

HW04_Code43e

```
#read in the data
arrivals <- readr::read_csv("https://www4.stat.ncsu.edu/online/datasets/gamma-arrivals.csv",
                           col_names = "arrival_time")
```

Rows: 3935 Columns: 1

-- Column specification -----

Delimiter: ","

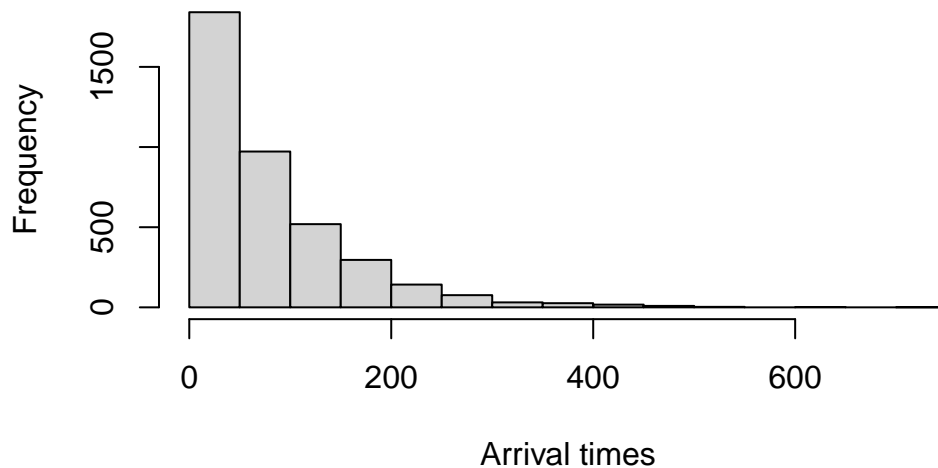
dbl (1): arrival_time

i Use `spec()` to retrieve the full column specification for this data.

i Specify the column types or set `show_col_types = FALSE` to quiet this message.

```
#a
hist(arrivals$arrival_time,
     xlab = "Arrival times",
     main = "Histogram of Arrival Times")
```

Histogram of Arrival Times



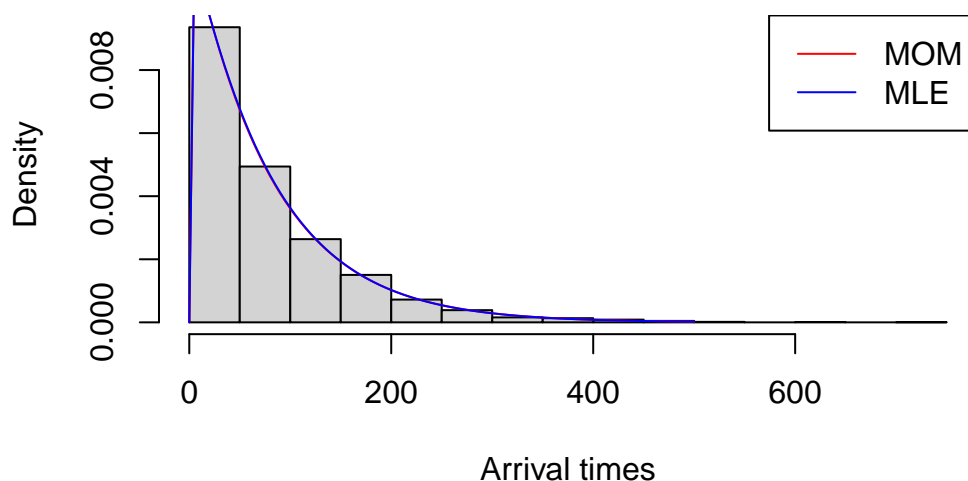
```
#b fit the parameters
mean_arrival <- mean(arrivals$arrival_time)
sd_arrival <- mean(arrivals$arrival_time^2)-(mean_arrival)^2
alpha_hat_MOM<- mean_arrival^2/sd_arrival

lambda_hat_MOM <- mean_arrival/sd_arrival
MOMs <- c(alpha_hat_MOM, lambda_hat_MOM)

#mle fits
MLEs <- MASS::fitdistr(arrivals$arrival_time, densfun = "gamma")

#c
hist(arrivals$arrival_time,
      xlab = "Arrival times",
      main = "Histogram of Arrival Times",
      freq = FALSE)
curve(dgamma(x, shape = MOMs[1], rate = MOMs[2]),
      col = "Red", from = 0, to = 500, add = TRUE)
curve(dgamma(x, shape = MLEs$estimate[1], rate = MLEs$estimate[2]),
      col = "Blue", from = 0, to = 500, add = TRUE)
legend("topright", legend = c("MOM", "MLE"),
      col = c("Red", "Blue"), lwd = 1)
```

