**O QUE DEVE SER IMPORTADO:**

%matplotlib inline

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import random

import scipy.stats

import intertools

from ipywidgets import interact, IntSlider, FloatSlider

**AULAS**

**Aula 1 – Introdução ao pandas**

* Funções básicas do pandas (sort\_values, head e etc)

**Aula 2 – Explorando variáveis qualitativas**

* Filtros em tabelas
* Construção de tabelas relativas e absolutas
* Construção de gráficos de barra, pizza e etc.

**Aula 3 – Explorando variáveis quantitativas**

* Tabela de frequência absoluta e relativa para variáveis contínuas
* Construção de histograma

**Aula 5 – Explorando duas variáveis quantitativas**

* Gráfico de dispersão (scatter plot)
* Ajuste de reta
* Uso de covariância e correlação

**Aula 6 – Aplicação de variáveis quantitativas**

* Como fazer boxplot
* Gráfico de barra

**Aula 8 – Algorítimo de Naive Bayes**

* Como calcular Naive Bayes através de um texto

**Aula 11 – Variáveis aleatórias discretas**

* Variância
* Valor esperado

**Aula 12 – Modelos probabilísticos discretos**

* Distribuição binomial
* Distribuição de Poisson

**Aula 13 – Poisson e Binomial**

* Comparação entre as funções
* Funções interativas para visualização

# Aula 14 - Distribuições contínuas – exponencial

# Comparação entre contínua e exponencial

# Aula 15 – Distribuição normal

# norm.\_\_

# ppf, pmf, etc

**FUNÇÕES DO PYTHON**

* Média = .mean()
* Mediana = .median()
* Desvio padrão = .std()
* Covariância = dados.variavel1.cov(dados.variavel2) – aula 5
* Correlação = dados.variavel1.corr(dados.variavel2) – aula 5
* Variância = lista.var()
* Valor máximo = lista.max()
* Valor mínimo = lista.min()
* Descritiva = .describe() – retorna média, mediana, valor máximo, valor mínimo, desvio padrão, primeiro quartil e terceiro quartil
* Boxplot = .boxplot(dados.coluna) – aula 6
* Conta número absoluto = value\_counts()
* Conta número relativo = value\_counts(True)
* Soma = .sum()
* Multiplica = .prod()
* Lista de possibilidades = intertools – aula 12

**DISCRETA X CONTÍNUA:**

* **Discreta:** Pontos em um gráfico

cdf, pmf

* **Contínua:** Uma linha (infinitos pontos)

cdf, ppf

**FUNÇÕES COM ESPECIFICAÇÃO**

* **Binomial – discreta (pontos em uma linha)**

**- Ou da certo ou não, n vezes**

n (Número de repetições) se n=1 será BERNOULLI

x (Número de sucessos)

P (Probabilidade de sucesso)

binom. \_ (x, n=\_\_, p=\_\_)

binom.\_\_(valor OU probabilidade, n=\_\_, p=\_\_)

* **Bernoulli – Sucesso/fracasso**

Ex**:** Sabendo que a probabilidade de Matheus ir para uma festa é p=0.5, qual a probabilidade de ela ir na MonteCarlo?

binom. \_ (x, n=1, p=\_\_)

norm.\_\_(valor OU probabilidade, n=1, p=\_\_)

* **Poisson – discreta (taxa por dia, taxa constante / Quando tem taxa no tempo)**

Ex: Se uma empresa fabrica 50 peças por mês, qual a probabilidade dela fabricar 60?

poisson. \_ (x, λ=média)

poisson.\_\_(valor OU probabilidade, λ=média)

**TIPO DE GRÁFICO**

* **Exponencial**

expon. \_ (x, scale=beta)

**β** = Taxa média = MÉDIA

**α** = Média de acontecimento por unidade de tempo

MÉDIA 🡪 **E(x) = β = 1/ α**

VARIÂNCIA 🡪 **Var(x) = DesvPadrão² = β² = 1/ α²**

expon.\_\_(valor OU prob, scale=média)

* **Normal**

norm. \_\_ (x, loc=**μ**, scale=**σ**)

norm.\_\_(valor OU prob, loc=média, scale=desviopadrão)

**O QUÊ PODE SER ESSE \_\_?**

**pmf** > entra com ponto retorna o valor da probabilidade naquele ponto, serve só para discreta.

EXEMPLO – Nem sempre é com norm!

norm.pmf(valor, loc=média, scale=desviopadrão)

**cdf** > é a cumulativa, entra com valor do ponto e retorna a probabilidade acumulada até ele (probabilidade de tudo antes dele)

