Aluna: Livia Barbosa Fonseca
DRE:118039721

Questão 1-)

Nesse exercício vamos determinar a curva da forma $T = c_0X^{C1}$ que melhor se ajusta aos dados da tabela a seguir com o método de mínimos quadrados com coeficientes não-lineares e usar o modelo para calcular T(0.3) com três casas decimais.

x	0.1	0.2	0.4	0.8	0.9
T(x)	22	43	84	210	320

Como estamos tratando de uma função não linear, vamos ter que realizar a troca de variáveis a fim de conseguirmos realizar a regressão. Assim, vamos considerar a função:

$$T(x) = c_0X^{C1}$$

Linearizando a função, obtemos:

$$Y = c_0X^{C1}$$
$$\ln(Y) = \ln(c1*X^{c2})$$
$$\ln(Y) = \ln(c1) + \ln(X^{c2})$$
$$\ln(Y) = \ln(c1) + c2\ln(X)$$

Realizando a troca de variáveis, obtemos:

$$y\_barra = \ln(Y)$$
$$x\_barra = \ln(X)$$
$$c1\_barra = \ln(c1)$$
$$c2\_barra = c2$$

Dessa forma, encontramos a seguinte função linear:

$$y\_barra = c1\_barra + x\_barra*c2\_barra$$

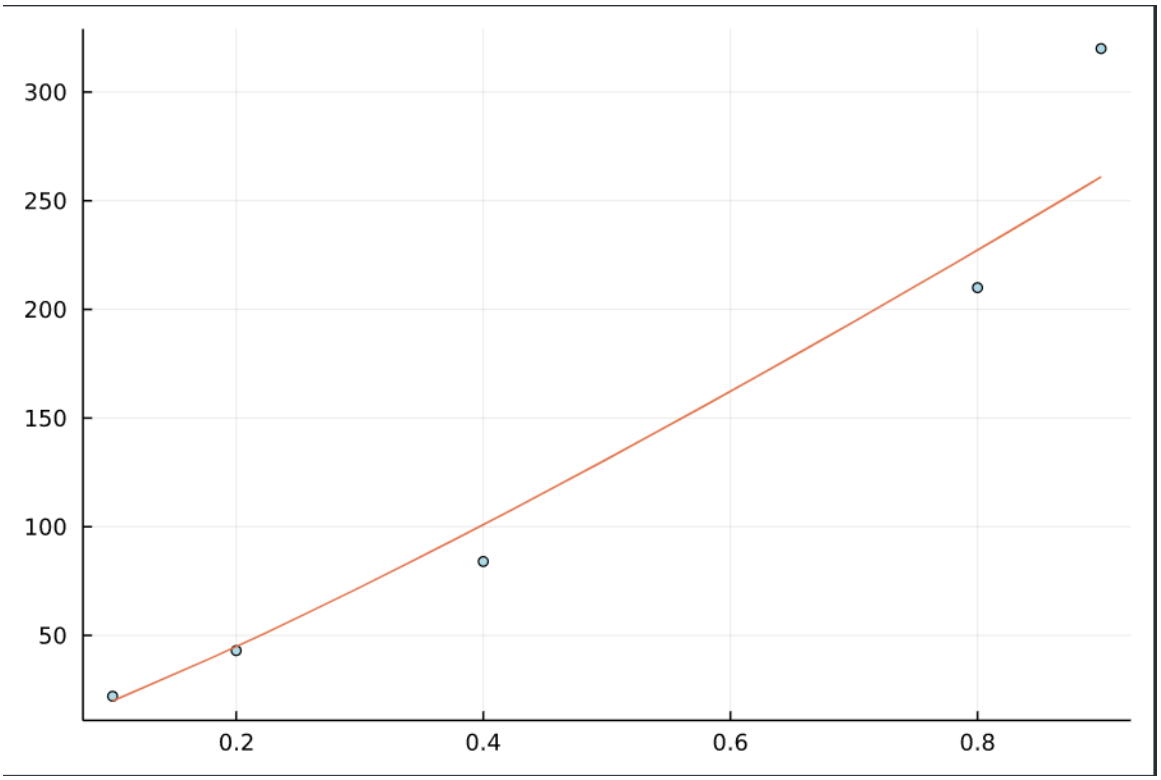
Agora, vamos utilizar o Julia para realizar os cálculos do Método da regressão:

```
using Plots
using Random
using LinearAlgebra

#Vetores de pontos x e y do problema dado
x = [0.1;0.2;0.4;0.8;0.9]
y = [22; 43;84;210;320]

#Troca de variáveis "indo para mundo linear"
x_barra=log.(x)
y_barra=log.(y)

#Função que recebe um conjunto de números x e y e o grau do polinômio
que desejado e cria a matriz de vandermonde dele
function vandermonde(x,y,grau)
```

```
