Lista 1, aula 1 em 20/9

Henrique Ap. Laureano
laureano@ufpr.br ∧ www.leg.ufpr.br/~henrique/
/UFPR/DEST/LEG/

September 25, 2019

Contents

1 Derivadas e aproximação em séries de Taylor **1**
1. Desenhe o gráfico das funções. Identifique o que o parâmetro controla 1
2. Calcule a derivada analitica e numericamente das funções 6
3. Determine a reta tangente de $f(x)$ no ponto requisitado e esboce o gráfico . . . 20
4. Aproxime as funções usando a expansão de Taylor de segunda ordem 23
5. Com as funções do 4. encontre a reta tangente em três pontos, esboce o gráfico 27
6. Com as funções do 4. identifique o ponto de inflexão e veja se é de max ou min 31
7. Função perda absoluta 32
8. Ajustar uma reta relacionando y_i com x_i, usando a função perda absoluta . . . 34

1 Derivadas e aproximação em séries de Taylor

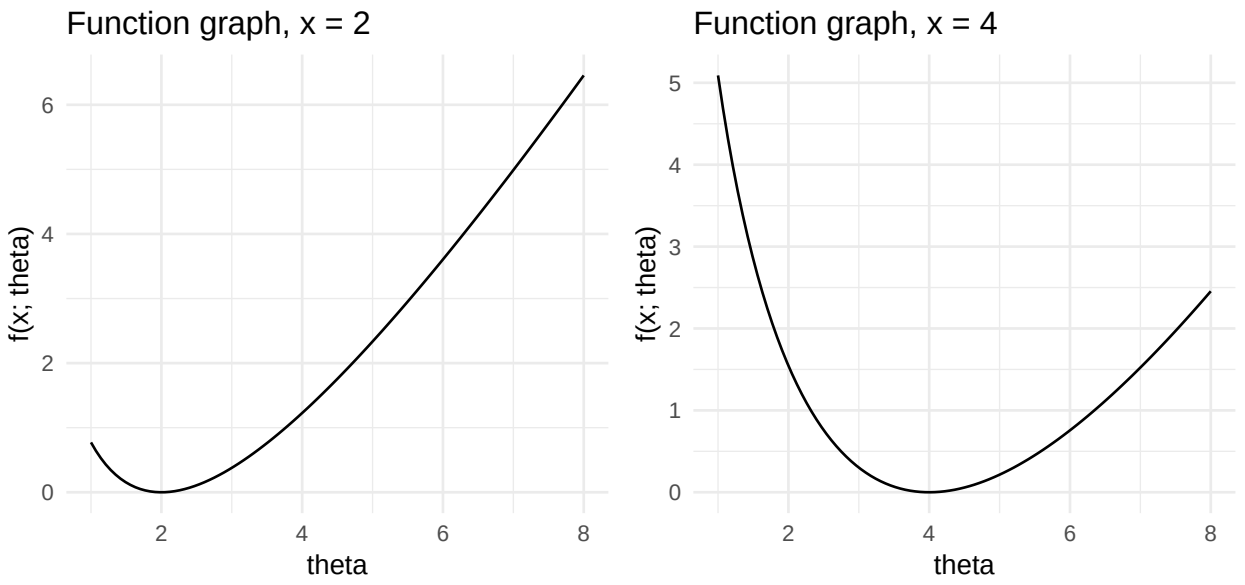
1. Desenhe o gráfico das seguintes funções. Identifique o que o parâmetro controla da função.

<r code>
```
library(ggplot2)
library(gridExtra)

fnplot <- function(fn, theta_seq, x, extra = FALSE, ...) {
    ggplot(data.frame(theta = theta_seq), aes(x = theta)) +
        theme_minimal() +
        stat_function(fun = fn, args = list(x = x, ...)) +
        labs(y = "f(x; theta)",
             title = ifelse(extra == FALSE,
                            paste("Function graph, x =", x),
                            paste("Function graph, x =", x, "and", extra)))
}
```

a. $f(x;\theta) = 2(x \log \frac{x}{\theta} - x + \theta)$

Deviance de uma Poisson:

<r code>
```
f1a <- function(theta, x) { 2 * (x * log(x/theta) - x + theta) }

grid.arrange(fnplot(f1a, theta_seq = seq(8), x = 2),
             fnplot(f1a, theta_seq = seq(8), x = 4), ncol = 2)
```



No caso da dist. Poisson, o parâmetro θ corresponde a taxa de ocorrência de um evento num dado intervalo de tempo. No gráfico da esquerda temos como se dois eventos foram observados, no gráfico da direita, quatro eventos. Concluímos que com a Deviance o mínimo da função é dado no valor mais provável, *likely*, para a tal taxa de ocorrência. Nestes casos, na quantidade de eventos observados, 2 e 4.

b. $f(x;\theta) = \binom{100}{x} \exp[x \log \frac{\theta}{1-\theta} + 100 \log(1-\theta)]$

Função densidade de uma Binomial escrita na forma da família Exponencial:

<r code>

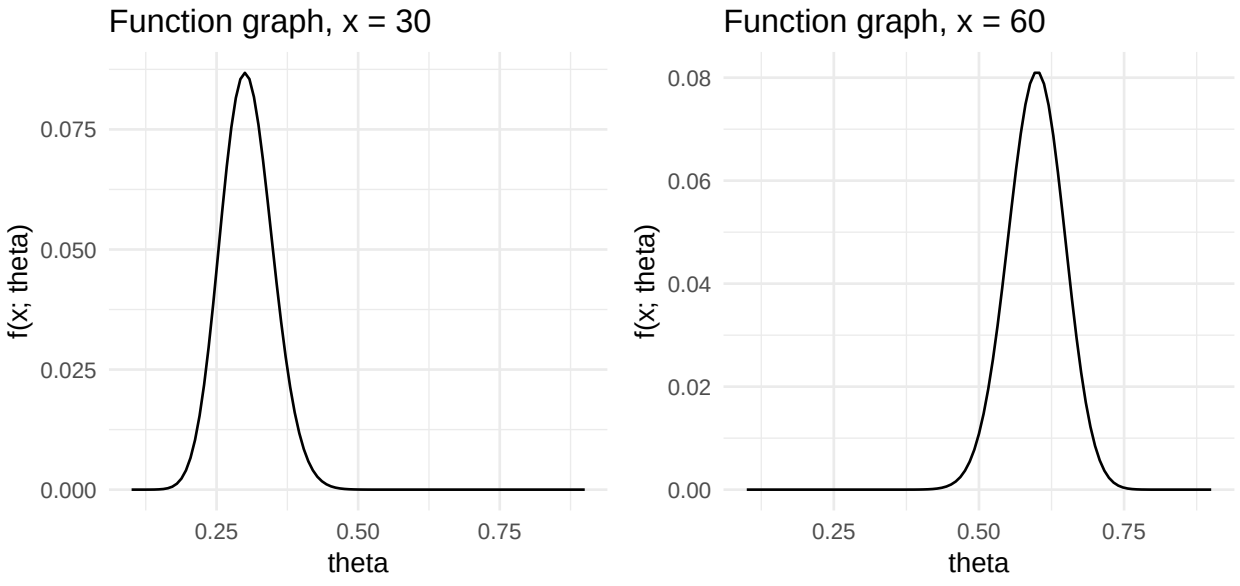
```r
f1b <- function(theta, x) {
    choose(100, x) * exp(x * log(theta/(1 - theta)) + 100 * log(1 - theta))
}
# equivalente a usar a dbinom(), behold:
f1b(theta = .5, x = 50)
```

```
[1] 0.07958924
```

```r
dbinom(x = 50, size = 100, prob = .5)
```

```
[1] 0.07958924
```

```r
grid.arrange(fnplot(f1b, theta_seq = seq(.1, .9, .1), x = 30),
             fnplot(f1b, theta_seq = seq(.1, .9, .1), x = 60), ncol = 2)
```

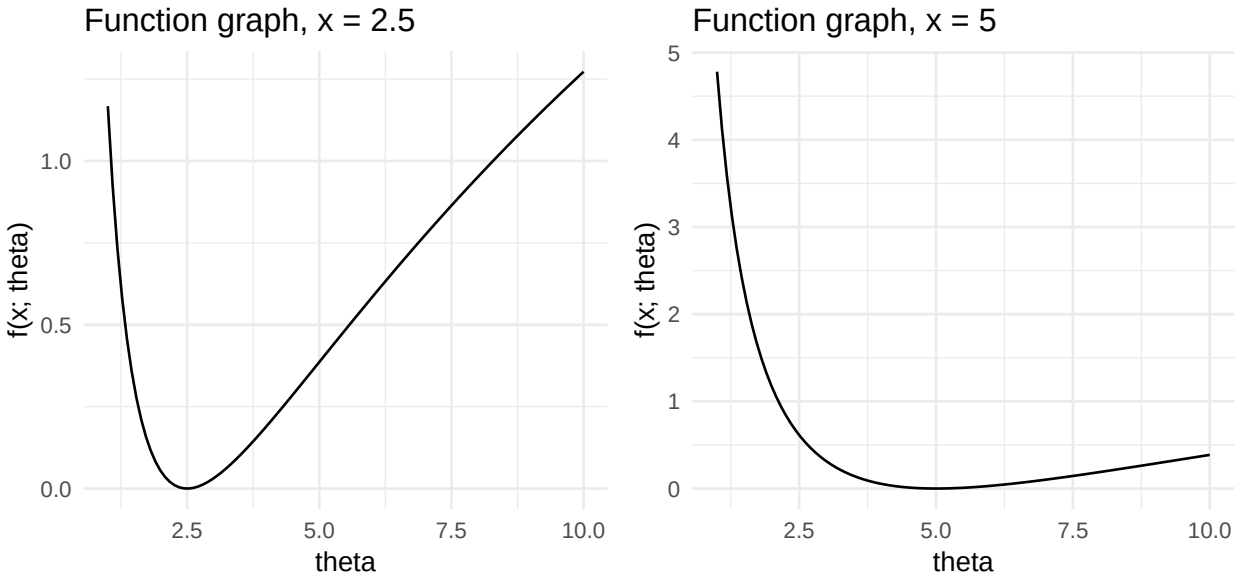


Com a dist. Binomial é simples, θ corresponde a probabilidade de ocorrência do dado observado. Consequentemente, temos as maiores probabilidades ao redor das quantidades x's estabelecidas.

c. $f(x;\theta) = 2(\frac{x}{\theta} - \log\frac{x}{\theta} - 1)$

Deviance de uma Gama(forma, escala):

<r code>
```r
f1c <- function(theta, x) { 2 * (x/theta - log(x/theta) - 1) }

grid.arrange(fnplot(f1c, theta_seq = seq(10), x = 2.5),
             fnplot(f1c, theta_seq = seq(10), x = 5), ncol = 2)
```



A função Deviance fornecida corresponde a uma dist. Gama com parâmetros de forma e escala, comumente denotados por k

e θ, respectivamente. Nos gráficos vemos uma clara relação do parâmetro de escala, θ, com o valor x fornecido.

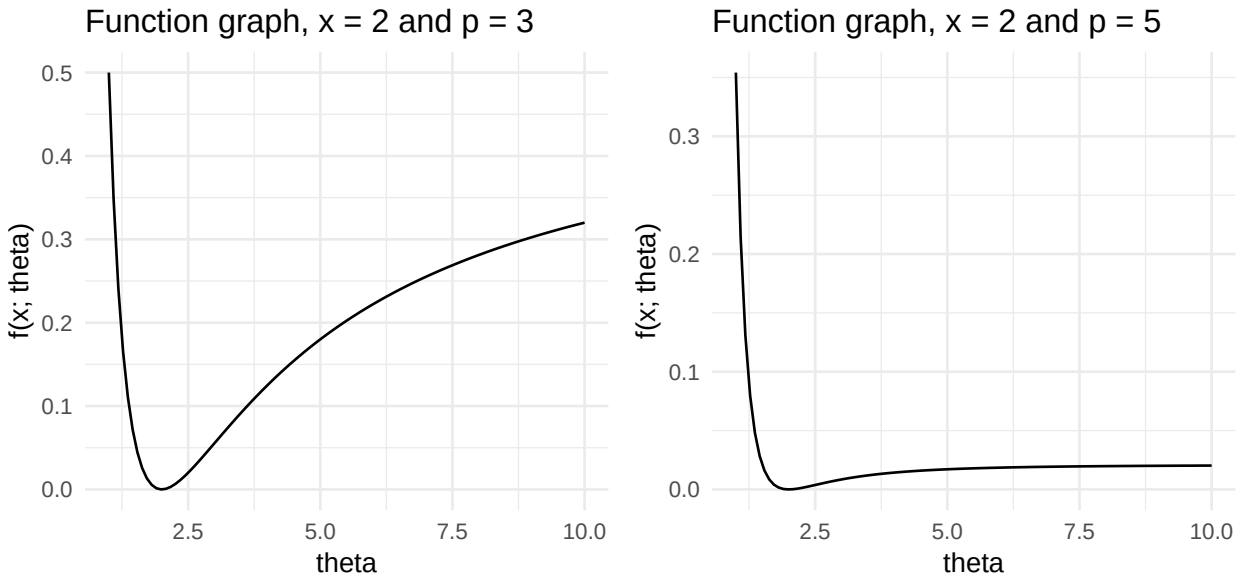
Para valores de x próximos de zero, apenas valores de θ bem próximos de x possuem $f(\cdot)$'s próximas de zero, lembrando que a dist. Gama não admite valores negativos. Quanto mais distante de zero, maiores são as equivalências dos θ's ao redor de x, i.e., mais parecidos, equivalentes, são os valores de $f(\cdot)$ retornados para diferentes θ's.

d. $f(x;\theta,p) = 2(\frac{x^{2-p}}{(1-p)(2-p)} - \frac{x\theta^{1-p}}{1-p} + \frac{\theta^{2-p}}{2-p})$

Deviance de uma Tweedie com $p \neq 0, 1, 2$, i.e., não caímos nos casos particulares: Normal, $p = 0$, Poisson, $p = 1$, e Gama, $p = 2$.

<r code>
```r
f1d <- function(theta, x, p) {
    2 * (x^(2 - p)/((1 - p) * (2 - p)) -
         x * theta^(1 - p)/(1 - p) +
         theta^(2 - p)/(2 - p))
}
```

```r
grid.arrange(
    fnplot(f1d, theta_seq = seq(10), x = 2, p = 3, extra = "p = 3"),
    fnplot(f1d, theta_seq = seq(10), x = 2, p = 5, extra = "p = 5"),
    ncol = 2)
```



Na dist. Tweedie θ é o parâmetro de média, e p é a potência aplicada na função de variância que indica em qual distribuição *caímos*.

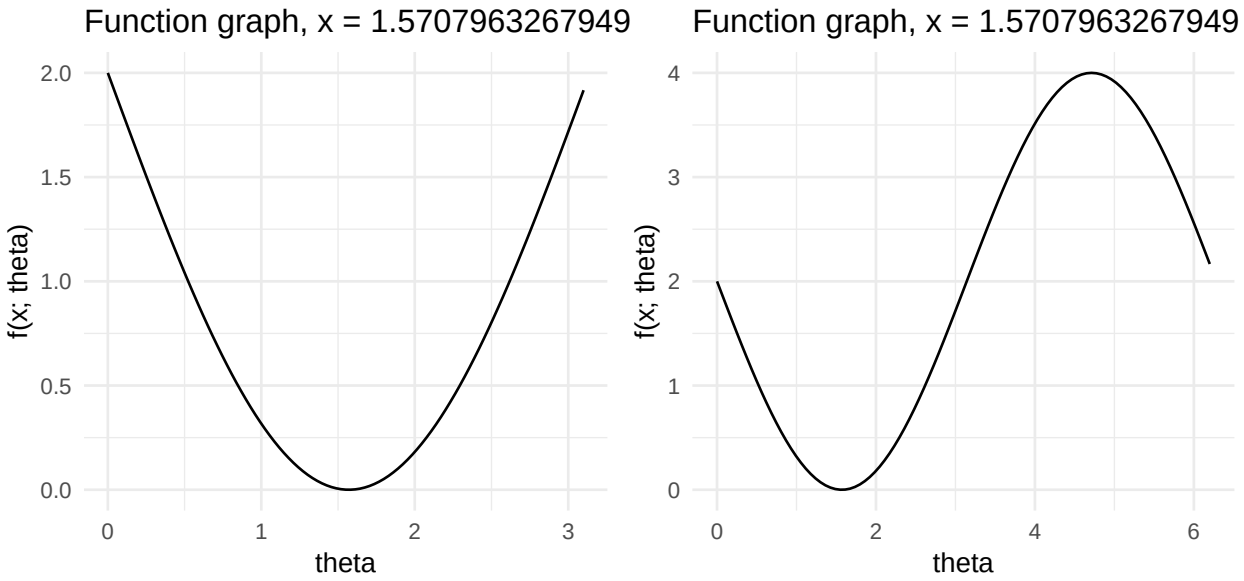
Em $p = 3$ temos uma dist. Normal Inversa, assim, a maior massa da função está ao redor de x. Com $p > 3$ temos uma dist. *Positive stable*, que é caracterizada por possuir uma cauda super pesada (*as p increases*). Tal cauda começa a se formar no ponto x estipulado.

e. $f(x;\theta) = 2(1 - \cos(x - \theta))$

Deviance de uma von Mises (2d, no círculo):

<r code>

```r
f1e <- function(theta, x) { 2 * (1 - cos(x - theta)) }

grid.arrange(fnplot(f1e, theta_seq = seq(0, pi, .1), x = pi/2),
             fnplot(f1e, theta_seq = seq(0, 2 * pi, .1), x = pi/2),
             ncol = 2)
```



Na dist. von Mises temos $0 \leq x < 2\pi$ e $\theta \in [0, 2\pi)$. Sua deviance é caracterizada por seu comportamento cíclico, dado que se trata de uma dist. para dados direcionais (medidas angulares ou direções geográficas). Como observado nas figuras, o parâmetro θ corresponde a média da dist.

