Assignment 3 - Part 2

Version 2 with some corrections. Use this version of the assignment

STA238

Winter 2020

Suppose Joe owns a pizza shop. We know it's the favourite shop of at least one STA238 student, so this question is of serious practical importance. Joe is concerned with the number of customers he serves during the lunch hour. To study this, one day his trusted assistant records the time between successive customers. Let X_i be the number of minutes until the i^{th} customer enters Joe's shop, measured from when the previous customer entered.

We model $X_1, \ldots, X_n$ as independent and identically distributed random variables from the Exponential (λ_0) distribution, which is sometimes an appropriate model for waiting times. (Note that we are using here an alternative parametrization of the Exponential distribution, which might be different than what you've seen before; throughout this assignment work with the Exponential density given below.) The parameter λ represents the average number of minutes Joe has to wait until the next customer enters his shop, measured from the time the previous customer entered. $\lambda = 2$ would be 2 minutes per customer, $\lambda = 0.1$ would be 10 customers per minute, and so on. Joe wants to estimate the true value of λ, λ_0. The exponential density is $f_\lambda(x_i) = \lambda^{-1}\exp(-x_i/\lambda)$.

1. Show that the Maximum Likelihood Estimator (MLE) for λ in this model is $\widehat{\lambda} = \bar{X}$.

2. The waiting times data are posted on the assignment page on Quercus in the file `assignment3-waiting.csv`. Consider the following code. Describe in words, in full detail, the bootstrap algorithm that is implemented in the code. Indicate the distribution that is being estimated by the bootstrap distribution.

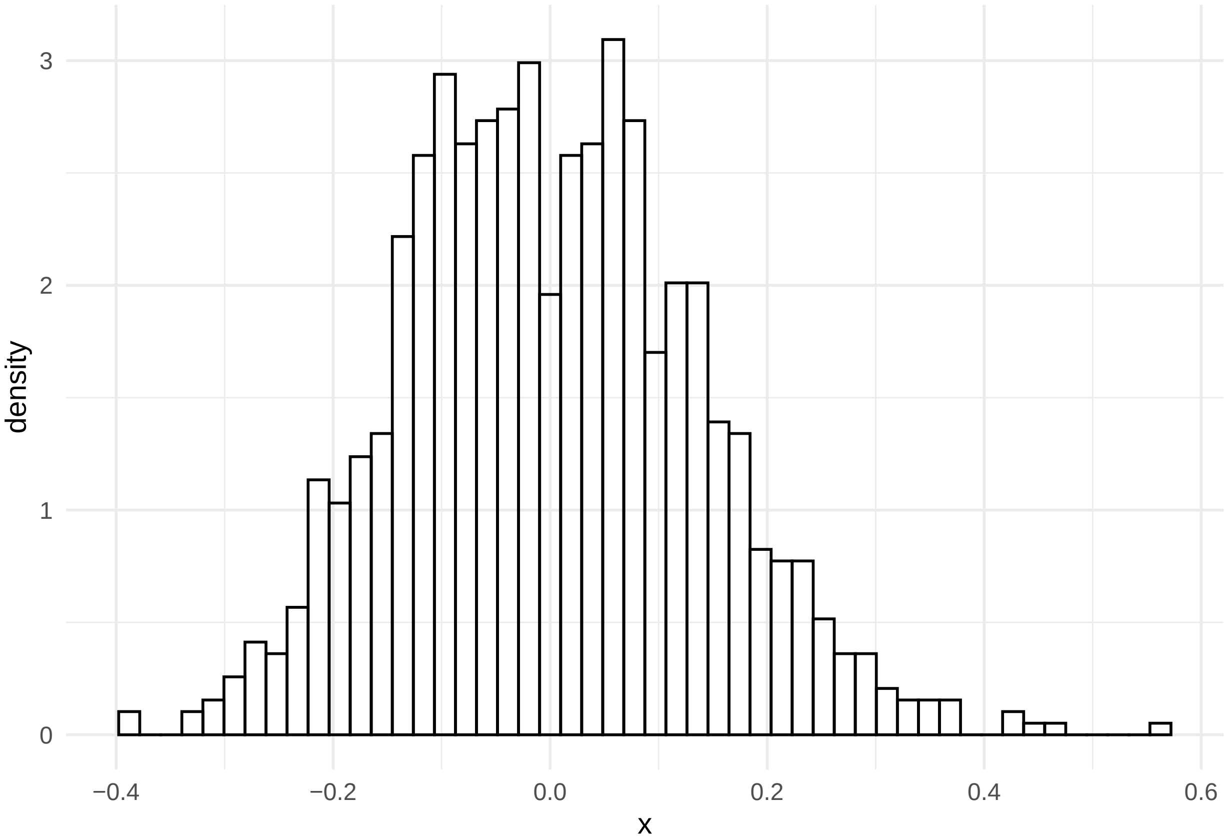
```r
set.seed(7886)
waiting <- readr::read_csv(
  # Include the path to where you saved the data on your computer.
  file = "assignment3-waiting.csv",
  col_names = TRUE,
  col_types = "n"
)
glimpse(waiting)
```

```
## Observations: 30
## Variables: 1
## $ waitingtime <dbl> 0.281, 0.330, 0.708, 0.463, 3.372, 0.008, 0.105, 1...
```

```r
n <- nrow(waiting)

B <- 1000 # Number of bootstrap samples to do
bootmle <- numeric(B)
for (b in 1:B) {
  samp <- sample(waiting$waitingtime,n,replace = TRUE)
  bootmle[b] <- mean(samp) - mean(waiting$waitingtime)
}

tibble(x = bootmle) %>%
  ggplot(aes(x = x)) +
  theme_minimal() +
  geom_histogram(aes(y = ..density..),bins = 50,colour = "black",fill = "transparent")
```