#Recebendo o tamanho do vetor x
n,=size(y)
#Criando uma matriz zerada do tamanho desejado
V=zeros(n,grau+1)
#Looping que vai até o número de linhas da matriz
    for i=1:n
#Looping que vai até o número de colunas desejadas na matriz (grau da
#matriz + 1)
     for j=1:(grau+1)
#Criando ponto a ponto da matriz (posição atual x elevado a j-1)
        V[i,j]=x[i]^(j-1)
        end
    end
#Retorna a matriz desejada
    return V
end

#Função que realiza a regressão dado um conjunto de pontos e o grau de
#polinômio
function regressão(x,y,grau)
#Chama a função de vondermonte para criar a matriz
V=vandermonde(x,y,grau)
#Resolvendo o sistema: quando o sistema a ser resolvido não é possível
#de ser resolvido, o contra barra  do Julia já disponibiliza a solução
#pelo método dos mínimos quadrados*
c=V\y
return c #Retorna os coeficientes do polinômio
end

#Realizando a regressão para encontrar os coeficientes c1 e c2 do
polinômio
c_barra = regressão(x_barra,y_barra,1)

#Criando o polinômio y_barra = c1_barra + x_barra*c2_barra passando os
#coeficientes encontrados
reta_barra(x)= c_barra[1]+c_barra[2]*x

#Voltando para a função do problema e encontrando os valores dos
#coeficientes desta função
c1=exp(c_barra[1])
c2=c_barra[2]

#Retornando o modelo anterior
```

```
funcao(x)=c1*(x^c2)

#Plotando os pontos dados
scatter(x, y, c=:lightblue, ms=3, leg=false)
#Plotando o gráfico da função
plot!(funcao)
```

