BDA - Assignment 3

Anonymous

Contents


```r
# To install aaltobda, see the General information in the assignment.
remotes::install_github("avehtari/BDA_course_Aalto", subdir = "rpackage", upgrade = "never")
```

```
## Skipping install of 'aaltobda' from a github remote, the SHA1 (38f34d35) has not changed since last i
##   Use `force = TRUE` to force installation
```

1. Inference for normal mean and deviation

```r
# Installing libraries and setting up dataset
library(aaltobda)
data("windshieldy1")
head(windshieldy1)  # Testing hardness
```

```
## [1] 13.357 14.928 14.896 15.297 14.820 12.067
```

```r
windshieldy_test <- c(13.357, 14.928, 14.896, 14.820)
```

A)

```r
data <- windshieldy1

mu_point_est <- function(data) {
  y <- data
  n <- length(y)
```

```r
  sims <- 10000

  ybar  <- (1/n)*sum(y)
  s_sq  <- (1/(n-1))*sum((y - ybar)^2)

  mu_s <- rtnew(sims, df = n - 1, mean = ybar, scale = s_sq/n)
  mu <- mean(mu_s)
  mu <- round(mu, digits = 1)
  return(mu)
}

mu_interval <- function(data, prob) {
  y <- data
  n <- length(y)
  sims <- 10000

  ybar  <- (1/n)*sum(y)
  s_sq  <- (1/(n-1))*sum((y - ybar)^2)

  lower <- (1-prob)/2
  upper <- prob + (1-prob)/2

  interval <- qtnew(c(lower, upper), df = n - 1, mean = ybar, scale = s_sq/n)
  round(interval, digits = 1)
  return(interval)
}
if(TRUE) {
  cat("\n")
  cat("Below is the point estimate E(mu|y)")
  cat("\n")
  print(mu_point_est(data = data))
}
```

```
##
## Below is the point estimate E(mu|y)
## [1] 14.6
```

```r
if(TRUE) {
  cat("\n")
  cat("Below is the interval at 95%")
  cat("\n")
  print(mu_interval(data = data, prob = .95))
}
```

```
##
## Below is the interval at 95%
## [1] 14.05441 15.16803
```

B)

```r
data <- windshieldy1

mu_pred_point_est <- function(data) {
```

```
  y <- data
  n <- length(y)
  sims <- 10000

  ybar  <- (1/n)*sum(y)
  s_sq  <- (1/(n-1))*sum((y - ybar)^2)

  density <- integrate(function(theta) theta*dtnew(theta,
                                      df = n - 1,
                                      mean = ybar,
                                      scale = s_sq/n),
            lower = -Inf,
            upper = Inf)[1]
  return(density)
}
mu_pred_interval <- function(data, prob) {

  y <- data
  n <- length(y)

  ybar  <- (1/n)*sum(y)
  s_sq  <- (1/(n-1))*sum((y - ybar)^2)

  scale <- sqrt((1+(1/n)))*sqrt(s_sq)

  lower <- (1-prob)/2
  upper <- prob + (1-prob)/2

  interval <- qtnew(c(lower, upper), df = n - 1, mean = ybar, scale = scale)
  round(interval, digits = 1)
  return(interval)
}

if(TRUE) {
  cat("\n")
  cat("Below is the expected hardness of the next windshield,")
  cat("\n")
  cat("E_p(theta|y)[theta] optained by integrating")
  cat("theta*p(theta|y) over theta.")
  cat("\n")
  print(mu_pred_point_est(data = data))
  cat("\n")
}
```

```
##
## Below is the expected hardness of the next windshield,
## E_p(theta|y)[theta] optained by integratingtheta*p(theta|y) over theta.
## $value
## [1] 14.61122
```

```
if(TRUE) {
  cat("\n")
  cat("Below is the predictive interval:")
  print(mu_pred_interval(data = data, prob = 0.95))
```

```r
  cat("\n")
}
```

```
##
## Below is the predictive interval:[1] 11.02792 18.19453
```

```r
cat("I also plot the distribution")
```

```
## I also plot the distribution
```

