

# SM-2302 R Individual Assignment Report

HAZIQJ Dr. Haziq Jamil

2022-10-25

## Total marks (out of 30): NA

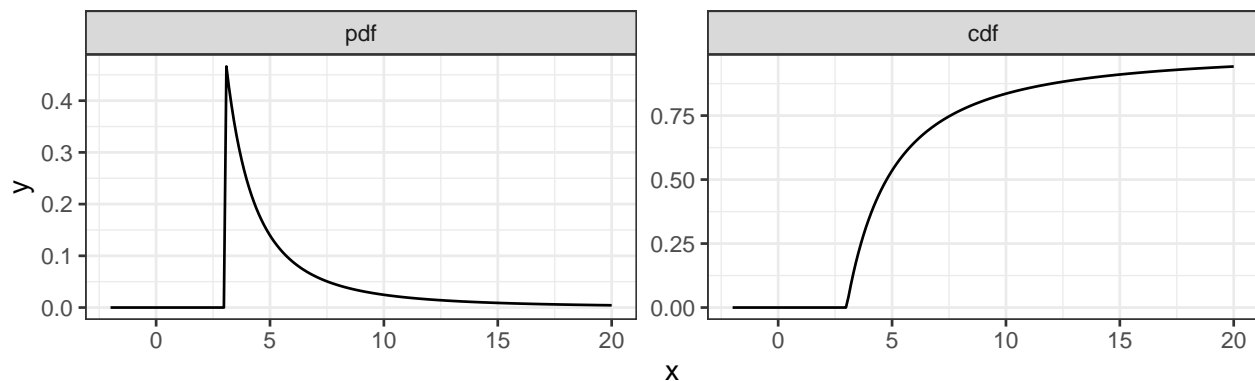
Latest commit time: [2022-10-25 09:20:38](#). Penalty marks: 0 (0 days late).

Item	Points	Weight	Marks
Code correctness (10)	NA	0.6	NA
Code styling (5)	NA	0.1	NA
Creativity (5)	NA	0.1	NA
Use of GitHub (5)	NA	0.1	NA
Following instructions (5)	NA	0.1	NA

## Checks

Context	No	Test	Result
Q1	1	Test pdf values	Success
	2	Returns 0 for values of $x < \beta$	Success
	3	Error for invalid parameter values	Success
Q2	4	Test deviance value (single $x$ )	Success
	5	Test deviance value (multiple $x$ )	Success
Q3	6	X correctly loaded	Success
	7	Correct MLE alpha value	Success
	8	Correct MLE beta value	Success
Q4	9	Test cdf values	Success
	10	Check quantile values	Success

Pareto distribution for  $\alpha = 1.5$  and  $\beta = 3$



```
## [ FAIL 0 | WARN 0 | SKIP 0 | PASS 10 ]
```

## Git log

```
## 9ad203f Haziq Jamil Tue Oct 25 09:20:38 2022 +0800 Update README.md
## cb60ece Haziq Jamil Mon Oct 24 21:03:50 2022 +0800 Update README.md
## 9cb3909 Haziq Jamil Mon Oct 24 20:47:52 2022 +0800 Update README.md
## 6404189 Haziq Jamil Mon Oct 24 20:35:11 2022 +0800 Update README.md
## 49d5a30 Haziq Jamil Mon Oct 24 20:34:39 2022 +0800 Update README.md
## 240e24e Haziq Jamil Mon Oct 24 20:30:18 2022 +0800 Update README.md
## b332536 Haziq Jamil Mon Oct 24 20:11:34 2022 +0800 Add data file to repo
## 76c593a Haziq Jamil Mon Oct 24 20:10:00 2022 +0800 Adjust tolerance for optim
## a98fbe7 Haziq Jamil Mon Oct 24 20:02:13 2022 +0800 Edit data import line
## bb84c38 Haziq Jamil Mon Oct 24 19:54:25 2022 +0800 Complete Q5
## aeee6b8 Haziq Jamil Mon Oct 24 19:53:21 2022 +0800 Complete Q4
## 1694ea2 Haziq Jamil Mon Oct 24 19:48:49 2022 +0800 Complete Q3
## 3999f54 Haziq Jamil Mon Oct 24 19:46:57 2022 +0800 Complete Q2
## 372e8cf Haziq Jamil Mon Oct 24 19:46:49 2022 +0800 Complete Q1
## ddb0408 Haziq Jamil Mon Oct 24 19:34:31 2022 +0800 First commit
## 17545a7 github-classroom[bot] Mon Oct 24 11:31:13 2022 +0000 Initial commit
```