Obtemos o gráfico da função modelada e os pontos dados:

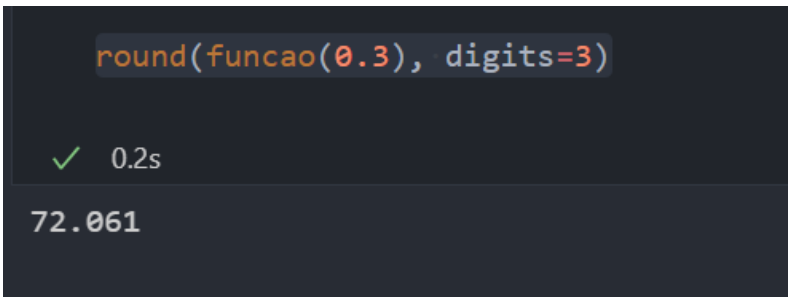


Agora, com a modelagem já realizada, vamos calcular T(0.3) com três casas decimais:

```
round(funcao(0.3), digits=3)
```

A função “round” serve para arredondar o nosso valor em 3 casas decimais (digits=3).

Dessa forma, obtemos:



Temos que T(0.3) é aproximadamente 72.061, com arredondamento de 3 dígitos.

Questão 2-)

(a) Vamos realizar a integral numérica utilizando o método dos trapézios de t = 0 a t = 125 e estimar a altura atingida pelo ônibus 125 segundos após o lançamento. Para isso, vamos utilizar as seguintes funções em julia para nos auxiliar nos cálculos:

Vamos começar utilizando o método da interpolação por partes para descobrir as funções que passam pelos pontos e depois vamos calcular a área do trapézio formado por elas. Para isso, vamos interpolando de dois pontos a dois pontos e encontrando retas que passam por esses pontos. Para utilizar o método da interpolação, vamos primeiro encontrar a matriz de vandermonde associada ao problema e em seguida realizar a interpolação:

```
#Função que recebe um conjunto de números x e y e o grau do polinômio
que desejado e cria a matriz de vandermonde dele
function vandermonde(x,y,grau)
#Recebendo o tamanho do vetor x
n,=size(y)
#Criando uma matriz zerada do tamanho desejado
V=zeros(n,grau+1)
#Looping que vai até o número de linhas da matriz
    for i=1:n
#Looping que vai até o número de colunas desejadas na matriz (grau da
#matriz + 1)
     for j=1:(grau+1)
#Criando ponto a ponto da matriz (posição atual x elevado a j-1)
        V[i,j]=x[i]^(j-1)
        end
    end
#Retorna a matriz desejada
    return V
end
```

```
#Função que recebe os pontos x e y e o grau desejado e realiza a
interpolação dos pontos dados
function interpolação(x,y,grau)
#Cria uma matriz de vondermonde com os pontos dados
    V = vandermonde(x,y,grau)
#Resolve o sistema
    a = V\y
#Cria a função com os coeficientes encontrados pelo sistema anterior
    f(x) =sum(a[i+1]*x^i for i in 0:grau)
```

```
#Retorna a função criada
    return f
end
```

Para realizarmos a integração das funções, vamos utilizar a seguinte função “trapezio” no Julia. Ela irá nos auxiliar no cálculo das integrais das funções já criadas anteriormente.

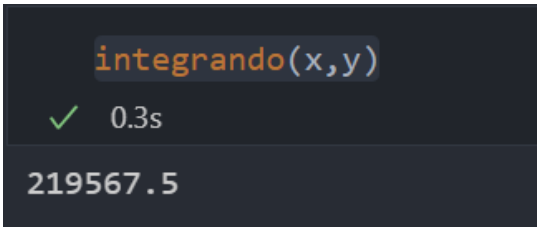
```
#Função que calcula a integral f(x) num intervalo a até b, passando a quantidade de intervalos de integração
function trapezio(f,a,b,n)
#Pegando o tamanho de cada intervalo de integração
    h = (b-a)/n
#Criando uma variável que irá guardar o valor da integral (area do trapezio)
    S=0.0
    for i=1:(n-1)  #Lopping for que calcula o "meio"
        x=a+i*h
        S+=2*f(x)
    end
#Calculando as “pontas”
    S=h/2*(S+f(a)+f(b))
#Retorna a área (valor da integral)
    return S
end
```

Na função “integrando” vamos chamar as funções de interpolação e do trapézio e realizar o somatório das áreas descobertas.

```
#Função que recebe os pontos x e y
function integrando(x,y)
#Variável que guarda o valor do somatório das áreas dos trapézios
    somatorio = 0.0
#Lopping que realiza a interpolação e a soma das áreas (temos 7 pares de pontos - logo temos 7 funções e 7 áreas)
    for i=1:7
#Realizando a interpolação
        polinomio = interpolação(x[i:i+1],y[i:i+1],1)
#Realizando o somatório das áreas
        somatorio = somatorio + trapezio(polinomio,x[i],x[i+1],1)
    end
#Retornando o somatório dos trapézios
    return somatorio
end
```

Agora vamos chamar a função integrando e passar os pontos x e y do problema:

```
#Vetores de pontos x e y do problema dado
x = [0;10;15;20;32;59;62;125]
y = [0;185;319;447;742;1325;1445;4151]
```



Descobrimos que a altura é 219567,5 pés.

(b) Para descobrirmos o erro máximo cometido no item anterior, precisamos da função “original” do tempo em relação a velocidade do ônibus espacial. Além disso, precisamos descobrir a derivada segunda dessa função e saber se ela é limitada no intervalo que temos interesse, pois o erro máximo é calculado da seguinte forma:

$$|\int_a^b f(x) \; - \; (método\ do\ trapézio)| \leq \frac{h^3 * n * M}{12}$$

Se |f''(x)| <=M, $a \leq x \leq b$.

Lembrando que h é o tamanho dos intervalos de integração e n a quantidade deles.

