In [620…

```
using CSV;
using PyCall;
using PyPlot;
using Printf;
using DataFrames
np = pyimport("numpy");
```

In [3]:

```
wrg.filterwarnings("ignore")
```

1.0. Capítulo 0

1.1. Funções

A e B são dois conjuntos e podemos associar eles a uma reta, e cada elemento a de A pode associar somente com um elemento de b de B, esse é o conceito de função e também o conceito de imágem.
f é aplicação de A em B se e somente se $\forall x \in A \exists | y \in B \; / \; (x, y) \in f$

1.1.1. Domínio & Imagem

Resumidamente, existe uma função por exemplo uma reta, onde existe o eixo **x** e o eixo **y** em um plano cartesiano, logo cada **x** tem uma relação com a função, logo $f(x)$ é a função avaliada nesse ponto **x** em específico.
A função do primeiro grau representada pela reta em **vermelho** esta no eixo X positivo e negativo. Logo o ponto em **vermelho** representado na reta quando o x é igual a 5, na função do primeiro grau $f(5)$ o resultado da função é 0.

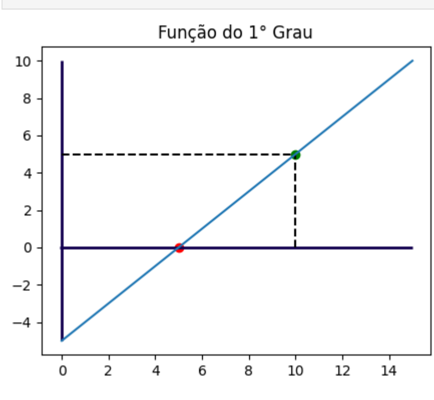
E o ponto em verde é quando o **x** é exatamente 10, colocando na função, logo $f(10)$ é igual a 5.

OBS: Geralmente nas escolas é utilizado y para notação de $f(x)$, nesse notebook vai ser utilizado os dois.

Pode também ser representada op conceito de domínio e imagem utilizando o Diagrama de Venn, recomendo pesquisar depois.

In [156…

```
fig, ax = plt.subplots( figsize=(5, 4) )
ax.plot( collect( -5:1:10 ) );
ax.vlines( 0, -5, 10, color="#12004f", linewidth=2, linestyle="-")
ax.hlines( 0, -.1, 15, color="#12004f", linewidth=2, linestyle="-");
ax.vlines( 10, 0, 5, color="k", linestyle="--" )
ax.hlines( 5, 0, 10, color="k", linestyle="--" )
ax.scatter( 5, 0, c="r" )
ax.scatter( 10, 5, c="g" )
ax.set_title("Função do 1° Grau");
```



1.1.1. Função Polinomial

As funções polinomias nada mais são que as funções com outros graus.
Onde $n \in \mathbb{N}$, a são números reais chamados de $coeficientes$ e o grau de um polinômio é n.
Exemplos:

Para as funções polinomias existem algumas fórmulas como:
$f(x) = 2x^3 + 2x^2 - 4x$
$g(x) = 3x^2 - 3x$

Somar os termos semelhantes, caso não tiver nenhum semelhante, então sómente copiar.
$f(x) + g(x) = 2x^3 + 5x^2 - 7x$

Subtrair os termos semelhantes, caso não tiver nenhum semelhante, então sómente copiar, porém, precisa multiplicar por -1 para mudar o sinal dos elementos da função.
$((2x^3 + 2x^2 - 4x) - (3x^2 - 3x)) = (2x^3 + 2x^2 - 4x - 3x^2 + 3x)$ Agorá é so aplicar as mesmas regrinhas da soma.
$f(x) - g(x) = 2x^3 - x^2 - x$

Para a multiplicação e a divisão vai ser tarefa de casa aprender :)

1.1.2. Função do 1° Grau

A função polinomial do primeiro grau, tambem conhecida como função afim, é descrita pela seguinte formula:
$f(x) = a \cdot x + b$ Onde temos os seguintes termos:

a Coeficiente Angular (Inclinação da Reta)

