STAT680_HW1

Yudi Zhang, Wangqian Ju

9/24/2021

Q1

a)

We have $g(X) = X^{\alpha-1}$ and $X \sim \text{Exp}(1)$

```r
n <- 10000
u <- runif(n)

# obtain the Exp(1) random variable
ee <- -log(1-u)

alpha <- 1.89

e.gamma <- ee^(alpha - 1) %>% mean

# check the result using gamma function
c(e.gamma, gamma(alpha))
```

```
## [1] 0.9667408 0.9583793
```

b)

$$\begin{aligned}
Var(\frac{1}{n}\sum X_i^{\alpha-1}) &= \frac{1}{n}Var(X^{\alpha-1}) \\
&= \frac{1}{n}(E[X^{2(\alpha-1)}] - [E(X^{\alpha-1})]^2) \\
&= \frac{1}{n}\left(\Gamma(2\alpha-1) - [\Gamma(\alpha)]^2\right)
\end{aligned}$$

c)

```r
estimate_gamma <- function(alpha, n = 1000) {
  u <- runif(n)
  ee <- -log(1-u)

  tt <- ee^(alpha - 1)
```

```r
  est <- mean(tt)
  sd <- sqrt(var(tt) / n)

  result <- c(est, sd)
  names(result) <- c("estimate", "sd")
  return(result)
}
```

For $\alpha = 1.5$, $\Gamma(\alpha) = 0.8862269$. And for different simulation sizes, we have:

```
##       n  estimate          sd
## 1    10 0.9197723 0.152759381
## 2   100 0.8690389 0.043931641
## 3  1000 0.8945611 0.014730002
## 4 10000 0.8900807 0.004675612
```

d)

```r
estimate_gamma_antithetic <- function(alpha, n = 1000) {
  u <- runif(n)
  u2 <- 1 - u
  u <- c(u, u2)
  ee <- -log(1-u)

  tt <- ee^(alpha - 1)
  est <- mean(tt)
  sd <- sqrt(var(tt) / (2*n))

  result <- c(est, sd)
  names(result) <- c("estimate", "sd")
  return(result)
}
```

We can see that the method of antithetic variates can reduce the standard error.

```r
ttd.df
```

```
##    estimate          sd       type
## 1 0.8826963 0.004641837         MC
## 2 0.8860397 0.003269658 Antithetic
```

e)

```r
tr_val <- 1 - exp(-1)
n <- 100000

### i)
u <- runif(n)
tt <- as.numeric(u < exp(-(1:n) / n))
mean(tt)
```

```
## [1] 0.63094
```

```r
sqrt(var(tt) / n)
```

```
## [1] 0.001525965
```

```r
### ii)
u <- runif(n)
tt <- exp(-u)
mean(tt)
```

```
## [1] 0.6316708
```

```r
sqrt(var(tt) / n)
```

```
## [1] 0.0005724503
```

```r
### iii)
u <- runif(n)
u2 <- 1 - u
u <- c(u, u2)
tt <- exp(-u)
mean(tt)
```

```
## [1] 0.6321535
```

```r
sqrt(var(tt) / (2*n))
```

```
## [1] 0.0004049526
```

```r
### iv)
is_beta <- function(a=2, b=2, n=1000) {
  bb <- rbeta(n, a, b)
  tt <- exp(-bb) / dbeta(bb, a, b)
  est <- mean(tt)
  sd <- sqrt(var(tt) / n)
  result <- c(est, sd)
  names(result) <- c("estimate", "sd")
  return(result)
}

grid <- readRDS("grid.rds")

grid
```

```
## # A tibble: 625 x 5
##         a     b estimate      sd    diff
##     <dbl> <dbl>    <dbl>   <dbl>   <dbl>
##   1   0.2   0.2    0.635 0.00226 0.00331
```

```
##  2   0.4   0.2    0.631 0.00208 0.00112
##  3   0.6   0.2    0.630 0.00231 0.00215
##  4   0.8   0.2    0.629 0.00273 0.00275
##  5   1     0.2    0.633 0.00334 0.000962
##  6   1.2   0.2    0.631 0.00418 0.00106
##  7   1.4   0.2    0.640 0.00563 0.00823
##  8   1.6   0.2    0.643 0.00802 0.0107
##  9   1.8   0.2    0.627 0.00888 0.00545
## 10   2     0.2    0.621 0.0129  0.0111
## # ... with 615 more rows
```

```
# grid <- expand.grid(a=seq(0.2,5,by=.2), b=seq(0.2,5,by=.2))
#
# grid %<>% as_tibble %>% mutate(
#   is_beta_result = purrr::map2(a, b, function(aa, bb) {
#     rr <- is_beta(aa, bb, n = 100000) %>% t %>% as_tibble
#     colnames(rr) <- c("estimate", "sd")
#     rr
#   })
# ) %>% unnest(is_beta_result) %>%
#   mutate(
#     diff = abs(tr_val - estimate)
#   )
#
# grid$sd %>% which.min() %>% {grid[.,]}
# grid$diff %>% which.min() %>% {grid[.,]}
# grid %>% filter(a==1, b==1)
# is_beta(1, 1, n = 100000)
#
# saveRDS(grid, "homework1/grid.rds")
```

Q2

a)

```
library(boot)
require(fitdistrplus)
```

```
## Loading required package: fitdistrplus
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survival'

## The following object is masked from 'package:boot':
##
##     aml
```

