STK1110 - Statistiske metoder og dataanalyse høsten 2022
Løsningsforslag til ekstraoppgave 3.1

Ekstraoppgave 3.1

a) Vi finner først den kumulative fordelingen til Y_i-ene:

$$F_Y(y;\theta) = P(Y_i \leq y) = P(g(X_i) \leq y) = P(h(g(X_i)) \leq h(y)) = P(X_i \leq h(y)) = F_X(h(y);\theta).$$

Vi derivere og finner at tettheten til Y_i-ene er gitt ved:

$$f_Y(y;\theta) = F'_Y(y;\theta) = F'_X(h(y);\theta)h'(y) = f_X(h(y);\theta)h'(y)$$

Likelihooden basert på X_i-ene er nå

$$\mathcal{L}_X(\theta) = \prod_{i=1}^{n} f_X(x_i;\theta),$$

og likelhooden basert på Y_i-ene er

$$\mathcal{L}_Y(\theta) = \prod_{i=1}^{n} f_Y(y_i;\theta) = \prod_{i=1}^{n} f_X(h(y_i);\theta)h'(y_i) = \left(\prod_{i=1}^{n} f_X(x_i;\theta)\right)\left(\prod_{i=1}^{n} h'(y_i)\right) = \left(\prod_{i=1}^{n} h'(y_i)\right)\mathcal{L}_X(\theta).$$

Vi merker oss at $\prod_{i=1}^{n} h'(y_i)$ er en konstant med hensyn til parameteren θ. Derfor er de to likelihoodene $\mathcal{L}_Y(\theta)$ og $\mathcal{L}_X(\theta)$ proporsjonale. Det betyr at θ som maksimerer $\mathcal{L}_X(\theta)$ også er den θ som maksimerer $\mathcal{L}_Y(\theta)$. Det vil si at maximum likelihood estimatoren basert på X_i-ene og maximum likelihood estimatoren basert på Y_i-er vil være den samme.

b) For normalfordelingen har vi at $\mathrm{E}[X] = \mu$ og $\mathrm{E}(X^2) = \mathrm{V}(X) + (E[X])^2 = \sigma^2 + \mu^2$.

Momentestimatorene er derfor gitt som løsningene av likningene

$$\overline{X} = \widehat{\mu}$$

$$\frac{1}{n}\sum_{i=1}^{n} X_i^2 = \widehat{\sigma}^2 + \widehat{\mu}^2$$

Det gir

$$\widehat{\mu} = \overline{X}$$

$$\widehat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n} X_i^2 - \overline{X}^2 = \frac{1}{n}\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2$$

c) Fra eksempel 7.18 i læreboka har vi at maximum likelihood estimatorene $\widehat{\mu}$ og $\widehat{\sigma}^2$ er lik moment-estimatorene fra fra punkt b). I punkt a) har vi vist at strengt voksende transformasjoner av X_i-ene

ikke endrer maximum likelihood estimatorene. Altså er maximum likelihood estimatorene for μ og σ^2 basert på Y_i-ene også lik momentestimatorene fra punkt b).

d) Vi har at $Y_i = \exp(X_i)$. Derfor er

$$\begin{aligned} \mathrm{E}[Y_i] &= \mathrm{E}[e^{X_i}] = M_X(1) = \exp[\mu + \sigma^2/2] \\ \mathrm{E}[Y_i^2] &= \mathrm{E}[e^{2X_i}] = M_X(2) = \exp[2\mu + 2\sigma^2] \end{aligned}$$

e) Ved å bruke resultatet i punkt d), får vi at momentestimatorene basert på Y_i-ene er løsningen av likningene

$$\begin{aligned} \overline{Y} &= \exp[\widehat{\mu} + \widehat{\sigma}^2/2] \\ \frac{1}{n}\sum_{i=1}^{n} Y_i^2 &= \exp[2\widehat{\mu} + 2\widehat{\sigma}^2] \end{aligned}$$

Vi tar logaritmen på begge sider av likningene over og får likningene:

$$\begin{aligned} \ln(\overline{Y}) &= \widehat{\mu} + \frac{\widehat{\sigma}^2}{2} \\ \ln\left(\frac{1}{n}\sum_{i=1}^{n} Y_i^2\right) &= 2\widehat{\mu} + 2\widehat{\sigma}^2 \end{aligned}$$