(c) Nesse exercício, vamos encontrar uma reta p1(x) = c0x+c1, no sentido dos mínimos quadrados, que melhor descreve a distribuição dos pontos. E vamos estimar a altura a partir dessa curva. Dessa forma, iremos calcular a seguinte integral:

$$\int_0^{125} p1(x)dx$$

Primeiramente, iremos calcular a reta pedida utilizando as seguintes funções no Julia, via método dos mínimos quadrados. Como estamos tratando de uma reta, teremos um polinômio de grau 1:

Para utilizarmos a função que realiza a regressão, temos que criar uma matriz de vandermonde com os pontos pedidos. Para isso, vamos utilizar a mesma função utilizada anteriormente:

```
#Função que recebe um conjunto de números x e y e o grau do polinômio
que desejado e #cria a matriz de vandermonde dele
function vandermonde(x,y,grau)
```

```
    #Recebendo o tamanho do vetor x
    n,=size(y)
    #Criando uma matriz zerada do tamanho desejado
    V=zeros(n,grau+1)
    #Looping que vai até o número de linhas da matriz
    for i=1:n
#Looping que vai até o número de colunas desejadas na matriz (grau da
matriz + 1)
        for j=1:(grau+1)
 #Criando ponto a ponto da matriz (posição atual x elevado a j-1)
            V[i,j]=x[i]^(j-1)
        end
    end
    #Retorna a matriz desejada
    return V
end
```

Também, iremos utilizar a função trapézio que foi utilizada na questão anterior:

```
#Função que calcula a integral f(x) num intervalo a até b, passando a
quantidade de intervalos de integração
function trapezio(f,a,b,n)
#Pegando o tamanho de cada intervalo de integração
    h = (b-a)/n
#Criando uma variável que irá guardar o valor da integral (area do
trapezio)
    S=0.0
    for i=1:(n-1)  #Lopping for que calcula o "meio"
        x=a+i*h
        S+=2*f(x)
    end
#Calculando as "pontas"
    S=h/2*(S+f(a)+f(b))
#Retorna a área (valor da integral)
    return S
end
```

Agora, vamos utilizar uma função que recebe o grau do polinômio interpolador que desejamos e realiza a regressão com os pontos dados e retorna esse polinômio.

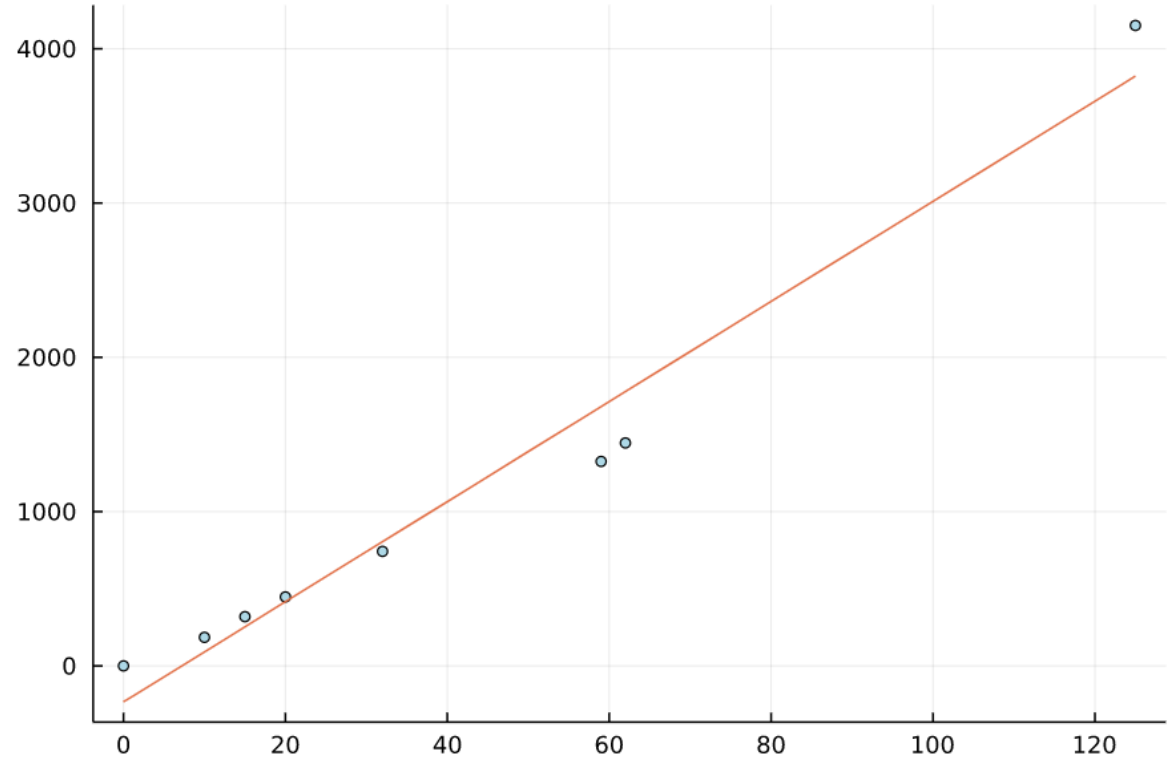
```
#Recebe o grau do polinômio desejado
function polinomio(grau)
    #Cria a matriz de vandermonde
    V = vandermonde(x,y,grau)
    #Realiza a regressão
```

```
    a = V\y
    #Monta a função
    f(x) =sum(a[i+1]*x^i for i in 0:grau)
    #Retorna a função
    return f
end
```

