STAT3023 Statiscal Inference

Lab Week 7

Tutor: Wen Dai

SID: 470408326

School of Mathematics and Statistics

The University of Sydney

Semester 2, 2021

Consider the test of $H_0 : \theta = \theta_0$ against $H_1 : \theta \neq \theta_0$ based on $X \sim f(.; \gamma_0, \theta)$ where

$$f(x; \gamma, \theta) = \begin{cases} \frac{1}{\Gamma(\gamma)\theta^{\gamma}} x^{\gamma-1} e^{-\frac{x}{\theta}} & \text{for } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

that is X has a gamma distribution with known shape parameter γ_0 but unknown scale parameter θ.

1. Consider first the exponential case where $\gamma_0 = 1$ and suppose the hypothesised value of the scale parameter (also the mean in this case) is $\theta_0 = 1$

(a) The "equal-tailed"test at level α rejects for $X \leq a$ or $X \geq b$ where

$$P_1(X \leq a) = P_1(X \geq b) = \frac{\alpha}{2}$$

Taking $\alpha = 0.05$, determine the value sof `a` and `b` satisfying (2) above (**hint:** use `qexp()`).

Solution

Since we take $\gamma_0 = 1$ and under the null $\theta_0 = 1$ we have that the the pdf is given by:

$$f(x) = e^{-x} \qquad \text{for } x > 0$$

```
a = qexp(0.025, rate = 1)
b = qexp(0.975, rate = 1)
c(a, b)

## [1] 0.02531781 3.68887945
```

(b) We shall plot the power function of the equal-tailed test. Define a vector of θ-values: `th=(250:1500)/1000` and obtain a corresponding vector of values of the power (the probability of rejecting) for each such θ-value; that is:

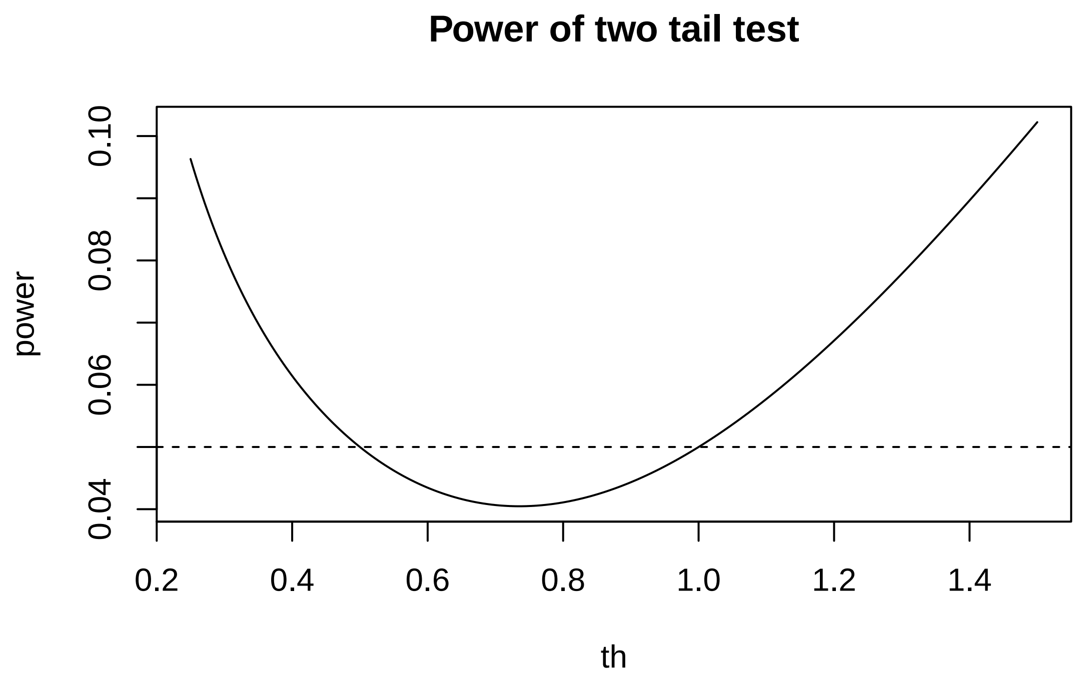
$$P_\theta(X \leq a) + P_\theta(X \geq b)$$