3. Joe comes in, angry. His assistant recorded the times *between* successive customers, when really, what he wanted was the number of customers that arrived each minute! Joe is furious at his hapless assistant.

Luckily his assistant is a statistics major, and knows that if $X_1, \ldots, X_n$ are independent Exponential(λ) waiting times in minutes, then the number of customers per minute $Y_1, \ldots, Y_m$ is an i.i.d. sample from a Poisson $\left(\lambda^{-1}\right)$ distribution. So the parameter representing the physical quantity of interest now is $\mu = \lambda^{-1}$. μ is the average number of customers per minute. So $\mu = 1/2$ would be one customer every two minutes and $\mu = 2$ would be one customer every 30 seconds, and so on.

Follow the steps below which will allow you to find an estimate of μ. First, you will need to convert the observed waiting times into a dataset containing the count of the number of customers arriving each minute. To do this, follow the following steps:

Step 1: Create a new dataframe with two new variables: `totaltime`, computed as the total time (continuous, fractional number of minutes) that has passed when each successive customer enters the pizza shop; and `numberofminutes`, which is the number of integer minutes that have passed (plus one) when each customer enters the shop. You can use `totaltime = cumsum(waitingtime)`, `numberofminutes = ceiling(totaltime)`, and remember you can create new variables in a dataframe with the `mutate` function in the `dplyr` package.

Your new dataframe should look like this (the numbers should match!):

```
## # A tibble: 30 x 3
##    waitingtime totaltime numberofminutes
##          <dbl>     <dbl>           <dbl>
##  1       0.281     0.281               1
##  2       0.330     0.611               1
##  3       0.708     1.32                2
##  4       0.463     1.78                2
##  5       3.37      5.15                6
##  6       0.008     5.16                6
##  7       0.105     5.27                6
##  8       1.52      6.79                7
##  9       0.056     6.85                7
## 10       0.834     7.68                8
## # ... with 20 more rows
```

Step 2: Now, add up the number of customers that arrived in each integer minute. You can `group_by(numberofminutes)` and then count the number of customers using `summarize(numberofcustomers = n())`. Your new dataframe should be called `customers` and should look like this (the numbers should match!):

```
customers1
```

```
## # A tibble: 18 x 2
##    numberofminutes numberofcustomers
##              <dbl>             <int>
##  1               1                 2
##  2               2                 2
##  3               6                 3
##  4               7                 2
##  5               8                 1
##  6               9                 2
##  7              10                 1
##  8              11                 2
##  9              12                 3
## 10              14                 1
## 11              16                 2
## 12              17                 1
## 13              19                 2
## 14              20                 1
## 15              22                 2
## 16              23                 1
## 17              25                 1
## 18              26                 1
```

