

## Aula 01 - Explorando o dataframe:

- Para ler o arquivo EXCEL:

`dados = pd.read_excel("Arquivo.xlsx")`

- Para ver a dimensão do DF:

`dados.shape` → (nº linhas, nº colunas)

- Para ver partes do DF:

`dados.head(nº linhas)` → mostra as n primeiras linhas

`dados.tail(nº linhas)` → mostra as n últimas linhas

- Para ordenar os dados pela coluna escolhida:

`dados.sort_values(by="coluna", ascending=True).head()`

- Para acessar diretamente 1 coluna:

`dados.COLUNA` ou `dados[ "COLUNA" ]`

- Para filtrar os dados pelas linhas/colunas:

`dados.loc[ "linhas", "colunas" ]`

OBS: ":" pega todos as linhas / colunas;

Para pegar mais de 1 linha / coluna usar lista.

- Para iterar no DF (dicionário):

`dados[ [ "coluna 1", "coluna 2" ] ][ linha 1 : linha n ]`

OBS: no slicing a última linha não conta.

- Para filtrar com condição:

`dados[ dados.COLUNA > n ]` → apenas as linhas em que a variável for maior que n

OBS: & é equivalente a and e | é equivalente a or

- Para contar valores de uma variável Qualitativa

`dados.COLUNA.value_counts( )`

↳ se True retorna as frequências relativas

- Para ouvir duas variáveis Qualitativas:

$ct = pd.crosstab(dados.COLUNA1, dados.COLUNA2)$

OBS: normalize = True retorna as frequências relativas

Verificar tipos de gráficos no Tutorial da aula 1.

## Aula 02 - Tipos de variáveis

- Para mudar o índice:

$dados.indx = dados.set_index("coluna")$

- Para juntar 2 DFs:

$df = dados_1.join(dados_2, how="inner", lsuffix="-coluna1", rsuffix="-coluna2")$

- Para iterar em uma variável específica:

$dados.loc["linha indice", "coluna"]$  ou

$dados["linha indice"]["coluna"]$

- Para descrever uma variável:

$dados["coluna"].describe()$

- Para descobrir o valor que separa os  $n\%$  maiores:

$dados["coluna"].quantile(0.9)$  → Vai de 0% até 100% ou

$mf.percentile(dados["coluna"], n)$  → Vai de 0% até 100%

- Para dividir o DF em  $n$  seções:

$df = pd.cut(dados["coluna"], bins=n)$

OBS: retorna uma lista

## Distribuições discretas:

- Binomial:  $P(Y=g): Prob = stats.binom.pmf(y, n, P)$

" $x$  sucessos em  $n$  repetições" →  $P(Y \leq y): Prob = stats.binom.cdf(y, n, P)$

$E(Y) = m, V = stats.binom.stats(n, P, moments="m V")$

$DP(Y) : dP = stats.binom.rstl(n, P)$

- Poisson:

" $k$  sucessos em evento que acontece com  $\rightarrow$  taxa fixa"

$\lambda \rightarrow$  média

$$P(X=k) = \frac{e^{-\lambda} \cdot \lambda^k}{k!}$$

$P(X=x)$ : Prob = stats.poisson.pmf( $x, \lambda$ )

$P(X \leq x)$ : Prob = stats.poisson.cdf( $x, \lambda$ )

$E(x)$ : E = stats.poisson.mean( $\lambda, loc=0$ )

$Var(x)$ : Var = stats.poisson.var( $\lambda, loc=0$ )

$DP(x)$ : dP = stats.poisson.std( $\lambda, loc=0$ )

- Bernoulli:

"Apenas 1 repetição"  $\rightarrow$

$$P(X=x) : P = p^x (1-p)^{1-x}; x=0 \text{ ou } x=1$$

$$E(x) = P$$

$$Var(x) = P \cdot (1-P)$$

## PROBABILIDADE SIMPLES !!!

### Distribuições Contínuas:

- Exponencial:

"Até  $x$  sucessos com taxa fixa"  $\rightarrow$

$\therefore$  A prob é cumulativa  
 $\therefore$  é a área sob o gráfico

$$F(x) = P(X \leq x) = 1 - e^{-\lambda x}, \text{ se } x \geq 0$$

$\hookrightarrow$  Cumulativa

$E(x)_e : loc, scale = expon.fit(dados['velma'])$

$DP(x) : \hookrightarrow beta$

$P(X=x) : P = expon.cdf(x, scale=beta)$

$X : x = expon.ppf(P, scale=beta)$

$f(x_i) : f_{xi} = expon.pdf(x_i, scale=beta)$

$\hookrightarrow$  Pdf só faz desenho!

- Normal:

"normalmente distribuída"  $\rightarrow$

$\therefore$  A prob é cumulativa  
 $\therefore$  é a área sob o gráfico

$E(x)_n : loc, scale = expon.fit(dados['velma'])$

$P(X=x) : P = norm.cdf(x, loc=loc, scale=scale)$

$X : x = norm.ppf(P, loc=loc, scale=scale)$

$f(x_i) : f_{xi} = norm.pdf(x_i, loc=loc, scale=scale)$

$\hookrightarrow$  Pdf só faz desenho!

Variáveis aleatórias:

- Discretas:

"Conjunto de possibilidades é um conjunto finito ou enumerável"

Ex: número de filhos, nº de salas de aula.

- Contínuas:

"Conjunto de possibilidades num intervalo ou conjuntos de intervalos contínuos"

Ex: tempo de duração, temperatura

Geração das funções:

Pmf  $\rightarrow$  Distribuição discreta  $\longrightarrow$  retorna a  $P(X=x)$

Polf  $\rightarrow$  Distribuição contínua  $\longrightarrow$  faz desenho

Cdf  $\rightarrow$  Cumulativa  $\longrightarrow$  retorna a P acumulada até x

PPf  $\rightarrow$  Cumulativa  $\longrightarrow$  retorna o x dado a P (oposto de cdf)

RVS  $\rightarrow$  qualquer  $\longrightarrow$  solta números aleatórios

Sintaxe:

From scipy.stats import:

binom.  $\hookrightarrow$  (n, m = , p = )

poisson.  $\hookrightarrow$  (n, mu =  $\lambda$ )

expovn.  $\hookrightarrow$  (n, scale = beta, loc = 0)

norm.  $\hookrightarrow$  (n, scale =  $\mu$ , scale =  $\sigma$ )

Eu acredito  
em Você!