Trabalho 1
Inferência — Prof. Vinícius D. Mayrink
Wyara Vanesa Moura e Silva — 25 de Outubro de 2022

Questão

Saiba que se $X_i \sim \text{Weibull}(\alpha, \beta)$ então

$$f(x_i|\alpha,\beta) = \frac{\alpha}{\beta}\left(\frac{x_i}{\beta}\right)^{\alpha-1} \exp\left\{-\left(\frac{x_i}{\beta}\right)^{\alpha}\right\}, \quad \text{para} \quad x_i \geq 0.$$

Nesta distribuição contínua temos que $\alpha > 0$ é o parâmetro de forma e $\beta > 0$ é o parâmetro de escala.

Suponha que $X_1, \cdots, X_n$ seja uma amostra aleatória proveniente da distribuição Weibull$(\alpha, 1)$. Implemente o método de Newton, usando o *software R*, para obter a estimativa de máxima verossimilhança de α. Utilize o conjunto de dados fornecido na página do curso.

Solução:

O gráfico da Figura 1, refere-se ao histograma dos dados fornecidos.

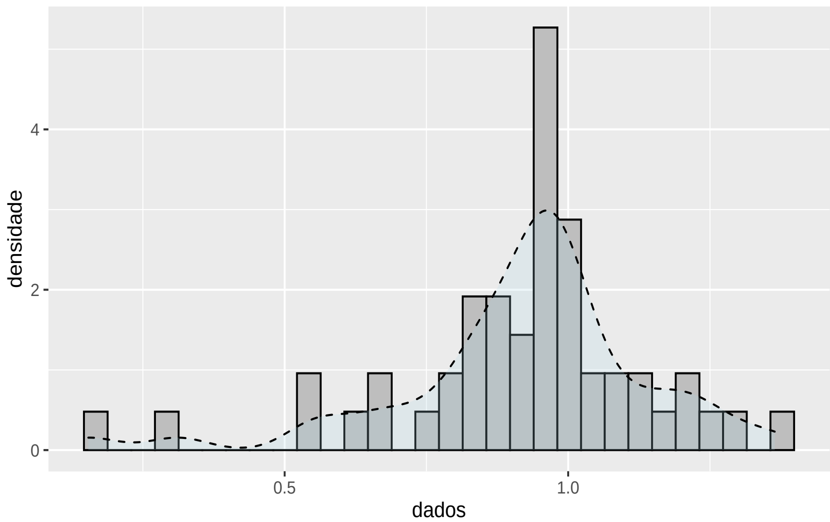

Figure 1: Histograma dos dados Weibull. A linha tracejada corresponde a densidade.

A função de verossimilhança, para α proveniente de uma amostra da distribuição Weibull$(\alpha, 1)$, é:

$$\begin{aligned} f(\mathbf{x}|\alpha) &= \prod_{i=1}^{n} \frac{\alpha}{1}\left(\frac{x_i}{1}\right)^{\alpha-1} \exp\left\{-\left(\frac{x_i}{1}\right)^{\alpha}\right\} \\ &= \alpha^n \left[\prod_{i=1}^{n} (x_i)^{\alpha-1}\right] \exp\left\{-\sum_{i=1}^{n} (x_i)^{\alpha}\right\} \end{aligned}$$

E assim, a log-verossimilhança, função que queremos otimizar, é dada por:

$$\ln f(\mathbf{x}|\alpha) = n \ln \alpha + (\alpha - 1)\sum_{i=1}^{n} (\ln x_i) - \sum_{i=1}^{n} (x_i)^{\alpha}$$

A fim de utilizar o método de Newton para encontrar uma estimativa de máxima verossimilhança para α, teremos que encontrar a primeira e segunda derivada da função log-verossimilhança. Desta forma, $f(\alpha)$ é a função que queremos encontrar as raizes utilizando o método de newton, e além dela também precisamos de $f'(\alpha)$ para implementar o algoritmo, tais funções são as respectivas derivadas da função log-verossimilhança:

$$\begin{aligned} f(\alpha) &= \frac{\partial}{\partial \alpha} \ln f(\mathbf{x}|\alpha) &= \frac{n}{\alpha} + \sum_{i=1}^{n} (\ln x_i) - \sum_{i=1}^{n} (\ln x_i)\,(x_i)^{\alpha} \\ f'(\alpha) &= \frac{\partial^2}{\partial \alpha^2} \ln f(\mathbf{x}|\alpha) &= -\frac{n}{\alpha^2} - \sum_{i=1}^{n} (\ln x_i)^2\,(x_i)^{\alpha} \end{aligned}$$

