Programa para obtenes EMV's y Perfiles de la distribución Gamma

Hecho por Isis Mociño

Consideraciones

Usamos las siguientes variables:

- `datos`: vector de datos.
- `n`: cantidad de datos.
- `mu`: μ.
- `alp`: α.
- `t1` y `t2`: estadísticas suficientes

Parametrización correcta

Contamos con la función `dgamma` en R que recibe los parámetros `shape` α y `scale` σ. Donde la densidad está dada por

$$f(x) = \frac{1}{\sigma^\alpha \Gamma(\alpha)} x^{\alpha-1} e^{-x/\sigma}$$

Así que para obtener la parametrización en términos de α y μ basta con tomar $\sigma = \mu/\alpha$. Esto es importante pues si usan `dgamma`, deberá ser de la siguiente manera:

```
dgamma(x, shape = alp, scale = mu/alp, log = FALSE)
```

Encontrar EMV's

Para obtener $\widehat{\mu}$ basta con

```
mu_g <- t1/n
```

En cambio, para obtener $\widehat{\alpha}$ es un poco más complicado. Denotemos por $l_{p\alpha}$ a la logverosimilitud perfil de α. Debemos encontrar la raíz de la derivada de $l_{p\alpha}$; es decir, $\widehat{\alpha}$ es tal que

$$l_{p\alpha}'(\alpha) = 0.$$

Paso 1: Declarar funciones necesarias

En el código $l_{p\alpha}$ y su derivada están dadas respectivamente como

```
lvp_alp <- function(alp){
  return(lv(alp,mu_g))
}

dlvp_alp <- function(alp){
  return(n*log(alp) - n*digamma(alp) - n*log(t1/n) + t2)
}
```

Aclaración: La función `digamma` hace lo siguiente:

digamma(x) calculates the digamma function which is the logarithmic derivative of the gamma function, ψ(x) = d(ln(Γ(x)))/dx = Γ'(x)/Γ(x).

Paso 2: Obtener raíz de $l_{p\alpha}{}'$

Usamos la función uniroot la cual recibe la función a la que se le quiere encontrar la raíz (`dlvp_alp`) y un vector que indica el intervalo de búsqueda (en mi caso, me es útil el intervalo $(0.1,\ 5)$). El valor obtenido es $\widehat{\alpha}$.

```
alp_g <- uniroot(dlvp_alp, c(0.1, 5))$root
```

Perfil relativa de α

La función correspondiente a la verosimilitud perfil relativa de α es

```
Rlvp_alp <- function(alp){
  return(exp(lvp_alp(alp) - lvp_alp(alp_g)))
}
```

La función correspondiente a la logverosimilitud perfil relativa de α es

```
rlvp_alp <- function(alp){
  return(lvp_alp(alp) - lvp_alp(alp_g))
}
```

Niveles e intervalos

Ahora, obtenemos los niveles del 90%, 95% y 99% de acuerdo con la tabla

```
c_90 <- 0.2585 - 1/(0.5171 + 1.6667*n)
c_95 <- 0.1465 - 1/(2.029 + 2.084*n)
c_99 <- 0.0362 - 1/(14.609 + 4.886*n)
```

Realizamos una funcion que sirve para encontrar la α (el lugar de μ considerar α) que satisface la siguiente igualdad

$$r_p\left(\mu; t_1, t_3\right) = \ln c$$

donde c es el nivel deseado. La función se muestra a continuación. Recibe el nivel y regresa una lista con el límite izquierdo y derecho del intervalo.

```
intervalo_alp <- function(nivel){
  aux <- function(alp){
    return(rlvp_alp(alp) - log(nivel))
  }
  l <- uniroot(aux, c(alp_g, alp_g*10)) //tal vez deban manipular el intervalo
  r <- uniroot(aux, c(0.1, alp_g)) //tal vez deban manipular el intervalo
  inter <- list("l" = l, "r" = r)
```

```
  return(inter)
}
```

Con esto podemos obtener los intervalos

```
int_90_alp <- intervalo(c_90)
int_95_alp <- intervalo(c_95)
int_99_alp <- intervalo(c_99)
```

Para acceder a los límites del intervalo basta lo siguiente

```
int_90_alp$l //Arroja el limite izquierdo
int_90_alp$r //Arroja el limite derecho
```

Con esto deberían poder graficar la perfil relativa de α y sus niveles.
Importante: Grafican `Rlvp_alp`, no `rlvp_alp`

Perfil relativa de μ

La función correspondiente a la perfil relativa de μ es

```
lvp_mu <- function(mu){
  aux2 <- function(alp){
    return(lv(alp, mu))
  }
  res <- optimize(aux2, c(0.001, alp_g+10), maximum = TRUE)
  return(res$maximum)
}
```

La función correspondiente a la verosimilitud perfil relativa de α es

```
Rlvp_mu <- function(mu){
  return(exp(lvp_mu(mu) - lvp_mu(mu_g)))
}
```

La función correspondiente a la logverosimilitud perfil relativa de α es

```
rlvp_mu <- function(mu){
  return(lvp_alp(mu) - lvp_alp(mu_g))
}
```

Niveles e intervalos

Usaremos los mismos niveles que para α. Realizamos una funcion que sirve para encontrar la μ que satisface la siguiente igualdad

$$r_p(\mu; t_1, t_3) = \ln c$$

donde c es el nivel deseado. La función se muestra a continuación. Recibe el nivel y regresa una lista con el límite izquierdo y derecho del intervalo.

```
intervalo_mu <- function(nivel){
  aux3 <- function(mu){
    return(rlvp_mu(mu) - log(nivel))
  }
  r <- uniroot(aux3, c(mu_g-0.5, mu_g + 1000), extendInt = "yes")$root
  l <- uniroot(aux3, c(0.1, mu_g))$root
  inter2 <- list("l" = l, "r" = r)
  return(inter2)
}
```

Con esto podemos obtener los intervalos

```
int_90_mu <- intervalo_mu(c_90)
int_95_mu <- intervalo_mu(c_95)
int_99_mu <- intervalo_mu(c_99) // dio error porque el intervalo es muy grande
```

Se accede a los límites de la misma manera

```
int_90_mu$l //Arroja el limite izquierdo
int_90_mu$r //Arroja el limite derecho
```

Importante: Grafican `Rlvp_mu`, no `rlvp_mu`

Contornos

Se hace igual que en las otras distribuciones, por lo que no hay código.