Step 3: Finally, you have to add zeroes for minutes in which customers didn't arrive. Here is the code to this. Run it and make sure you have created the data frame below.

```
zeroes <- tibble(numberofminutes = 1:max(customers1$numberofminutes))
customers <- zeroes %>%
  left_join(customers1,by = "numberofminutes") %>%
  replace_na(list(numberofcustomers = 0))
```

```
customers
```

```
## # A tibble: 26 x 2
##    numberofminutes numberofcustomers
##              <dbl>             <dbl>
##  1               1                 2
##  2               2                 2
##  3               3                 0
##  4               4                 0
##  5               5                 0
##  6               6                 3
##  7               7                 2
##  8               8                 1
##  9               9                 2
## 10              10                 1
## # ... with 16 more rows
```

The column `numberofcustomers` now represents independent realizations of random variables $Y_1, \ldots, Y_m$ from the Poisson $\left(\lambda_0^{-1}\right)$ distribution, where λ_0 is the **same** λ_0 from question 1.

4. (a) The density of a Poisson $\left(\lambda^{-1}\right)$ random variable is

$$f_\lambda(y_j) = \frac{\lambda^{-y_j} e^{-1/\lambda}}{y_j!}$$

Use this density to find the MLE for λ and the MLE for $\mu = \lambda^{-1}$. *Hint*: you might want to find the MLE for μ first, and then apply the *principle of invariance* to find the MLE for λ.

(b) Find the MLE for $\mu = \lambda^{-1}$ using the *original sample of waiting times* and the *principle of invariance* of the MLE. Do you expect your answer to be exactly the same as your answer to part (a)? Why or why not?

5. Use your answer from question 4 to create a similar plot to that given in question 2. That is, construct a bootstrap estimate of the sampling distribution of $\lambda^{-1} - \lambda_0^{-1}$ and plot a histogram of this distribution. You might wish to rely heavily on the code given in question 2.

6. (a) For the random sample $X_1, \ldots, X_n$, it can be shown that

$$\frac{\hat{\lambda} - \lambda_0}{\hat{\lambda}} \sim \mathrm{N}\left(0, \frac{1}{n}\right)$$

Compute a 95% confidence interval for λ using this asymptotic distribution, using the observed values of the random sample $X_1, \ldots, X_n$.

(b) For the random sample $Y_1, \ldots, Y_m$, it can be shown that

$$\frac{\hat{\mu} - \mu_0}{\sqrt{\hat{\mu}}} \sim \mathrm{N}\left(0, \frac{1}{n}\right)$$

Compute a 95% confidence interval for μ using this asymptotic distribution, using the observed values of the random sample $Y_1, \ldots, Y_m$.

(c) Compare your confidence intervals in part (a) part (b). Interpret both of them: that is, "we are 95% confident that ____________ is between _____ and _____." Remember that $\mu_0 = \lambda_0^{-1}$.

7. (a) Compute the likelihood ratio Λ based on the sample $Y_1, \ldots, Y_m$ and evaluate the claim that $\mu = 2$, that is, two customers enter the shop per minute. Is this claim supported by the data?

(b) Compute the likelihood ratio Λ based on the sample $X_1, \ldots, X_n$ and evaluate the claim that $\lambda = 1/2$, that is, one customer enters the shop every 30 seconds. Is this claim supported by the data?

8. Use a bootstrap procedure to estimate a p-value corresponding to the claim that $\mu = 2$ based on $Y_1, \ldots, Y_m$. Is your value different than what you got in question 7 (a)? Say why you would expect them to be the same OR explain why they are different. (You may wish to use the code from the lectures for week 11 for this question.)

9. (a) Consider the observed values of the random sample of customers per minute, $y_1, \ldots, y_m$. Suppose we put a Gamma(a, b) prior distribution on $\mu = \lambda^{-1}$ with density

$$\pi(\mu) = \frac{b^a}{\Gamma(a)} \mu^{a-1} e^{-b\mu}$$

Derive the posterior distribution for $\lambda | y_1, \ldots, y_m$. State the name of the distribution and give expressions for its parameters.

(b) Compute a 95% posterior credible interval for μ. You may wish to use the `pgamma` function in `R` to calculate cumulative probability from the Gamma distribution. Interpret your interval and compare it to the confidence interval for μ that you obtained in question 6 (a). Comment on whether they are different. Compare their interpretations, and state whether you would expect them to be the same and why/why not.