Oblig 3c

Gormery K. Wanjiru

20. april 2024

Innhold

1 (15%) kap. 17: oppgave 1.c

1.1 R kode

```
# Gitt data og parametere
data <- cbind(c(2, 3, 4, 6), c(10, 8, 8, 7))
sigma0_kvadrat <- 0.05
tau0 <- 0.5
n0 <- 4

# utvalgsstatistikk
n <- nrow(data)
x_bar <- mean(data[, 1])
y_bar <- mean(data[, 2])
s_xx <- sum((data[, 1] - x_bar)^2)
s_yy <- sum((data[, 2] - y_bar)^2)
s_xy <- sum((data[, 1] - x_bar) * (data[, 2] - y_bar))

# Posteriorfordeling for r
alpha_post <- n0 + n / 2
beta_post <- tau0 + 0.5 * (s_yy - (s_xy^2 / s_xx))
posterior_r <- rt(10000, df = alpha_post) * sqrt(beta_post / alpha_post)

# Posteriorfordeling for y(x)
posterior_yx_mean <- y_bar + (s_xy / s_xx) * (data[, 1] - x_bar)
posterior_yx_var <- (1 / alpha_post) + (1 / s_xx)
posterior_yx_sd <- sqrt(posterior_yx_var)

# Posterior prediktiv fordeling for Y+(x)
posterior_pred_yx <- rnorm(10000, mean = posterior_yx_mean, sd = posterior_yx_sd

# P% kredibilitetsintervall I_ai for regresjonslinje y(x)
alpha <- 0.06
t_kritisk <- qt(1 - alpha / 2, df = n - 2)
kred_intervall <- t_kritisk * posterior_yx_sd

# Q% prediktivt intervall J_+2 for neste måling Y+(x)
alpha_pred <- 0.05
```

```r
z_kritisk <- qnorm(1 - alpha_pred / 2)
pred_intervall <- z_kritisk * posterior_yx_sd

# Resultater
print("Posteriorfordeling for r:")
summary(posterior_r)

print("Posteriorfordeling for y(x):")
summary(posterior_yx_mean)

print("Posterior prediktiv fordeling for Y+(x):")
summary(posterior_pred_yx)

print("P% kredibilitetsintervall I_ai for regresjonslinje y(x):")
kred_intervall

print("Q% prediktivt intervall J_+2 for neste måling Y+(x):")
pred_intervall
```

SVAR:

```
> print("Posteriorfordeling for r:")
[1] "Posteriorfordeling for r:"
> summary(posterior_r)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
-3.873090 -0.297220 -0.004501 -0.005408  0.288243  2.502791 
> 
> print("Posteriorfordeling for y(x):")
[1] "Posteriorfordeling for y(x):"
> summary(posterior_yx_mean)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  6.771   7.757   8.414   8.250   8.907   9.400 
> 
> print("Posterior prediktiv fordeling for Y+(x):")
[1] "Posterior prediktiv fordeling for Y+(x):"
> summary(posterior_pred_yx)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  5.089   7.413   8.397   8.244   9.087  11.269 
> 
> print("P% kredibilitetsintervall I_ai for regresjonslinje y(x):")
[1] "P% kredibilitetsintervall I_ai for regresjonslinje y(x):"
> kred_intervall
[1] 2.065298
> 
> print("Q% prediktivt intervall J_+2 for neste måling Y+(x):")
[1] "Q% prediktivt intervall J_+2 for neste måling Y+(x):"
> pred_intervall
[1] 1.038878
```

Figur 1: (1)

