BDA - Assignment 3

Anonymous

Contents

Inference for normal mean and deviation 1

A) 1

B) 2

```
# 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
##   Use `force = TRUE` to force installation
```

Inference for normal mean and deviation

```
# 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
```

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

A)

```
data <- windshieldy1

mu_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)

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

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

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

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

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

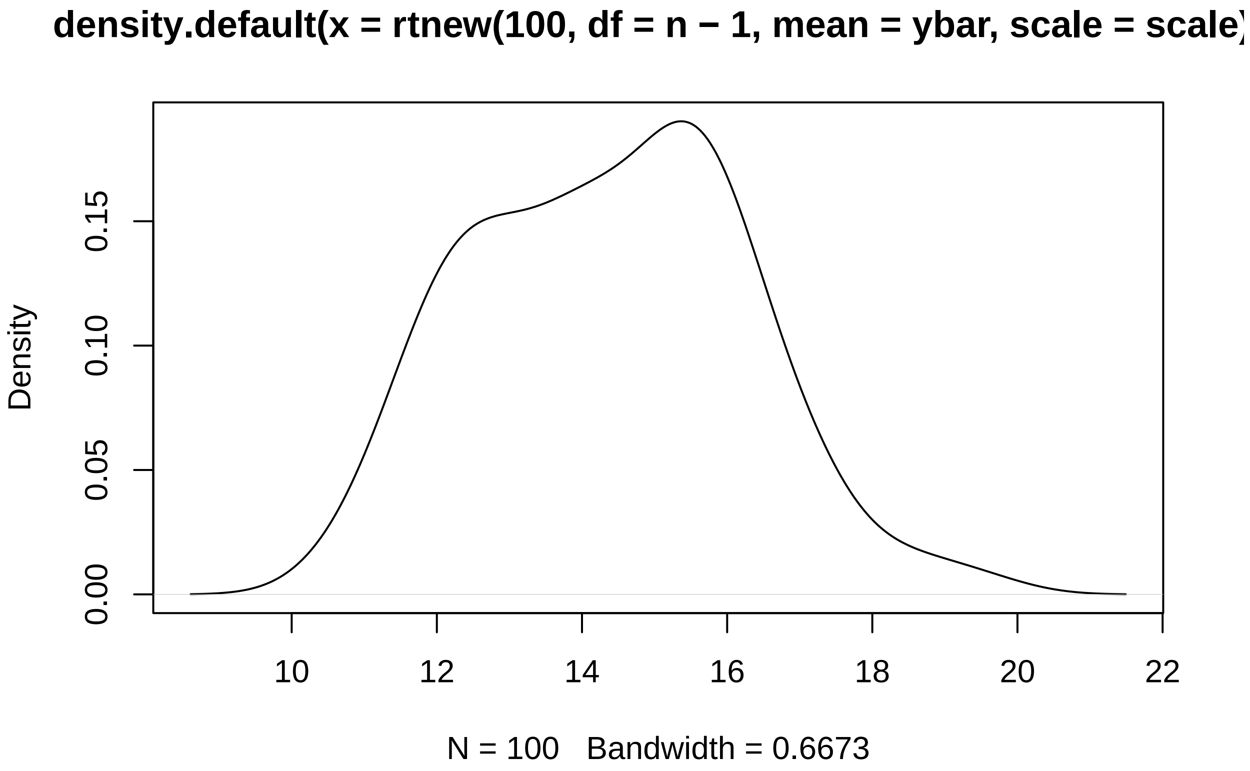
```
y <- data
n <- length(y)
```

```
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)))
```


density.default(x = rtnew(100, df = n − 1, mean = ybar, scale = scale)
Density
N = 100   Bandwidth = 0.6673