Finally plot the power against `th` and add a horizontal dashed line at $\gamma = 0.05$. Add an informative heading, etc and remember to use `type = 'l'`

Solution

We plot the power function of the equal-tailed test. Note that for $P_\theta(X \geq b)$ we note that $P_\theta(X \geq b) = 1 - P_\theta(X \leq b)$. We alse use the `abline(.)` function to plot a horizontal line

```
th = (250:1500)/1000
power =  pexp(a, rate = 1/th, lower.tail = TRUE) + pexp(b, rate = 1/th, lower.tail = FALSE)
# Plot theta against the power:
plot(th, power, type = "l", main = "Power of two tail test")
# add a horizontal dashed line at gamma = 0.05
abline(h = 0.05, lty = 2)
```



(c) This is a 1-parameter exponential family with sufficient statistic X and so (since it is continuous) the *uniformly most powerful unbiased* (UMPU) test is of the form

$$\delta(X) = \begin{cases} 1 & \text{for } X \geq d \\ 0 & \text{for } c < X < d \\ 1 & \text{for } X \leq c \end{cases}$$

Where c and d are chosen so that:

$$\mathbb{E}[_]\theta_0 [\delta(X)] = \alpha$$

$$\mathbb{E}[_]\theta_0 [X\delta(X)] = \alpha\, \mathbb{E}[_]\theta_0 (X) = \alpha$$

Since $\mathbb{E}[_]\theta_0 (X) = 1.$ We show in a tutorial exercise these are equivalent to

$$\begin{aligned} 1 - e^{-c} + e^{-d} &= \alpha \\ ce^{-c} &= de^{-d} \end{aligned}$$

From the first equation we get:

$$\begin{aligned} e^{-d} &= \alpha - 1 + e^{-c} \\ \implies d &= -\log\left[\alpha - 1 + e^{-c}\right] \end{aligned}$$

Hence, once c is determined we can compute d. To determine c we need to solve the equation:

$$ce^{-c} - de^{-d} = ce^{-c} + \left\{\log\left[\alpha - 1 + e^{-c}\right]\right\}\left[\alpha - 1 + e^{-c}\right] = 0$$

We can use the R function `uniroot()` to determine c *numerically*.

(i) Write an R function which computes the middle member of the equation above (i.e. the function whose root we wish to find).

Solution

We write a function which computes the root c to the equation

$$ce^{-c} + \left\{\log\left[\alpha - 1 + e^{-c}\right]\right\}\left[\alpha - 1 + e^{-c}\right] = 0$$

```
fn = function(c, alpha) {
  term = alpha - 1 + exp(-1 * c)
  return(c * exp(-c) + (log(term) * term))
}
```

(ii) Noting that c can be no bigger than the lower 0.05-quantile of the exponential(1) distribution, execute a certain command involving `eps = 1e-5`. Note: the use of `eps` here is to stay away from the upper bound, since there the function is trying to evaluate `log(0)`

```
eps = 1e-5
uniroot(f = fn, lower = 0, upper = qexp(0.05) - eps, alpha = 0.05)

## $root
## [1] 0.04235629
##
## $f.root
## [1] -3.187136e-05
##
## $iter
## [1] 4
##
## $init.it
## [1] NA
##
## $estim.prec
## [1] 6.103516e-05
```

(iii) Write an R function which takes as an argument the level `alpha` and returns a list with elements `c` and `d`, corresponding to the desired values c and d defining the UMPU test (3) above for $\theta_0 = 1$ and $\alpha = 0.05$

Solution

We write a function `expon.umpu` (exponential uniformly most powerful unbiased test) which takes in α and returns the optimal c and d which solve the above equations.