Com as funções necessárias já criadas, agora vamos plotar a reta (polinômio de grau 1) e os pontos x e y desejados.

```
scatter(x, y, c=:lightblue, ms=3, leg=false) #Pontos
plot!(polinomio(1)) #polinômio de grau=1
```

Assim, obtemos:



Agora, vamos utilizar o método do trapézio para realizar a interpolação dessa função encontrada. Essa função também já foi utilizada no item anterior:

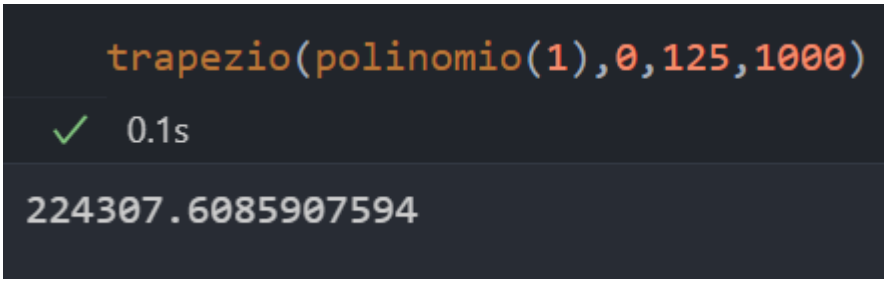
```
#Função que calcula a integral f(x) num intervalo a até b, passando a
quantidade de intervalos de integração
function trapezio(f,a,b,n)
    #Pegando o tamanho de cada intervalo de integração
    h = (b-a)/n
    #Criando uma variável que irá guardar o valor da integral (area do
trapezio)
    S=0.0
    for i=1:(n-1)  #Lopping for que calcula o "meio"
```

```
        x=a+i*h
        S+=2*f(x)
    end
    #Calculando as "pontas"
    S=h/2*(S+f(a)+f(b))
    #Retorna a área (valor da integral)
    return S
end
```

Calculando a integral dessa função, onde temos o polinômio de grau 1 encontrado, os intervalos de integração que são 0-125, considerando 1000 intervalos de integração:

```
trapezio(polinomio(1),0,125,1000)
```

Onde encontramos:



Dessa forma, descobrimos que a aproximação da integral calculada via métodos dos trapézios é de 224307.6085907594, ou seja, a altura é de 224307.6085907594 pes.

Questão 3-)

(a) Nesse exercício, queremos determinar uma aproximação para a área limitada pelo círculo $x^2 + y^2 = 1$ (que é igual a π) no primeiro quadrante utilizando o método dos trapézios com h = 0,1 e determinar uma estimativa para π a partir disto.