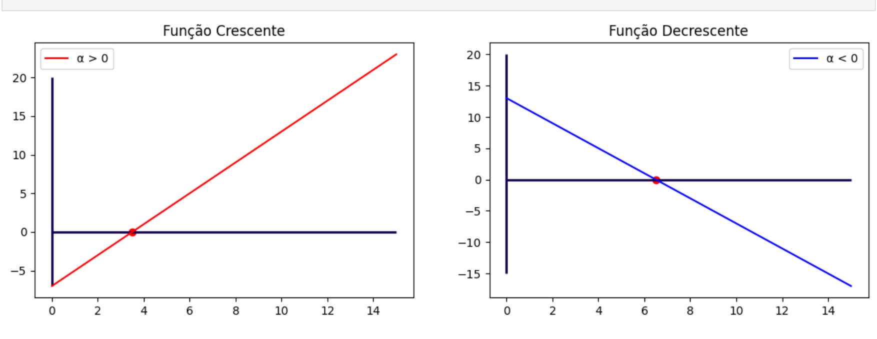
b Coeficiente Linear (Onde a reta corta o eixo Y)

OBS: $a, b \in \mathbb{R}, a \neq 0$, O coeficiente angular tem que ser diferente de zero, logo ele sendo 0 é uma função **constante**.
Pode ser representada também representado a relação de **x** e **y** em um gráfico, conhecido como reta.
Se o coeficiente angular for positivo, logo a retra é crescente, se for negativo, a reta é decrescente.

In [119…

```
fig, (ax1, ax2) = plt.subplots( 1, 2, figsize=(13, 4) )
for i in zip( [[a * p + b for p in collect( -5:1:10 )], [-a * p + b for p in collect( -5:1:10 )]],
              [ax1, ax2], ["r", "b"], ["Função Crescente", "Função Decrescente"], [-7, -15], ["α > 0", "α < 0'
    i[2].plot( i[1], c=i[3], label=i[6] );
    i[2].scatter( i[7], 0, c="r" )
    i[2].set_title(i[4])
    i[2].hlines( 0, 0, 15, color="#12004f", linewidth=2, linestyle="-");
    i[2].vlines( 0, i[5], 20, color="#12004f", linewidth=2, linestyle="-");
    i[2].legend();
end

plt.savefig("Linear")
```




```
I) Determine a expressão que define a função Afim que passe pelos pontos (1, 5) e (3, -1).
```

Foi dado um par $(x_0, y_0), (x_1, y_1)$, sempre o primeiro número do par é o **x** e o segundo é o **y**.
Agora o objetivo é e, um plano cartesiano conseguir desenvolver a fórmula que passe por esses pontos, para isso usa-se a fórmula:
$y - y_0 = m(x - x_0)$ e para descobrir o m, utiliza-se a fórmula: $m = \frac{y_1 - y_0}{x_1 - x_0}$.

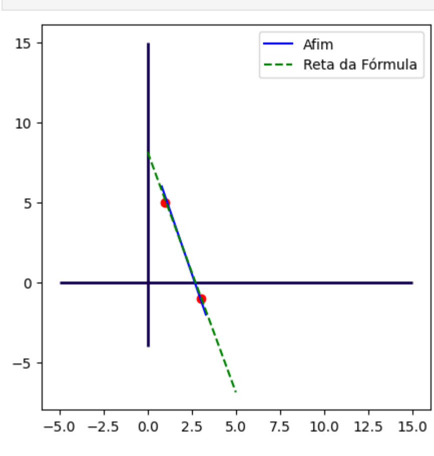
- Primeiro pásso é desenhar um plano cartesiano com dois eixos e setar os pontos.
- Segundo passo é encontrar o m para aplicar na fórmula.

$m = \frac{-1-5}{3-1} = \frac{-6}{2} = -3$, se for fazer utilizando o inverso dos pontos da o mesmo resultado, logo $m = \frac{5-(-2)}{1-3} = \frac{6}{-2} = -3$

Agora basta utilizar na outra fórmula o resultado de $m = -3$
$y - y_0 = m(x - x_0) = y - 5 = -3(x - 1)$
$y = -3x + 8 \;\; ou \;\; f(x) = -3x + 8$

In [208…

```
fig, ax = plt.subplots( figsize=(5, 5) );
ax.hlines( 0, -5, 15, color="#12004f", linewidth=2, linestyle="-");
ax.vlines( 0, -4, 15, color="#12004f", linewidth=2, linestyle="-");
ax.scatter( [1, 3], [5, -1], color="r" )
ax.plot( [.8, 3.3], [6, -2], c="b", label="Afim" )
ax.plot( [(-3) * p + 8 for p in collect(-.05:5)], c="g", linestyle="--", label="Reta da Fórmula" ) #-3x + 8
ax.legend();
```



1.1.3. Função do 2° Grau

A funçaõ do segundo grau é a clássica função da fórmula de bháskara e é representada quando o maior grau do polinômio é 2, logo a fórmula é: $f(x) = ax^2 + bx + c$, onde possui três parâmetros, $a, b, c \in \mathbb{R}, a \neq 0$ e o a tem que ser diferente de zero, se for zero, logo a função vai ser uma função do primeiro grau.

