav.txt",
    header=T)
tall = data$X7.708
sigma2 = 1+(var(tall)/mean(tall)^2)
mu = log(mean(tall)) - 1/2 * sigma2^2
c("Forventingsverdi:",mu,"Variansen:",sigma2)
```

Fikk følgende output:

```
[1] "Forventingsverdi:" "2.78772"  "Variansen:" "
    0.89015"
```

Altså de beregnede estimatene ble $\mu = 2.78772$ og $\sigma^2 = 0.89015$.

c) $X_1 ... X_n$ er lognormal fordelt og har tetthetsfunksjonen

$$f(x;\mu,\sigma) = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma x} e^{\frac{-[\ln x - \mu]^2}{2\sigma^2}} & \text{if } x \geq 0 \\ -x & \text{if } x < 0 \end{cases}$$

Dette gir Likelihood-funksjonen:

$$\mathcal{L}(\mu,\sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma x_i} e^{\frac{-[\ln x_i - \mu]^2}{2\sigma^2}}$$

Tar logaritmen til denne:

$$\ell = \ln \mathcal{L}(\mu,\sigma^2) = \sum_{i=1}^{n} \ln \frac{1}{\sqrt{2\pi}\sigma x_i} - \sum_{i=1}^{n} \frac{[\ln x_i - \mu]^2}{2\sigma^2}$$

Deriverer denne først med hensyn på μ og setter den lik 0:

$$\frac{\partial}{\partial \mu}\ell = \sum_{i=1}^{n} \frac{\ln x_i - \mu}{\sigma^2}$$

$$\sum_{i=1}^{n} \frac{\ln x_i - \mu}{\sigma^2} = 0 \implies \sum_{i=1}^{n} \frac{\ln x_i}{\sigma^2} - \frac{n\mu}{\sigma^2} = 0$$

$$\implies \hat{\mu} = \frac{1}{n} \sum_{i=0}^{n} \ln(x_i)$$

Gjør det samme for σ^2. Bare for å gjøre det lettere så setter $\theta = \sigma^2$:

$$\ell = \ln \mathcal{L}(\mu, \sigma^2) = \sum_{i=1}^{n} \ln \frac{1}{\sqrt{2\pi}\sigma x_i} - \sum_{i=1}^{n} \frac{[\ln x_i - \mu]^2}{2\sigma^2}$$

$$= \sum_{i=1}^{n} \ln \frac{1}{\sqrt{2\pi\theta} x_i} - \sum_{i=1}^{n} \frac{[\ln x_i - \mu]^2}{2\theta}$$

$$= -n \ln \left[\sqrt{2\pi\theta} \right] - \sum_{i=1}^{n} \ln(x_i) - \sum_{i=1}^{n} \frac{[\ln x_i - \mu]^2}{2\theta}$$

Derivasjon gir:

$$\frac{\partial}{\partial \theta} \ell = -\frac{n}{2\theta} + \frac{1}{2\theta^2} \sum_{i=1}^{n} [\ln x_i - \mu]^2$$

Setter den lik 0:

$$-\frac{n}{2\theta} + \frac{1}{2\theta^2} \sum_{i=1}^{n} [\ln x_i - \mu]^2 = 0$$

$$\implies \hat{\theta} = \hat{\sigma^2} = \frac{1}{n} \sum_{i=1}^{n} [\ln x_i - \mu]^2$$

Beregner de tilsvarende estimatene for bilforsikringskravene med R, ved følgende kode:

```
m = 1/length(tall) * sum(log(tall))
s2 = 1/length(tall) * sum((log(tall) - m)^2)
c("Forventingsverdi:",m,"Variansen:",s2)
```

Får følgende output:

```
[1] "Forventingsverdi:" "2.78239"  "Variansen:"
 "0.76580"
```

Altså de beregnede estimatene ble $\mu = 2.78239$ og $\sigma^2 = 0.76580$. Vi ser at det er en forskjell mellom MLE-estimatene og moment-estimatene. Forventningsverdiene er 99.80% like, mens variansene er 86.03% like. MLE er det meste brukte metoden for å finne estimatorer til en fordeling siden den maksimerer log-likelihood funksjonen, siden den antar litt "strengere" krav og derfor er nøyaktigere enn moment-estimatorene. Men i dette tilfellet så er faktisk forventningsverdiene ganske like, men er en liten forskjell mellom variansene.

d) Vi husker at $Y_i = \ln(X_i)$. En alternativ måte for å finne MLE-estimatorene til log-normalfordeling er å finne MLE-estimatorene til en normalfordeling og deretter bruke det til å finne MLE-estimatorene for log-normalfordeling. Fra s.355 i læreboka har vi MLE-estimatorene for parametere i en normalfordeling:

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} Y_i$$

$$\hat{\sigma^2} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{\mu})^2$$

Vi setter inn funksjonen for Y_i og får

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} \ln(X_i)$$

$$\hat{\sigma^2} = \frac{1}{n}\sum_{i=1}^{n}(\ln(X_i) - \hat{\mu})^2$$

e) Vi har at

$$E(X_i) = e^{\mu + \frac{1}{2}\sigma^2} = \phi(\mu, \sigma^2)$$

Funksjonen $\phi(\mu, \sigma^2)$ inneholder parameterene μ, σ^2 som er MLE-estimatorer av $\hat{\mu}, \hat{\sigma^2}$. Derfor, i følge invarians-prinsippet på s.357, blir MLE-estimator for ϕ:

$$\hat{\phi} = e^{\hat{\mu} + \frac{1}{2}\hat{\sigma^2}}$$

Den beregnede verdien blir da:

$$\hat{\phi} = 23.695$$

En alternativ estimator for $E(X_i)$ er $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$. Beregning av estimatet $\bar{x}$ gir

$$\bar{x} = 24.141$$

Disse estimatene er 98.15% like. MLE pleier mer nøyaktig enn andre metoder brukt for å finne estimatorer, men i dette tilfellet er den ikke så ulik $\bar{x}$-estimatet.

f) I denne oppgaven bruker vi faktumet om at n er stor. Generelt sett har vi konfidensintervallet gitt ved:

$$P(\bar{x} - z_{\alpha/2}\frac{\sigma}{\sqrt{n}} \leq \mu \leq \bar{x} + z_{\alpha/2}\frac{\sigma}{\sqrt{n}}) = 1 - \alpha$$

Siden vi skal finne 95% intervallet så gir dette:

$$1 - \alpha = 0.95 \implies \alpha = 0.05$$

Finner verdien til $z_{\alpha/2} = z_{0.025=1.96}$. Bruker R til å beregne intervallet med følgende kode:

```
x_strek = mean(tall)
sigma = sd(tall)
z = 1.96
n = length(tall)
low = x_strek - (z*sigma /sqrt(n))
high = x_strek + (z*sigma /sqrt(n))
c("Lav: ",low,"Hoy:",high)
```

Får følgende output:

```
[1] "Lav: "    "23.431" "Hoy:"    "24.851"
```

95% konfidensintervall for $E(X_i)$ ble $(23.431, 24.851)$.

g) Vi antar X_i-ene er normalfordelt og dermed lager 95% konfidensintervall for Var(X). Generelt sett har vi følgende konfidensintervall for varians:

$$P(\chi_{1-\alpha/2,n-1} \leq \frac{(n-1)s^2}{\sigma^2} \leq \chi_{\alpha/2,n-1}) = 1 - \alpha$$

$$\implies \frac{\chi_{1-\alpha/2,n-1}}{(n-1)s^2} \leq \frac{1}{\sigma^2} \leq \frac{\chi_{\alpha/2,n-1}}{(n-1)s^2}$$

$$\implies \frac{(n-1)s^2}{\chi_{1-\alpha/2,n-1}} \geq \sigma^2 \geq \frac{(n-1)s^2}{\chi_{\alpha/2,n-1}}$$

Altså konfidensintervallet ble:

$$\left(\frac{(n-1)s^2}{\chi_{\alpha/2,n-1}}, \frac{(n-1)s^2}{\chi_{1-\alpha/2,n-1}}\right)$$

Beregner dette med R:

```
nedre = (n-1)*var(tall)/qchisq(0.025, df=n-1)
ovre = (n-1)*var(tall)/qchisq(0.975, df=n-1)
c("Nedre: ",low,"Ovre:",high)
```

Som ga følgende output:

```
[1] "Nedre: "          "23.431"    "Ovre:"      "24.851"
```

Altså 95% konfidensintervallet blir $(23.431, 24.851)$.

h)

Oppgave 2

a) I denne oppgaven så har vi en "liten" n-verdi, nemlig n=15, og derfor må vi bruke konfidensintervaller med t-fordeling. Generelt har vi at:

$$P\left(\hat{x} - t_{\alpha/2,n-1}\frac{s}{\sqrt{n}} \leq \mu \leq \hat{x} + t_{\alpha/2,n-1}\frac{s}{\sqrt{n}}\right) = 1 - \alpha$$

For 95% har vi:

$$1 - \alpha = 0.95 \implies \alpha = 0.05$$

Da har vi:

$$P\left(\hat{x} - t_{0.025,14}\frac{s}{\sqrt{15}} \leq \mu \leq \hat{x} + t_{\alpha/2,14}\frac{s}{\sqrt{15}}\right) = 1 - \alpha$$

der $t_{\alpha/2,14} = 2.145$, $\bar{x}$ og s er henholdssvis gjennomsnittet og empirisk standardavvik.

b) Tolker "10000 datasett hvet med størrelse" som en matrise M av størrelse 15x10000. Starter med å danne en matrise av denne størrelsen, og samtidig bruker funksjonen **rnorm()** til å generere stokastiske variablene $X_1 ... X_15$. I tillegg til det så settes det opp en for-løkke som løper gjennom alle kolonne i matrisene og fyller den opp med de stokastiske variablene :

```
matrise_str = 10000
M = matrix(data = NA, nrow = matrise_str, ncol = 15,
    byrow = FALSE, dimnames = NULL)
for(i in 1:matrise_str){
    M[i,] = rnorm(15,558,30)
}
```

Deretter settes inn uttrykkene for nedre og øvre grensene for intervallet:

```
matrise_str = 10000
M = matrix(data = NA, nrow = matrise_str, ncol = 15,
    byrow = FALSE, dimnames = NULL)
ovre_vektor = vector()
nedre_vektor = vector()
for(i in 1:matrise_str){
   M[i,] = rnorm(15,558,30)
    nedre_vektor[i] = mean(M[i,])-qt(0.975,14)*sd(M[i,])
        /sqrt(15)
    ovre_vektor[i] = mean(M[i,])+qt(0.975,14)*sd(M[i,])/
        sqrt(15)
 }
```

For å tell opp andelen av disse intervallene som inneholder verdien 558, setter vi opp en if-test som sjekker om 558 er innenfor nedre og øvre-grensene. Deretter lager vi en variabel *num* som blir addert med 1 hvis 558 er inne intervallet:

```
matrise_str = 10000
M = matrix(data = NA, nrow = matrise_str, ncol = 15,
    byrow = FALSE, dimnames = NULL)
ovre_vektor = vector()
nedre_vektor = vector()
num = 0
for(i in 1:matrise_str){
    M[i,] = rnorm(15,558,30)
    #intervallet_fra_opp_a
    nedre_vektor[i] = mean(M[i,])-qt(0.975,14)*sd(M[i
        ,])/sqrt(15)
    ovre_vektor[i] = mean(M[i,])+qt(0.975,14)*sd(M[i,])
        /sqrt(15)
    if(nedre_vektor[i] < 558 && ovre_vektor[i] > 558){
      num = num +1 }
}
print(c("Antall 558: ", num,"Prosent",num/matrise_str *
    100))
```

Får følgende output:

```
"Antall 558: " "9543"          "Prosent"       "95.43"
```

Det vil si at ved en tilfeldig kjøring av programmet så inneholdt intervallet forventningsverdien ca.95.43% av gangene. Dette stemmer ganske godt med at vi var ute etter 95% konfidensintervall.

c) Denne oppgaven blir helt likt som ovenfor, men det eneste forskjellen er at nå har vi noen andre uttrykk for nedre og øvre-grensene. Alt utenom det blir helt likt:

```
for(i in 1:matrise_str){
    M[i,] = rnorm(15,558,30)
     nedre_vektor[i] = mean(M[i,]) -1.96*sd(M[i,])/sqrt
         (15)
     ovre_vektor[i] = mean(M[i,]) +1.96*sd(M[i,])/sqrt
         (15)
     if(nedre_vektor[i] < 558 && ovre_vektor[i] > 558){
       num = num +1 }
}
print(c("Antall 558: ", num,"Prosent",num/matrise_str *
    100))
```

Får følgende output:

```
[1] "Antall 558: " "9284"       "Prosent"      "92.84"
```

Vi ser at vi fikk mindre prosentandel enn oppgave b). Dette skyldes at disse tilnærmede intervallene gir gode resultater for større n-verdier, og er ikke så presis for "små" n-verdier. Men vårt tilfellet så er den faktisk ikke så langt unna.

d) For å lage konfidensintervaller for σ, bruker vi resultatet fra oppgave 1c:

$$P\left(\frac{(n-1)s^2}{\chi_{\alpha/2,n-1}} \leq \sigma^2 \leq \frac{(n-1)s^2}{\chi_{1-\alpha/2,n-1}}\right) = 1-\alpha$$

$$\implies P\left(\sqrt{\frac{(n-1)s^2}{\chi_{\alpha/2,n-1}}} \leq \sigma \leq \sqrt{\frac{(n-1)s^2}{\chi_{1-\alpha/2,n-1}}}\right) = 1-\alpha$$

Setter inn verdien for $\alpha = 0.05$, $n = 15$, $\chi_{0.025,n-1} = 26.119$ og $\chi_{0.975,n-1} = 5.629$. Vi får følgende intervaller:

$$\left(\sqrt{\frac{(14)s^2}{26.119}}, \sqrt{\frac{(14)s^2}{5.629}}\right)$$

R-koden blir helt samme som i oppgave b og c, men forskjellen blir uttrykkene for øvre og nedre-grensene, og i tillegg til det så er vi nå opptatt av å finne om intervallet inneholder 30 og ikke 558. Slik ble koden:

```
for(i in 1:matrise_str){
    M[i,] = rnorm(15,558,30)
    nedre_vektor[i] = sqrt((n-1)*sd(M[i,])^2 /(26.119 )
        )
    ovre_vektor[i] =sqrt( (n-1)*sd(M[i,])^2 /(5.629) )
    if(nedre_vektor[i] < 30 && ovre_vektor[i] > 30){
      num = num +1 }
 }
 print(c("Antall 30: ", num,"Prosent",num/matrise_str *
     100))
```

Slik ble outputtet:

```
[1] "Antall 30: " "9509"        "Prosent"     "95.09"
```

Som vi ser det, under denne kjøringen, så var det hele 9509 av intervallene som inneholdt standardavviket.

e) Vi antar at $Z_1 ... Z_15 \overset{uif}{\sim} t_7$. I denne deloppgaven fortsetter vi koden fra oppgave b) og c), men legger til noen flere linjer. Vi starter med å definere definere to matriser M og Z. Matrisen Z blir brukt til å lage variablene når vi trekker $z_1 ... z_15$ fra t_7 med R-funksjonen **rt()**. Deretter brukes det formelen, fra oppgaveteksten, $x_i = \mu + \sigma z_i$. Slik ser implementasjonen:

```
matrise_str = 10000
M = matrix(data = NA, nrow = matrise_str, ncol = 15,
    byrow = FALSE, dimnames = NULL)
Z = matrix(data = NA, nrow = matrise_str, ncol = 15,
    byrow = FALSE, dimnames = NULL)
for(i in 1:matrise_str){
    Z[i,] = rt(n = 15, df = 7)
    M[i,] = mu + sigma*Z[i,]
}
```

Deretter gjentar vi samme prosessen som i oppgave b):

```
for(i in 1:matrise_str){
    Z[i,] = rt(n = 15, df = 7)
    M[i,] = mu + sigma*Z[i,]
    nedre_vektor[i] = mean(M[i,])-qt(0.975,14)*sd(M[i
        ,])/sqrt(15)
    ovre_vektor[i] = mean(M[i,])+qt(0.975,14)*sd(M[i,])
        /sqrt(15)
    if(nedre_vektor[i] < 558 && ovre_vektor[i] > 558){
```

```
        num = num +1 }
  }
print(c("Antall 558: ", num,"Prosent",num/matrise_str *
    100))
```

For å sjekke om hvor robust denne metoden er for å lage konfidensintervall for forventningsverdien for antakelsen om normalfordeling, så sammenlignes prosentandel fra oppgave b) med denne oppgaven. Listen under 10 kjøringer av begge metodene, samt prosentandelene.

```
[1] "Kjoring:" "1"         "Opp_b:"   "95.15"    "Opp_e:
    "   "95.16"
[1] "Kjoring:" "2"         "Opp_b:"   "94.99"    "Opp_e:
    "   "95.22"
[1] "Kjoring:" "3"         "Opp_b:"   "95.31"    "Opp_e:
    "   "95.25"
[1] "Kjoring:" "4"         "Opp_b:"   "95.17"    "Opp_e:
    "   "95.24"
[1] "Kjoring:" "5"         "Opp_b:"   "94.76"    "Opp_e:
    "   "95.23"
[1] "Kjoring:" "6"         "Opp_b:"   "95.59"    "Opp_e:
    "   "95.22"
[1] "Kjoring:" "7"         "Opp_b:"   "94.77"    "Opp_e:
    "   "95.03"
[1] "Kjoring:" "8"         "Opp_b:"   "94.68"    "Opp_e:
    "   "94.96"
[1] "Kjoring:" "9"         "Opp_b:"   "95.01"    "Opp_e:
    "   "95.38"
[1] "Kjoring:" "10"        "Opp_b:"   "95.19"    "Opp_e:
    "   "95.36"
```

Som vi ser her så har metoden fra oppgave e) bare en kjøring der den har under 95%, mens metoden fra oppgave b) har 4 kjøringer med andelprosent under 95%. Utfra dette så kan vi konkludere med at denne metoden fra oppgave e) er mer robust enn metoden fra oppgave b).

f) Gjentar den samme prosedyren som i oppgaven overnfor. Forkjellen er at vi har nå en annen verdi for variansen, $\tilde{\sigma^2} = 1.4\sigma$. Koden ble:

```
matrise_str = 10000
M = matrix(data = NA, nrow = matrise_str, ncol = 15,
    byrow = FALSE, dimnames = NULL)
Z = matrix(data = NA, nrow = matrise_str, ncol = 15,
    byrow = FALSE, dimnames = NULL)
```

```
sigma_tilde = 30*1.4
for(i in 1:matrise_str){
   Z[i,] = rt(n = 15, df = 7)
   M[i,] = mu + sigma*Z[i,]
    nedre_vektor[i] = sqrt((n-1)*(sd(M[i,]*1.4))^2
        /(26.119 ))
    ovre_vektor[i] = sqrt((n-1)*(sd(M[i,]*1.4))^2
        /(5.629) )
    if(nedre_vektor[i] < sigma_tilde && ovre_vektor[i]
       > sigma_tilde){
      num = num +1 }
 }
```

Får følgende output:

```
[1] "Antall 1.4*30: " "7851"            "Prosent"
          "78.51"
```

Antall intervaller som inneholdt 1.4σ var på 7851. Som vi ser her så ser vi at denne metoden har lavere antall som inneholder standardavviket.