2. Calcule a derivada analitica e numericamente das seguintes funções:

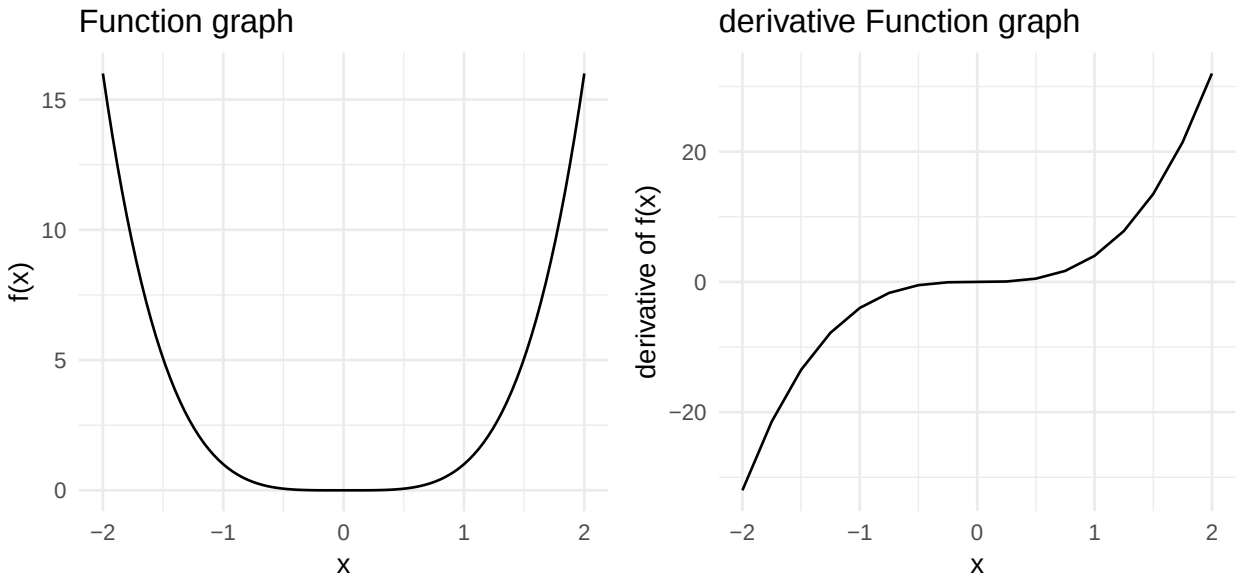
As derivadas analiticas foram feitas no papel e estão presentes no final desse exercício.

<r code>

```r
fn2plot <- function(fn, x_seq) {
    ggplot(data.frame(x = x_seq), aes(x = x)) +
        theme_minimal() +
        stat_function(fun = fn) +
        labs(y = "f(x)", title = "Function graph")
}

library(numDeriv)

fnlplot <- function(fn, x_seq) {
    fnl <- grad(fn, x_seq)
    ggplot(data.frame(x = x_seq, y = fnl), aes(x = x, y = y)) +
        theme_minimal() +
        geom_line() +
        labs(y = "derivative of f(x)", title = "derivative Function graph")
}
```

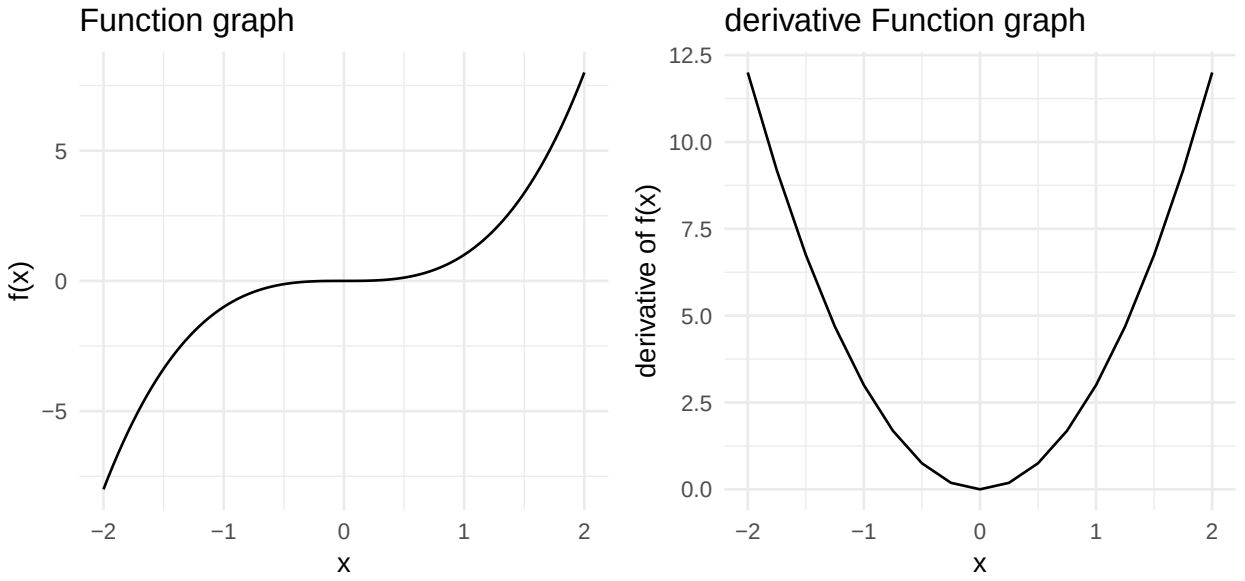
a. $f(x) = x^4$

```r
f2a <- function(x) { x^4 }

x <- seq(-2, 2, .25)
grid.arrange(fn2plot(f2a, x), fnlplot(f2a, x), ncol = 2)
```
<r code>



b. $f(x) = x^3$

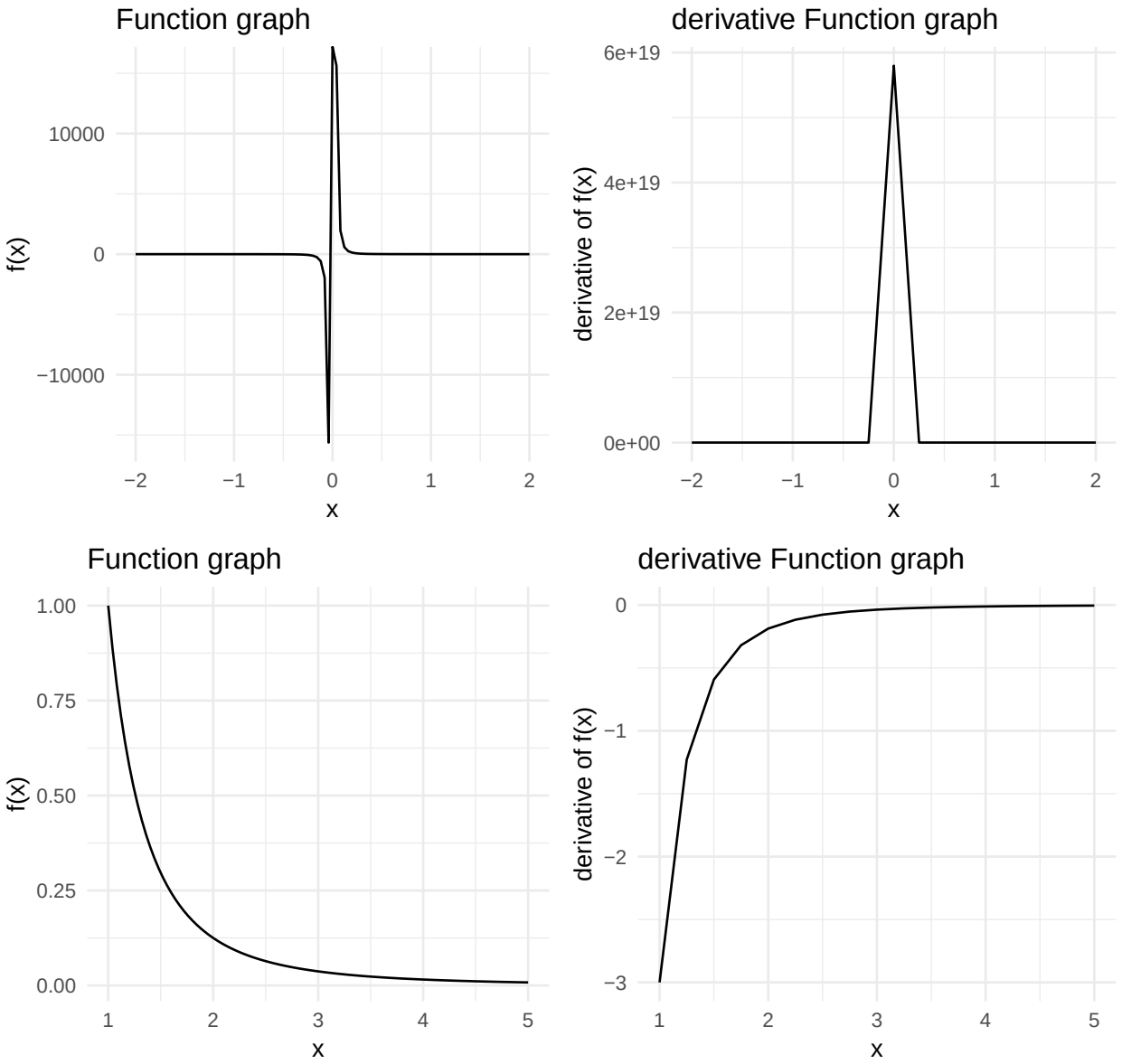
```r
f2b <- function(x) { x^3 }

grid.arrange(fn2plot(f2b, x), fnlplot(f2b, x), ncol = 2)
```
<r code>



c. $f(x) = x^{-3}$

```r
f2c <- function(x) { x^(-3) }

xp <- seq(1, 5, .25)
grid.arrange(fn2plot(f2c, x), fnlplot(f2c, x),
             fn2plot(f2c, xp), fnlplot(f2c, xp), ncol = 2, nrow = 2)
```

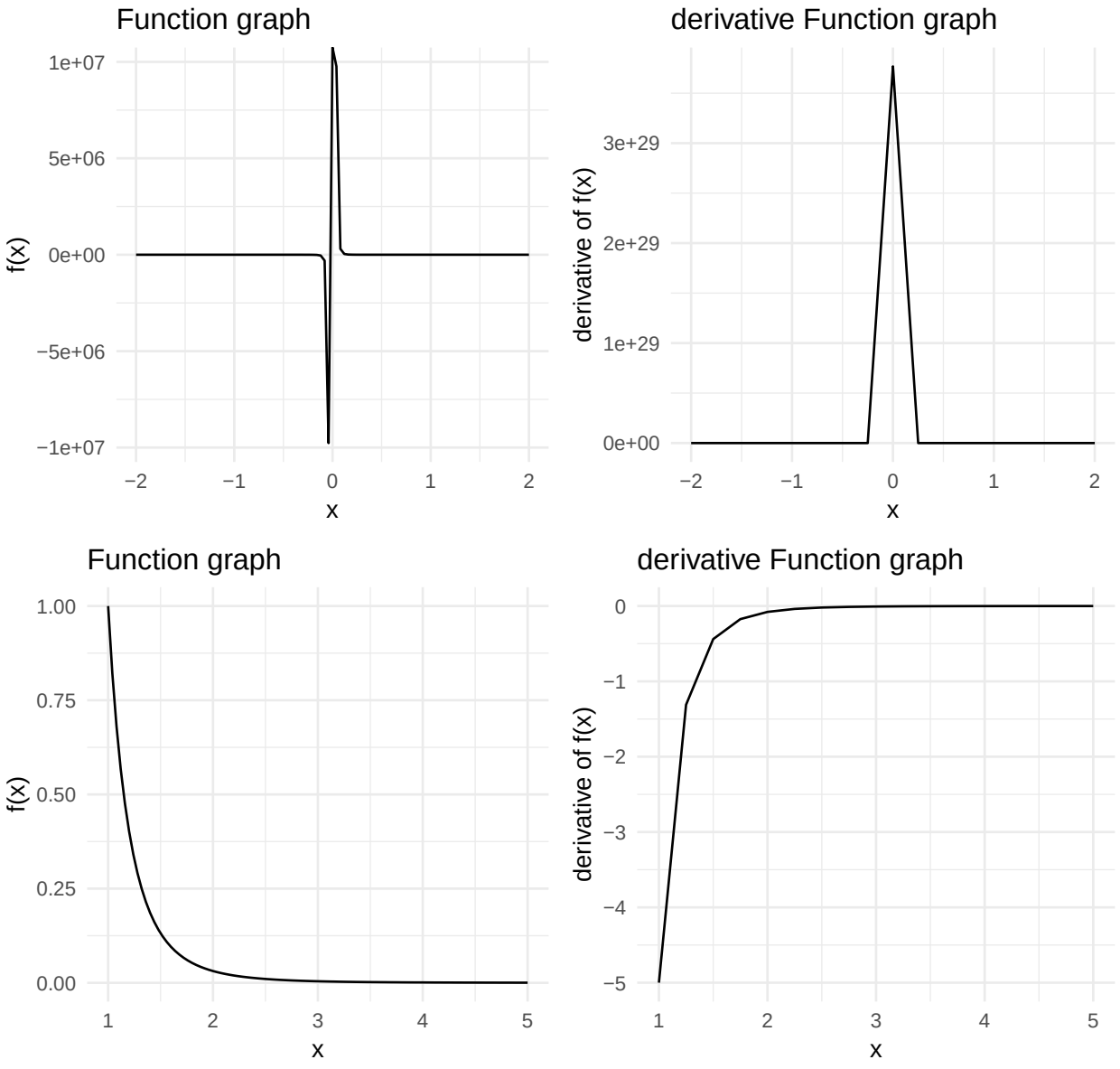
<r code>



d. $f(x) = \frac{1}{x^5}$

```r
f2d <- function(x) { 1/x^5 }
```
<r code>

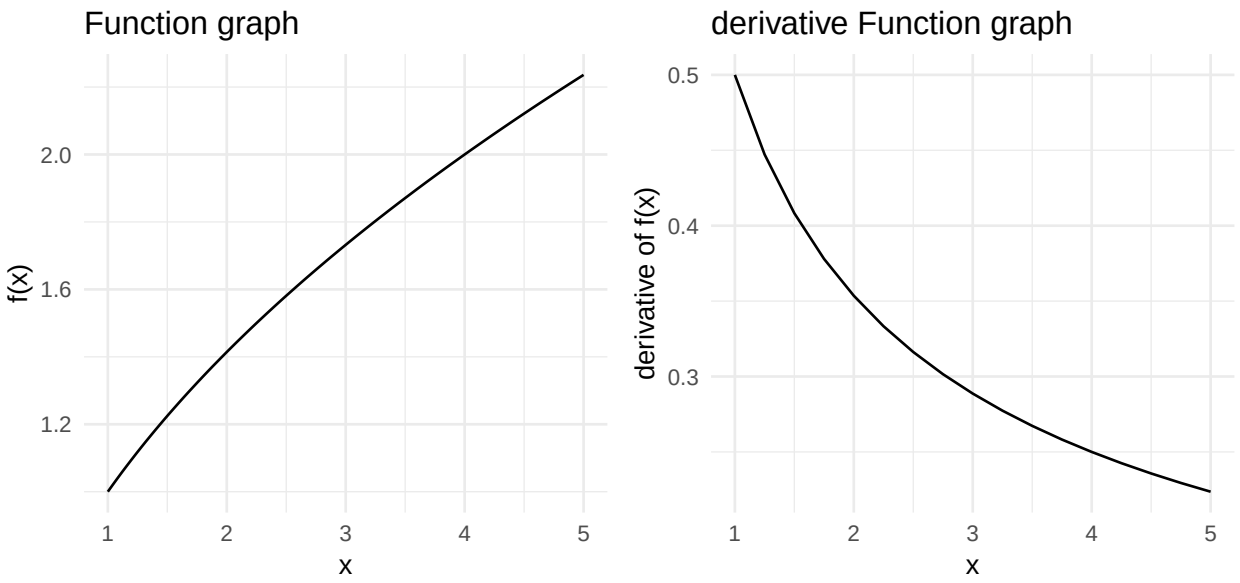
```r
grid.arrange(fn2plot(f2d, x), fnlplot(f2d, x),
             fn2plot(f2d, xp), fnlplot(f2d, xp), ncol = 2, nrow = 2)
```