Coeficiente a: Como é a abertura dessa parabola, se o a for muito grande, mais fina a concavidade para cima vai ficar, quando menor próximo de 0 mais aberta é a concavidade, logo a mesma coisa para números negativos, porém para a concavidade para baixo.

Coeficiente b: É em qual quadrante vai estar a parábola.

Coeficiente c: É onde a parábola corta o eixo Y

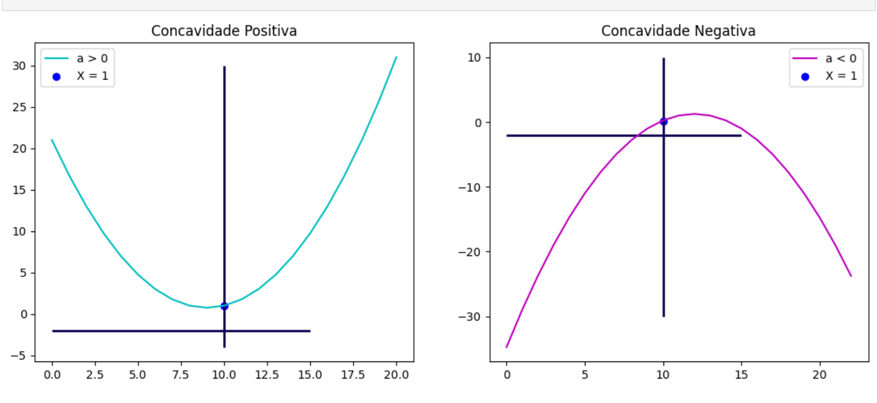
O nome do gráfico dessa função tambem é chamada de Parábola onde possui os pontos de vértice minimo e máximo.

Por exemplo a função $f(x) = 1x^2 + 1x + 1$, quando o termo que multiplica o a é positivo, logo a concavidade é para cima, se o a for negativo, logo a concavidade é para baixo.
Os eixos do matplotlib começam em 0, logo a verdadeira carinha da função $f(x) = 1x^2 + 1x + 1$ é como eu desenhei abaixo, onde os ponstos estão mais centralizados, recomendo usar outra ferramenta para desenhar a parábola como o Geogebra.

In [314…

```
fig, (ax1, ax2) = plt.subplots( 1, 2, figsize=(13, 5) )
for i in zip( [[1p^2 + 1p + 1 for p in collect(-5:.5:5)], [-1p^2 + 1p + 1 for p in collect(-5.5:.5:5.5)]],
              [ax1, ax2], [1, .09], ["c", "m"], ["a > 0", "a < 0"], ["Concavidade Positiva", "Concavidade Nega
    i[2].plot( i[1], c=i[4], label=i[5] );
    i[2].hlines( -2, 0, 15, color="#12004f", linewidth=2, linestyle="-");
    i[2].scatter( 10, i[3], c="b", label="X = 1" )
    i[2].set_title(i[6])
    i[2].legend();
end
ax1.vlines( 10, -4, 30, color="#12004f", linewidth=2, linestyle="-");
ax2.vlines( 10, -30, 10, color="#12004f", linewidth=2, linestyle="-");

plt.savefig("Parábola")
```




```
II) Calcúle as Raizes da seguinte função do segundo grau:
```

a) $f(x) = x^2 + x - 2$
b) $g(x) = x^2 - 10x + 24$
c) $h(x) = x^2 - 15x$

Exístem inúmeras formas de calcular as raizes de uma função do 2° Grau, as raizes são justamente os pontos onde a função corta o eixo **X**, geralmente quando se tem duas raizes, a função corta em dois pontos, com uma raiz somente em um ponto no eixo **x** e sem raizes a parábola esta flutuando em algum quadrante tendo apenas nos múeros complexos.

Uma das ferramentas para calcular as raizes é justamente o Bháskara, dado pela fórmula: $x = \frac{-b \pm \sqrt{\Delta}}{2 \cdot a}$ e o Delta a fórmula é $\Delta = b^2 - 4 \cdot a \cdot b$ e para os vértices é dever de casa :D

Logo a respósta para a pergunta **a)** utilizando bháskara é:
$\sqrt{\Delta = 1^2 - 4 \cdot 1 \cdot -2}$ Logo o delta é: $\sqrt{9}$: **3**

