Lista 3 - Introdução a Análise de Dados
Funções
Gabarito

Guilherme Masuko

April 2023

Curso de Cálculo: Takhe
Professor: Douglas Bokliang

Questão 1

A convexidade de uma função é dada pela segunda derivada (isso sob a condição de continuidade da função). A definição formal de convexidade de função pode ser encontrada clicando aqui (pode assustar, mas não se preocupe, utilizaremos um teorema para resolver o exercício).

Teorema[1]:

Seja I um intervalo contido no domínio de uma função f. Suponha que f, f' e f'' sejam contínuas em I.

- Se $f''(x) > 0$ para todo $x \in I$, então f é uma função convexa no intervalo I.
- Se $f''(x) < 0$ para todo $x \in I$, então f é uma função côncava no intervalo I.

Isso quer dizer que para funções do segundo grau contínuas, o coeficiente a determina a convexidade da função. Seja $f(x) = ax^2 + bx + c$.

[1]<https://www.professores.uff.br/hjbortol/wp-content/uploads/sites/121/2017/09/gma00108-parte-21.pdf>

Figure 1: Função Convexa

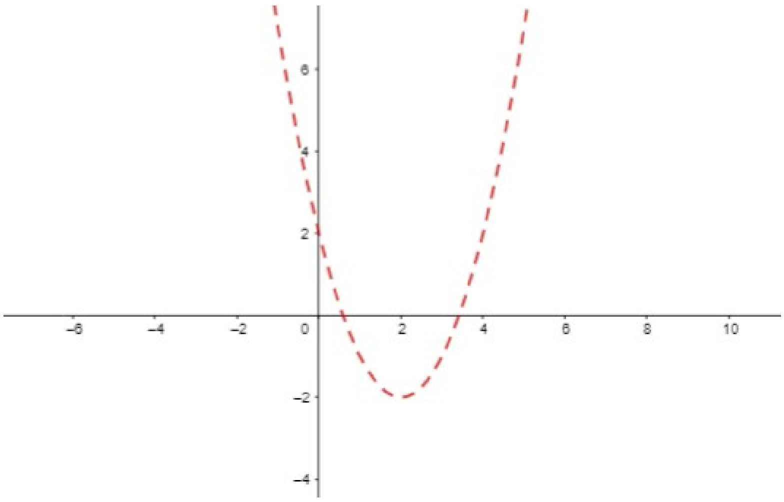
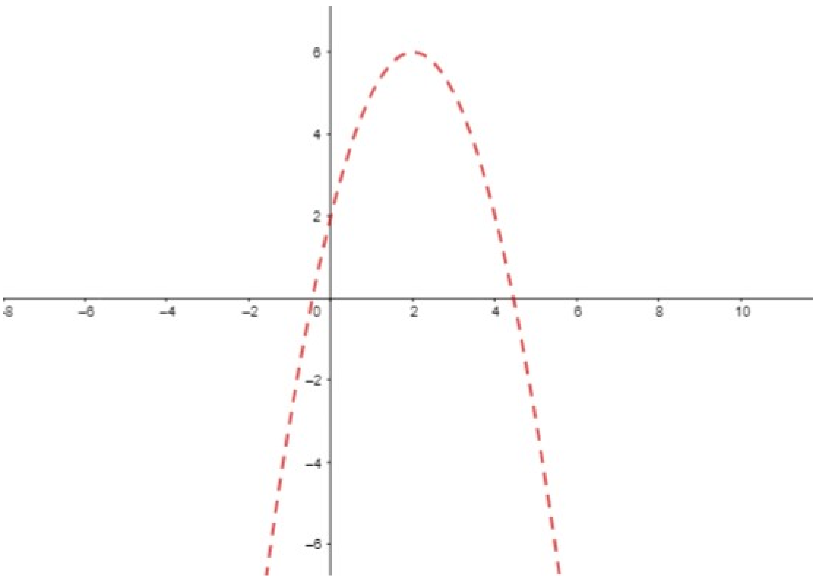


Figure 2: Função Côncava



$$f'(x) = 2ax + b$$
$$f''(x) = 2a$$

Ou seja, se $a > 0$, então a função f é convexa e se $a < 0$, então a função é côncava.