Para descobrir os pontos depois do cdf, usar o 1 – norm.cdf

EXEMPLO – Nem sempre é com norm!

norm.cdf(valor, loc=média, scale=desviopadrão)

1 - norm.cdf(valor, loc=média, scale=desviopadrão)

**ppf** > dada uma probabilidade acumulada, encontra o ponto (inversa da cdf).

EXEMPLO – Nem sempre é com norm!

norm.ppf(probabilidade, loc=média, scale=desviopadrao)

**pdf** > criar uma distribuição (criar gráficos)

NESSE CASO: Usar fit para descobrir loc e scale

Usar linspace

EXEMPLO – Nem sempre é com norm!

media, desviopadrao = norm.fit(variável)

media, desviopadrao

x = np.linspace(ValorInicial, ValorFinal, númerodepontos)

y = norm.pdf(x, loc=media, scale=desviopadrao)

plt.figure()

plt.hist(variável, density=**True**, bins=15);

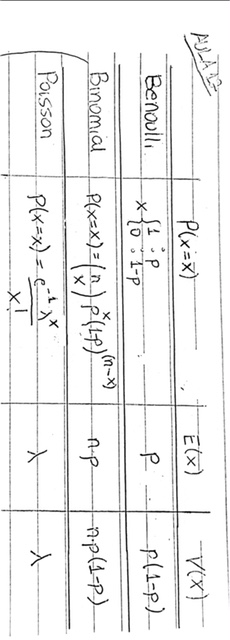
plt.plot(x, y, 'r-')

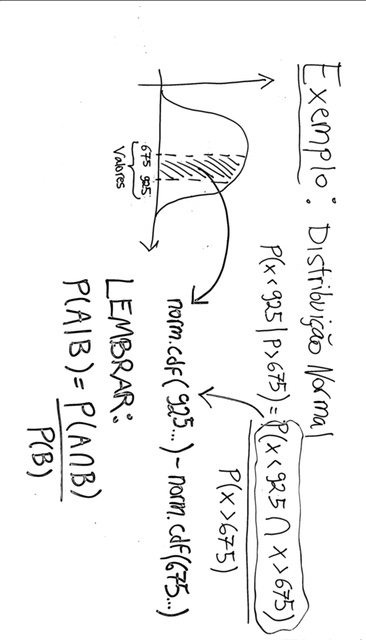
plt.show();

**OBS:**

* Quando a média e o desvio padrão não são dados, usar os valores padrão:

norm.pdf(x, loc=0, scale=1)



****

**IMPORTANTE:**

* Quando estivermos trabalhando com amostras (“amostral”) o scale passa a ser: **desviopadrao/n\*\*0.5**

**REGRESSÕES:**

**- Lembrar:**

**(Ho = Hipótese Nula)**

**(H1 = Outra hipótese)**

Resíduo

* Resíduo = “Erro”

F-statistic

* Ho: Assume que TODOS Bi(Beta)=0

H1: Existe algum Bi (i>1)

* Quando prob F-statistic < 10%(0,1), você rejeita Ho

Quando prob(F-statistic) >10%(0,1), você aceitamos Ho

P ItI

* Quando P ItI = 0 para UMA variável:
* Ho: Bi = 0 (Coeficiente não ajuda a prever o resultado)

H1: Bi ≠ 0 (Coeficiente ajuda a prever o resultado)

* Quando P ItI < 10%(0,1) você rejeita hipótese nula, aceitando e variável
* Quando P ItI > 10%(0,1) você aceita hipótese nula, rejeitando a variável