$x = \frac{-1 \pm 3}{1 \cdot 2}$ Logo as raizes são: -2 e 1

Porém não é so essa a forma de calcular a equação do 2° grau, existe a fórmula dos produtos notáveis, para isso, vamos seguir esses passos:
$x^2 + \frac{bx}{a} + \frac{c}{a} = 0$, Dividir todos os lados da equação por a.
Após a divisão, transformar em um produto notável e movimentar a $\frac{c}{a}$ para o outro lado, então:
$x + (\frac{b}{2a})^2 = -\frac{c}{a} + (\frac{b}{2a})^2$ Para transformar em produto notável, foi utilizado a seguinte fórmula: $(x + \frac{b}{2a})^2 = x^2 + \frac{2b}{2a}x + (\frac{b}{2a})^2$ o primeiro a quadrado, mais duas vezes o primeiro vezes o segundo e o quadrado do segundo, porém não tem o segundo termo ao quadrado, para isso foi preciso colocar ele ali, mas tem que colocar do outro lado também, simplificando, temos então:
$x + \frac{b}{2a} = \pm \frac{\sqrt{\Delta}}{2a}$ Já foi tirado a raiz também e simplificado para Δ a expressão $b^2 - 4 \cdot a \cdot c$, e como requer o módulo na raiz, por isso o $\pm$
Finalizando nossa formula ficaria: $x = \frac{-b \pm \sqrt{\Delta}}{2a}$

E oque tudo isso quer nos dizer, simples, pode ser possível transformar qualquer equação do 2° grau em um produto notável quando possível, geralmente vai facilitar as manipualões.

b) Para a alternativa B, vai ser utilizado o Produto Notável.
$x^2 - 10x + 24 = 0$ pode ser escrita como $(x - 5)^2 = \pm 1$, basta mover o $+24$ para o outro lado e somar 25 em ambos os lados para aparecer o produto notável, pois $(x - 5)^2 = x^2 + 2 \cdot 5x + 5^2$
Então, passando o 5 para o outro lado, consegue as duas raizes, sendo $x_1 = 6$ e $x_2 = 4$

c) Para a alternativa C, vai ser utilizado a Fatoração.
$h(x) = x^2 - 15x$ pode ser escrita como $h(x) = x(x - 15)$ basta fazer a distributiva para encontrar o mesmo resultado.
Lembra que eu tinha citado que o c é o ponto em que a parábola corta o eixo Y, logo essa parábola esta flutuando em um quadrante, sendo assim uma de suas raizes é **0**, e a outra é **+15**.

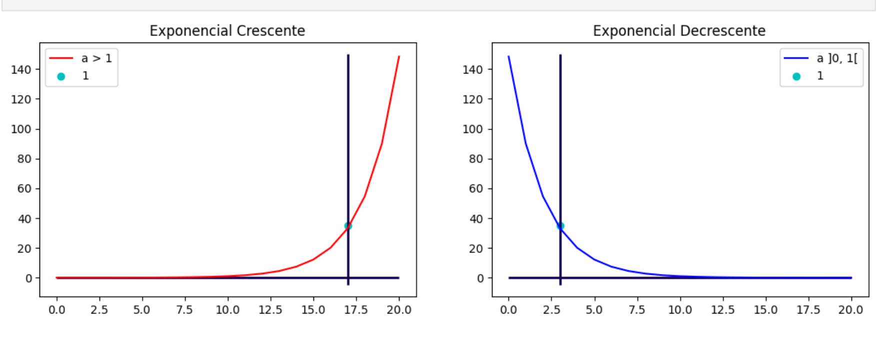
1.1.4. Função Exponencial

A função exponencial é dada pela fórmula $f(x) = a^x$
A carinha dessa função é como um lado da função do 2° grau, onde sempre vai cortar o eixo **Y** em 1, pois todo $a^0 = 1$.

In [363…

```
fig, (ax1, ax2) = plt.subplots( 1, 2, figsize=(13, 4) )
for i in zip( [[exp(p) for p in collect(-5:.5:5)], [exp(-p) for p in collect(-5:.5:5)]],
              [ax1, ax2], ["r", "b"], ["Exponencial Crescente", "Exponencial Decrescente"], ["a > 1", "a ]0, 1
    i[2].plot( i[1], color=i[3], label=i[5] );
    i[2].hlines( 0, 0, 20, color="#12004f", linewidth=2, linestyle="-");
    i[2].scatter( i[6], 35, c="c", label="1" )
    i[2].set_title(i[4])
    i[2].legend()
end
ax1.vlines( 17, -5, 150, color="#12004f", linewidth=2, linestyle="-");;
ax2.vlines( 3, -5, 150, color="#12004f", linewidth=2, linestyle="-");

plt.savefig("Exp.png")
```