Histogram of Arrival Times



```
#d
B <- 10000
n <- length(arrivals$arrival_time)
MOM_boots <- replicate(B, {
  MOM_data <- rgamma(n, shape = alpha_hat_MOM, rate = lambda_hat_MOM)
  mean_MOM_data <- mean(MOM_data)
  sd_MOM_data <- mean(MOM_data^2) - (mean(MOM_data))^2
  alpha_hat_MOM_boot <- mean_MOM_data^2 / sd_MOM_data
  lambda_hat_MOM_boot <- mean_MOM_data / sd_MOM_data
  c(alpha_hat_MOM_boot, lambda_hat_MOM_boot)
})
MLE_boots <- replicate(B, {
  MLE_data <- rgamma(n, shape = MLEs$estimate[1], rate = MLEs$estimate[2])
  MLE_boot <- MASS::fitdistr(MLE_data, "gamma")
  c(MLE_boot$estimate[1], MLE_boot$estimate[2])
})
```

```
Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```
Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
```

```
c(sd(MOM_boots[1,]), sd(MOM_boots[2,]))
```

```
[1] 0.0318999948 0.0004483089
```

```
c(sd(MLE_boots[1,]), sd(MLE_boots[2,]))
```

```
[1] 0.0204118570 0.0003263246
```

```
#####
#e. form approximate confidence intervals for the parameters.
# Bootstrap confidence intervals for Method of Moments
mom_ci_alpha <- quantile(MOM_boots[1,], probs = c(0.025, 0.975))
mom_ci_lambda <- quantile(MOM_boots[2,], probs = c(0.025, 0.975))
cat("Method of Moments - CI for MON Aloha:", mom_ci_alpha, "CI for MOM Lambda:", mom_ci_lambda)
```

```
Method of Moments - CI for MON Aloha: 0.9509891 1.075989 CI for MOM Lambda: 0.01180998 0.01359998
```

```
# Bootstrap confidence intervals for MLE
mle_ci_alpha <- quantile(MLE_boots[1,], probs = c(0.025, 0.975))
mle_ci_lambda <- quantile(MLE_boots[2,], probs = c(0.025, 0.975))
cat("Maximum Likelihood - CI for MLE Alpha:", mle_ci_alpha, "CI for MLE Lambda:", mle_ci_lambda)
```

```
Maximum Likelihood - CI for MLE Alpha: 0.9876939 1.0672 CI for MLE Lambda: 0.01222593 0.01359998
```

```
#the true value of is likely between 0.951 and 1.076.
#the true value of is likely between 0.0118 and 0.0136.
#the true value of is likely between 0.987 and 1.068.
#the true value of is likely between 0.0122 and 0.0135.

#Both the Method of Moments and MLE produce relatively similar confidence intervals for and
```

```
#which indicates that both methods are giving consistent estimates for these parameters based on
#MOM uses sample moments (mean, variance), while MLE maximizes the likelihood of the data given
```

```
#####
library(tidyverse)
```

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr      1.1.4      v readr      2.1.5
v forcats    1.0.0      v stringr    1.5.1
v ggplot2    3.5.1      v tibble     3.2.1
v lubridate  1.9.4      v tidyr      1.3.1
v purrr      1.0.2