Para implementação do algoritmo no R foi utilizado a fórmula de recorrência do método de newton:

$$\alpha_{k+1} = \alpha_k - \frac{f(\alpha_k)}{f'(\alpha_k)}, \quad k = 0, 1, 2, \cdots$$

Além disso, foi escolhido uma precisão de 10^{-7}, ou seja, o critério de parada do programa será quando o valor absoluto da diferença entre o valor anterior e o atual forem menores que esta precisão. Outra característica importante do programa é a necessidade da entrada de um chute inicial para o parâmetro (ou seja, é necessário que esteja dentro do espaço paramétrico) que se deseja realizar a estimação, neste caso α.

Os códigos em R estão na parte final deste trabalho, a partir daqui serão mostrados alguns resultados utilizando esses algoritmos.

Para os dados fornecidos, o $\hat{\alpha}$ estimado que irá maximizar a função de verossimilhança será: $\hat{\alpha} = 4.965997$. O chute inicial utilizado foi $\hat{\alpha}_0 = 10$.

A Tabela 1 mostra alguns resultados de estimações do parâmetro α utilizando o algoritmo desenvolvido, foram geradas amostras com diferentes tamanhos de n, e com $\alpha = 5$. O chute inicial utilizado nestas simulações foram todos iguais, $\hat{\alpha}_0 = 2$.

Table 1: Simulação para diferentes tamanhos de amostras.

n	10	30	100	1000
α	5.595741	4.692904	5.088293	5.0562

Podemos observar que quanto maior a amostra, o algoritmo tem um melhor desempenho quanto a estimação do verdadeiro valor para o parâmetro α. Não foram obtidos problemas na estimação do parâmetro, quando foram introduzidos diferentes valores nos chutes iniciais de α.

Código R

Veja abaixo os códigos em R utilizados.

```
rm(list=ls(all=TRUE))
set.seed(22)
setwd("C:\\Users\\Wyara Silva\\Meu Drive\\disciplina-isolada-ufmg-2022\\
inferencia-estatistica\\trabalho-1")

dados.f = read.table("Dados_Weibull.txt")
dados = unlist(dados.f, use.names = FALSE)

library(ggplot2)

ggplot(dados.f, aes(x=V1)) +
  geom_histogram(aes(y=..density..), bins = 30, color="black", fill="grey")+
  geom_density(alpha=.2, linetype = 2, fill="light blue") +
  labs(x="dados", y = "densidade")

newton.raphson.w = function(x.dados, alpha.0=10, precisao = 1e-7, n=100){

  #primeira e segunda derivada da log-verossimilhança com relação a alpha
  dlogLikW = function(y){(length(x.dados)/y) + sum(log(x.dados)) -
  sum(log(x.dados)*(x.dados^y))}
  ddlogLikW = function(z){-(length(x.dados)/z^2) -
  sum((log(x.dados)^2)*(x.dados^z))}

  for (i in 1:n) {
    #equação de recorrência de newton-raphson
    alpha.1 = alpha.0 - dlogLikW(alpha.0)/ddlogLikW(alpha.0)

    #verificar se convergiu
    if(abs(alpha.1 - alpha.0) < precisao){
      res = list(alpha.1,i)
      names(res) = c("alpha.estimado","n.iter")
      return(res)
    }

    #nova iteração
    alpha.0 = alpha.1
  }
  print("com o número de iterações não houve convergência")
}

newton.raphson.w(dados)

#simulação
n = c(10,30,100,1000)

estimativas = array(NA, dim=c(1, length(n)))
```

```
for (i in 1:length(n)){
  #a parametrização da distribuição weibull na função 'rweibull'
  #é igual a utilizada neste trabalho.
  dat =  rweibull(n[i], shape=5, scale=1)
  estimativas[,i] = newton.raphson.w(dat,2)$alpha.estimado
}

estimativas
```