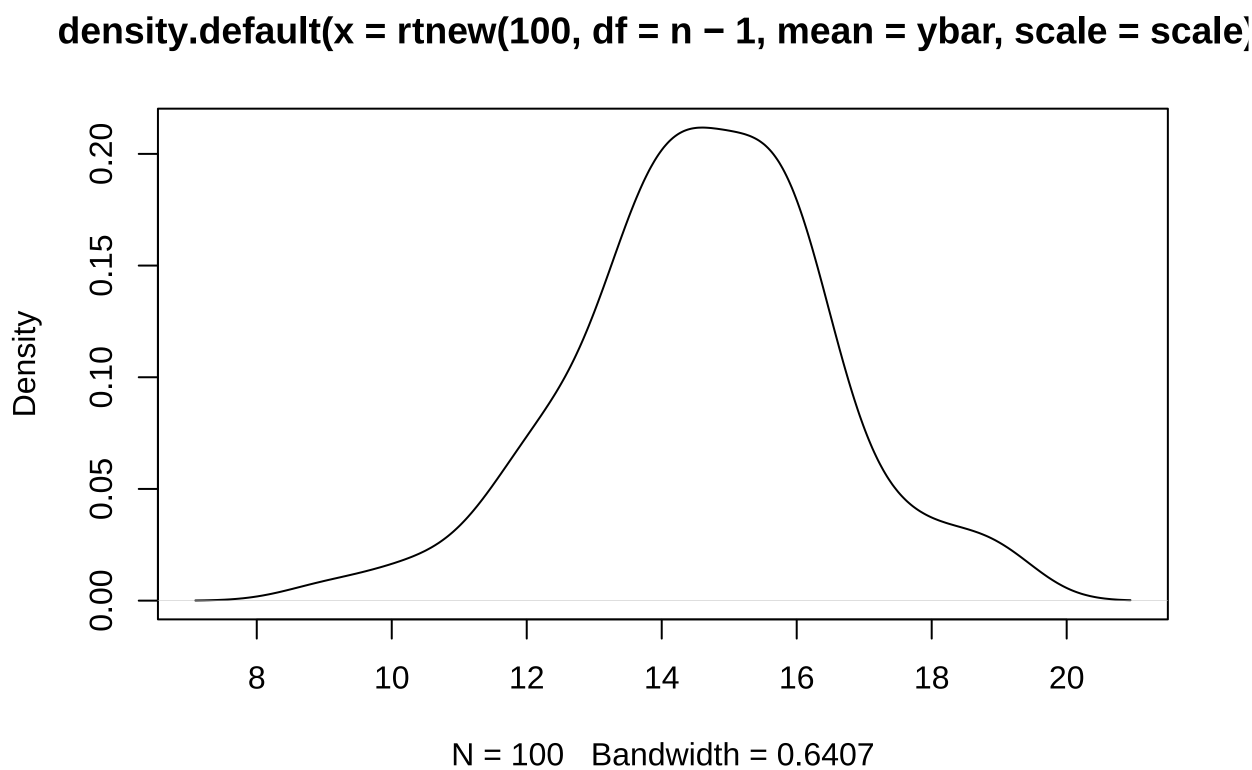
```r
y <- data
n <- length(y)

ybar  <- (1/n)*sum(y)
s_sq  <- (1/(n-1))*sum((y - ybar)^2)

scale <- sqrt((1+(1/n)))*sqrt(s_sq)

prob <- 0.95
lower <- (1-prob)/2
upper <- prob + (1-prob)/2

plot(density(rtnew(100, df = n - 1, mean = ybar, scale = scale)))
```



2. Inference for difference between proportions

A.

Assuming a conjugate prior of $Beta(\alpha, \beta)$, we find that the posterior distribution for each group is $P(\theta|y_i)Beta(\theta|\alpha + y_i, \beta + n_i - y_i)$. Notice that neither α or β are indexed since I assume the same prior for each group posterior distribution. The priors come from a logistic regression of the form $y = \alpha + x\beta$, where $y = (y_c, y_t)$ and $x = 1$ if treated.

```
library(aod)
# Control
n0 <- 674
y0 <- rep(0, n0)
y0[1:39] <- 1
y0tot <- sum(y0)

# Treatment
n1 <- 680
y1 <- rep(0, n1)
y1[1:22] <- 1
y1tot <- sum(y1)

# Creating variables
t <- rep(0, length(y0) + length(y1))
t[(length(y0) + 1):length(t)] <- 1
y <- c(y0, y1)

# odds ration
out <- glm(y ~ t, family = "binomial")
alpha <- exp(coef(out))[1] # Use as priors (0.06141732, 0.54438469)
beta  <- exp(coef(out))[2] # Use as priors (0.06141732, 0.54438469)

p0 <- rbeta(n0, alpha + y0tot, beta + n0 - y0tot)
p0hat <- mean(p0)

p1 <- rbeta(n1, alpha + y1tot, beta + n1 - y1tot)
p1hat <- mean(p1)

#set.seed(4711)
#sims <- 100000
#p0 <- rbeta(sims, 5, 95)
#p1 <- rbeta(sims, 10, 90)

#p0hat <- mean(p0)
#p1hat <- mean(p1)

oddsRatio.formula <- function(p0, p1) {

  numerator    <- p1/(1-p1)
  denominator  <- p0/(1-p0)

  oddsRatio <- numerator/denominator

  return(oddsRatio)
```

```r
}

posterior_odds_ratio_interval <- function(p0, p1, prob) {

  numerator    <- p1/(1-p1)
  denominator  <- p0/(1-p0)

  oddsRatios <- numerator/denominator

  lower <- (1-prob)/2
  upper <- prob + (1-prob)/2

  lower <- quantile(oddsRatios, lower)
  upper <- quantile(oddsRatios, upper)

  Interval <- list(lower, upper)
  return(Interval)

}

estimate <- oddsRatio.formula(p0hat, p1hat)
interval <- posterior_odds_ratio_interval(p0, p1, prob = 0.95)
```

```
## Warning in numerator/denominator: longer object length is not a multiple of
## shorter object length
```

```r
print(estimate)
```

```
## [1] 0.5436883
```

```r
print(interval)
```

```
## [[1]]
##      2.5%
## 0.2943856
##
## [[2]]
##     97.5%
## 0.9117585
```

B.