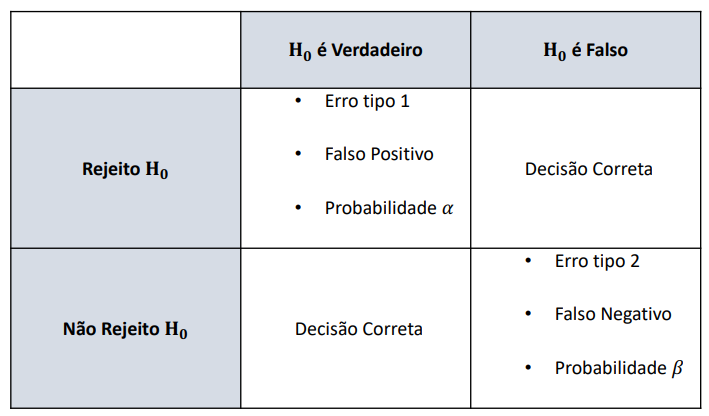
Omnibus e JB

* Teste de normalidade dos erros
* Ho: Resíduos são normais

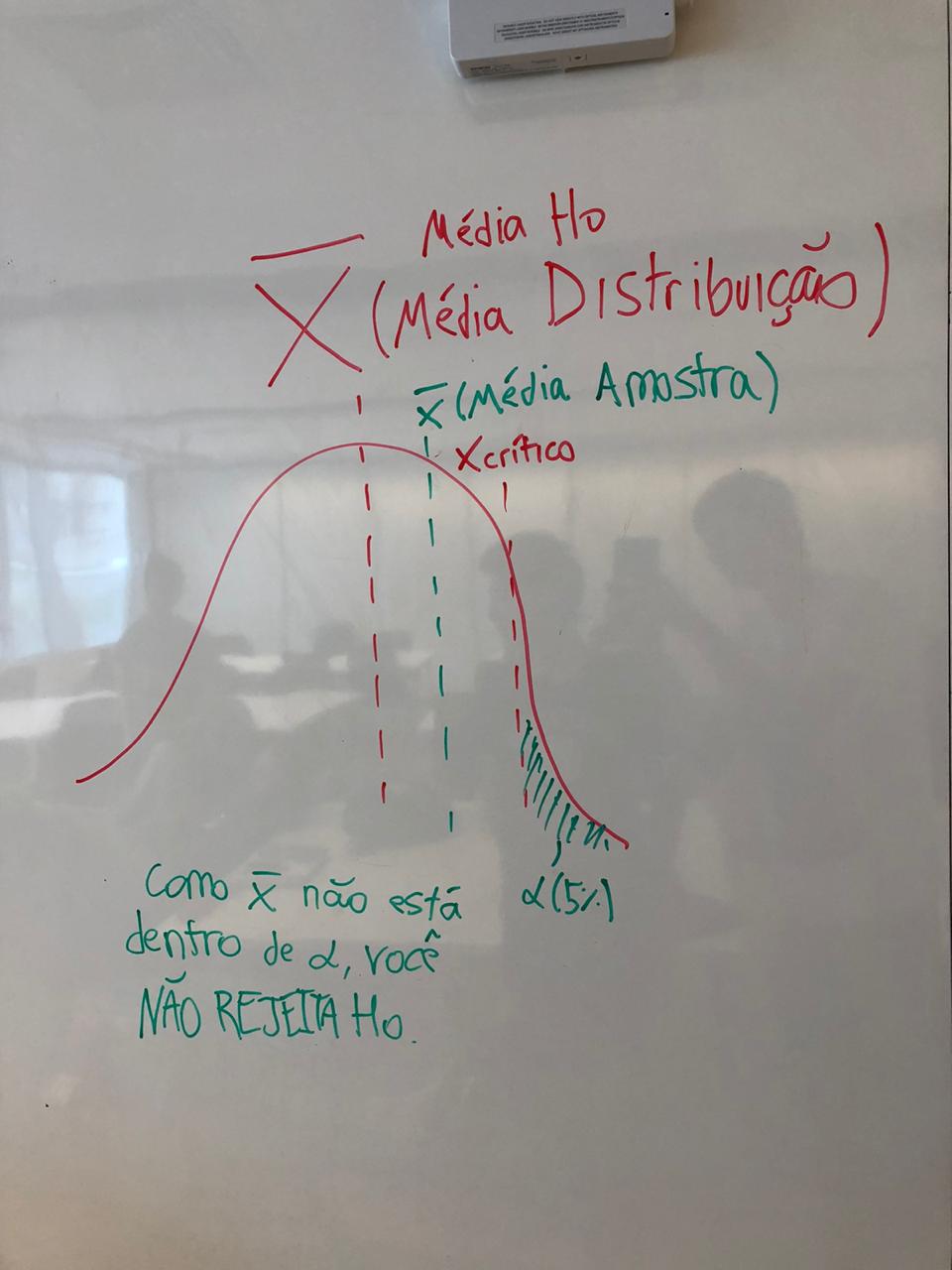
H1: Resíduos não são normais

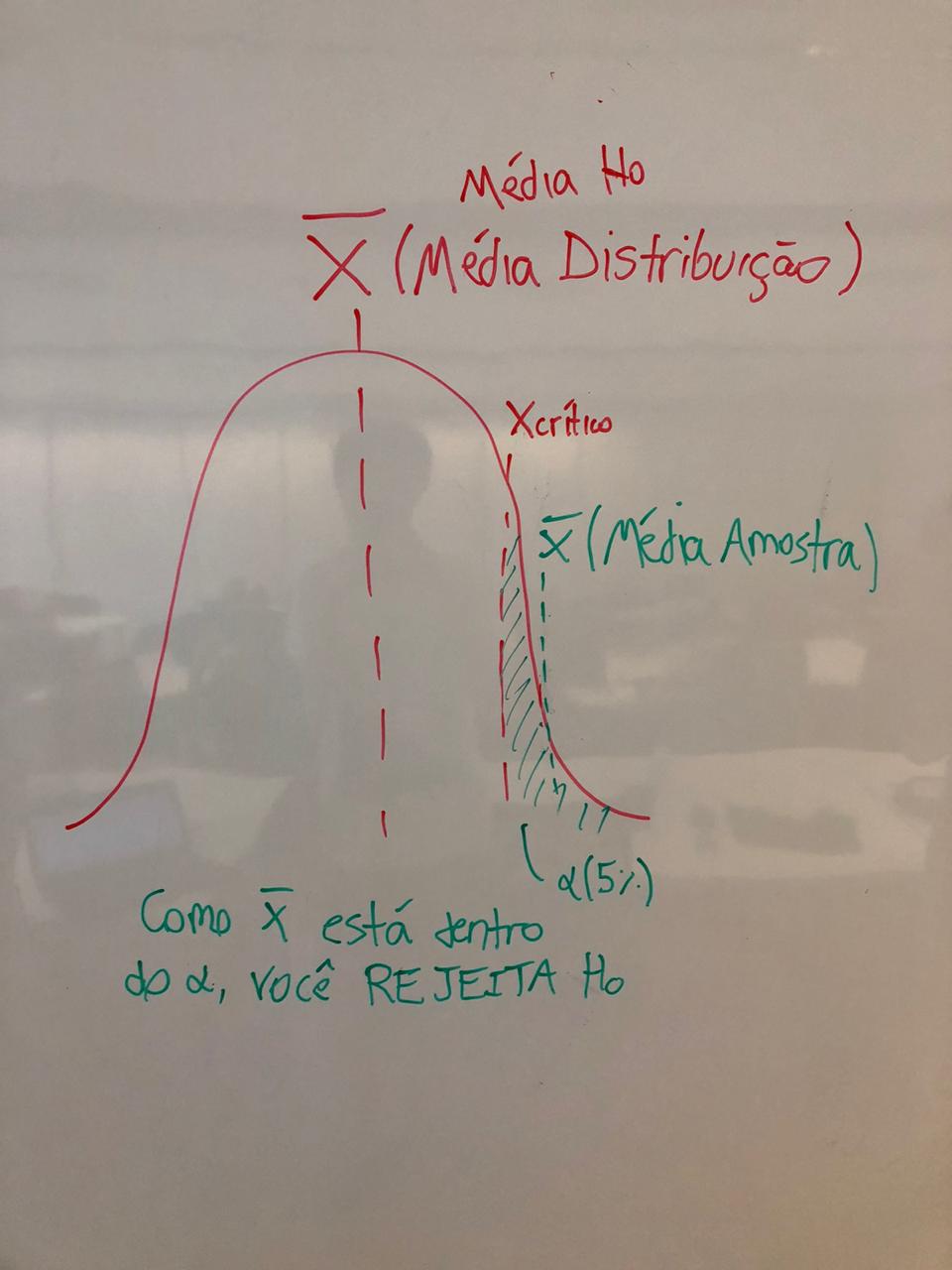
* Quando P(Omnubis) e Prob(JB) > 10%(0,1) você aceita hipótese nula
* Quando P(Omnubis) e Prob(JB) < 10%(0,1) você aceita hipótese nula

**TESTE DE HIPÓTESES:**

****

* **Falso Positivo**: Ho verdadeiro, mas você rejeita (Erro Tipo 1) - Prob Alpha
* **Falso Negativo**: Ho é falso, mas você aceita (não rejeita) (Erro Tipo 2)- Prob Beta





**DISTRIBUIÇÃO T**

*Quando usar:*

* Quando não se tem o desvio padrão da distribuição (pode ter o desvio padrão da amostra)
* Calculando o desvio padrão (dp) da distribuição t

dp = variável\_tabela["VARIÁVEL"].std(ddof=1)

dp\_distribuicao = dp/(n\*\*(1/2))

* Para calcular o cdf/ppf, usar a distribuição t

t.cdf(media\_amostra, loc=media\_distribuicao, scale=dp\_distribuicao, df=n-1)

* OBS:

Quando n>120, podemos considerar t como distribuição normal

Quando n<120, usar distribuição t