Vamos isolar o y na função para podermos utilizar no método dos trapézios:

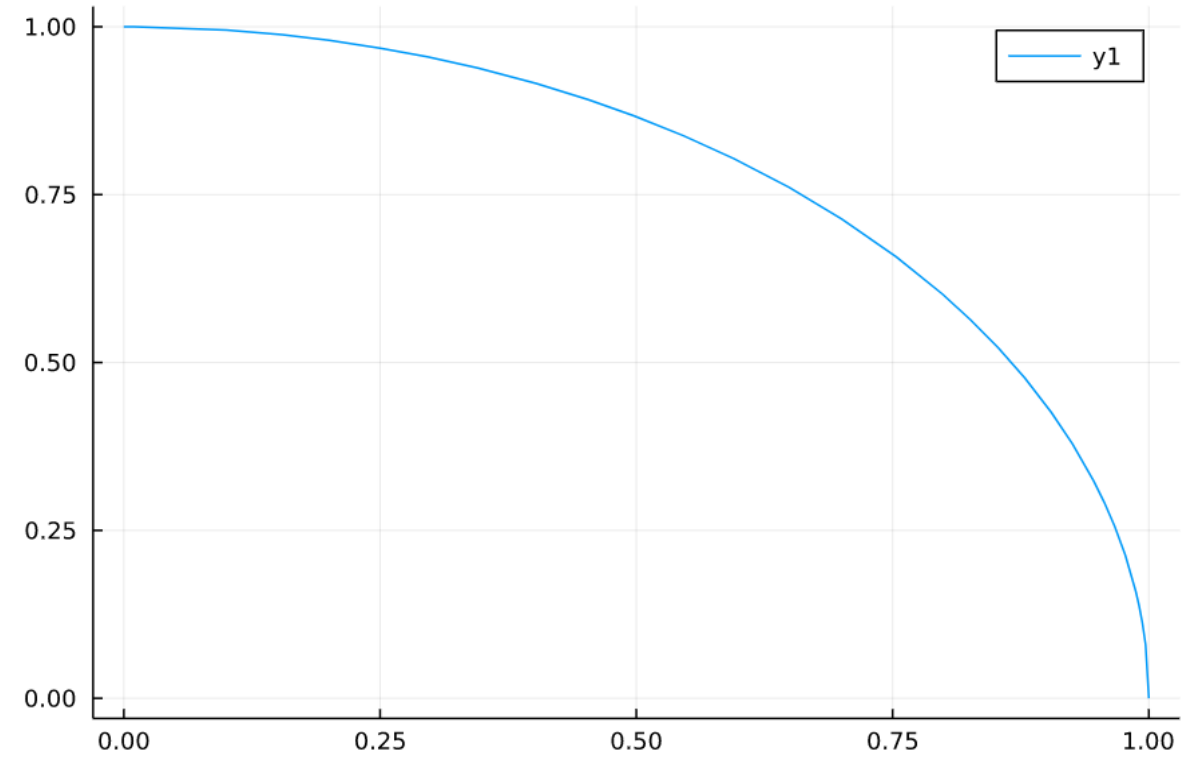
$$x^2 + y^2 = 1$$
$$y^2 = 1 - x^2$$
$$y = \sqrt{1 - x^2}$$

Logo temos que a função é f(x) = $\sqrt{1 - x^2}$ e o intervalo de integração será de 0 a 1, visto que o raio do círculo é 1. Teremos:

$$\int_0^1 \sqrt{1 - x^2}\,dx$$

Agora, vamos observar o gráfico da função:

```
l(x) = sqrt(1 - x^2) #Função dada
plot(l,0,1) #Plotando o gráfico da função no intervalo pedido
```



Podemos notar que o gráfico fica como o esperado no intervalo de 0 até 1.

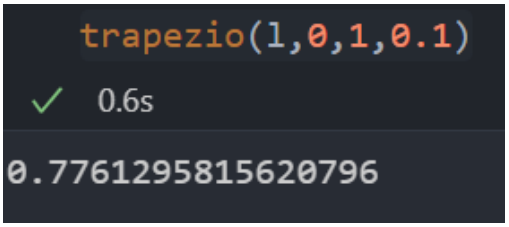
Agora, vamos calcular a integral dessa função via método dos trapézios. Para isso, vamos utilizar a mesma função utilizada na questão anterior, porém ao invés de passarmos o número de intervalos de integração, iremos passar o tamanho do intervalo:

```
#Função que calcula a integral f(x) num intervalo a até b, passando o
tamanho do intervalo de integração
function trapezio(f,a,b,h)
#Pegando o número de intervalos
    n = (b-a)/h
#Criando uma variável que irá guardar o valor da integral (area do
trapezio)
    S=0.0
    for i=1:(n-1)  #Lopping for que calcula o "meio"
        x=a+i*h
        S+=2*f(x)
    end
#Calculando as "pontas"
```

```
    S=h/2*(S+f(a)+f(b))
#Retorna a área (valor da integral)
    return S
end
```