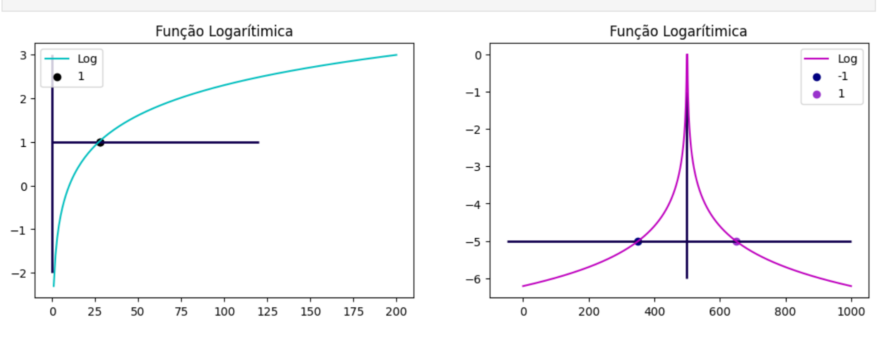
1.1.5. Função Logarítimica

A função logarítimica é dada pela fórmula $f(x) = logx$
A carinha dessa função é ao contrario da função exponencial onde corta o exiso **x** em 1.
Tem as mesmas propriedades da exponencial, porem agora em relação ao eixo x.

In [459…

```
fig, (ax1, ax2) = plt.subplots( 1, 2, figsize=(13, 4) )
for i in zip( [[log(i) for i in 0:.1:20], [-log(Complex(i)) for i in -50*10:1:50*10]], [ax1, ax2], ["c", "m"],
              [350, 650], [-1, 1], ["navy", "darkorchid"] )
    i[2].plot( i[1], color=i[3], label="Log" );
    i[2].set_title("Função Logarítimica")
    ax2.scatter( i[4], -5, c=i[6], label=i[5] )
    ax1.scatter( 28, 1, c="k", label="1" )
    i[2].legend();
end
ax1.vlines( 0, -2, 3, color="#12004f", linewidth=2, linestyle="-")
ax1.hlines( 1, 0, 120, color="#12004f", linewidth=2, linestyle="-")
ax2.hlines( -5, 10*100, -50, color="#12004f", linewidth=2, linestyle="-")
ax2.vlines( 499, -6, -1, color="#12004f", linewidth=2, linestyle="-");

plt.savefig("Log.png")
```



1.2. Limites

Limites nada mais é que um conceito para se estudar em mais detalhes sobre o infinito, vou passar apenas a explicação intuitiva sobre os limites.
Como que uma função se comporta dado um certo valor , é útil quando se tem um polinômio a saber qual é o termo dominante na expressão, a noção mais básica é que quanto mais eu me aproximo de um certo valo, não importa quanto meu me aproxime eu nuca vou chegar no destino, tipo um zoom infinito.

Imagina-se os pontos amerelhos estarem se aproximando do ponto em vermelho pela esquerda e os pontos em azuis pela direita o mais próximo possível dele mas nunca chegando nele, esse é o conceito de limite, resumidamente, o objetivo é chegar o mais próximo possível do ponto em vermelho mas nunca estar exatamente no ponto vermelho.