Her multipliserer vi den første likningen med 2 og trekker den fra den andre likningen. Da får vi

$$\widehat{\sigma}^2 = \ln\left(\frac{1}{n}\sum_{i=1}^{n} Y_i^2\right) - 2\ln(\overline{Y})$$

Så setter vi dette inn i den første likningen og får:

$$\widehat{\mu} = 2\ln\overline{Y} - \frac{1}{2}\ln\left(\frac{1}{n}\sum_{i=1}^{n} Y_i^2\right)$$

f)

```
> # f)
> # For reproduserbarhet angit vi seed:
> set.seed(1)
> # Simuler data
> x = rnorm(n=10, mean=1, sd=sqrt(3))
> y = exp(x)
>
> # Lag en ramme for å skrive inn resultatet.
> result.matrix = as.data.frame(matrix(NA, nrow=2, ncol=2))
> colnames(result.matrix) = c("mu", "sigma.sq")
> rownames(result.matrix) = c("X", "Y")
>
> # Momentestimatorene basert på X.
> result.matrix["X","mu"] = mean(x)
> result.matrix["X","sigma.sq"] = mean(x^2) - mean(x)^2
>
> # Momentestimatorene basert på Y.
> result.matrix["Y","mu"] = 2*log(mean(y)) - (1/2)*log(mean(y^2))
> result.matrix["Y","sigma.sq"] = log(mean(y^2)) - 2*log(mean(y))
>
> # Vis resultatet.
> print(result.matrix)
         mu sigma.sq
X 1.228982 1.645149
Y 1.460774 1.209689
```

Vi ser at momentestimatene basert på X_i-ene og Y_i-ene er forskjellige.

g)

```
> # g)
> repetition = 1000
>
> # Lag en ramme for å skrive inn resultatet.
> result.array = array(data=NA, dim=c(2,2,repetition))
> dimnames(result.array) = list(
+   c("X", "Y"),
+   c("mu", "sigma.sq"),
+   paste("rept.", 1:repetition, sep="")
+ )
>
> # Gjenta simulasjonen 1000 ganger.
> for (i in 1:repetition) {
+
+   # Simuler data.
+   x = rnorm(n=10, mean=1, sd=sqrt(3))
+   y = exp(x)
+
+   # Momentestimatorene basert på X.
+   result.array["X","mu",i] = mean(x)
+   result.array["X","sigma.sq",i] = mean(x^2) - mean(x)^2
+
+   # Momentestimatorene basert på Y.
+   result.array["Y","mu",i] = 2*log(mean(y)) - (1/2)*log(mean(y^2))
+   result.array["Y","sigma.sq",i] = log(mean(y^2)) - 2*log(mean(y))
+ }
>
> # Skjevhet
> bias.matrix = apply(X=result.array, MARGIN=1:2, FUN=mean)
> bias.matrix[,"mu"] = bias.matrix[,"mu"] - 1
> bias.matrix[,"sigma.sq"] = bias.matrix[,"sigma.sq"] - 3
> print(bias.matrix)
           mu   sigma.sq
X -0.0122316 -0.2390005
Y  0.5920356 -1.8587630
>
> # Standardavvik
> sd.matrix = apply(X=result.array, MARGIN=1:2, FUN=sd)
> print(sd.matrix)
         mu  sigma.sq
X 0.5602708 1.2738232
Y 0.6664552 0.4239948
```

For μ viser simuleringene at estimatoren basert på X_i-ene både har mindre skjevhet og standardfeil enn den basert på Y_i-ene.