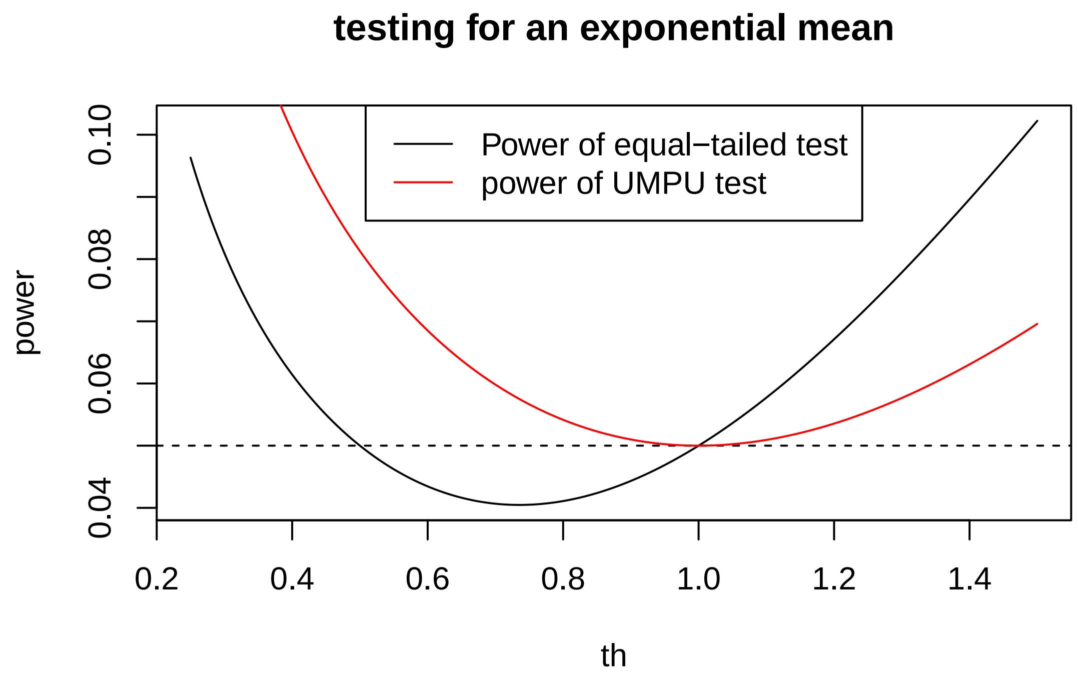
```
expon.umpu = function(alpha) {
  eps = 1e-8
  c = uniroot(f = fn, lower = 0, upper = qexp(alpha) - eps, alpha = alpha)$root
  term = alpha - 1 + exp(-c)
  d = -log(term)
  list(c_val = c, d_val = d)
}
expon.umpu(0.05)

## $c_val
## [1] 0.04235611
##
## $d_val
## [1] 4.764356
```

(d) Recreate your plot from part (b) above and add to it a red curve of the power of the UMPU test. Add an informative heading and legend. **Comment** on what feature of the plot indicates that the UMPU test is unbiased. The power function of the UMPU test never goes below the 0.05 level, thus it is unbiased.

Solution

We now plot the power of the UMPU test in red:

```
th = (250:1500)/1000
# for the two tailed test
power = pexp(a, 1/th) + pexp(b, 1/th, lower.tail = FALSE)
plot(th, power, type = "l", main = "testing for an exponential mean")
abline(h = 0.05, lty = 2)
# for the uniformly most powerful test:
umpu = expon.umpu(0.05)
umpu.power = pexp(umpu$c, 1/th) + pexp(umpu$d, 1/th, lower.tail = FALSE)
lines(th, umpu.power, col = "red")
legend("top", legend = c("Power of equal-tailed test", "power of UMPU test"),
       col = c("black", "red"),
       lty = c(1, 1))
```



2. Consider now the case where $Y_1, ..., Y_n$ are iid exponential with mean θ and we again wish to test $H_0 : \theta = \theta_0$ against a two sided $H_1 : \theta \neq \theta_0$. The likelihood is:

$$\Pi_{i=1}^n \left[\frac{1}{\theta} e^{-\frac{Y_i}{\theta}}\right] = \exp\left(-\frac{1}{\theta}\sum_{i=1}^n Y_i - n\log\theta\right)$$

and so clearly $X = \sum_{i=1}^n Y_i$ is a sufficient statistic; indeed this is a 1-parameter exponential family. The UMPU test is thus of the form (3) where c and d are chosen to satisfy the two conditions (4) and (5).