-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()     masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```
library(plotrix)
```

```
N <- 10000
n <- length(arrivals$arrival_time)
head(arrivals$arrival_time,10)
```

```
[1] 38.02663 409.44521 272.46339 98.67453 87.85801 79.83959 211.38774
[8] 99.32829 181.32278 103.85056
```

```
mu <- mean(arrivals$arrival_time)
mu
```

```
[1] 79.93522
```

```
observed_CIs <- replicate(N, {
  sample_data <- sample(arrivals$arrival_time, size = n, replace = FALSE)
  c(mean(sample_data)-qnorm(0.975)*sd(sample_data)/sqrt(n),
    mean(sample_data)+qnorm(0.975)*sd(sample_data)/sqrt(n))
})
observed_CIs[, 1:5]
```

```
      [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 77.45264 77.45264 77.45264 77.45264 77.45264
[2,] 82.41780 82.41780 82.41780 82.41780 82.41780
```



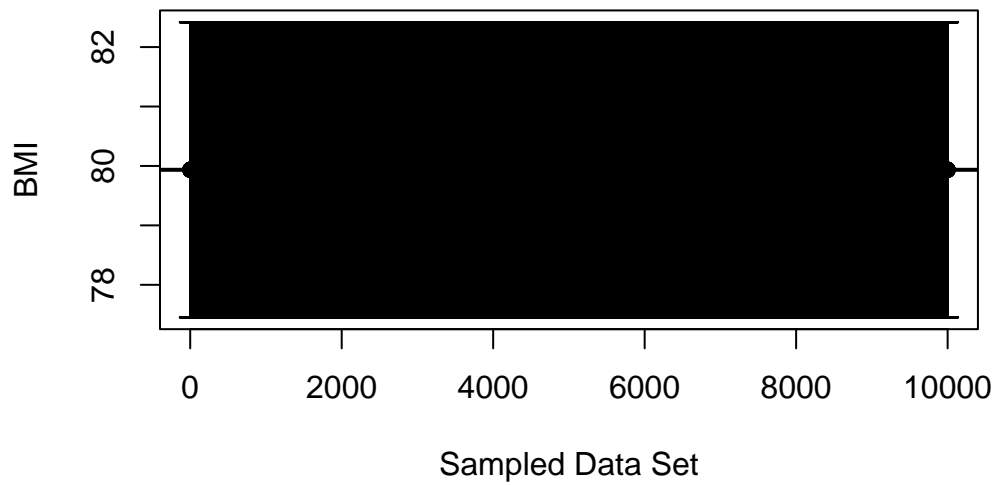
```
#check how many contained the true value
mean((observed_CIs[1, ] < mu) & (observed_CIs[2, ] > mu))
```

```
[1] 1
```

```
#quick function to color our intervals based on how they hit or miss
mycolor <- function(endpoints, par) {
  if (par < endpoints[1])
    "Red" # if the mean is below the left endpoint of the confidence interval
  else if (par > endpoints[2])
    "Orange" # if the mean is above the right endpoint of the confidence interval
  else "Black" # if the mean lies between the endpoints
}

#Load the plotrix package, which contains the plotCI function.
require(plotrix)
plotCI(x = 1:N,
       y = colMeans(observed_CIs),
       li = observed_CIs[1, ],
       ui = observed_CIs[2, ],
       col = apply(FUN = mycolor, X = observed_CIs, MARGIN = 2, par = mu),
       ylab = "BMI",
       xlab = "Sampled Data Set",
       main = paste0("Visualization of 100 CIs\nProportion containing mu = ",
                     mean((observed_CIs[1, ] < mu) & (observed_CIs[2, ] > mu)))
)
#draw a line for true mean
abline(h = mu, lwd = 2)
```

Visualization of 100 CIs Proportion containing $\mu = 1$



#####