For σ^2 har estimatoren basert på X_i-ene mindre skjevhet enn den basert på Y_i-ene. Men estimatoren basert på Y_i-er har mindre standardfeil enn den basert på X_i-ene.

h)

```
> # h)
> repetition = 1000
>
> # Lag en ramme for å skrive inn resultatet.
> result.array.2 = array(data=NA, dim=c(2,2,repetition))
> dimnames(result.array.2) = list(
+   c("X", "Y"),
+   c("mu", "sigma.sq"),
+   paste("rept.", 1:repetition, sep="")
+ )
>
```

```
> # Gjenta simulasjonen 1000 ganger.
> for (i in 1:repetition) {
+
+   # Simuler data.
+   x = rnorm(n=1000, mean=1, sd=sqrt(3))
+   y = exp(x)
+
+   # Momentestimatorene basert på X.
+   result.array.2["X","mu",i] = mean(x)
+   result.array.2["X","sigma.sq",i] = mean(x^2) - mean(x)^2
+
+   # Momentestimatorene basert på Y.
+   result.array.2["Y","mu",i] = 2*log(mean(y)) - (1/2)*log(mean(y^2))
+   result.array.2["Y","sigma.sq",i] = log(mean(y^2)) - 2*log(mean(y))
+ }
>
> # Skjevhet
> bias.matrix = apply(X=result.array.2, MARGIN=1:2, FUN=mean)
> bias.matrix[,"mu"] = bias.matrix[,"mu"] - 1
> bias.matrix[,"sigma.sq"] = bias.matrix[,"sigma.sq"] - 3
> print(bias.matrix)
              mu      sigma.sq
X 0.0005411164 -0.002528499
Y 0.2221945841 -0.463651350
>
> # Standardavvik
> sd.matrix = apply(X=result.array.2, MARGIN=1:2, FUN=sd)
> print(sd.matrix)
           mu   sigma.sq
X 0.05388164 0.1356096
Y 0.19488271 0.5258223
```

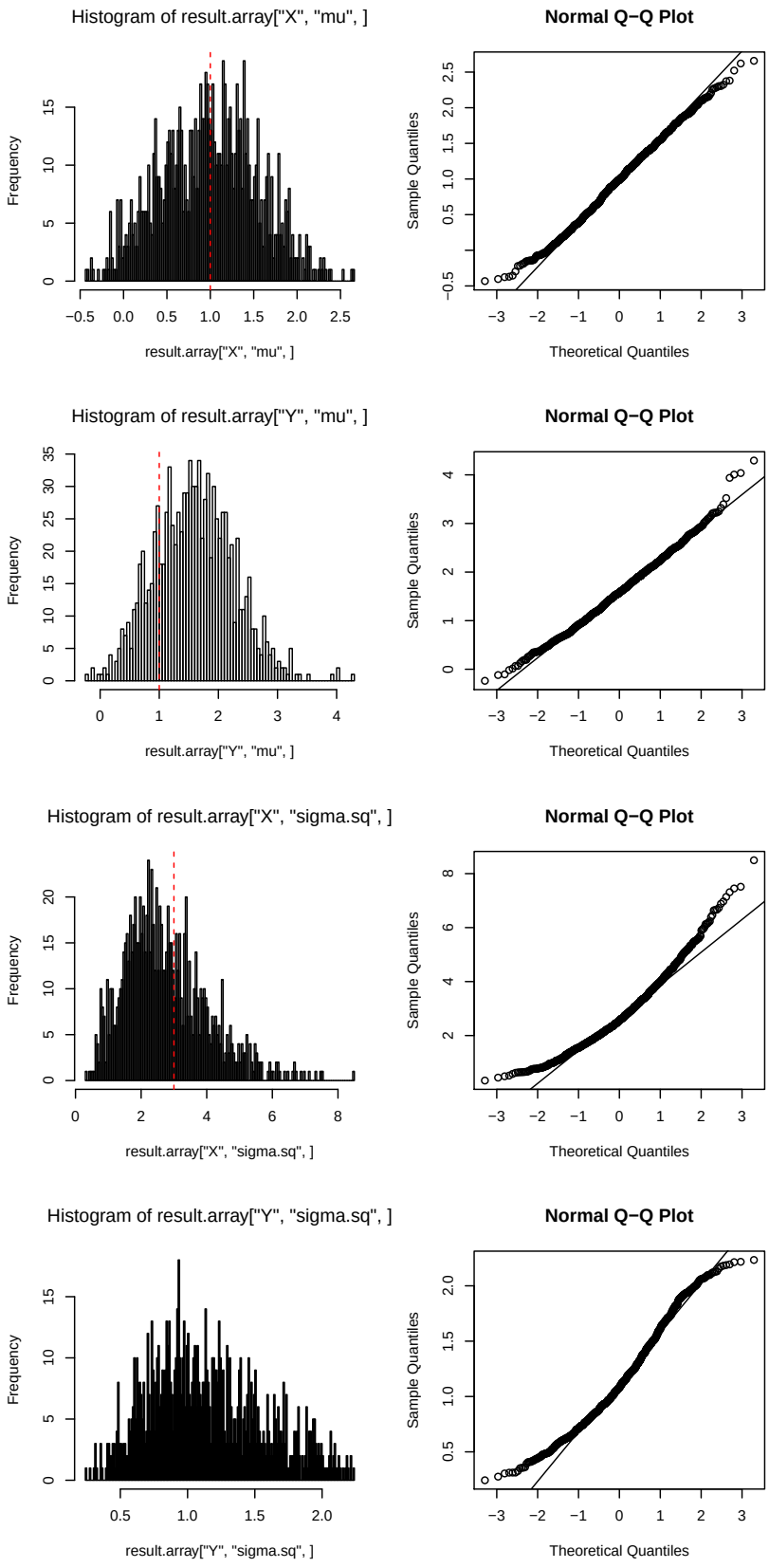
Vi har økt n fra 10 til 1000. For både μ og σ^2 har nå estimatorene basert på X_i-ene mindre skjevhet og standardfeil enn estimatorene basert på Y_i-ene.