e. $f(x) = \sqrt{x}$

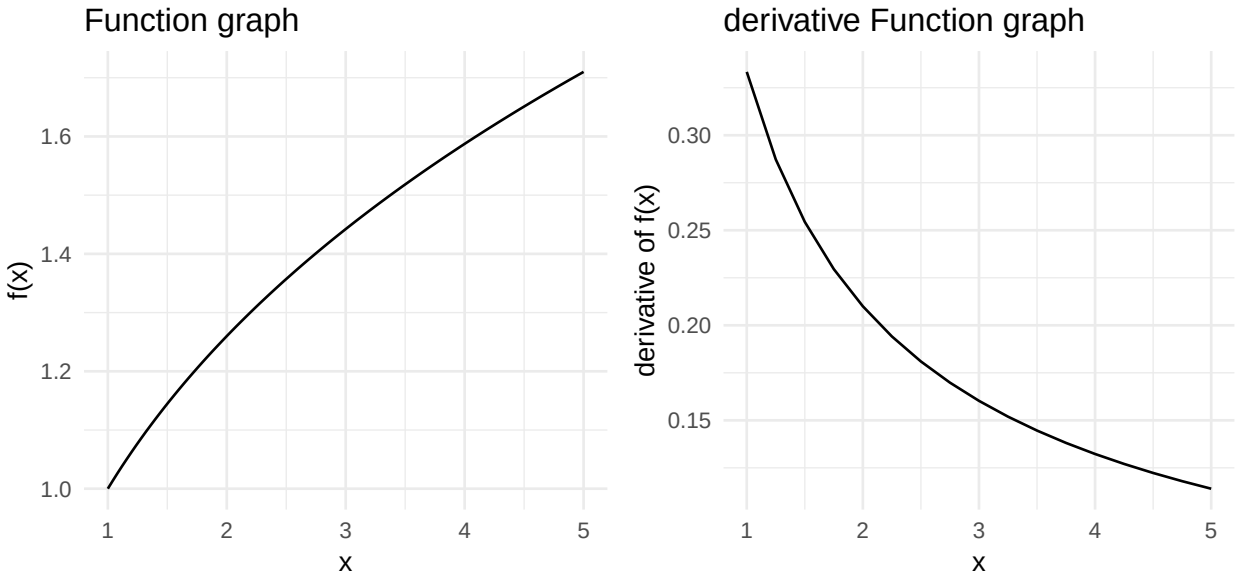
```r
f2e <- function(x) { sqrt(x) }
```
<r code>

```r
grid.arrange(fn2plot(f2e, xp), fnlplot(f2e, xp), ncol = 2)
```



f. $f(x) = \sqrt[3]{x}$

<r code>
```r
f2f <- function(x) { x^(1/3) }

grid.arrange(fn2plot(f2f, xp), fnlplot(f2f, xp), ncol = 2)
```



g. $f(x) = x^{1/3}$

Temos aqui exatamente a mesma função que na letra **f.**

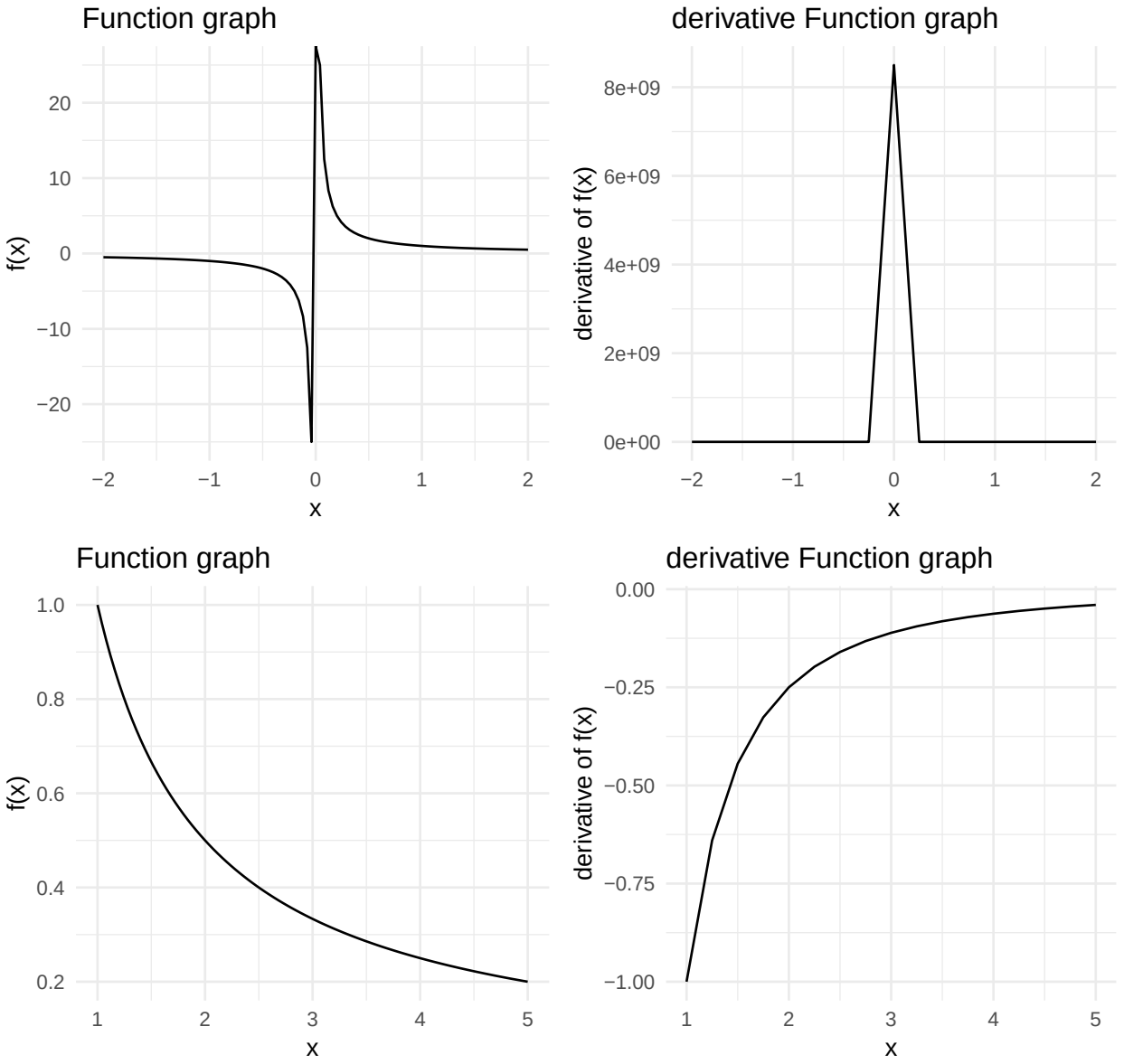
<r code>

```
f2g <- f2f
```

h. $f(x) = \frac{1}{x}$

<r code>

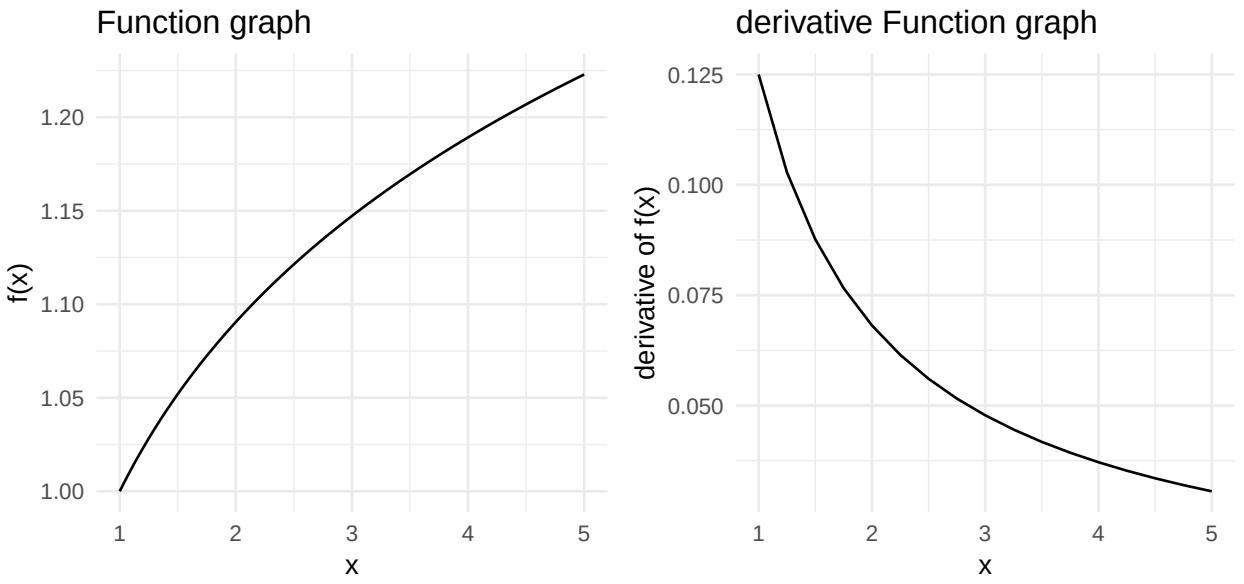
```
f2h <- function(x) { 1/x }

grid.arrange(fn2plot(f2h, x), fnlplot(f2h, x),
             fn2plot(f2h, xp), fnlplot(f2h, xp), ncol = 2, nrow = 2)
```



i. $f(x) = \sqrt[8]{x}$

```r
f2i <- function(x) { x^(1/8) }

grid.arrange(fn2plot(f2i, xp), fnlplot(f2i, xp), ncol = 2)
```

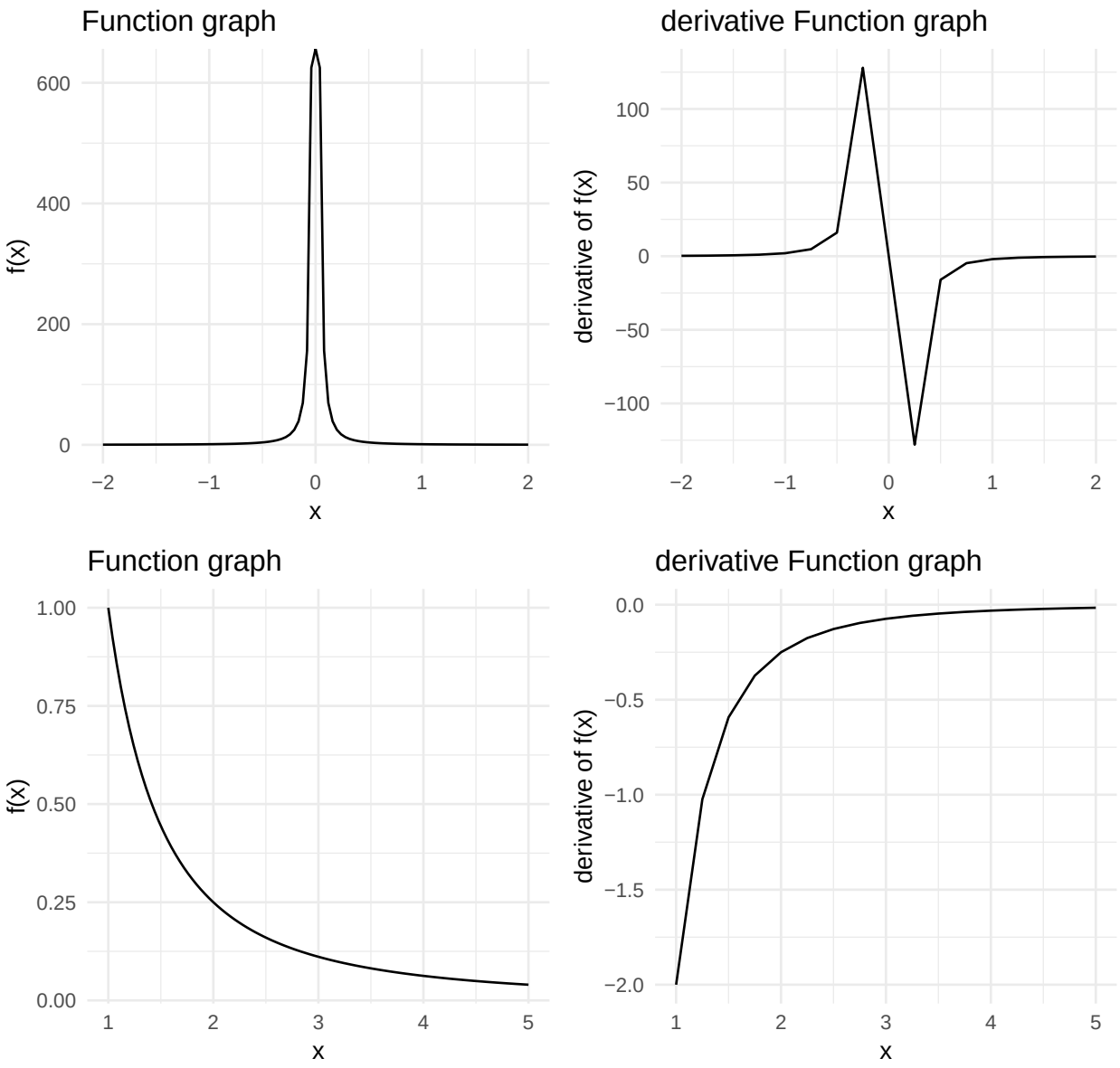
<r code>



j. $f(x) = \frac{1}{x^2}$

```r
f2j <- function(x) { 1/x^2 }

grid.arrange(fn2plot(f2j, x), fnlplot(f2j, x),
             fn2plot(f2j, xp), fnlplot(f2j, xp), ncol = 2, nrow = 2)
```

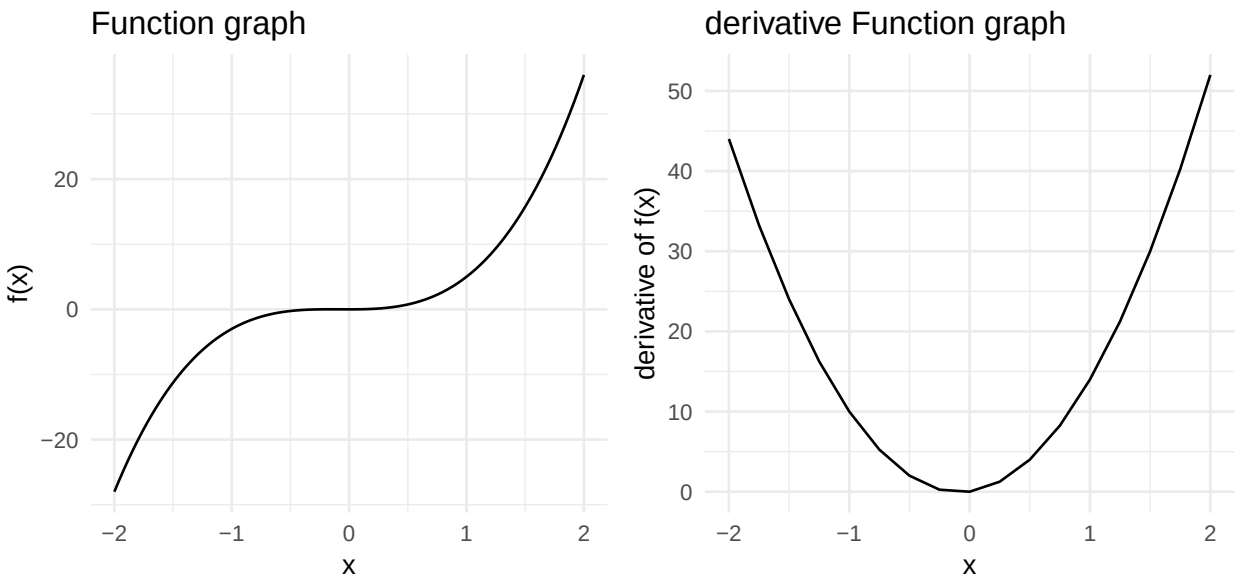
<r code>



k. $f(x) = 4x^3 + x^2$

```r
f2k <- function(x) { 4 * x^3 + x^2 }

grid.arrange(fn2plot(f2k, x), fnlplot(f2k, x), ncol = 2)
```