Assuming a prior distribution of $Beta(\alpha = 0.06, \beta = 0.54)$ obtained from a logistic regression, there are lower odds of mortality if using a beta-blocker. In fact, the probability that the beta-blockers reduced the odds of dying when compared to no using them is 95%.

3. Inference for difference between normal means

In exercise 1, we learned that the posterior distribution is t-student if we assumed the conjugate prior $p(\theta) = \sigma^{-2}$.

```r
data("windshieldy1")
data("windshieldy2")
y1 <- windshieldy1
```

```
y2 <- windshieldy2
```

A.

```
sims <- 1000
n1 <- length(y1)
n2 <- length(y2)

ybar1  <- (1/n1)*sum(y1)
ybar2  <- (1/n2)*sum(y2)

s_sq1  <- (1/(n1-1))*sum((y1 - ybar1)^2)
s_sq2  <- (1/(n2-1))*sum((y2 - ybar2)^2)

mu1 <- rtnew(sims, df = n1 - 1, mean = ybar1, scale = s_sq1/n1)
mu2 <- rtnew(sims, df = n2 - 1, mean = ybar2, scale = s_sq2/n2)

dmu <- mu1 - mu2
```

The estimate:

```
estimate <- mean(dmu)
print(estimate)
```

```
## [1] -1.201385
```

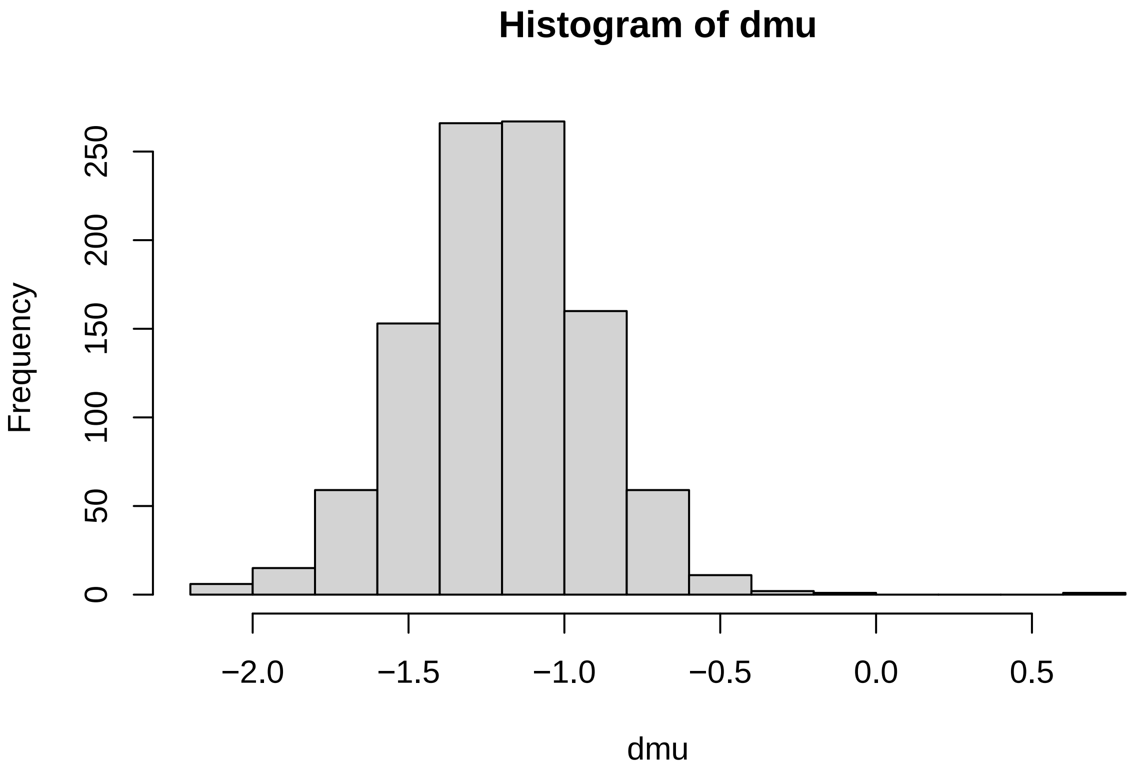
The interval:

```
prob <- 0.95
lower <- (1-prob)/2
upper <- prob + (1-prob)/2
lower <- quantile(dmu, lower)
upper <- quantile(dmu, upper)
interval <-list(lower, upper)
print(interval)
```

```
## [[1]]
##      2.5%
## -1.763285
##
## [[2]]
##      97.5%
## -0.6629618
```

The histogram:

```
hist <- hist(dmu)
```



The estimated probability that the second windshield factory producers harder windshields is at least 95% as the previous histogram shows.

B

The probability of a single point is always zero. That is $p(\mu_2 - \mu_1 = 0) = 0$.