## Source

```
1  # Load libraries -----
2  library(tidyverse)
3
4  # Q1 -----
5  pareto_pdf <- function(x, alpha, beta) {
6    # First test for invalid parameter values
7    if (alpha <= 0 | beta <= 0)
8      stop("Parameters alpha and beta must be > 0.")
9
10   # The Pareto pdf
11   res <- alpha * beta ^ alpha / x ^ (alpha + 1)
12   res[x < beta] <- 0 # pdf is zero when x < beta
13   return(res)
14 }
15
16 # Some tests:
17 # pareto_pdf(10, 5, 5)
18 # pareto_pdf(10:15, 5, 5) # it's vectorised too
19
20 # Q2 -----
21 pareto_dev <- function(alpha, beta, x) {
22   # First test for invalid parameter values
23   if (alpha <= 0 | beta <= 0)
24     stop("Parameters alpha and beta must be > 0.")
25
26   # Return the log-likelihood
27   res <- -2 * log(alpha) - 2 * alpha * log(beta) +
28     2 * (alpha + 1) * log(x[x >= beta]) # only interested in values for which
29                                           # x >= beta
30   sum(res)
31 }
32
```

```

33 # Some tests:
34 # pareto_dev(2, 2, 2:10)
35 # sum(-2 * log(pareto_pdf(x = 2:10, alpha = 2, beta = 2)))
36
37 # Note: It's fine to do sum(-2 * log(pareto_pdf)), but it breaks down easily
38 # because the pdf explodes with large values of alpha. Better to use log scale
39 # directly.
40
41 # Q3 -----
42 # Read in the data set
43 X <- scan("haziqj.txt") # other functions like read.table() are fine
44                          # as long as it's vectorised
45
46 # Compute the MLE
47 beta_hat <- min(X)
48 res <- optim(5, pareto_dev, method = "L-BFGS-B", lower = 0, x = X,
49             beta = beta_hat)
50 alpha_hat <- res$par
51
52 # Alternatively, can code the alpha_hat directly based on the formulae from
53 # Wikipedia or differentiation by hand.
54
55 # Q4 -----
56 pareto_cdf <- function(x, alpha, beta) {
57   # First test for invalid parameter values
58   if (alpha <= 0 | beta <= 0)
59     stop("Parameters alpha and beta must be > 0.")
60
61   # Return the cdf
62   res <- 1 - (beta / x) ^ alpha
63   res[x < beta] <- 0 # F(x) >= 0
64   return(res)
65 }
66
67 # If using integrate(), one can do the following:
68 # pareto_cdf <- function(x, alpha, beta) {
69 #   res <- integrate(pareto_pdf, lower = -Inf, upper = x, alpha = alpha,
70 #                     beta = beta)
71 #   return(res$value)
72 # }
73
74 # Since the cdf is available in closed form (either from Wikipedia or
75 # integration by hand), best to code that directly rather than use a numerical
76 # integrator.
77
78 # Q5 -----
79 B <- 1000
80 xvals <- seq(min(X), max(X), length = B)
81 cdf_vals <- rep(NA, B)
82 for (i in seq_along(xvals)) {
83   cdf_vals[i] <- pareto_cdf(xvals[i], alpha = alpha_hat, beta = beta_hat)
84 }
85

```

```
86 qvals <- rep(NA, 5)
87 quantiles <- c(0.05, 0.25, 0.5, 0.75, 0.95)
88 for (j in seq_along(quantiles)) {
89   qvals[j] <- xvals[which.min(abs(cdf_vals - quantiles[j]))]
90 }
```

## Feedback

NA