<r code>

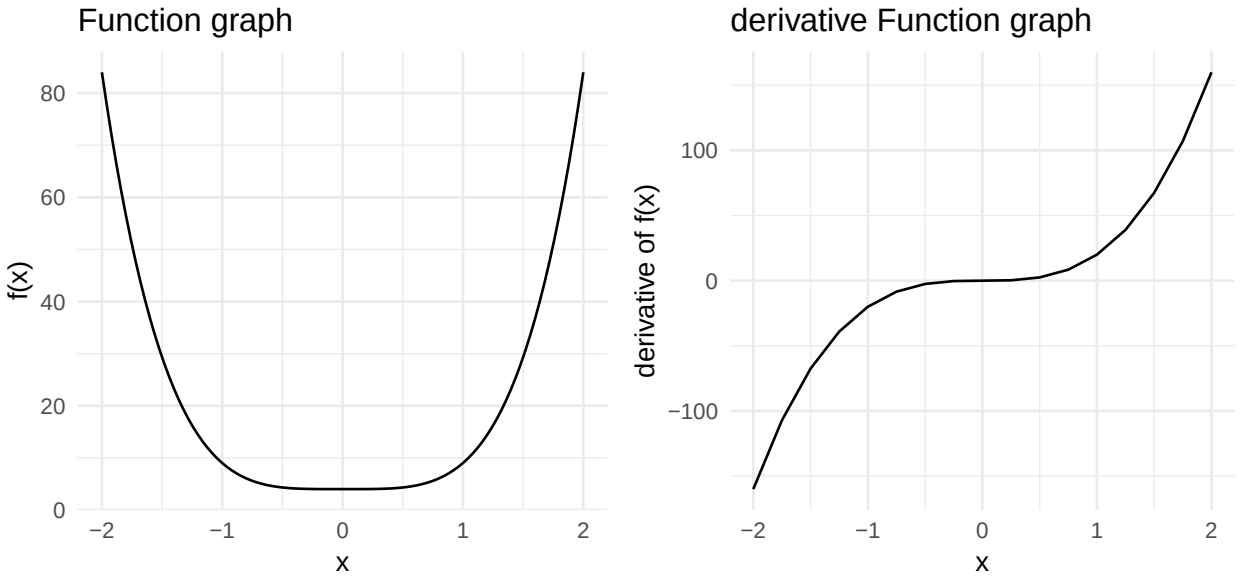


l. $f(x) = 5x^4 + 4$

```r
f2l <- function(x) { 5 * x^4 + 4 }
```

<r code>

```r
grid.arrange(fn2plot(f2l, x), fnlplot(f2l, x), ncol = 2)
```

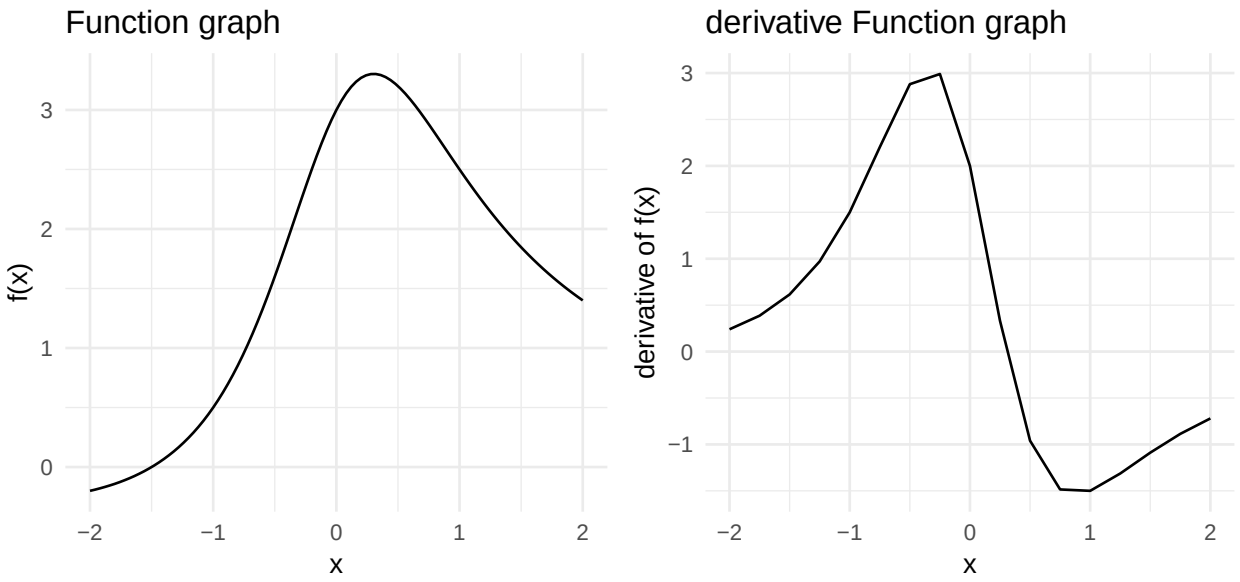


m. $f(x) = \frac{2x+3}{x^2+1}$

```r
f2m <- function(x) { (2 * x + 3)/(x^2 + 1) }
```

<r code>

```r
grid.arrange(fn2plot(f2m, x), fnlplot(f2m, x), ncol = 2)
```

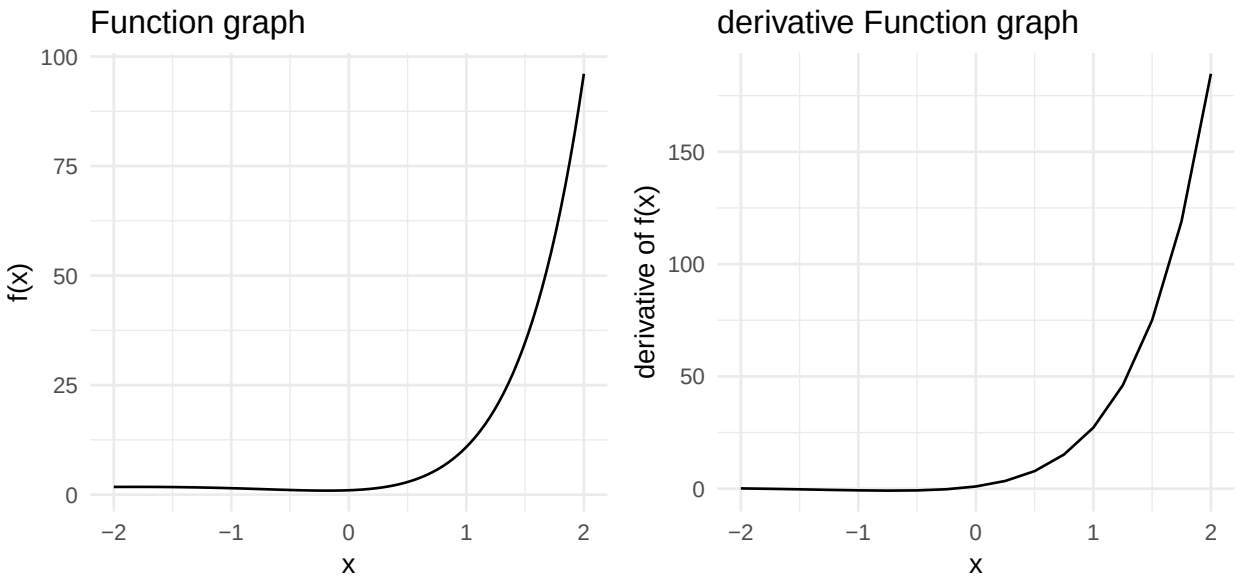


n. $f(x) = (3x^2 + 1)e^x$

```r
f2n <- function(x) { (3 * x^2 + 1) * exp(x) }
```

<r code>

```r
grid.arrange(fn2plot(f2n, x), fnlplot(f2n, x), ncol = 2)
```



o. $f(x) = \sqrt[3]{x}$

Repetida, igual a letra **f.** (e por consequencia, a letra **g.** também)

<r code>

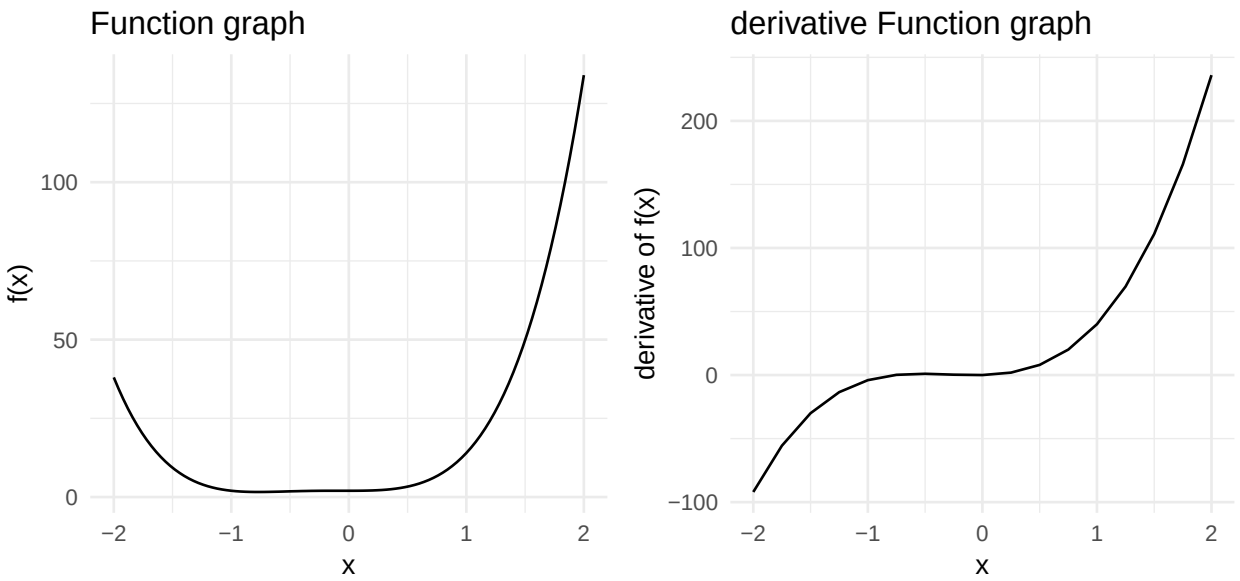
```
f2o <- f2f
```

p. $f(x) = 5x^4 + 6x^3 + x^2 + 2$

<r code>

```
f2p <- function(x) { 5 * x^4 + 6 * x^3 + x^2 + 2 }
```

```
grid.arrange(fn2plot(f2p, x), fnlplot(f2p, x), ncol = 2)
```

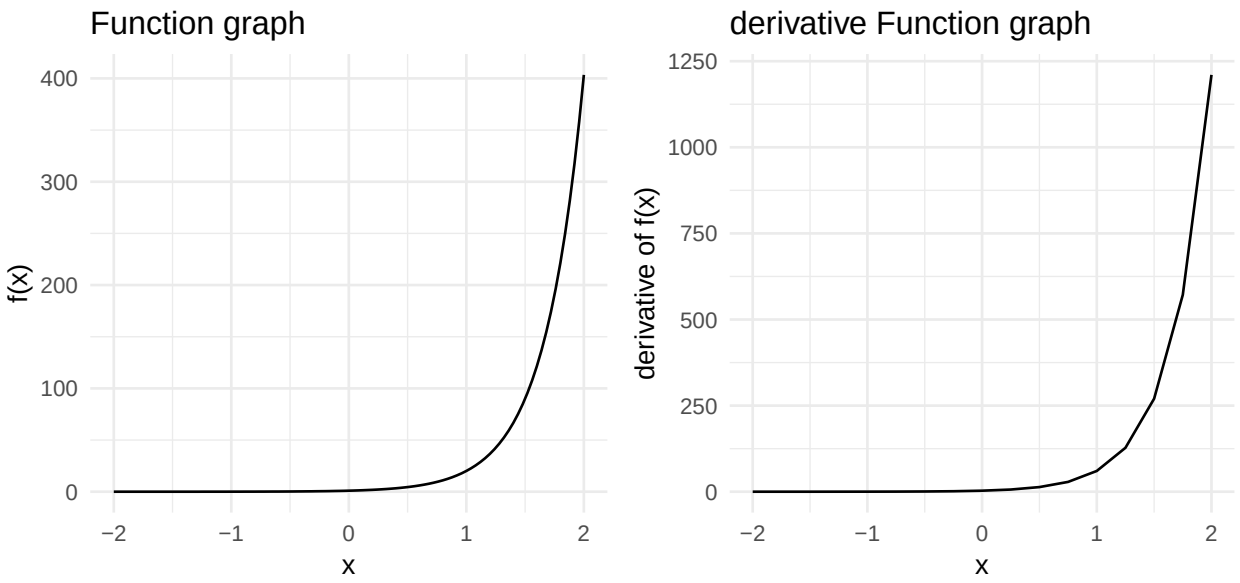


q. $f(x) = e^{3x}$

<r code>

```
f2q <- function(x) { exp(3 * x) }
```

```
grid.arrange(fn2plot(f2q, x), fnlplot(f2q, x), ncol = 2)
```

