

# Resolução Lista de exercicio 1

Aluno: Gerson

pdf/Lista R - PGMAT0061.pdf

## Resolução Lista de exercicio 1

### 1. Questão 1

Escreva uma função no R para calcular o coeficiente de correlação linear de Pearson entre duas variáveis. Compare os resultados de sua função com aqueles obtidos pelo R mediante o uso da função `cor(.)`. Para isso, considere o conjunto de dados a seguir:

#### 1.1 Criar a função

```
correlacao <- function(x, y) {  
  # numerador  
  numerador = sum(  
    (x - mean(x)) *  
    (y - mean(y))  
  )  
  # denominador  
  denominador <- sqrt(  
    sum((x - mean(x))**2) *  
    sum((y - mean(y))**2)  
  )  
  r_person = numerador / denominador  
  
  return(r_person)  
}
```

## 1.2 Teste da função

```
x <- c(0.06,-0.55,-1.41,-1.57,0.07,-0.65,0.73,0.73,-0.22,0.27)
y <- c(-0.46, 0.1, -2.51, -2.31, -1.06, -0.67, 0.72, 0.5, 0.4, 0.77)

a <- correlacao(x, y) |> round(7)
b <- cor(x, y) |> round(7)

glue::glue("A correlação calculada é {a} e a correlação do R é {b}")
```

A correlação calculada é 0.8497228 e a correlação do R é 0.8497228

## 2. Questão 2

Numa determinada localidade, a distribuição de renda (em unidades monetárias, u.m.) é uma variável aleatória  $X$  com função de distribuição de probabilidade:

**f.d.p**

```
f_X <- function(x) {
  dplyr::case_when(
    x >= 0 & x <= 2 ~ (1/10*x + 1/10)
    ,x > 2 & x <= 6 ~ (-3/40*x + 9/20)
    ,TRUE ~ 0
  )
}
```

**2.a Mostre que  $f(x)$  é uma f.d.p**

**1. Para qualquer  $x_i$ ,  $f(x) \geq 0$**

```
min_x = 0
f_X(min_x) > 0 # TRU
```

[1] TRUE

```
max_x = 6
f_X(max_x) > 0 # TRUE
```

```
[1] TRUE
```

Como  $f(x)$  é maior que zero para o valor mínimo e máximo da função, então  $f(x) \geq 0$

## 2. $F(x_i) = 1$

```
F_X <- function(x) {
  if_else(x >= 0 & x <= 6, integrate(f_X, lower = 0, upper = x)$value, 0)
}

glue::glue("Seja x = 6 o limite superior de X,
  então F({max_x}) = {F_X(max_x)} => f(x) é f.d.p")
```

Seja  $x = 6$  o limite superior de  $X$ ,  
então  $F(6) = 1 \Rightarrow f(x)$  é f.d.p

### 2.b Gráfico de $f(x)$ .

```
x <- seq(0, 6, by = 0.1)
y <- purrr::map_dbl(x, f_X)

df <- data.frame(x = x, y = y)

ggplot(df, aes(x = x, y = y)) +
  geom_line() +
  labs(
    title = "Função de distribuição de probabilidade",
    x = "x",
    y = "f(x)"
  )
```



**2.c  $P(x \geq 4.5)$**

```
p_4.5 = (1 - F_X(4.5)) * 100

glue::glue("A probabilidade de encontrar uma pessoa
           com renda superior a 4.5 u.m é {p_4.5 |> round(2)}%")
```

A probabilidade de encontrar uma pessoa  
com renda superior a 4.5 u.m é 8.44%

**2.d Calcular  $E(X)$  e  $Var(X)$**

**$E(X)$**

```
E_X <- integrate(\(x) {x * f_X(x)}, lower = 0, upper = 6)$value
E_X |> round(2)
```

[1] 2.47

## Var(X)

```
Var_X <- integrate(\(x) {(x - E_X)^2 * f_X(x)}, lower = 0, upper = 6)$value  
Var_X |> round(2)
```

```
[1] 1.78
```

## 3. Questão 3

Seja a amostra abaixo obtida de uma distribuição Poisson de parâmetro  $\lambda$ :

```
x <- c(5,4, 6,2, 2,4, 5, 3, 3, 0, 1, 7, 6, 5, 3, 6, 5, 3, 7, 2)
```