However since X itself has a gamma distribution with shape parameter n and scale parameter θ, the UMPU above is the same as for a single observation from the density (1) above with $\gamma_0 = n$. We show in the tutorial that in the present case the two conditions (4) and (5) are equivalent to

$$\int_0^c f(x; n, 1)\, \mathrm{d}x + \int_d^\infty f(x; n, 1)\, \mathrm{d}x = \alpha = \int_0^c f(x; n+1, 1)\, \mathrm{d}x + \int_d^\infty f(x; n+1, 1)\, \mathrm{d}x$$

Below we shall write a function to determine the UMPU test, for the case $n = 5$ and $\alpha = 0.05$.

(a) By adapting your solution to part (c) of the previous question, write a function (playing the same role as the function `fn()` above; it will use `pgamma()` and `qgamma()`) the root of which gives the desired value of c to solve the above equations (**hint**: in the body of this function you will need to first find `d` in terms of `c` and `alpha` using one of the two constraints). Then use `uniroot()` to actually find the root. Wrap this all in an appropriate function which takes as input values of `alpha` and `n` and outputs a list with elements `c` and `d`, the lower and upper critical values of the desired UMPU test.

Solution

```
gamma.root = function(c, n, alpha) {
  lower = pgamma(q = c, shape = n, scale = 1) # P (X < c)
  # P(X < c) + P(X > d) = alpha
  # P(X > d) = alpha - P(x < c)
  # P(X < d) = 1 - alpha + P(X < c)
  # d = F^-1(1 - alpha + P(X < c))
  d = qgamma(p = 1 - alpha + lower, shape = n, scale = 1)

  # return the equation to solve--> the latter half of the integral equation
  return(pgamma(c, shape = (n + 1), scale = 1) +
           pgamma(d, shape = (n + 1), scale = 1, lower.tail = FALSE) -
           alpha)
}
```

Show the roots:

```
eps = 1e-8
upper = qgamma(0.05, shape = 5, scale = 1) - eps
uniroot(f = gamma.root, lower = 0, upper = upper, n = 5, alpha = 0.05)

## $root
## [1] 1.758069
##
## $f.root
## [1] 1.352949e-06
##
## $iter
## [1] 6
##
## $init.it
## [1] NA
##
## $estim.prec
## [1] 6.103516e-05
```

And finally wrap all this up in a function:

```r
gamma.umpu = function(alpha, n) {
  eps = 1e-8
  upper = qgamma(alpha, shape = n, scale = 1) - eps
  c = uniroot(f = gamma.root, lower = 0, upper = upper, n = n, alpha = alpha)$root
  lower = pgamma(c, shape = n, scale = 1)
  d = qgamma(1 - (alpha - lower), shape = n, scale = 1)
  return(list(c_val = c, d_val = d))
}
```

(b) Use your `gamma.umpu()` function to determine the appropriate c and d for the UMPU test for this problem with $n = 5$ and $\alpha = 0.05$. Plot the power as a function of θ and graphically verify that the test is unbiased and of level 0.05.

```r
gamma.umpu(0.05, 5)

## $c_val
## [1] 1.758069
##
## $d_val
## [1] 10.86438

gu = gamma.umpu(0.05, 5) # obtain c and d
g.power = pgamma(gu$c, shape = 5, scale = th) +
  1 - pgamma(gu$d, shape = 5, scale = th)
plot(th, g.power, type = "l")
abline(h = 0.05, lty = 2)
```