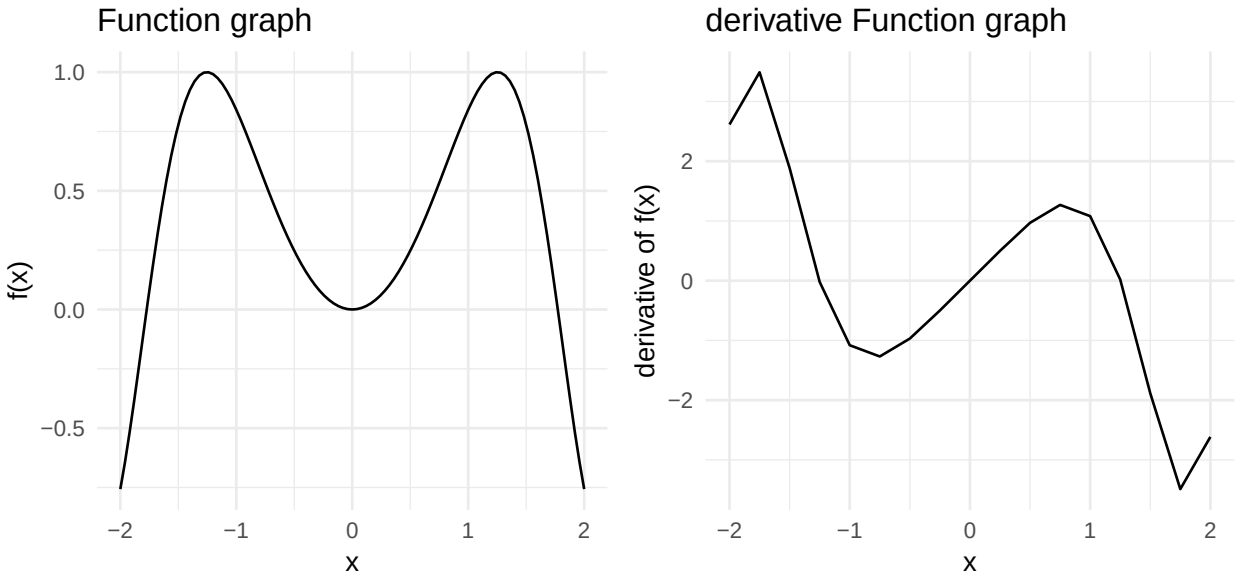


r. $f(x) = \sin x^2$

```r
f2r <- function(x) { sin(x^2) }
```

<r code>

```r
grid.arrange(fn2plot(f2r, x), fnlplot(f2r, x), ncol = 2)
```

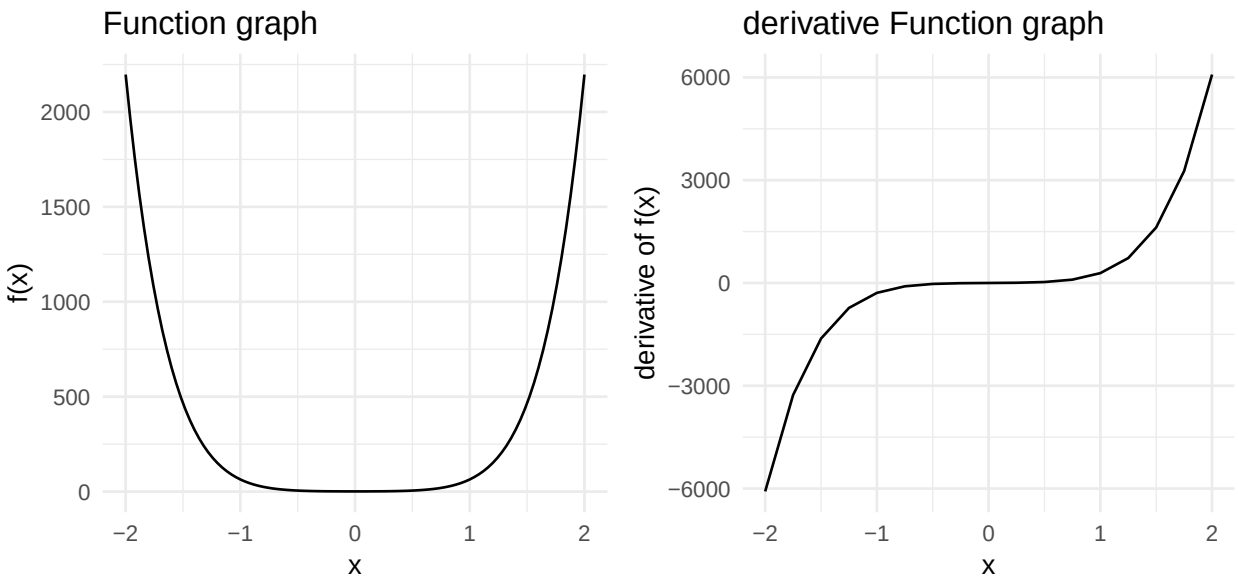


s. $f(x) = (3x^2 + 1)^3$

```r
f2s <- function(x) { (3 * x^2 + 1)^3 }
```

<r code>

```
grid.arrange(fn2plot(f2s, x), fnlplot(f2s, x), ncol = 2)
```

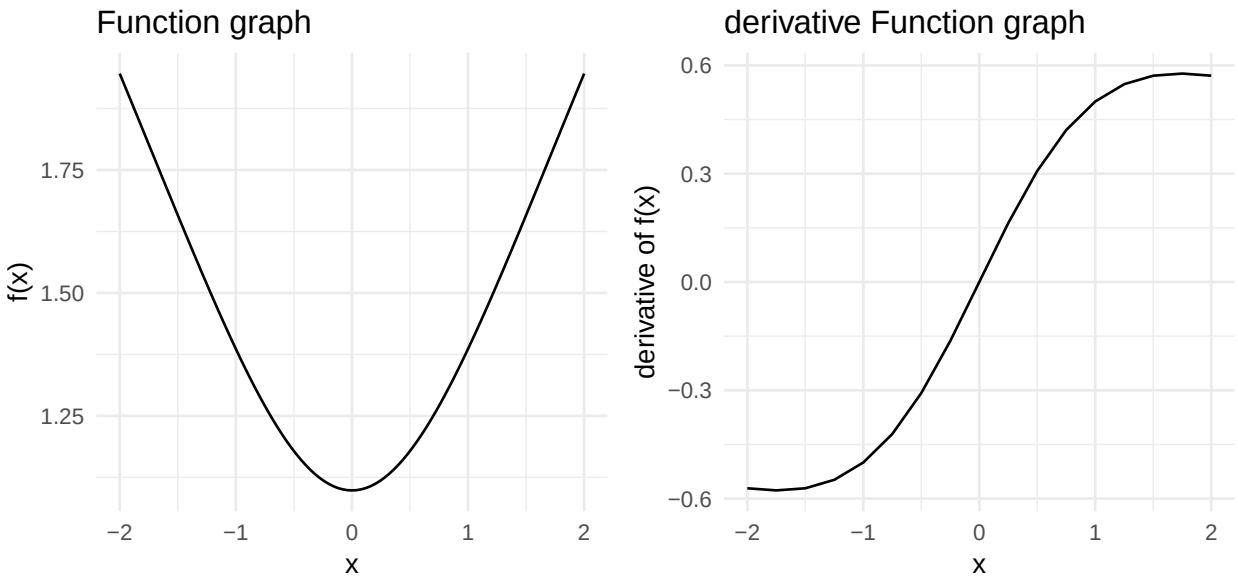


t. $f(x) = \log(x^2 + 3)$

```
f2t <- function(x) { log(x^2 + 3) }
```

<r code>

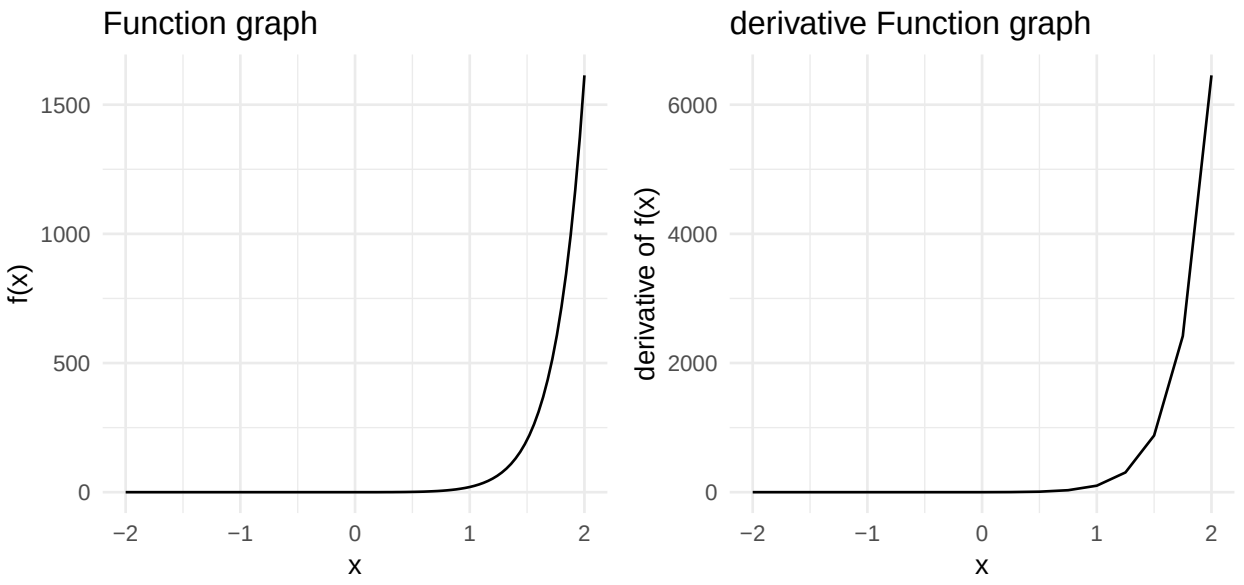
```
grid.arrange(fn2plot(f2t, x), fnlplot(f2t, x), ncol = 2)
```



u. $f(x) = x^2 e^{3x}$

```r
f2u <- function(x) { x^2 * exp(3 * x) }
```
<r code>

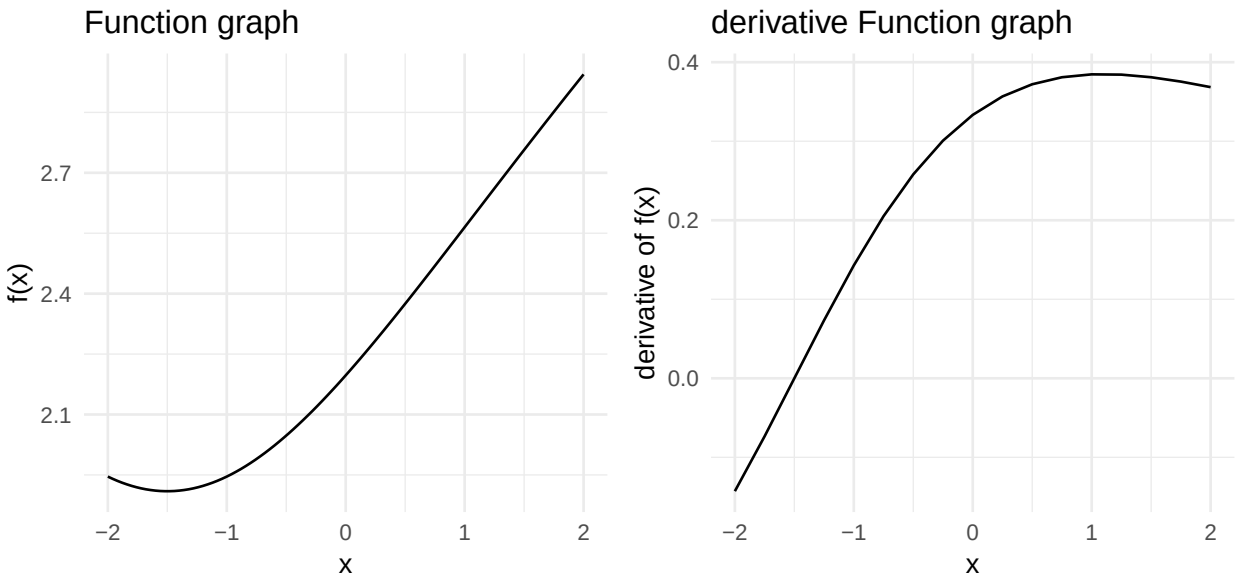
```r
grid.arrange(fn2plot(f2u, x), fnlplot(f2u, x), ncol = 2)
```



v. $f(x) = \log(x^2 + 3x + 9)$

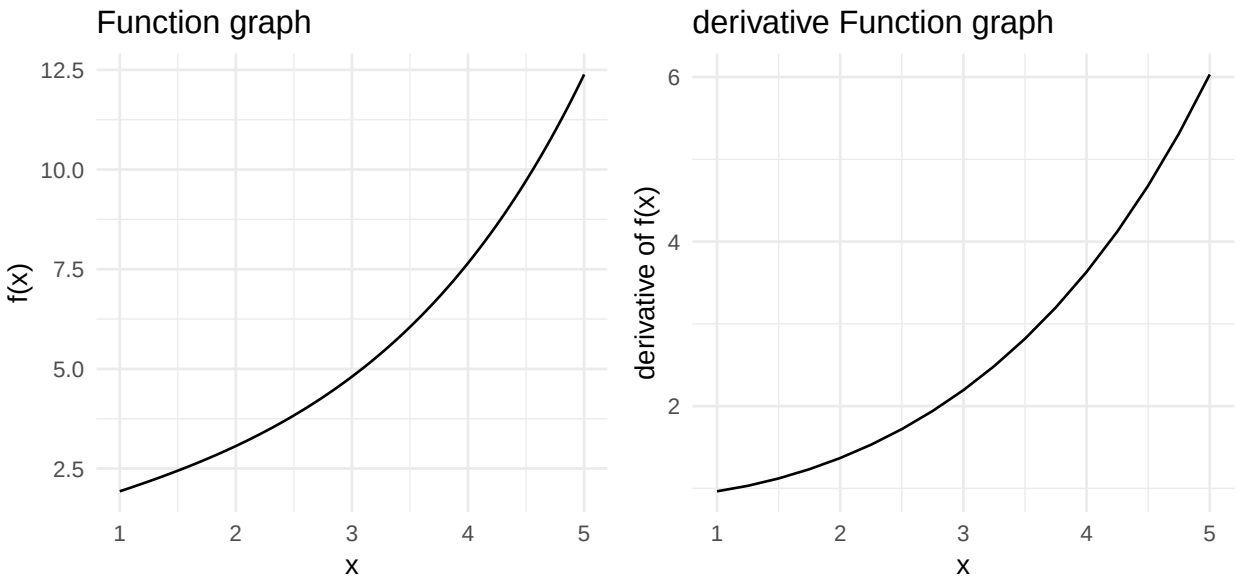
```r
f2v <- function(x) { log(x^2 + 3 * x + 9) }
```
<r code>

```r
grid.arrange(fn2plot(f2v, x), fnlplot(f2v, x), ncol = 2)
```



w. $f(x) = \sqrt{x + e^x}$

```r
f2w <- function(x) { sqrt(x + exp(x)) }

grid.arrange(fn2plot(f2w, xp), fnlplot(f2w, xp), ncol = 2)
```
<r code>



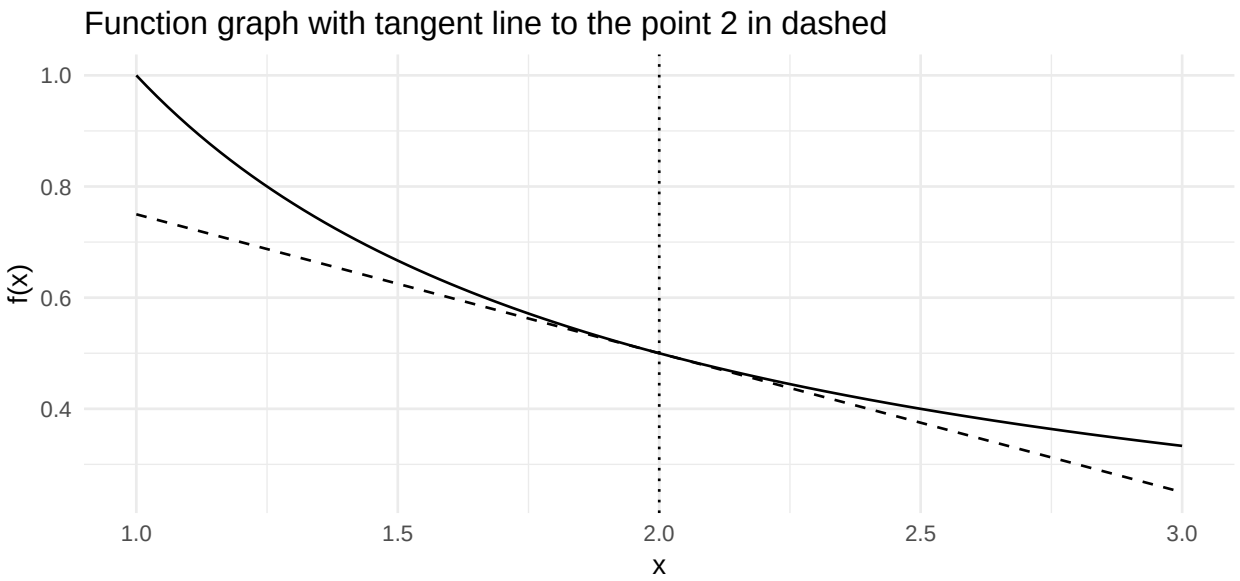
3. Determine a reta tangente ao gráfico de $f(x)$ no ponto requisitado e esboce o gráfico

```r
tangent <- function(fn, point, x_seq) {
    fnlp <- grad(fn, point)
    tang <- fn(point) + fnlp * (x_seq - point)
    main <- paste("Function graph with tangent line to the point",
                  point, "in dashed")
    ggplot(data.frame(x = x_seq), aes(x = x)) +
        theme_minimal() +
        stat_function(fun = fn) +
        geom_line(aes(y = tang), linetype = "dashed") +
        geom_vline(xintercept = point, linetype = "dotted") +
        labs(y = "f(x)", title = main)
}
```
<r code>

a. $f(x) = \frac{1}{x}$ **no ponto de abscissa 2.**

```r
f3a <- function(x) { 1/x }

tangent(f3a, point = 2, x_seq = seq(3))
```

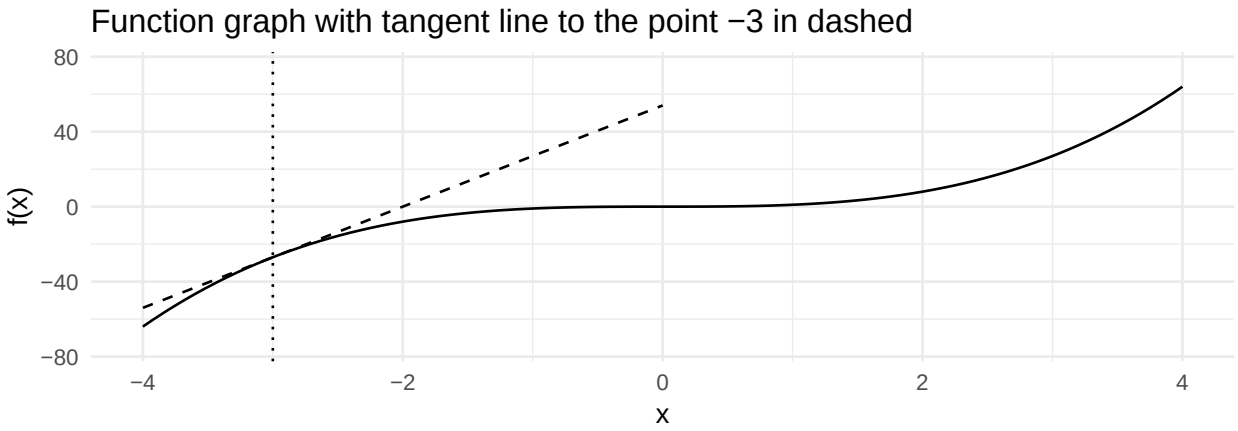
<r code>

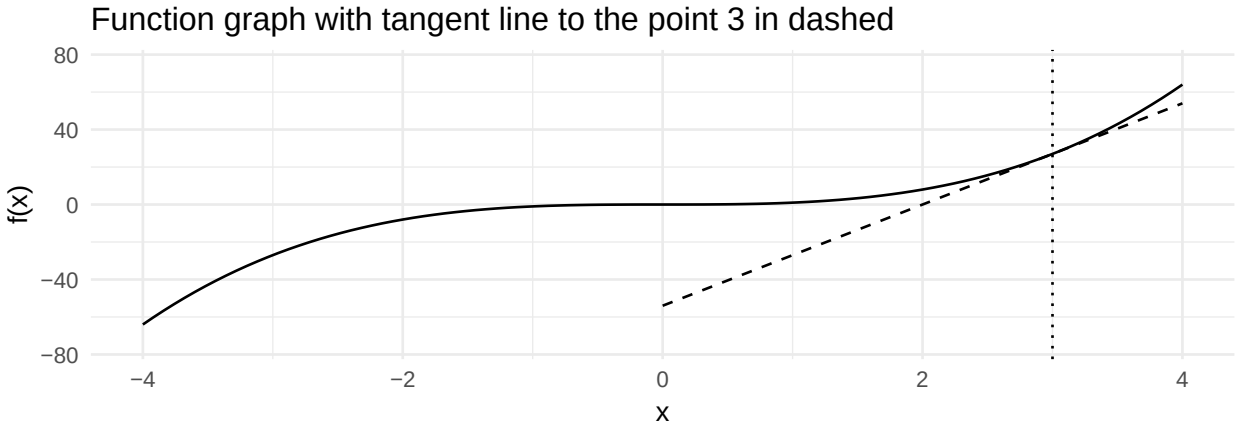


b. $f(x) = x^3$ **nos pontos de abscissa -3 e 3.**

```r
f3b <- function(x) { x^3 }

grid.arrange(tangent(f3b, point = -3, x_seq = seq(-4, 4, 1)) +
             scale_y_continuous(limits = c(-75, 75)),
             tangent(f3b, point = 3, x_seq = seq(-4, 4, 1)) +
             scale_y_continuous(limits = c(-75, 75)), nrow = 2)
```