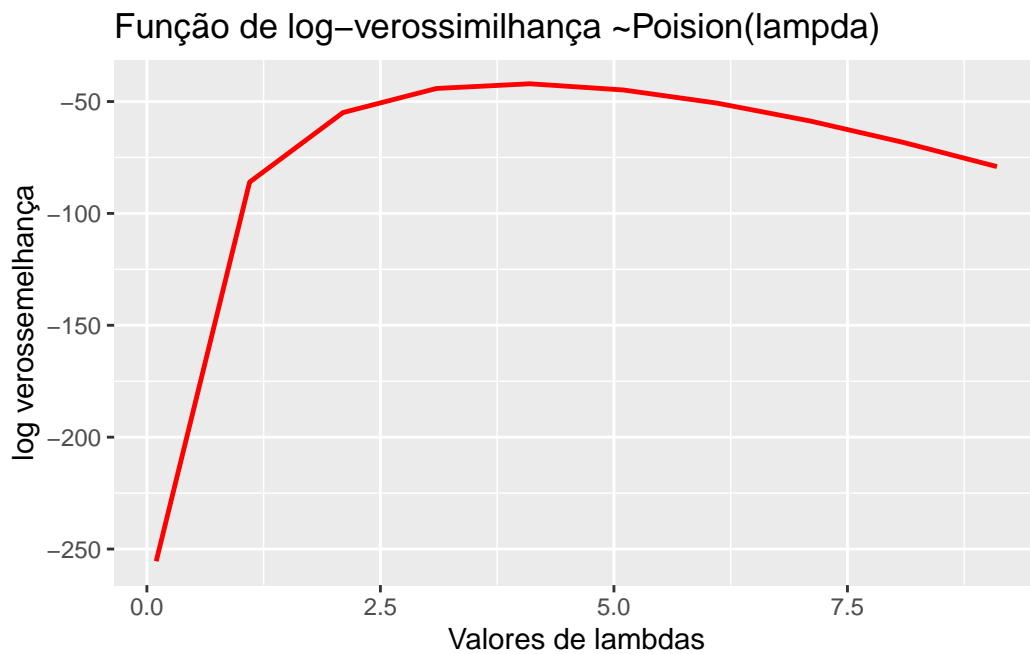
### 3.1 Obtenha o gráfico da função de log-verossimilhança

```
# Função da log verossimilhança  
log_vero_poisson <- function(lambda, x) {  
  sum(  
    x * log(lambda)  
    - lambda  
    - log(factorial(x))  
  )  
}  
  
# Vetor de lambdas  
lambdas <- seq(0.1, 10)  
  
# Aplicar a função para um vetor de lambda  
valores_log_vero <- map_dbl(lambdas, ~log_vero_poisson(.x, x))  
  
df <- data_frame(lambdas, valores_log_vero)
```

Warning: `data\_frame()` was deprecated in tibble 1.1.0.  
i Please use `tibble()` instead.

```
ggplot(df, aes(x = lambdas, y = valores_log_vero)) +
  geom_line(color = "red", size = .85) +
  labs(
    title = 'Função de log-verossimilhança ~Poisson(lampda)',
    x = 'Valores de lambdas',
    y = 'log verossemelhança'
  )
```

Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
i Please use `linewidth` instead.

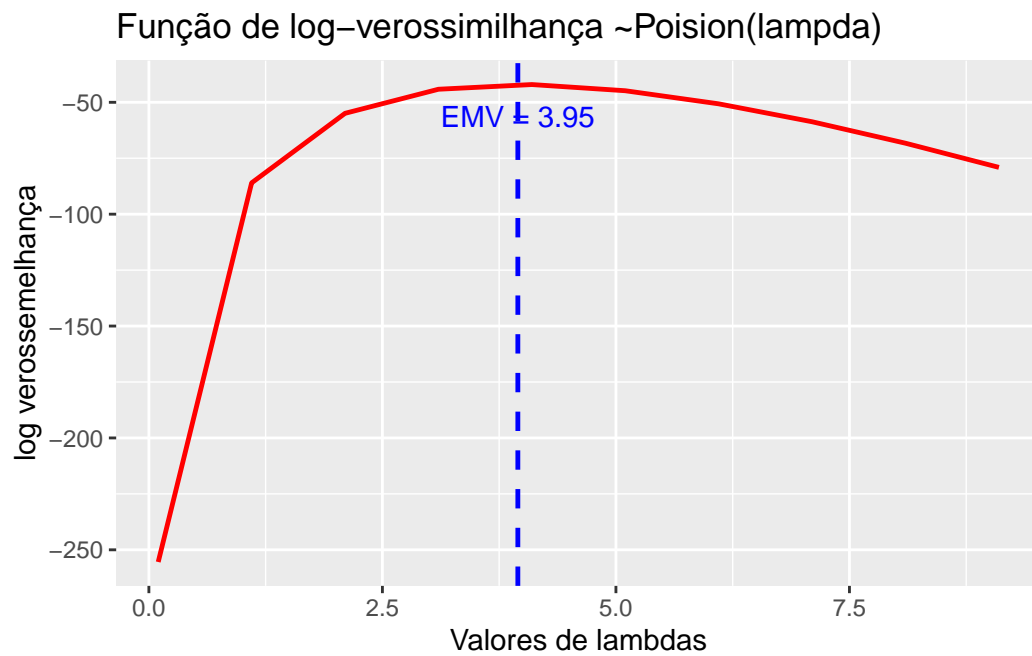


### 3.2 Encontre a E.M.V. de $\lambda$ e represente-a no gráfico anterior

```
EMV = mean(x)

ggplot(df, aes(x = lambdas, y = valores_log_vero)) +
  geom_line(color = "red", size = .85) +
  geom_vline(xintercept = EMV, color = "blue", linetype = "dashed", size = .85) +
  annotate("text", x = EMV, y = max(valores_log_vero),
    label = paste("EMV =", round(EMV, 2)),
    vjust = 2, color = "blue") +
```

```
labs(
  title = 'Função de log-verossimilhança ~Poisson(lampda)',
  x = 'Valores de lambdas',
  y = 'log verossemelhança'
)
```



#### Questão 4