Agora, passando os valores do nosso problema para a função, temos:

```
trapezio(1,0,1,0.1) Tamanho do intervalo: 0.1; Intervalo de integração:
0 até 1
```



Encontramos a aproximação da área da função do círculo no primeiro quadrante. Agora, vamos encontrar uma aproximação para π. Sabendo que a função do círculo é dada por:

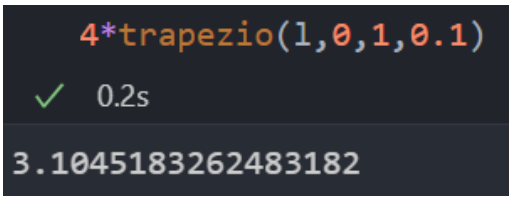
$$areaCirculo = \pi * R^2$$

Vamos utilizar essa fórmula para descobrir o valor aproximado de π. Com a integração anterior obtemos o valor de ¼ do círculo, já que temos a área do primeiro quadrante. Dessa forma obtemos:

$$areaCirculo/4 = \pi * R^2/4$$

Substituindo areaCirculo/4 por 0.7761295815620796 e sabendo que o raio é 1, obtemos:

$$0.7761295815620796 = \pi/4$$
$$\pi = 0.7761295815620796 * 4$$
$$\pi = 3.1045183262483182$$



Obtemos que a aproximação de π é de 3.1045183262483182.

(b) Não podemos utilizar a fórmula do erro da regra do trapézio para estimar o erro no item anterior visto que para estimarmos o erro máximo temos que ter a segunda derivada da função limitada no intervalo de integração em questão. Nesse caso, não teremos a derivada limitada. Vamos analisar a segunda derivada dessa função:

$$f(x) = \sqrt{1 - x^2}$$

Primeira derivada: $f'(x) = -\frac{x}{\sqrt{1-x^2}}$

Segunda derivada: $f''(x) = \frac{-1}{(1-x^2)*\sqrt{1-x^2}}$

Com a segunda derivada em mão, vamos substituir x = 1 e analisar se a função está definida:

$$f''(x) = \frac{-1}{(1-x^2)*\sqrt{1-x^2}}$$

$$f''(1) = \frac{-1}{(1-1^2)*\sqrt{1-1^2}}$$

$$f''(1) = \frac{-1}{0} \text{ (indefinido)}$$

Como temos uma divisão por zero, essa função não está definida em todo o intervalo que deveria. Sendo assim, não podemos calcular o erro máximo do exercício do item anterior.

Questão 4-)

Nesse exercício, vamos adaptar a função integral dupla da aula 18 em Julia para aproximar a seguinte integral numericamente

$$\int_a^b \int_{h(y)}^{g(y)} f(x,y)\, dxdy$$

Podemos observar que a única coisa que irá mudar do caso que temos para esse são os limites de integração da primeira integral. Agora será passado uma função e não somente pontos. Sendo assim, nossa função trapézio não será alterada e ficará da seguinte forma:

```
#Função que calcula a integral f(x) num intervalo a até b, passando o
número dos intervalo de integração
function trapezio(f,a,b,n)
#Pegando o tamanho dos intervalos
    h = (b-a)/n
#Criando uma variável que irá guardar o valor da integral (area do
trapezio)
    S=0.0
    for i=1:(n-1)  #Lopping for que calcula o "meio"
        x=a+i*h
      #Incrementando a soma
        S+=2*f(x)
    end
#Calculando as "pontas"
    S=h/2*(S+f(a)+f(b))
#Retorna a área (valor da integral)
```

```
    return S
end
```