<r code>

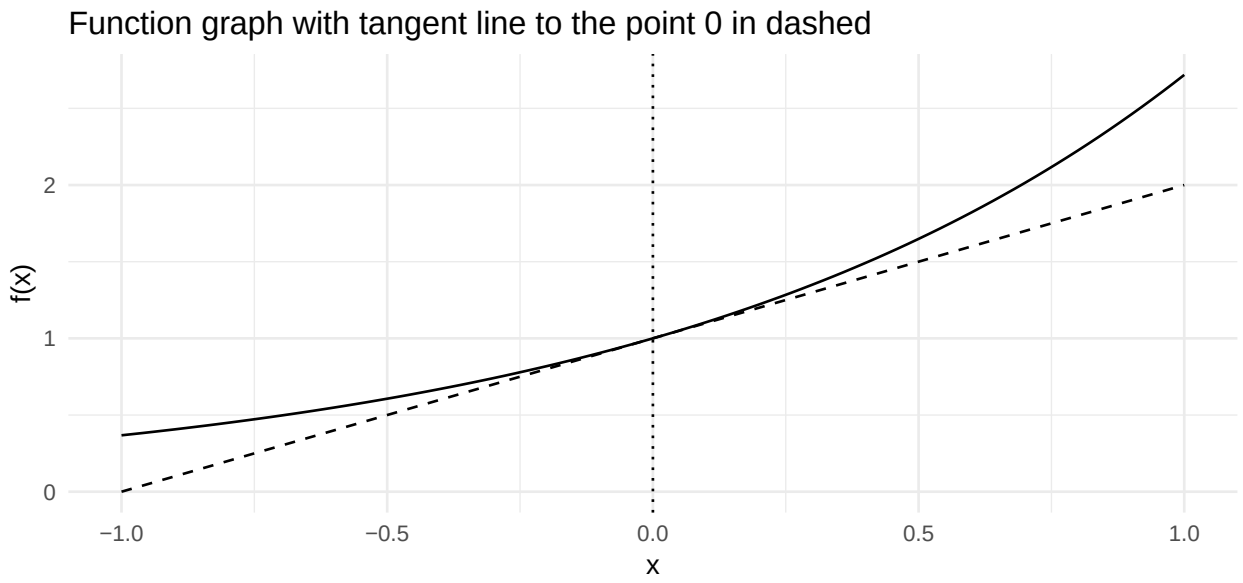




c. $f(x) = e^x$ **no ponto de abscissa 0.**

```r
f3c <- function(x) { exp(x) }

tangent(f3c, point = 0, x_seq = seq(-1, 1, 1))
```

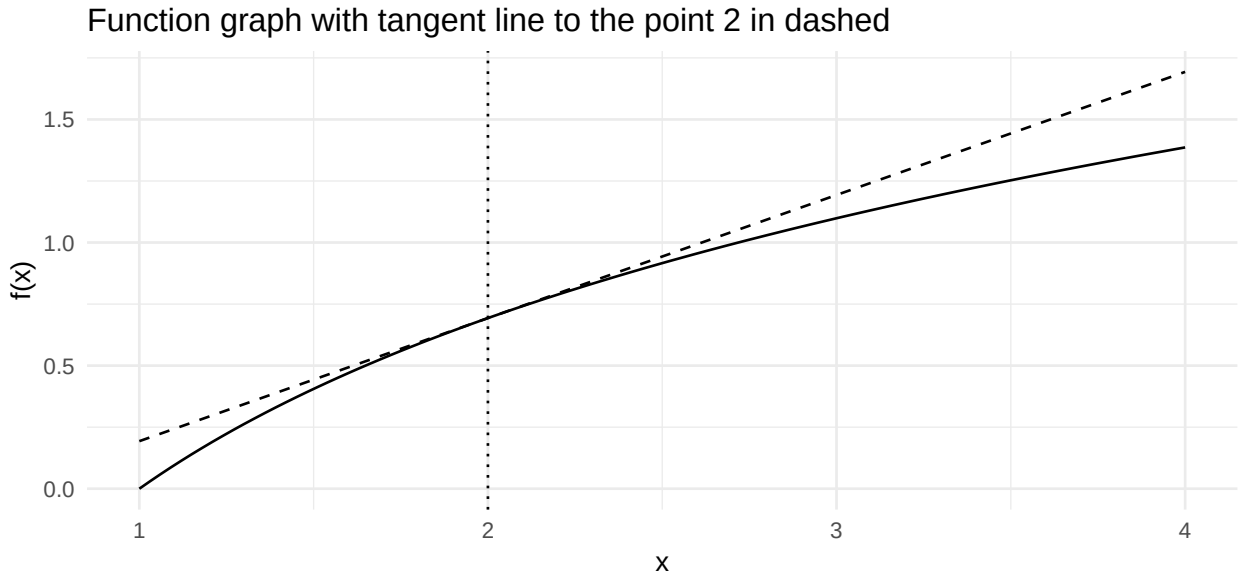
<r code>



d. $f(x) = \log x$ **no ponto de abscissa 2.**

```r
f3d <- function(x) { log(x) }

tangent(f3d, point = 2, x_seq = seq(4))
```

<r code>



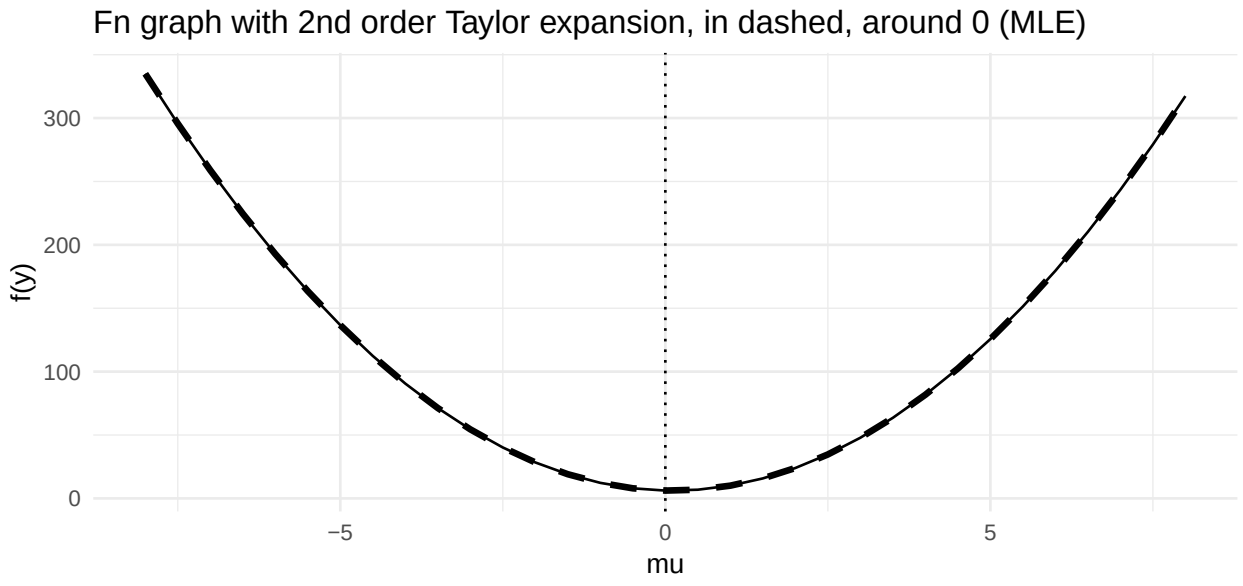
4. Aproxime as seguintes funções usando a expansão de Taylor de segunda ordem. Esboce o gráfico da função e da aproximação.

<r code>
```r
taylor <- function(fn, mu_seq, y) {
    fn_vec <- Vectorize(fn, "mu")
    fn_eval <- fn_vec(mu = mu_seq, y = y)
    point <- mu_seq[which.min(fn_eval)]
    prime <- grad(fn_vec, x = point, y = y)
    dprime <- hessian(fn_vec, x = point, y = y)
    approx <- min(fn_eval) +
        (mu_seq - point) * prime + .5 * (mu_seq - point)^2 %*% dprime
    main <- paste(
        "Fn graph with 2nd order Taylor expansion, in dashed, around",
        point, "(MLE)")
    ggplot(data.frame(mu = mu_seq, fn = fn_eval), aes(x = mu, y = fn)) +
        theme_minimal() +
        geom_line() +
        geom_line(aes(y = approx), linetype = "dashed", size = 1.25) +
        geom_vline(xintercept = point, linetype = "dotted") +
        labs(y = "f(y)", title = main)
}
```

a. $\sum_{i=1}^{n}(y_i - \mu)^2$**. Fixe** $y_i = \{2.09, -1.32, -0.20, 0.05, -0.07\}$

Deviance de uma dist. Normal:

<r code>
```r
y <- c(2.09, -1.32, - 0.20, 0.05, -0.07)

f4a <- function(mu, y) { t(y - mu) %*% (y - mu) }

taylor(f4a, mu_seq = seq(-8, 8, .5), y = y)
```

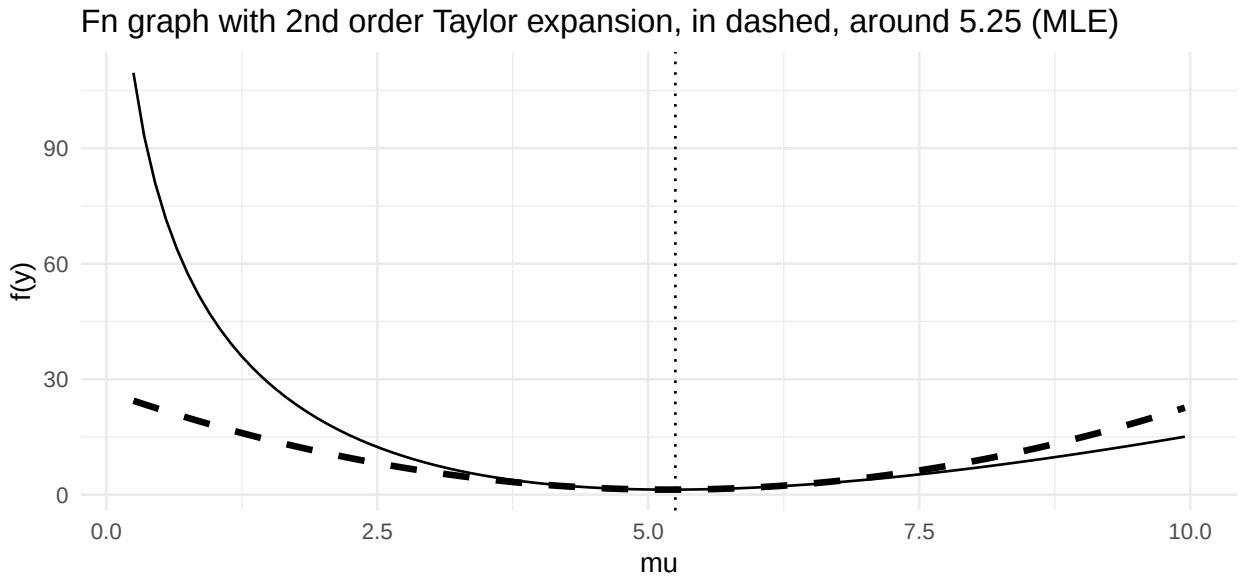


b. $\sum_{i=1}^{n} 2(y_i \log \frac{y_i}{\mu} + \mu - y_i)$**. Fixe** $y_i = \{7, 4, 4, 6, 5\}$

Deviance de uma dist. Poisson:

<r code>

```
y <- c(7, 4, 4, 6, 5)

f4b <- function(mu, y) { sum(2 * (y * log(y/mu) + mu - y)) }

taylor(f4b, mu_seq = seq(.25, 10, .1), y = y)
```

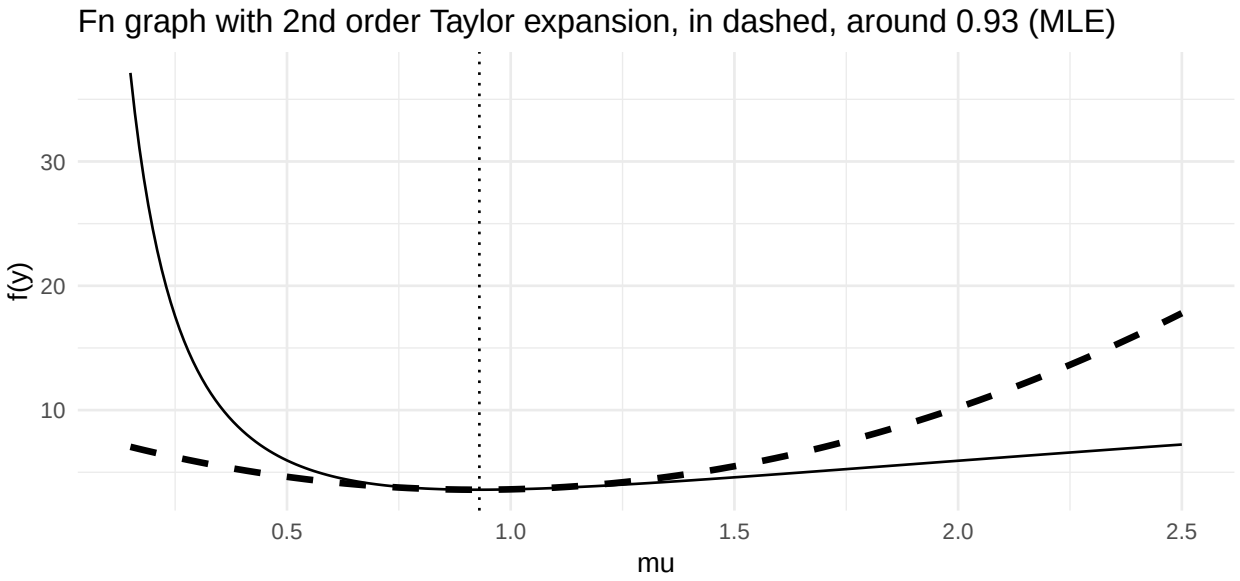


c. $\sum_{i=1}^{n} 2(\frac{y_i}{\mu} - \log \frac{y_i}{\mu} - 1)$**. Fixe** $y_i = \{2.35, 0.16, 0.56, 1.05, 0.51\}$

Deviance de uma dist. Gama:

<r code>

```
y <- c(2.35, 0.16, 0.56, 1.05, 0.51)

f4c <- function(mu, y) { sum(2 * (y/mu - log(y/mu) - 1)) }

taylor(f4c, mu_seq = seq(.15, 2.5, .01), y = y)
```