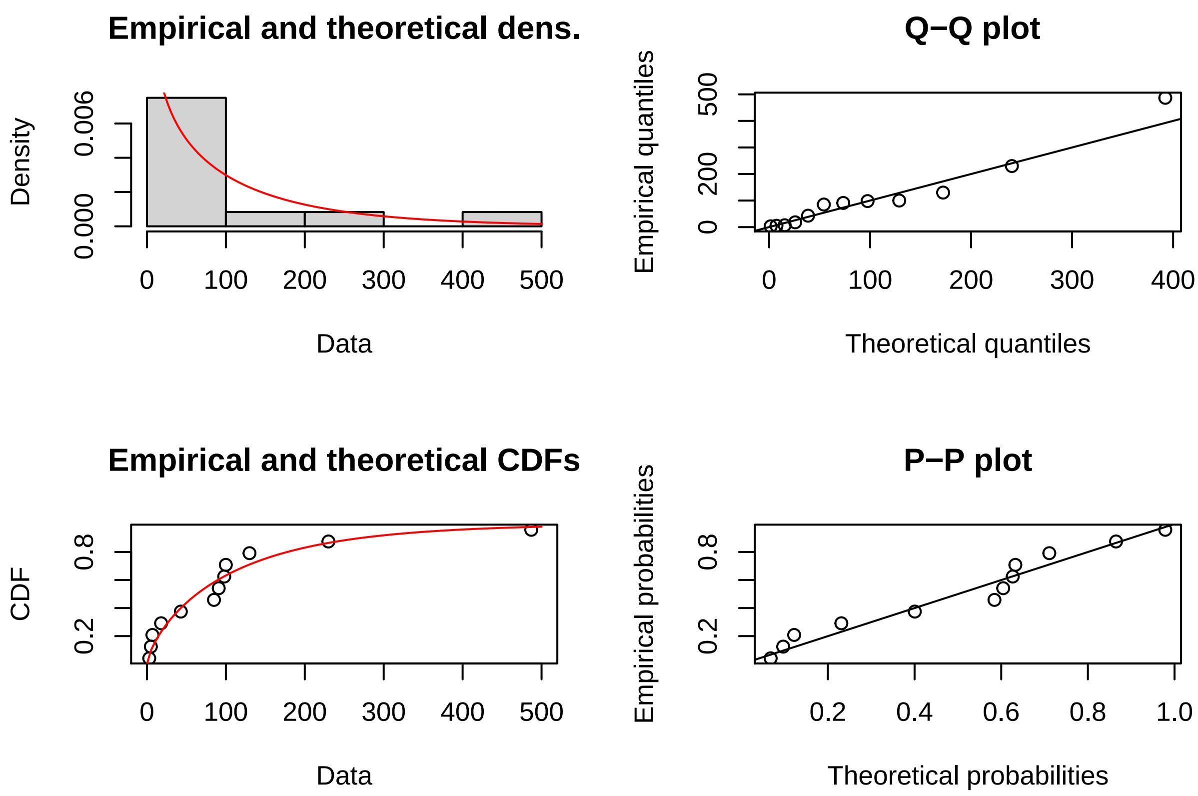
```
data("aircondit")

fit.gamma <- fitdist(aircondit$hours, distr = "gamma", method = "mle")
```

```
## $start.arg
## $start.arg$shape
## [1] 0.6866689
##
## $start.arg$rate
## [1] 0.006353143
##
##
## $fix.arg
## NULL
```

```
summary(fit.gamma)
```

```
## Fitting of the distribution ' gamma ' by maximum likelihood
## Parameters :
##          estimate Std. Error
## shape 0.706402862  0.2419358
## rate  0.006535456  0.0030685
## Loglikelihood:  -67.64542   AIC:  139.2908   BIC:  140.2607
## Correlation matrix:
##          shape     rate
## shape 1.000000 0.700236
## rate  0.700236 1.000000
```

```
plot(fit.gamma)
```




```
tb.find <- function(alpha, data) {
  xbar <- mean(data)
  result <- log(alpha) - log(xbar) - digamma(alpha) + mean(log(data))
  return(result)
}

alpha.est.result <- uniroot(tb.find, interval = c(0.001, 4), data = aircondit$hours)
alpha.est.result
```

```
## $root
## [1] 0.7064924
##
## $f.root
## [1] 1.100713e-06
##
## $iter
## [1] 10
##
## $init.it
## [1] NA
##
## $estim.prec
## [1] 6.103516e-05
```

```r
alpha.est <- alpha.est.result$root
beta.est <- alpha.est / mean(aircondit$hours)
c(alpha.est, beta.est)
```

```
## [1] 0.706492376 0.006536552
```

```r
# https://www.math.arizona.edu/~jwatkins/O3_mle.pdf

## b)
jk.est <- sapply(seq_along(aircondit$hours), function(i) {
  data <- aircondit$hours
  data.tmp <- data[-i]

  est <- uniroot(tb.find, interval = c(0.001, 4), data = data.tmp)
  est$root
})
n <- length(aircondit$hours)
eiv <- (n-1) * (alpha.est - jk.est)
bias.alpha <- - mean(eiv)

# var(jk.est) * (n-1)^2 / n

(sum(eiv^2) - n * bias.alpha^2) / (n * (n-1))
```

```
## [1] 0.06605922
```