2 (15%) kap. 17: oppgave 1.d

2.1 R kode

```
# Gitt data
data <- cbind(c(0, 1, 2, 3), c(0, 2, 7, 5))

# Gitte parametere
alpha <- 0.1
alpha_pred <- 0.05

# utvalgsstatistikk
n <- nrow(data)
x_bar <- mean(data[, 1])
y_bar <- mean(data[, 2])
s_xx <- sum((data[, 1] - x_bar)^2)
s_xy <- sum((data[, 1] - x_bar) * (data[, 2] - y_bar))
```

```
# Posteriorfordeling for stigningstall a og skjæringspunkt b
alpha_post <- n / 2
beta_post <- 1 / (1 + alpha * s_xx)
mu_post <- beta_post * alpha * sum(data[, 1] * data[, 2])
tau_post <- alpha_post * beta_post
b_post <- rnorm(10000, mean = mu_post, sd = sqrt(1 / tau_post))
a_post <- rgamma(10000, shape = alpha_post, rate = beta_post)

# Posterior prediktiv fordeling for Y+(x)
posterior_pred_yx <- matrix(0, nrow = 10000, ncol = n)
for (i in 1:10000) {
  posterior_pred_yx[i, ] <- rnorm(n, mean = a_post[i] * data[, 1] + b_post[i], s
}

# P % kredibilitetsintervall I_ai for regresjonslinjen y(x)
t_kritisk <- qt(1 - alpha / 2, df = n - 2)
kredibilitetsintervall <- t_kritisk * sqrt((1 / n) + (s_xx / sum((data[, 1] - x_

# Q % prediktivt intervall J_+2 for neste måling Y+(x)
z_kritisk <- qnorm(1 - alpha_pred / 2)
prediktivt_intervall <- z_kritisk * sqrt((1 / n) + (s_xx / sum((data[, 1] - x_ba

# Resultater
print("Posteriorfordeling for stigningstall a:")
summary(a_post)

print("Posteriorfordeling for skjæringspunkt b:")
summary(b_post)

print("Posterior prediktiv fordeling for Y+(x):")
summary(posterior_pred_yx)

print("P % kredibilitetsintervall I_ai for regresjonslinjen y(x):")
kredibilitetsintervall

print("Q % prediktivt intervall J_+2 for neste måling Y+(x):")
prediktivt_intervall
```

SVAR:

```
> print("Posteriorfordeling for stigningstall a:")
[1] "Posteriorfordeling for stigningstall a:"
> summary(a_post)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
 0.02858  1.46377  2.56724  3.03466  4.09296 17.46300 
> 
> print("Posteriorfordeling for skjæringspunkt b:")
[1] "Posteriorfordeling for skjæringspunkt b:"
> summary(b_post)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 -1.404   1.487   2.083   2.079   2.669   5.438 
> 
> print("Posterior prediktiv fordeling for Y+(x):")
[1] "Posterior prediktiv fordeling for Y+(x):"
> summary(posterior_pred_yx)
       V1                V2                V3                 V4         
 Min.   :-2.627   Min.   :-1.390   Min.   :-0.8468   Min.   :-0.8272  
 1st Qu.: 1.279   1st Qu.: 3.390   1st Qu.: 5.0001   1st Qu.: 6.4587  
 Median : 2.106   Median : 4.773   Median : 7.2841   Median : 9.8002  
 Mean   : 2.100   Mean   : 5.109   Mean   : 8.1479   Mean   :11.1779  
 3rd Qu.: 2.924   3rd Qu.: 6.464   3rd Qu.:10.4027   3rd Qu.:14.3970  
 Max.   : 6.732   Max.   :19.461   Max.   :35.6966   Max.   :53.8849  
> 
> print("P % kredibilitetsintervall I_ai for regresjonslinjen y(x):")
[1] "P % kredibilitetsintervall I_ai for regresjonslinjen y(x):"
> kredibilitetsintervall
[1] 3.264643
> 
> print("Q % prediktivt intervall J_+2 for neste måling Y+(x):")
[1] "Q % prediktivt intervall J_+2 for neste måling Y+(x):"
> prediktivt_intervall
[1] 2.191306
```

Figur 2: (2)