Crie uma função que receba os três coeficientes de uma função do segundo grau como parâmetros e retorne a convexidade da função (`'convexa'` ou `'côncava'`).

Teste sua função para as seguintes equações:

- $f(x) = -2x^2 + 4x + 6$

- $f(x) = x^2 - 4x + 10$

Solução

```
rm(list=ls())

convexidade <- function (a, b, c) {
  # criando a variável que será retornada
  conv <- 'afim'
  # criando os casos para cada valor de a
  if (a > 0) {
    conv <- 'convexa'
  } else if (a < 0) {
    conv <- 'côncava'
  }
  return(conv)
}


# testando para alguns casos

convexidade(-2, 4, 6)

convexidade(1, -4, 10)
```

Questão 2

Um ponto crítico[2] de uma função é um ponto no domínio de uma função onde a primeira derivada não existe ou é nula. Os pontos críticos serão sempre pontos de máximos ou mínimos locais ou pontos de inflexão.

No caso de funções do segundo grau, $f(x) = ax^2 + bx + c$, o ponto crítico da função é encontrado da seguinte maneira.

$$f'(x^\star) = 2ax^\star + b = 0$$
$$x^\star = -\frac{b}{2a}$$

Crie uma função que receba os três coeficientes de uma função do segundo grau como parâmetros e retorne o ponto crítico e o valor da função avaliado no ponto crítico, respectivamente, $x^\star$ e $f(x^\star)$.

[2] <https://pt.wikipedia.org/wiki/Ponto_cr%C3%ADtico_(fun%C3%A7%C3%B5es)>

Teste sua função para as seguintes equações:

- $f(x) = -2x^2 + 4x + 6$
- $f(x) = x^2 - 4x + 10$

Solução

```
rm(list=ls())

pontoCritico <- function (a, b, c) {
  # usando a fórmula para encontrar o ponto crítico
  x_star <- (-b)/(2*a)

  # usando o ponto crítico para achar o valor da função neste ponto
  f_x_star <- a*x_star^2 + b*x_star + c

  # retornamos os dois resultados
  result <- c(x_star, f_x_star)
  return(result)
}


# testando para alguns casos

pontoCritico(-2, 4, 6)

pontoCritico(1, -4, 10)
```

Questão 3

Utilizando as funções criadas nas questões 1 e 2, crie uma função que receba os coeficientes de uma função do segundo grau como parâmetros e retorne o ponto crítico, se ele é um ponto de máximo ou mínimo e o valor da função neste ponto.

Dica: Se uma função é estritamente côncava em um intervalo I, um ponto crítico nesse mesmo intervalo é um ponto de máximo local. Se uma função é estritamente convexa em um intervalo I, um ponto crítico nesse mesmo intervalo é um ponto de mínimo local.

Teste sua função para as seguintes equações:

- $f(x) = -2x^2 + 4x + 6$
- $f(x) = x^2 - 4x + 10$

Solução

```r
rm(list=ls())

# precisamos rodar as funções para gravar na memória do programa


# função convexidade

convexidade <- function (a, b, c) {
  # criando a variável que será retornada
  conv <- 'afim'
  # criando os casos para cada valor de a
  if (a > 0) {
    conv <- 'convexa'
  } else if (a < 0) {
    conv <- 'côncava'
  }
  return(conv)
}



# função ponto crítico

pontoCritico <- function (a, b, c) {
  # usando a fórmula para encontrar o ponto crítico
  x_star <- (-b)/(2*a)

  # usando o ponto crítico para achar o valor da função neste ponto
  f_x_star <- a*x_star^2 + b*x_star + c

  # retornamos os dois resultados
  result <- c(x_star, f_x_star)
  return(result)
}


localMaxMin <- function (a, b, c) {
  # declarando as variáveis utilizando as funções criadas
  f_x_star <- pontoCritico(a, b, c)
```

```
  conv <- convexidade(a, b, c)

  # criando uma variável que indica se é máximo ou mínimo
  max_min <- ''

  # note que o condicional precisa ser escrito da mesma forma que na
     função
  if (conv == 'convexa') {
    max_min <- 'mínimo'
  } else if (conv == 'côncava') {
    max_min <- 'máximo'
  }
  return(paste('O ponto',f_x_star[1],'é um ponto de',max_min,'local.
     O valor da função neste ponto é',f_x_star[2]))
}


# testando para alguns casos

localMaxMin(-2, 4, 6)

localMaxMin(1, -4, 10)
```