In [610…

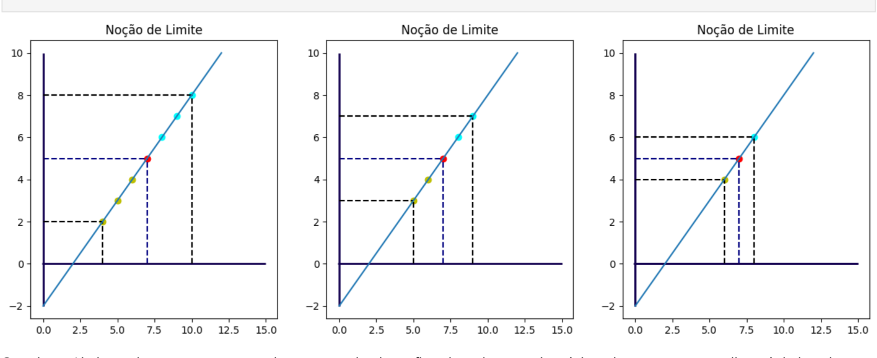
```
fig, (ax1, ax2, ax3) = plt.subplots( 1, 3, figsize=(15, 5)) # Precisa Melhorar KKKKKKKKK
for i in [ax1, ax2, ax3]
    i.plot( collect( -2:1:10 ) );
    i.vlines( 7, 0, 5, color="navy", linestyle="--" )
    i.hlines( 5, 0, 7, color="navy", linestyle="--" )
    i.vlines( 0, -2, 10, color="#12004f", linewidth=2, linestyle="-")
    i.hlines( 0, -.1, 15, color="#12004f", linewidth=2, linestyle="-");

    if i == ax1
        i.vlines( 10, 0, 8, color="k", linestyle="--" )
        i.hlines( 8, 0, 10, color="k", linestyle="--" )
        i.vlines( 4, 0, 2, color="k", linestyle="--" )
        i.hlines( 2, 0, 4, color="k", linestyle="--" )
        i.scatter( 7, 5, c="r" )
        i.scatter( 6, 4, c="y" )
        i.scatter( 5, 3, c="y" )
        i.scatter( 4, 2, c="y" )
        i.scatter( 8, 6, c="aqua" )
        i.scatter( 9, 7, c="aqua" )
        i.scatter( 10, 8, c="aqua" )
        i.set_title("Noção de Limite");

    elseif i == ax2
        i.vlines( 9, 0, 7, color="k", linestyle="--" )
        i.hlines( 7, 0, 9, color="k", linestyle="--" )
        i.vlines( 5, 0, 3, color="k", linestyle="--" )
        i.hlines( 3, 0, 5, color="k", linestyle="--" )
        i.scatter( 7, 5, c="r" )
        i.scatter( 6, 4, c="y" )
        i.scatter( 5, 3, c="y" )
        i.scatter( 8, 6, c="aqua" )
        i.scatter( 9, 7, c="aqua" )
        i.set_title("Noção de Limite");

    else
        i.vlines( 7, 0, 5, color="navy", linestyle="--" )
        i.hlines( 5, 0, 7, color="navy", linestyle="--" )
        i.vlines( 0, -2, 10, color="#12004f", linewidth=2, linestyle="-")
        i.hlines( 0, -.1, 15, color="#12004f", linewidth=2, linestyle="-");
        i.vlines( 8, 0, 6, color="k", linestyle="--" )
        i.hlines( 6, 0, 8, color="k", linestyle="--" )
        i.vlines( 6, 0, 4, color="k", linestyle="--" )
        i.hlines( 4, 0, 6, color="k", linestyle="--" )
        i.scatter( 7, 5, c="r" )
        i.scatter( 6, 4, c="y" )
        i.scatter( 8, 6, c="aqua" )
        i.set_title("Noção de Limite")
    end
end

plt.savefig("Limite")
```



Ou seja, no Limite onde o meu ponto amarelo a esta tendendo ou ficando cada vez mais próximo do ponto em vermelho v é dado pela seguinte expressão: $\lim_{a \to v} f(x)$ a palavra lim da essa noção, e o a é o número ou ponto que esta sendo aproximádo para o outro ponto que no caso é o vermelho v.
Matematicamente: Conforme o a se aproxima do v pera direita ou esquerda, a $f(a)$ se aproxima da $f(v)$.

A seguinte função $f(x) = x + 1$, no limite quando essa função tende a 1, fica como no DataFrame, logo escrita, fica $\lim_{x \to 1}(x + 1)$, isso é verdade para outros números também.

In [553…

```
DataFrame(x=[2, 1.5, 1.1, 1.01, 1.001], y=[3, 2.5, 2.1, 2.01, 2.001])
```

Out[553… 5 rows × 2 columns

	x	y
	Float64	Float64
1	2.0	3.0
2	1.5	2.5
3	1.1	2.1
4	1.01	2.01
5	1.001	2.001

1.3. Derivadas

Resumidamente a derivada é a reta tangente ao gráfico de uma função em um ponto escolhido, nome grande para uma coisa simples.
Essa ferramenta esta ligada a otimização, taxa de variação, onde uma variável depende de outra., ou seja, se eu tiver a velocidade e quiser a aceleração, vai ser necessário a derivação, resumidamente, vários probelmas caem na derivada.

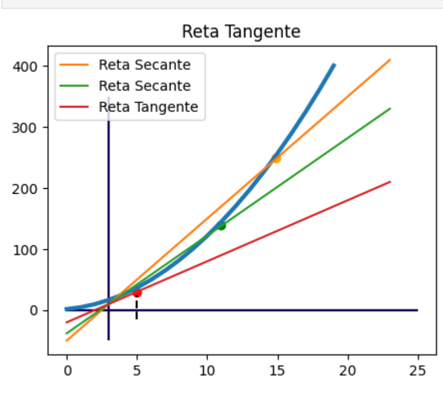
E a reta tangente é justamente uma reta em um ponto específico que corta uma função somente em uma parte, logo as demais retas que cortam em mais posições são chamadas de retas secantes.