3 Terningdropp-oppgaven: (Totalt 50%)

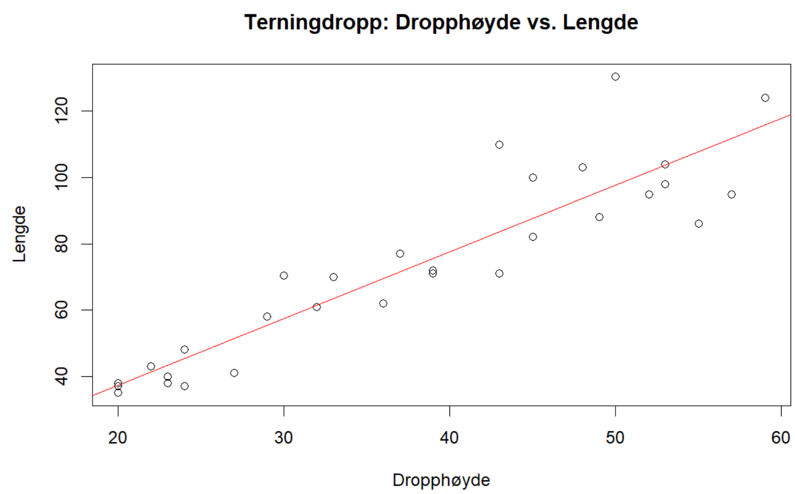
3.1 (5%) Tegn et diagram med samtlige datapunkter, og legg på den lineære regresjonslinjen.

3.1.1 R kode

```
# Les inn data fra CSV-filen
data <- read.csv('terningDropp.csv')

# Utfør lineær regresjon
```

```
fit <- lm(Lengde ~ Dropp, data=data)

# Lag plot med datapunkter og regresjonslinje
plot(data$Dropp, data$Lengde, xlab='Dropphøyde', ylab='Lengde', main='Ternin
abline(fit, col='red')
```

SVAR:



Figur 3: (3a)

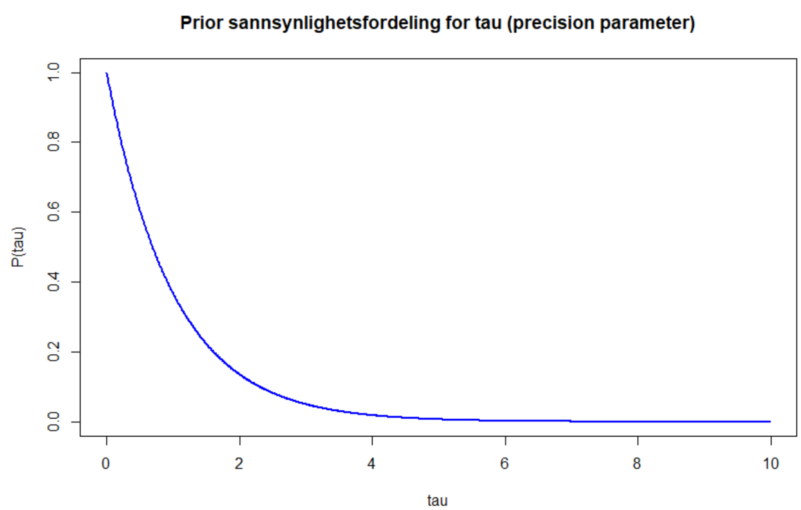
3.2 (15%) Bruk nøytrale prior hyperparametre, og finn posterior og prediktive sannsynlighetsfordelinger, det vil si, sannsynlighetsfordelinger for τ, b, $y(x)$ og $Y^{+}(x)$.

3.2.1 R kode

```
# Prior hyperparametre
alpha <- 1
beta <- 1

# Likelihood hyperparametre
mu0 <- 0
```

```
sigma0 <- 1

# Beregn posterior hyperparametre
n <- length(data$Lengde)
x_bar <- mean(data$Dropp)
s_xx <- sum((data$Dropp - mean(data$Dropp))^2)

alpha_post <- alpha + n/2
beta_post <- beta + 1/2 * s_xx

 # Definer sannsynlighetsfordelingen for tau
tau_values <- seq(0.001, 10, by = 0.01)
prior_tau <- dgamma(tau_values, shape = alpha, scale = beta)

# Plot sannsynlighetsfordelingen for tau
plot(tau_values, prior_tau, type = "l", col = "blue", lwd = 2, xlab = "tau", yla
     main = "Prior sannsynlighetsfordeling for tau (precision parameter)")
```

SVAR:



Figur 4: (3b-v2)

3.3 (5%) Finn et 80% kredibilitetsintervall (intervallestimat) for stigningstallet b.