Questão 4

É possível esboçar o gráfico de uma função quando se sabe características importantes dela, por exemplo, suas raízes, sua convexidade, seu ponto crítico e o valor da função neste ponto. Crie uma função que receba os coeficientes de uma função do segundo grau como parâmetros e retorne suas raízes (use a função feita na Lista 1), seu ponto de máximo ou mínimo e o valor da função neste ponto (use a função feita na questão anterior).

Teste sua função para as seguintes equações:

- $f(x) = -2x^2 + 4x + 6$
- $f(x) = x^2 - 4x + 10$

Solução

```
rm(list=ls())
```

```
# precisamos rodar as funções para gravar na memória do programa


# função que encontra as raízes

raizes_funcao_2grau <- function (a, b, c){
  # primeiro definimos o delta
  delta = b^2 - 4*a*c
  # apos isso, vamos quebrar nos três casos
  if(delta < 0){
    # se o delta é negativo, a parabola não tem raíz
    return(paste("Essa função não tem raízes. A parábola não corta o
        eixo das abscissas!"))
  } else if(delta == 0){
    # se o delta é igual a zero, a raíz é unitária
    x = (-b)/(2*a)
    return(paste("Essa função tem apenas uma raíz. Seu valor
        é",x,"."))
  } else{
    # se o delta é positivo, então temos duas raízes
    x_1 = (-b + sqrt(delta))/(2*a)
    x_2 = (-b - sqrt(delta))/(2*a)
    return(paste("Essa função tem duas raízes. As duas raízes dessa
        função são respectivamente",x_1,"e",x_2,"."))
  }
}


# função convexidade

convexidade <- function (a, b, c) {
  # criando a variável que será retornada
  conv <- 'afim'
  # criando os casos para cada valor de a
  if (a > 0) {
    conv <- 'convexa'
  } else if (a < 0) {
    conv <- 'côncava'
  }
  return(conv)
```

```
}


# função ponto crítico

pontoCritico <- function (a, b, c) {
  # usando a fórmula para encontrar o ponto crítico
  x_star <- (-b)/(2*a)

  # usando o ponto crítico para achar o valor da função neste ponto
  f_x_star <- a*x_star^2 + b*x_star + c

  # retornamos os dois resultados
  result <- c(x_star, f_x_star)
  return(result)
}


# função que verifica se é máximo ou mínimo

localMaxMin <- function (a, b, c) {
  # declarando as variáveis utilizando as funções criadas
  f_x_star <- pontoCritico(a, b, c)
  conv <- convexidade(a, b, c)

  # criando uma variável que indica se é máximo ou mínimo
  max_min <- ''

  # note que o condicional precisa ser escrito da mesma forma que na
     função
  if (conv == 'convexa') {
    max_min <- 'mínimo'
  } else if (conv == 'côncava') {
    max_min <- 'máximo'
  }
  return(paste('O ponto',f_x_star[1],'é um ponto de',max_min,'local.
     O valor da função neste ponto é',f_x_star[2]))
}
```

```
equacao_2grau <- function (a, b, c) {
  # declarando os resultados
  raizes <- raizes_funcao_2grau(a, b, c)
  maxMin <- localMaxMin(a, b, c)

  return(paste(raizes, maxMin))
}


# testando para alguns casos

equacao_2grau(-2, 4, 6)

equacao_2grau(1, -4, 10)
```