A função integral também não será alterada.

```
#Função que recebe uma f(x) e os limites a e b e realiza a integração
via método dos trapézios
function Integral(f,a,b)
#Retornando a integral com 1000 intervalos
    return trapezio(f,a,b,1000)
end
```

Porém, a função integraldupla irá ser alterada.

```
#integral dupla de h(x,y) de a(y) até b(y) no x e de c até d no y
function Integral_Dupla(h,a,b,c,d)
    function g(y)
        f(x)=h(x,y)
#Realizando a integração de f de a(y) até b(y)
        return Integral(f,a(y),b(y))
    end
#Retorna a integral
    return Integral(g,c,d)
end
```

Podemos observar que agora, a nossa função “g(x)”, que faz o papel de calcular a integral que se encontra no interior, terá os limites alterados. Onde anteriormente, eram pontos (a,b), que foram alterados por funções do tipo a(y), b(y). Dessa forma alteramos a seguinte linha de código:

```
return Integral(f,a,b)
```

E alteramos por:

```
  return Integral(f,a(y),b(y))
```

Agora vamos analisar se de fato a nossa função está funcionando como o desejado calculando a seguinte integral:

$$\int_0^1 \int_y^{y^2} x + y \, dx \, dy$$

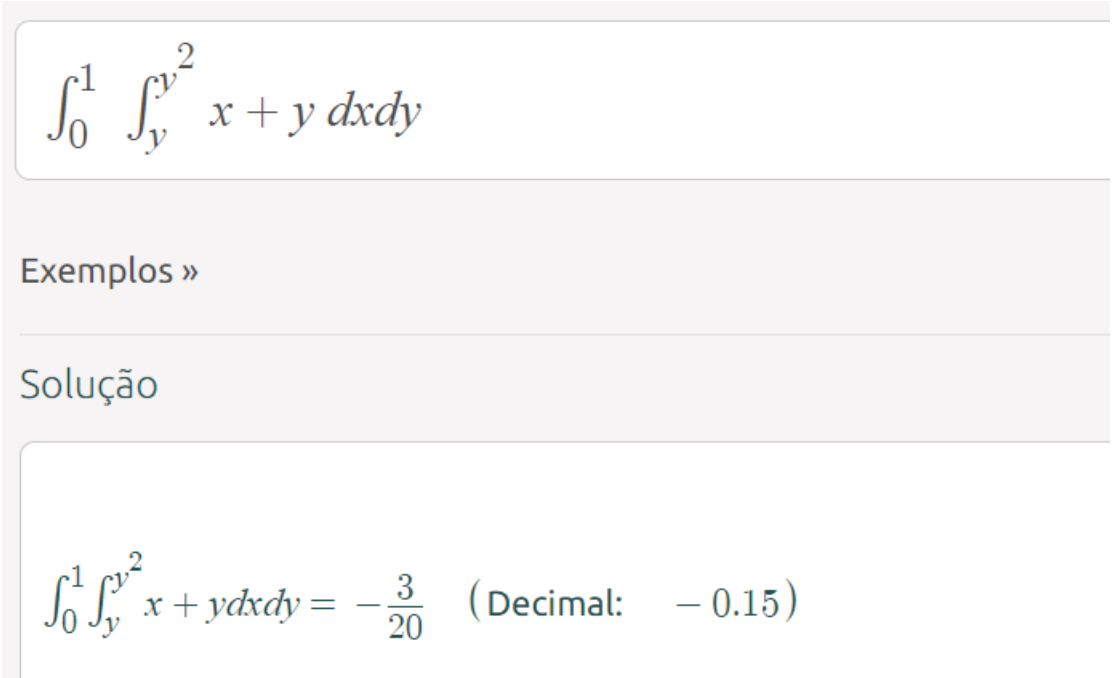
Realizando esse cálculo pelas funções que escrevemos no Julia:

```
h(x,y) = x + y
a(y) = y
b(y) = y^2
round(Integral_Dupla(h,a,b,0,1), digits=3)
```

✓ 0.7s

```
-0.15
```

Calculando essa integral pela calculadora para conferirmos se o valor está batendo encontramos:



Podemos observar que de fato nosso cálculo está correto e o valor da integral via método dos trapézios é de -0,15.