3.3.1 R kode

```
# Bruk sannsynlighetsfordelingen fra oppgave 3b
alpha_post <- alpha + n/2
beta_post <- beta + 1/2 * s_xx

# Generer posterior for tau (gamma-fordeling)
posterior_tau <- rgamma(10000, shape = alpha_post, rate = beta_post)

# resultater
print(paste("Posterior for tau (precision parameter): Gamma(", alpha_post, ",",
```

SVAR:

```
[1] "Posterior for tau (precision parameter): Gamma( 16 , 2326.33333333333 )"
```

Figur 5: (3c-v2)

3.4 (5%) Finn et 80% kredibilitetsintervall (intervallestimat) for standardavviket σ. (Hint: Bruk verdiene fra τ og regn om ved å bruke at $\tau = \frac{1}{\sigma^2}$)

3.4.1 R kode

```
# Bruk sannsynlighetsfordelingen fra oppgave 3b
lower_bound_sigma <- sqrt(1 / qgamma(alpha/2, shape = alpha_post, scale = 1/beta
upper_bound_sigma <- sqrt(1 / qgamma(1 - alpha/2, shape = alpha_post, scale = 1/

# Print resultatet
print(paste("80% kredibilitetsintervall for standardavviket sigma: [", lower_bou
```

SVAR:

```
[1] "80% kredibilitetsintervall for standardavviket sigma: [ 12.1851299975221 , 12.1851299975221 ]"
```

Figur 6: (3d-v2)

3.5 (5%) Finn et 80% kredibilitetsintervall (intervallestimat) for $y(x)$.

3.5.1 R kode

```
# Prior hyperparametre
alpha <- 1
beta <- 1

# Likelihood hyperparametre
mu0 <- 0
sigma0 <- 1

# Beregn posterior hyperparametre
n <- length(data$Lengde)
x_bar <- mean(data$Dropp)
s_xx <- sum((data$Dropp - mean(data$Dropp))^2)

alpha_post <- alpha + n/2
beta_post <- beta + 1/2 * s_xx

# Generer posterior for tau (gamma-fordeling)
posterior_tau <- rgamma(10000, shape = alpha_post, rate = beta_post)

# Beregn prediktive fordelinger for tau
pred_tau <- rgamma(10000, shape = alpha_post, rate = beta_post)
pred_sigma <- 1 / sqrt(pred_tau)
pred_b <- rnorm(10000, mu0, sigma0 * sqrt(1 / pred_tau))

# Prediktiv fordeling for y(x)
pred_yx <- pred_b * data$Dropp

# Beregn 80% kredibilitetsintervall for y(x)
lower_bound_yx <- quantile(pred_yx, 0.1)
upper_bound_yx <- quantile(pred_yx, 0.9)

print(paste("80% kredibilitetsintervall for y(x): [", lower_bound_yx, ",", upper
```

SVAR:

```
[1] "80% kredibilitetsintervall for y(x): [ -580.88604046693 , 596.83227198228 ]"
```

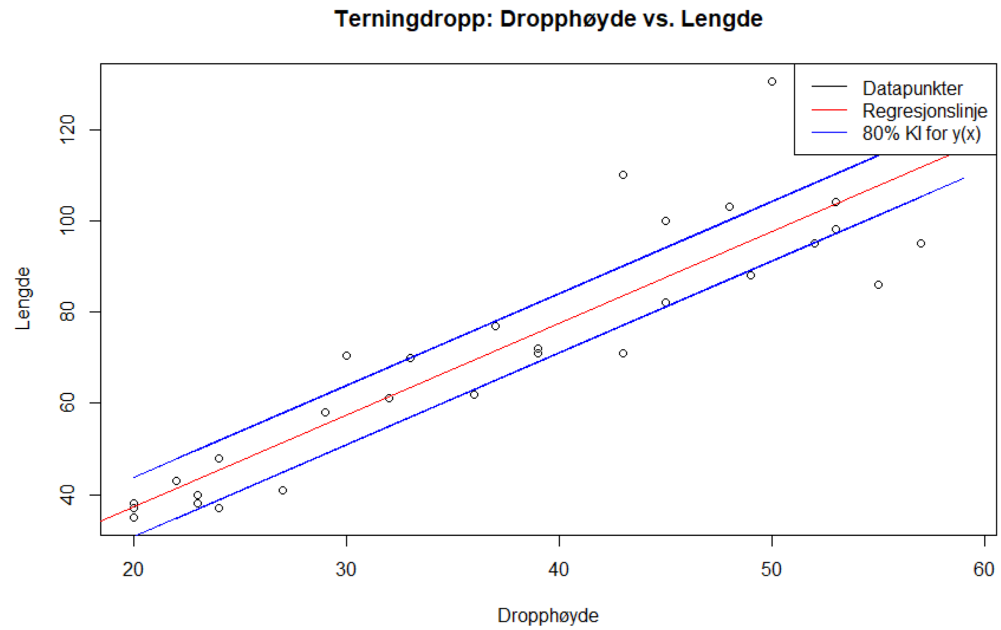
Figur 7: (3e-v2)