In [616…

```
fig, ax = plt.subplots( figsize=(5, 4) )
ax.plot( [1p^2 + 1p + 1 for p in collect(:.5:20)], linewidth=3 );

for i in zip( [20, 16, 10], ["Reta Secante", "Reta Secante", "Reta Tangente"],
              [14.9, 11, 5], [250, 140, 30], ["orange", "g", "r"] )
    ax.plot( [i[1]*p + 10 for p in collect(-3:1:20)], label=i[2] )
    ax.hlines( 0, -.1, 25, color="#12004f" )
    ax.vlines( 3, -50, 350, color="#12004f" )
    ax.scatter( i[3], i[4], c=i[5] )
    ax.legend();
end
ax.vlines( 5, -15, 25, linestyle="--", color="k")
ax.set_title("Reta Tangente");

plt.savefig("Tangente");
```



1.3.1. Como surgiu a Derivada.

E com toda reta é possivel desenhar um triângulo e calcular a tangente dado os pontos, que é a diferença da altura pela diferença da base, no gráfico acima, vamos calcular essa diferença.

Ponto em Laranja: $tg = \frac{f(15) - f(5)}{15 - 5}$ a função avaliada no ponto 15 - a função avaliada no ponto 5 e a diferença, porém vai dar um erro grande essa diferença, logo precisa de um triângulo menor.
Ponto em Verde: $tg = \frac{f(11) - f(4)}{11 - 5}$ mesmo assim ainda é possível pegar um triangulo menor, e no limite, temos:
Ponto em Vermelho: $tg = \frac{f(5.00005) - f(5)}{5.00005 - 5}$, logo essa aqui esta mais perto da verdadeira mas é necessário formular no limite, pois alguem pode vim pegar um ponto menor, que consequentemente vai ser melhor e assim sucessivamente.

$f'(v) = \lim_{p \to v} \frac{f(p) - f(v)}{p - v}$ No limite em que o os outros pontos se aproximan do ponto vermelho, existe essa função, logo essa é a derivada da função no ponto vermelho.
Mas, geralmente é utilizada outra fórmula:
$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$ essa fórmula basicamente fala que depende do h que seria o passo, ou seja, o passo indo a 0, partindo dai, surge a derivada com o limite.

1.3.2. Derivada com o Limite.

```
III) Calcule a derivada utilizando o Limite.
```

a) $f(x) = 50x + 10$

$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$ agora colocando os pontos $f'(x) = \lim_{h \to 0} \frac{50(x+h) + 10 - (50x + 10)}{h}$
$f'(x) = \lim_{h \to 0} \frac{50x + 50h + 10 - 50x - 10)}{h}$ Agora é so cortar os termos e o h, acabou a derivação de uma função do primeiro grau.
$f'(x) = \lim_{h \to 0} 50$ ou seja, a $f(x) = ax + b = f'(x) = a$.

1.3.2. Regras de Derivação.

$f(x) = \mathbb{N}$ **a regra é** $f'(x) = 0$
$f(x) = x^n$ **a regra é** $f'(x) = n \cdot x^{n-1}$

$f(x) = sin(x)$ **a regra é** $f'(x) = cos(x)$
$f(x) = cos(x)$ **a regra é** $f'(x) = -sin(x)$

$f(x) = e^x$ **a regra é** $f'(x) = e^x$
$f(x) = ln(x)$ **a regra é** $f'(x) = \frac{1}{x}$

$(f(x) + g(x))'$ **a regra é** $f'(x) + g'(x)$

$(n \cdot f(x))'$ **a regra é** $n \cdot f'(x)$

$f(x) \cdot g(x)$ **a regra é** $f'(x) \cdot g(x) + f(x) \cdot g(x)'$

$f(x) \div g(x)$ **a regra é** $\frac{f'(x) \cdot g(x) - f(x) \cdot g'(x)}{g(x)^2}$

1.3.3. Exemplos das Regras de Derivação.

$f(x) = 2x^5 + ln(x) - sin(x)$ No primeiro elemento o 5 cai multiplicando o 2 e subtrai 1 do expoente (Regra 2), o $ln(x)$ é justamente $\frac{1}{x}$ e por fim, o $sin(x)$ é o $cos(x)$.