i) Vi plotter histogram og qq-plott med følgende R-kode:

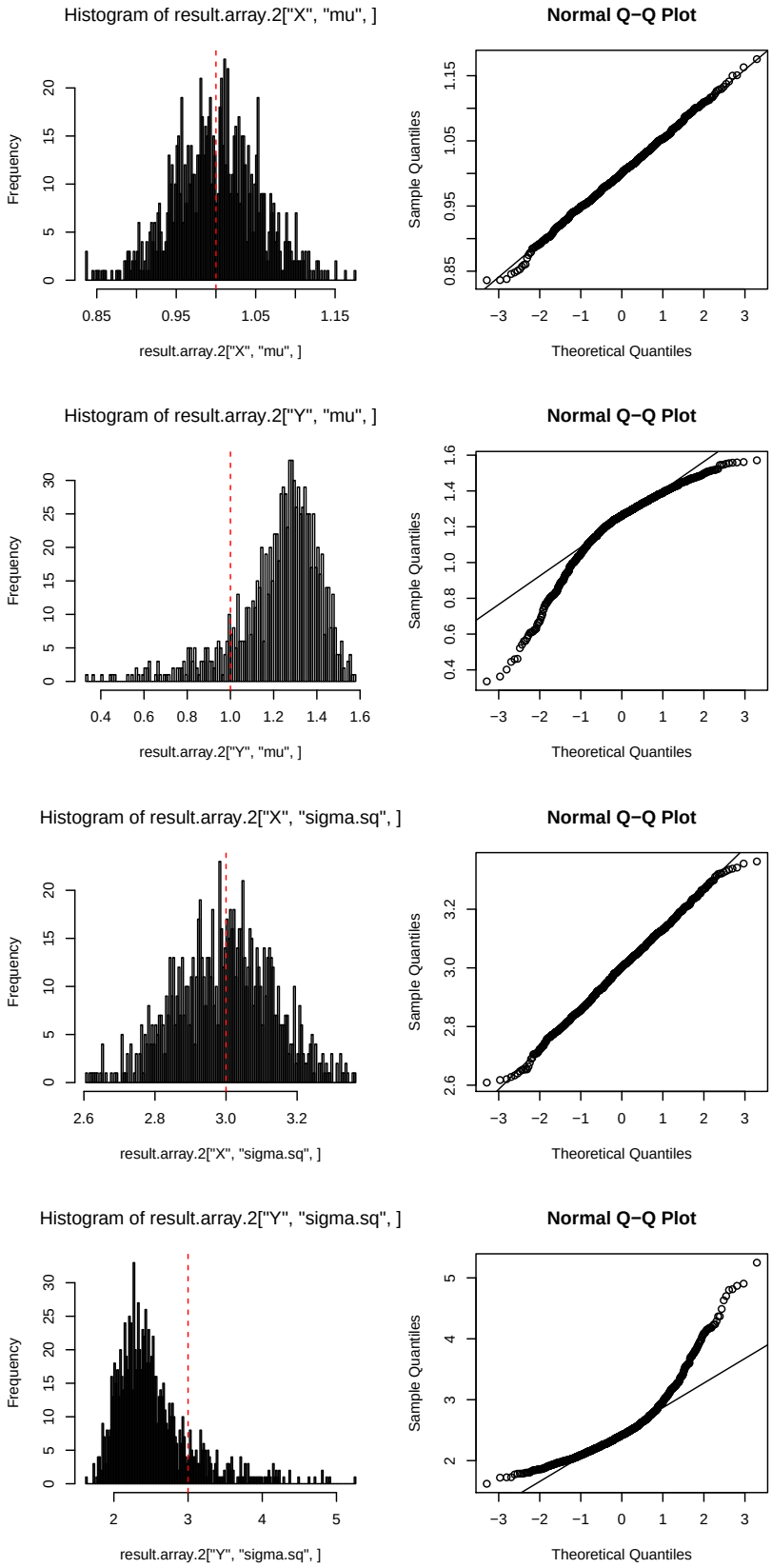
```
# i)
# Plot
pdf(file = paste("./plot/n10.pdf",sep=""), width = 6, height = 12)
par(mfrow=c(4,2))

# Histogram
hist(result.array["X","mu",], breaks=150, font.main=1)
# Display the correct value.
abline(v=1, col="red", lty=2, lwd=1)
# QQ plot
qqnorm(result.array["X","mu",])
qqline(result.array["X","mu",])

# Histogram
hist(result.array["Y","mu",], breaks=150, font.main=1)
# Display the correct value.
abline(v=1, col="red", lty=2, lwd=1)
# QQ plot
qqnorm(result.array["Y","mu",])
qqline(result.array["Y","mu",])

# Histogram
hist(result.array["X","sigma.sq",], breaks=150, font.main=1)
# Display the correct value.
abline(v=3, col="red", lty=2, lwd=1)
# QQ plot
qqnorm(result.array["X","sigma.sq",])
qqline(result.array["X","sigma.sq",])

# Histogram
hist(result.array["Y","sigma.sq",], breaks=150, font.main=1)
# Display the correct value.
```

```
abline(v=3, col="red", lty=2, lwd=1)
# QQ plot
qqnorm(result.array["Y","sigma.sq",])
qqline(result.array["Y","sigma.sq",])

dev.off()


# Plot
pdf(file = paste("./plot/n1000.pdf",sep=""), width = 6, height = 12)
par(mfrow=c(4,2))