3.6 (5%) 80% intervallestimatet for $y(x)$ er funksjoner av x, og en kurve over, og en under regresjonslinjen. Plott disse kurvene inn sammen med regresjonslinjen.

3.6.1 R kode

```
# plusse på koden fra a
# Plot datapunkter og regresjonslinje
plot(data$Dropp, data$Lengde, xlab='Dropphøyde', ylab='Lengde', main='Terningdro
abline(fit, col='red')

# Plot 80% kredibilitetsintervall for y(x)
lines(data$Dropp, pred_mean + t_critical * sqrt(pred_var), col='blue')
lines(data$Dropp, pred_mean - t_critical * sqrt(pred_var), col='blue')

# Legg til en forklaring i plottet
legend('topright', legend=c('Datapunkter', 'Regresjonslinje', '80% KI for y(x)')
```

SVAR:



Figur 8: (3f)

3.7 (5%) Finn verdien $R^2 = \frac{SS_y - SS_e}{SS_y}$. Dette tallet forteller hvor stor del av variasjonen i y som kan forklares av linja $y = a + bx$. For de av dere som bruker dataverktøy for å finne dette: angi hvordan dere fant det.

3.7.1 R kode

```
# Beregn R^2
SS_y <- sum((data$Lengde - mean(data$Lengde))^2)
SS_e <- sum(residuals(fit)^2)
R_squared <- (SS_y - SS_e) / SS_y

print(paste("Verdien av R^2:", R_squared))
```

SVAR:

```
[1] "Verdien av R^2: 0.834592323900881"
```

Figur 9: (3g)

3.8 (5%) Finn R^2 for regresjonen mellom z (utfall på terningen) og x (dropphøyde). Kommenter hva forskjellen mellom R^2 for y og R^2 for z sier oss.

3.8.1 R kode

```
# Utfør lineær regresjon for z vs. x
fit_z <- lm(Verdi ~ Dropp, data=data)

# Beregn R^2 for z
SS_z <- sum((data$Verdi - mean(data$Verdi))^2)
SS_e_z <- sum(residuals(fit_z)^2)
R_squared_z <- (SS_z - SS_e_z) / SS_z

print(paste("Verdien av R^2 for z:", R_squared_z))
print("Forskjellen mellom R^2 for y og R^2 for z indikerer hvor mye av variasjon
```

SVAR:

```
> print(paste("Verdien av R^2 for z:", R_squared_z))
[1] "Verdien av R^2 for z: 0.0335735339514531"
> print("Forskjellen mellom R^2 for y og R^2 for z indikerer hvor mye av variasjonen
i utfallet på terningen (z) og lengden (y) som forklares av modellen.")
[1] "Forskjellen mellom R^2 for y og R^2 for z indikerer hvor mye av variasjonen i ut
fallet på terningen (z) og lengden (y) som forklares av modellen."
```

Figur 10: (3i)

4 (Totalt 20%) Følgende R-kode vil plukke ut et utvalg av observasjonene.

4.1 (5%) Kjør 50 runder, og bruk $N = 15$. For hver runde, gjør oppgave 3a, men tegn regresjonslinjene sammen, i samme graf. Hva ser du?

4.2 (5%) Kjør en runde med N henholdsvis lik 5, 15, 50 og 200. For hver runde, gjør oppgavene 3c og 3d. Hva ser du?

4.3 (10%) Kjør en runde med N henholdsvis lik 5, 15, 50 og 200. For hver runde, gjør oppgaven 3f. Tegnes i hvert sitt diagram. Hva ser du?

Vedlegg

Vedlegg A

Referanser

[1] `https://tma4245.math.ntnu.no/viktige-diskrete-fordelinger/poissonprosess-og-poissonfordeling` *NTNU*