$f'(x) = 10x^4 + \frac{1}{x} - cos(x)$
$(f(x) = x^2) \cdot (g(x) = sin(x))$ Nesse caso vai ser utilizado a regrinha. $f'(x) \cdot g(x) + f(x) \cdot g'(x)$

$(x^2)' \cdot sin(x) + x^2 \cdot (sin(x))'$ agora deriva individualmente, logo $2x \cdot sin(x) + x^2 \cdot cos(x)$
$\frac{x^2}{sin(x)}$ ou $x^2 \div sin(x)$

$\frac{f'(x) \cdot g(x) - f(x) \cdot g'(x)}{g(x)^2}$ mudando os elementos $\frac{(x^2)' \cdot sin(x) - x^2 \cdot (sin(x))'}{sin(x)^2}$

$\frac{2x \cdot sin(x) - x^2 \cdot cos(x)}{sin(x)^2}$ para simplificar pode separar o $sin(x)^2$ em duas divisões e cortar o $sin(x)$ de cima da fração.

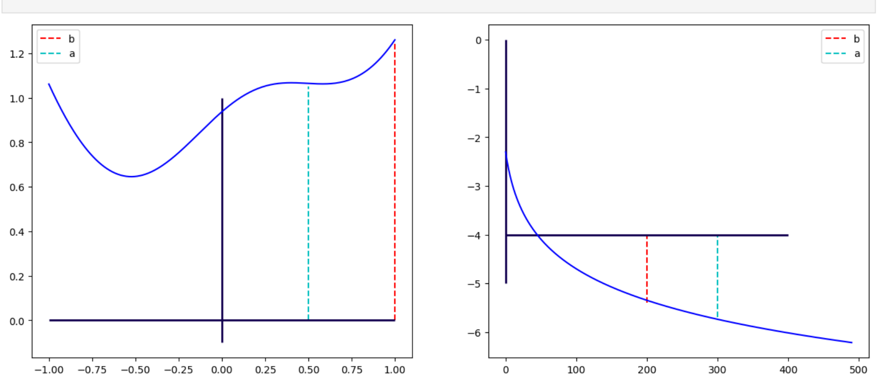
1.3. Integrais

A Integral é outra ferramenta matemática que pode ser utilizada para calcular a área de baixo de uma curva de uma função em pontos delçimitados por a e b.

In [806…

```
fig, (ax1, ax2) = plt.subplots( 1, 2, figsize=(15, 6))
for k in [ax1, ax2]
    if k == ax1
        x = collect(-1:.01*.8:1)
        k.plot(x, x.^2 + np.exp(-2*(x .- .18).^2), color="b");
        for i in zip( [1, .5], [1.25, 1.05], ["r", "c"], ["b", "a"] )
            k.vlines( i[1], 0, i[2], color=i[3], label=i[4], linestyle="--" )
        end
        k.hlines( 0, -1, 1, color="#12004f", linewidth=2 )
        k.vlines( 0, -.1, 1, color="#12004f", linewidth=2 )
    else
        x = [-log(i) for i in 1*10:1:50*10]
        k.plot( x, color="b" );
        for i in zip( [200, 300], [-5.4, -5.7], ["r", "c"], ["b", "a"] )
            k.vlines( i[1], i[2], -4, color=i[3], label=i[4], linestyle="--" )
        end
        k.hlines( -4, 0, 400, color="#12004f", linewidth=2 )
        k.vlines( 0, 0, -5, color="#12004f", linewidth=2 )
    end
    k.legend()
end

plt.savefig("Integrais")
```



$\int_a^b f(x)\, dx$ Integral da área acima do eixo X
$|\int_a^b f(x)\, dx|$ Integral da áream abaixo do eixo X

2.0. Referências

Hamilton Luiz Guidorizzi **Um cúrso de Cálculo Vol 1. 5 Edição**.
Link:
https://edisciplinas.usp.br/pluginfile.php/5580657/mod_resource/content/2/Um%20Curso%20de%20C%C3%A1lculo%20Vol%2001.pdf

Douglas Maiolli **Cálculo I - Limites Derivadas & Integrais**
Link: https://www.youtube.com/watch?v=EJ5FGPZbKeo&list=PLrOyM49ctTx8go5KFpSr-EMScIPygZNob

Geogebra **Geogebra Calculator**
Link: https://www.geogebra.org/calculator

THAIS MAGALHÃES SANTOS **Regras de Derivação**
Link: http://www.matematiques.com.br/conteudo.php?id=803

Geogebra **Geogebra Calculator**
Link: https://www.geogebra.org/calculator

Canal da USP **Cálculo I**
Link: https://www.youtube.com/playlist?list=PLAudUnJeNg4tr-aiNyYCXE46L3qEZ2Nzx