# Histogram
hist(result.array.2["X","mu",], breaks=150, font.main=1)
# Display the correct value.
abline(v=1, col="red", lty=2, lwd=1)
# QQ plot
qqnorm(result.array.2["X","mu",])
qqline(result.array.2["X","mu",])

# Histogram
hist(result.array.2["Y","mu",], breaks=150, font.main=1)
# Display the correct value.
abline(v=1, col="red", lty=2, lwd=1)
# QQ plot
qqnorm(result.array.2["Y","mu",])
qqline(result.array.2["Y","mu",])

# Histogram
hist(result.array.2["X","sigma.sq",], breaks=150, font.main=1)
# Display the correct value.
abline(v=3, col="red", lty=2, lwd=1)
# QQ plot
qqnorm(result.array.2["X","sigma.sq",])
qqline(result.array.2["X","sigma.sq",])

# Histogram
hist(result.array.2["Y","sigma.sq",], breaks=150, font.main=1)
# Display the correct value.
abline(v=3, col="red", lty=2, lwd=1)
# QQ plot
qqnorm(result.array.2["Y","sigma.sq",])
qqline(result.array.2["Y","sigma.sq",])

dev.off()
```

Vi kan se fra figurene på de neste sidene at estimatorene basert på X_i-ene har mindre skjevhet enn estimatorene basert på Y_i-ene. Når vi får flere observasjoner blr normaltilnærmingen bedre for estimatorene basert på X_i-ene.



Figur 1: Resultat fra ekstraoppgave 7 (g) ($n = 10$)



Figur 2: Resultat fra ekstraoppgave 7 (h) ($n = 1000$)