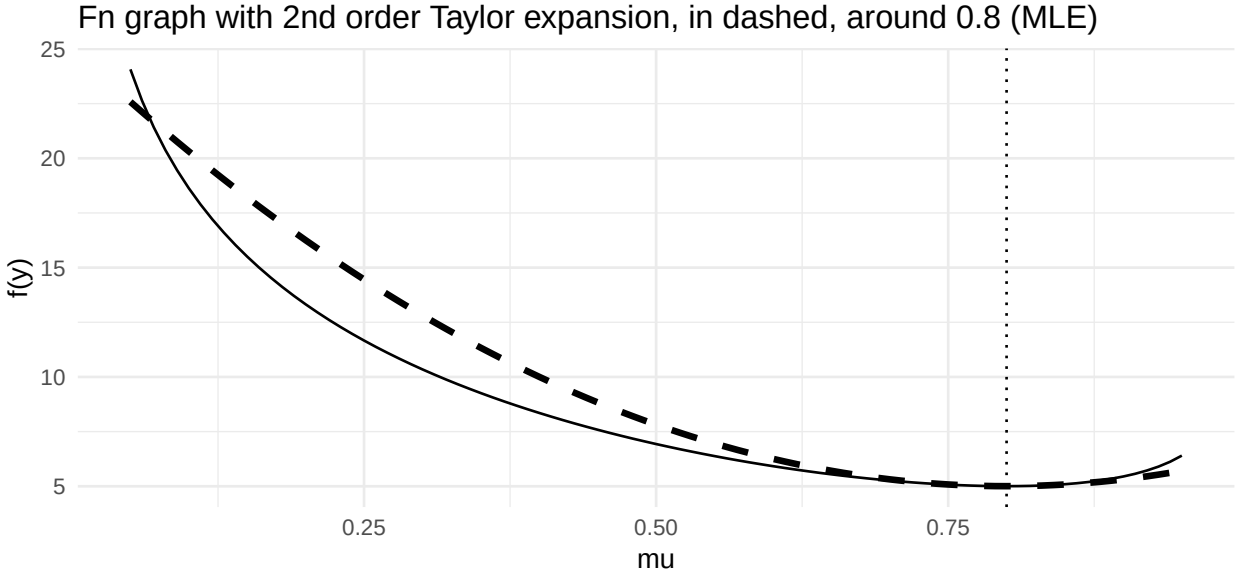


d. $\sum_{i=1}^{n} 2(y_i \log \frac{y_i}{\mu} + (1 - y_i) \log \frac{1-y_i}{1-\mu})$**. Fixe** $y_i = \{1, 0, 1, 1, 1\}$

Deviance uma dist. Binomial:

<r code>

```r
y <- c(1, 0, 1, 1, 1)

f4d <- function(mu, y) {
    sum(2 * ifelse(y == 1, -log(mu), -log(1 - mu)))
}
taylor(f4d, mu_seq = seq(.05, .95, .01), y = y)
```

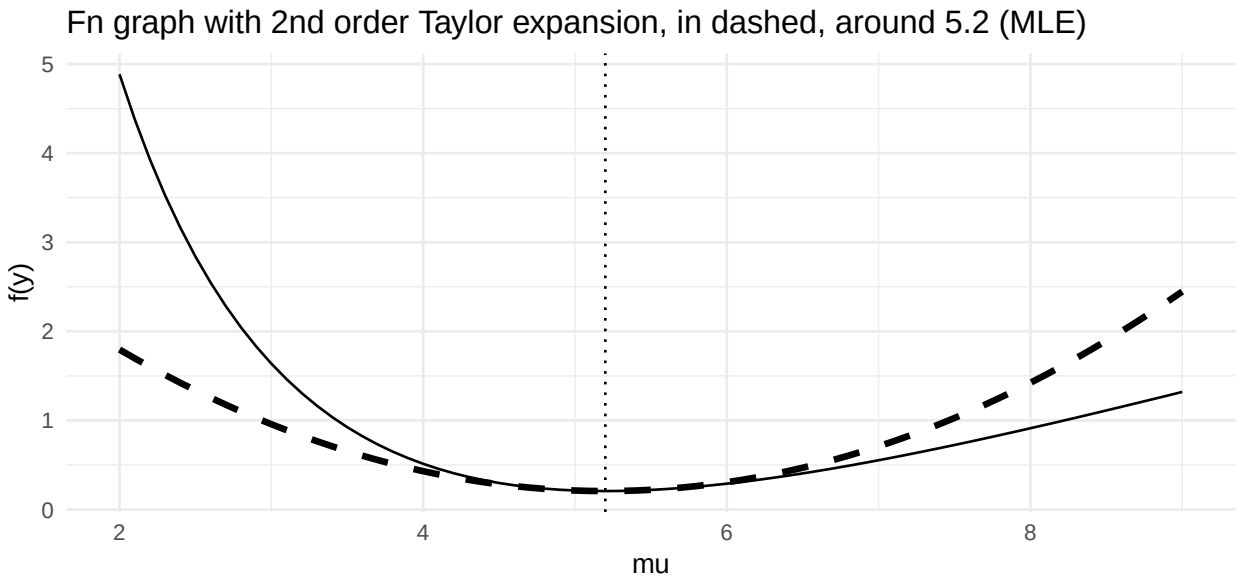


e. $\sum_{i=1}^{n} 2(y_i \log \frac{y_i}{\mu} + (m + y_i) \log \frac{m+\mu}{m+y_i})$**. Fixe** $m = 1$ **e** $y_i = \{7, 4, 4, 6, 5\}$

Deviance de uma dist. Binomial Negativa:

<r code>

```
y <- c(7, 4, 4, 6, 5)

f4e <- function(mu, y) {
    sum(2 * (y * log(y/mu) + (1 + y) * log((1 + mu)/(1 +
                                                     y))))
}
taylor(f4e, mu_seq = seq(2, 9, .1), y = y)
```



5. Considerando as funções do exercício 4 encontre a reta tangente e esboce o gráfico da função com a reta em pelo menos três pontos que julgue adequado.

<r code>

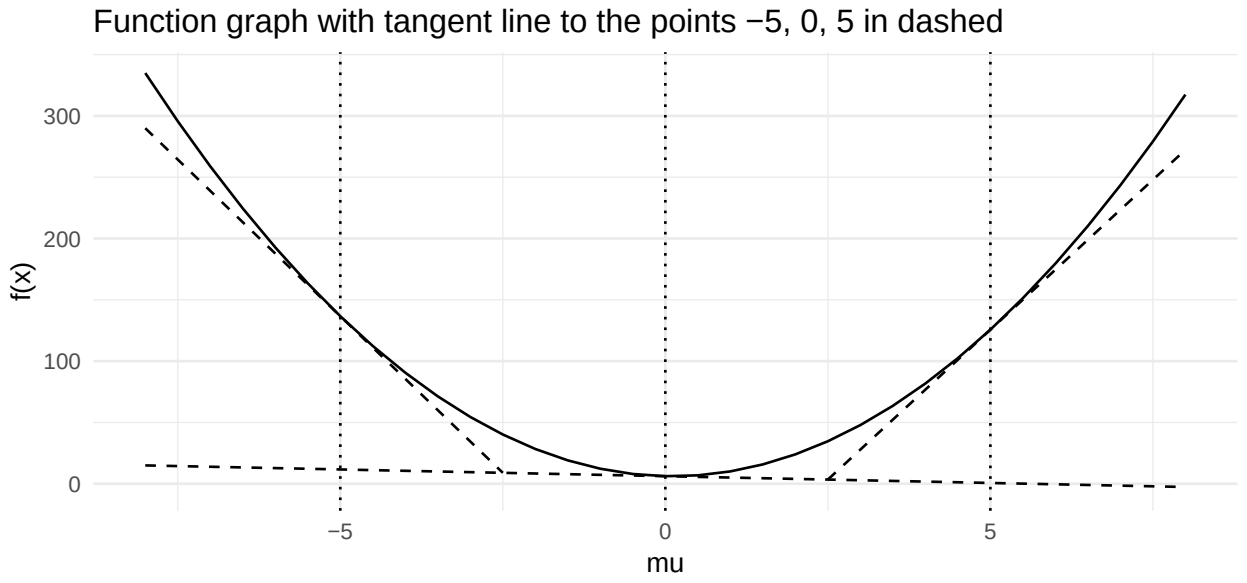
```
tangentplus <- function(fn, y, points, mu_seq, minx) {
    fn_vec <- Vectorize(fn, "mu")
    fn_eval <- fn_vec(mu = mu_seq, y = y)
    npoints <- length(points)
    primes <- sapply(points, function(i) grad(fn_vec, x = i, y = y))
    tang <- sapply(seq(npoints),
                   function(i) {
                       fn_vec(mu = points[i], y = y) +
                           primes[i] * (mu_seq - points[i])
                   })
    main <- paste("Function graph with tangent line to the points",
```

```
                paste(points, collapse = ", "), "in dashed")
    ggplot(data.frame(mu = mu_seq, fn = fn_eval), aes(x = mu, y = fn)) +
        theme_minimal() +
        geom_line() +
        ylim(minx, max(fn_eval)) +
        geom_line(aes(y = tang[ , 1]), linetype = "dashed") +
        geom_line(aes(y = tang[ , 2]), linetype = "dashed") +
        geom_line(aes(y = tang[ , 3]), linetype = "dashed") +
        geom_vline(xintercept = points, linetype = "dotted") +
        labs(y = "f(x)", title = main)
}
```

a. $\sum_{i=1}^{n}(y_i - \mu)^2$**. Fixe** $y_i = \{2.09, -1.32, -0.20, 0.05, -0.07\}$

<r code>

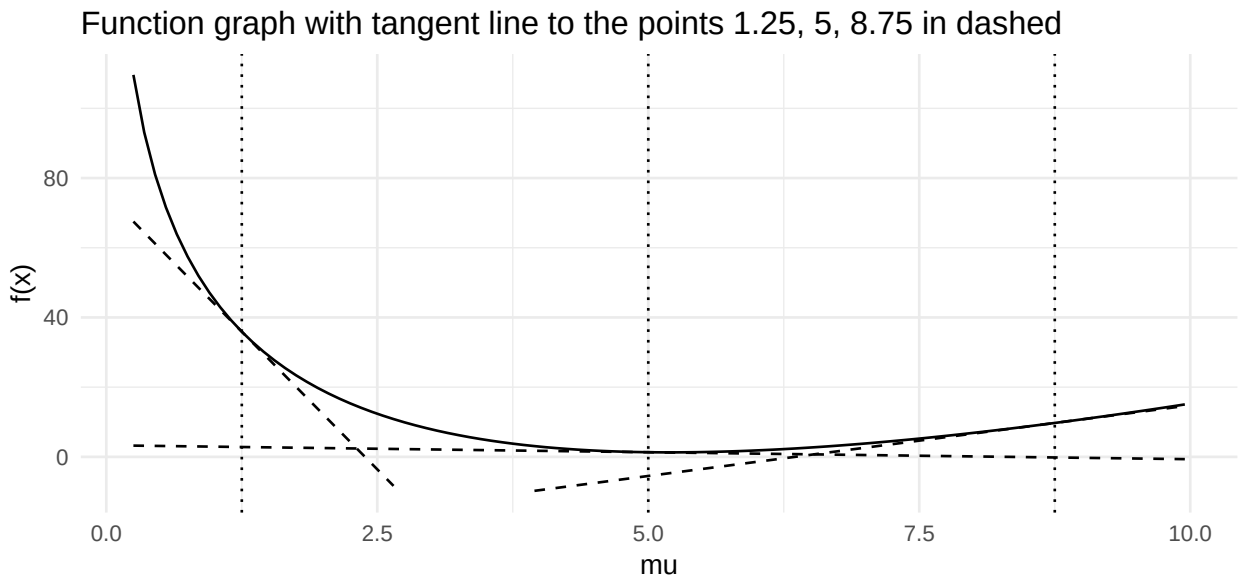
```
y <- c(2.09, -1.32, - 0.20, 0.05, -0.07)

tangentplus(f4a, y = y,
            points = c(-5, 0, 5), mu_seq = seq(-8, 8, .5), minx = -5)
```



b. $\sum_{i=1}^{n} 2(y_i \log \frac{y_i}{\mu} + \mu - y_i)$**. Fixe** $y_i = \{7, 4, 4, 6, 5\}$

<r code>

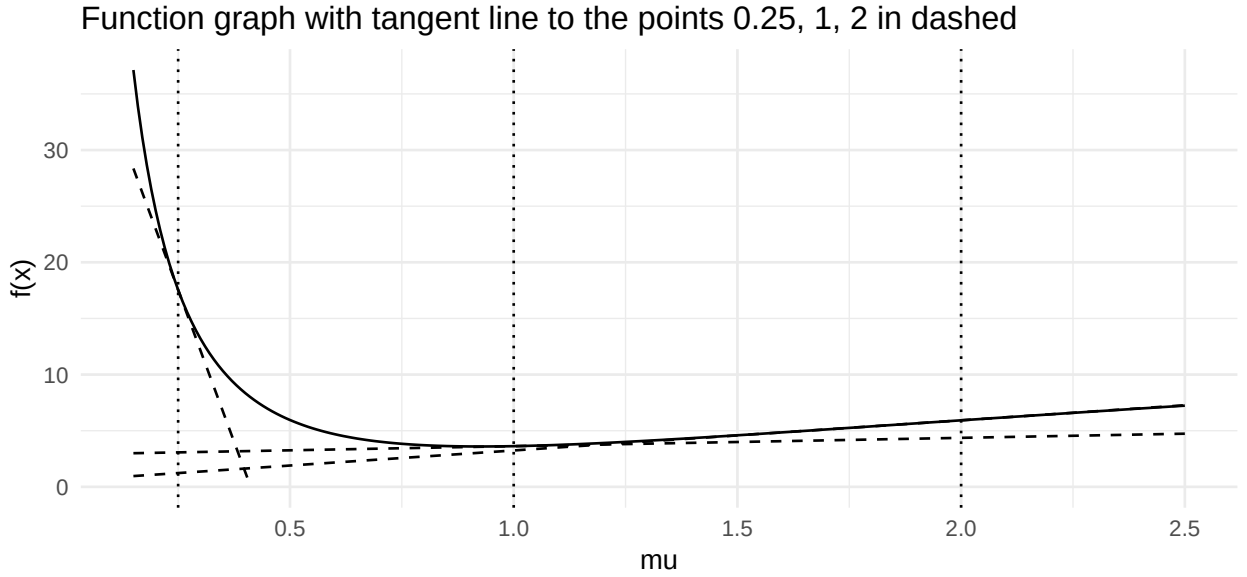
```
y <- c(7, 4, 4, 6, 5)

tangentplus(f4b, y = y, points = c(1.25, 5, 8.75),
            mu_seq = seq(.25, 10, .1), minx = -10)
```



c. $\sum_{i=1}^{n} 2(\frac{y_i}{\mu} - \log \frac{y_i}{\mu} - 1)$**. Fixe** $y_i = \{2.35, 0.16, 0.56, 1.05, 0.51\}$

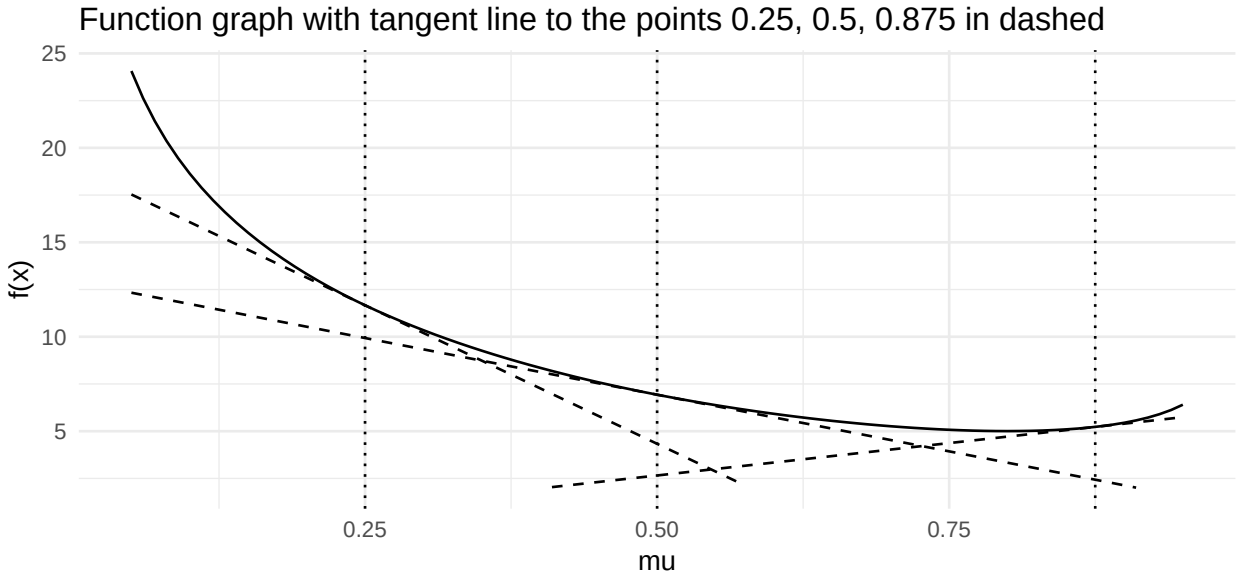
<r code>

```
y <- c(2.35, 0.16, 0.56, 1.05, 0.51)

tangentplus(f4c, y = y, points = c(.25, 1, 2),
            mu_seq = seq(.15, 2.5, .01), minx = 0)
```



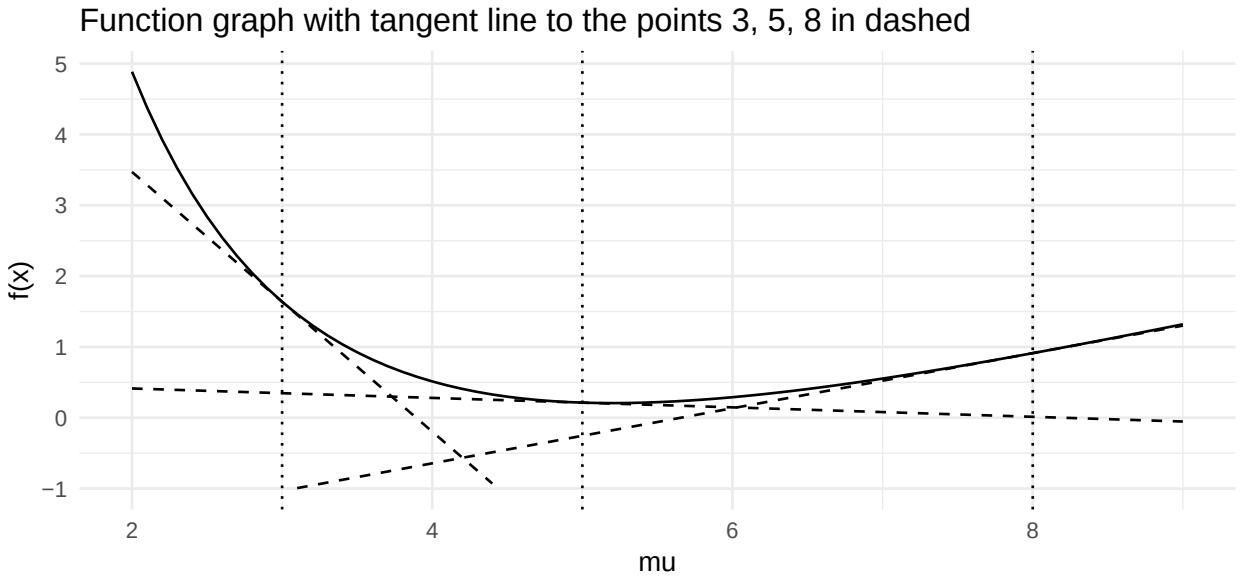
d. $\sum_{i=1}^{n} 2(y_i \log \frac{y_i}{\mu} + (1 - y_i) \log \frac{1-y_i}{1-\mu})$**. Fixe** $y_i = \{1, 0, 1, 1, 1\}$

<r code>
```r
y <- c(1, 0, 1, 1, 1)

tangentplus(f4d, y = y, points = c(.25, .5, .875),
            mu_seq = seq(.05, .95, .01), minx = 2)
```



e. $\sum_{i=1}^{n} 2(y_i \log \frac{y_i}{\mu} + (m + y_i) \log \frac{m+\mu}{m+y_i})$**. Fixe** $m = 1$ **e** $y_i = \{7, 4, 4, 6, 5\}$

<r code>
```r
y <- c(7, 4, 4, 6, 5)

tangentplus(f4e,y = y,
            points = c(3, 5, 8), mu_seq = seq(2, 9, .1), minx = -1)
```



6. Considerando as funções apresentadas no exercício 4 identifique o ponto de inflexão e verique se é um ponto de máximo ou mínimo.

Em tais funções, e nos intervalos considerados para μ, não temos nenhum ponto de inflexão, estritamente falando. i.e. não se verifica nenhuma mudança de concavidade.

Contudo, temos pontos, um para cada função, em que se observa o seguinte: até o tal ponto a função decresce, a partir do tal ponto a função cresce. Provavelmente é no tocante a esta questão que a pergunta se refere. Tais pontos são os chamados EMVs.

Para uma mais formal verificação precisamos checar duas coisas: (1.) a primeira derivada avaliada nestes pontos é zero, e (2.) a segunda derivada avaliada nestes pontos é positiva (caracterizando tais pontos como de mínimo).

a. $\sum_{i=1}^{n}(y_i - \mu)^2$**. Fixe** $y_i = \{2.09, -1.32, -0.20, 0.05, -0.07\}$

<r code>
```
y <- c(2.09, -1.32, - 0.20, 0.05, -0.07)

grad(f4a, x = mean(y), y = y)

[1] 7.655679e-11

hessian(f4a, x = mean(y), y = y) > 0

     [,1]
[1,] TRUE
```

b. $\sum_{i=1}^{n} 2(y_i \log \frac{y_i}{\mu} + \mu - y_i)$**. Fixe** $y_i = \{7, 4, 4, 6, 5\}$

<r code>
```
y <- c(7, 4, 4, 6, 5)

grad(f4b, x = mean(y), y = y)

[1] 3.202491e-11

hessian(f4b, x = mean(y), y = y) > 0

     [,1]
[1,] TRUE
```

c. $\sum_{i=1}^{n} 2(\frac{y_i}{\mu} - \log \frac{y_i}{\mu} - 1)$**. Fixe** $y_i = \{2.35, 0.16, 0.56, 1.05, 0.51\}$

<r code>
```r
y <- c(2.35, 0.16, 0.56, 1.05, 0.51)

grad(f4c, x = mean(y), y = y)

[1] 1.249948e-11

hessian(f4c, x = mean(y), y = y) > 0

     [,1]
[1,] TRUE
```

d. $\sum_{i=1}^{n} 2(y_i \log \frac{y_i}{\mu} + (1-y_i) \log \frac{1-y_i}{1-\mu})$**. Fixe** $y_i = \{1,0,1,1,1\}$

<r code>
```r
y <- c(1, 0, 1, 1, 1)

grad(f4d, x = mean(y), y = y)

[1] -5.520178e-11

hessian(f4d, x = mean(y), y = y) > 0

     [,1]
[1,] TRUE
```

e. $\sum_{i=1}^{n} 2(y_i \log \frac{y_i}{\mu} + (m+y_i) \log \frac{m+\mu}{m+y_i})$**. Fixe** $m = 1$ **e** $y_i = \{7,4,4,6,5\}$

<r code>
```r
y <- c(7, 4, 4, 6, 5)

grad(f4e, x = mean(y), y = y)

[1] 5.162475e-11

hessian(f4e, x = mean(y), y = y) > 0

     [,1]
[1,] TRUE
```

7. Sejam y_i valores observados para $i = 1, \ldots, n$. Considere a função perda absoluta dada por

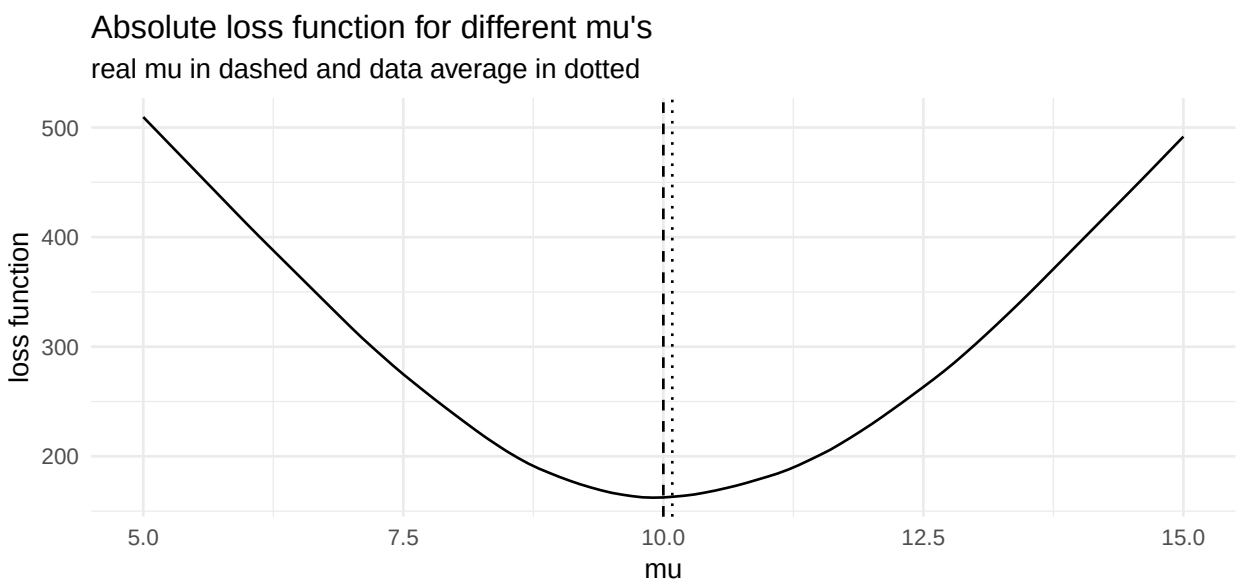
$$f(\mu) = \sum_{i=1}^{n} |y_i - \mu|.$$

a. Simule um conjunto de valores adequado para y_i.

<r code>
```r
n <- 100 ; mu <- 10 ; sd <- 2

y <- rnorm(n = n, mean = mu, sd = sd)
```

b. Esboce o gráfico da função perda para este conjunto de dados e diferentes valores de μ.

<r code>
```r
perda <- function(mu, y) {
    sum(abs(y - mu))
}
perda_vec <- Vectorize(perda, "mu")

mu_seq <- seq(5, 15, .1)
perda_eval <- perda_vec(mu = mu_seq, y = y)

ggplot(data.frame(mu = mu_seq, perda = perda_eval),
       aes(x = mu, y = perda)) +
    theme_minimal() +
    geom_line() +
    geom_vline(xintercept = mu, linetype = "dashed") +
    geom_vline(xintercept = mean(y), linetype = "dotted") +
    labs(y = "loss function",
         title = "Absolute loss function for different mu's",
         subtitle = "real mu in dashed and data average in dotted")
```



c. Encontre o valor de μ que minimiza a função perda absoluta.

<r code>
```
mu_seq[which.min(perda_eval)]
```

```
[1] 9.9
```

d. Discuta quando a função perda absoluta pode ser mais conveniente do que a função perda quadrática.

Se a amostra possui *outliers*, a função perda absoluta é mais vantajosa em relacção a função perda quadrática, pelo fato da última ter a tendência de ser dominada pelos outliers. i.e. a soma final tende a ser o resultado destas poucas observações discrepantes, invés de ser uma expressão da média dos valores.

8. Sejam y_i e x_i valores observados para $i = 1, \cdots, n$. Considere o problema de ajustar uma reta relacionando y_i com x_i, usando a função perda absoluta

$$f(\beta_0, \beta_1) = \sum_{i=1}^{n} |y_i - (\beta_0 + \beta_1 x_i)|.$$

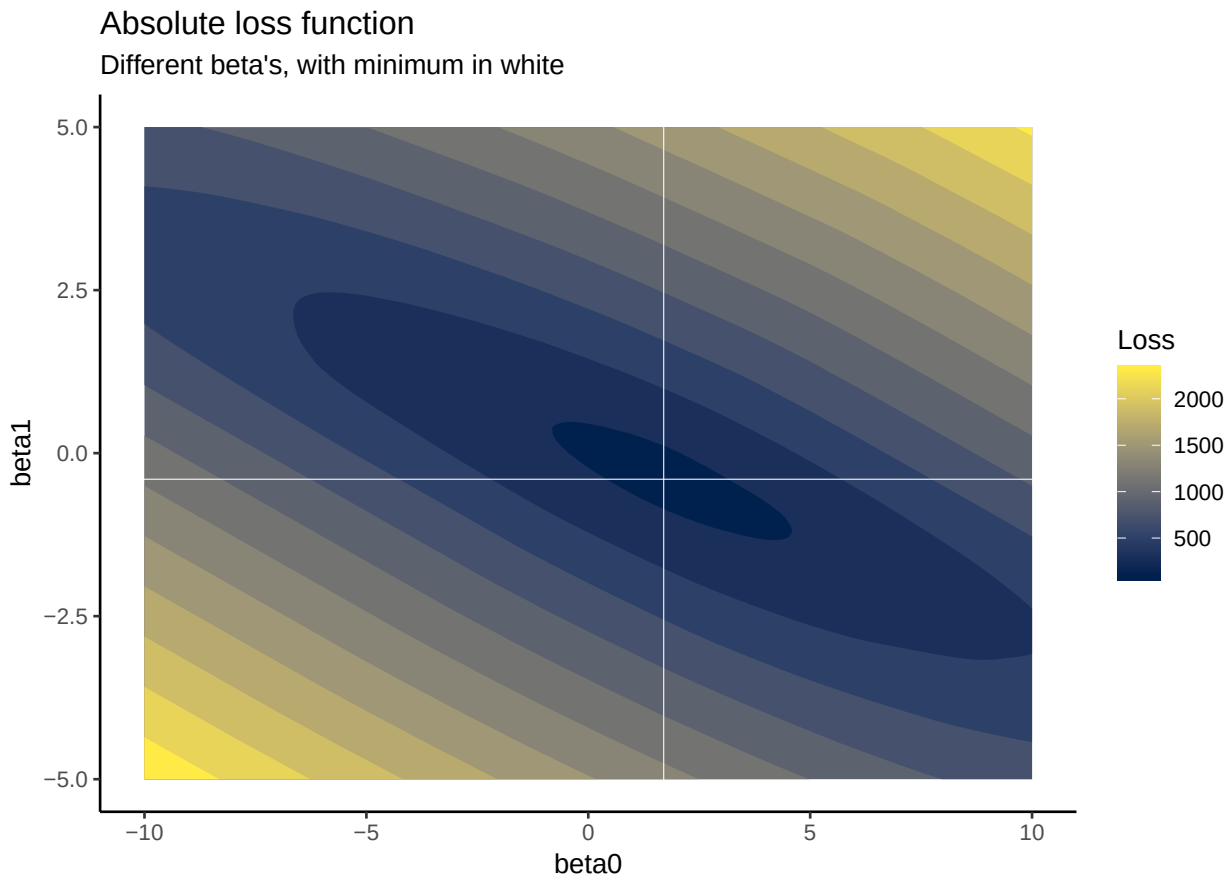
a. Simule um conjunto de valores adequado para y_i fixado um vetor para x_i.

<r code>
```
n <- 100 ; mu <- 5 ; sd <- 2

x <- rnorm(n = n, mean = mu/2, sd = sd)

y <- rnorm(n = n, mean = 2 - .5 * x, sd = sd)
```

b. Esboce o gráfico da função perda para este conjunto de dados e diferentes valores de β_0 e β_1.

<r code>
```
perdaplus <- function(beta0, beta1, y, x) {
    sum(abs(y - beta0 - beta1 * x))
}
perdaplus_vec <- Vectorize(perdaplus, c("beta0", "beta1"))

beta0_seq <- seq(-10, 10, .1)
beta1_seq <- seq(-5, 5, .1)

perdaplus_eval <- outer(beta0_seq, beta1_seq, perdaplus_vec, y = y, x = x)
```

```
rownames(perdaplus_eval) = beta0_seq
colnames(perdaplus_eval) = beta1_seq

longformat <- reshape2::melt(perdaplus_eval)

beta_opt <- longformat[which.min(longformat$value), 1:2]

library(viridis)

ggplot(longformat, aes(Var1, Var2, z = value)) +
    theme_classic() +
    metR::geom_contour_fill() +
    scale_fill_viridis_c(option = "cividis") +
    labs(x = "beta0", y = "beta1",
         title = "Absolute loss function",
         subtitle = "Different beta's, with minimum in white",
         fill = "Loss") +
    geom_vline(xintercept = beta_opt[[1]], col = "white", size = .2) +
    geom_hline(yintercept = beta_opt[[2]], col = "white", size = .2)
```



c. Encontre o valor de β_0 e β_1 que minimiza a função perda absoluta.

<r code>
```
# lembrando que Var1 = beta0, Var2 = beta1, absolute loss fn = value
longformat[which.min(longformat$value), ]
```

```
     Var1 Var2    value
9364  1.7 -0.4 159.6102
```

d. Discuta quando a função perda absoluta pode ser mais conveniente do que a função perda quadrática.

A função perda desta questão é muito similar com a da questão anterior, com a diferença de que aqui decompomos a média numa estrutura linear envolvendo uma covariável.

Além do ponto tocante aos outliers mencionado na questão anterior, uma outra diferença importante entre essas funções perdas é o fato de que a média minimiza o valor esperado da função perda quadrática, e a mediana minimiza o valor esperado da função perda absoluta. Portanto, em situações em que se deseja modelar a mediana invés da média, a perda absoluta se mostra a mais conveniente.

Last modification on ...

```
[1] "2019-09-25 21:22:17 -03"
```