

## Funções de XMAC02 - Prova 1

-Médias, medianas, modas, desvio padrão e quartis

```
import statistics as st
import seaborn as sns
df['x'].mean()
df['x'].median()
df['x'].mode()
df['x'].std()
q = st.quantiles(df['x'])
q
np.round -> arredonda o número que você possui
```

-Arrays

```
np.arange(número por onde começar, quantidade de números que a lista
vai ter)
np.linspace(número inicial, número final, quantidade de números no
intervalo deles)
np.x(número em inglês)((x,y)) -> matriz composta pelos mesmo número que
foi escrito no np
```

-Probabilidade

```
np.random.rand(X) -> irá gerar x números randômicos entre 0 e 1
np.random.randint(número inicial, número final + 1) -> irá gerar número
randômico entre os números que foram escritos. Se no número final for
escrito 2 e no inicial 0 por exemplo, irá gerar um número randômico
entre 0 e 1
np.random.randint(número inicial, número final + 1, número de números
que se quer gerar) -> pode-se colocar o total que se quer gerar
```

-Distribuição binomial(utilizada quando só há duas possibilidades de acontecimento)

```
np.random.binomial(quantidade de variáveis, taxa variável, número de
vezes que se quer testar)
binom.cdf(até onde se quer calcular(número mínimo até esse número),
quantidade de amostras, taxa de erro ou acerto) -> função de soma de
taxa acumulada, do número mínimo da amostra, até o número que foi
escrito no primeiro parâmetro
```

```
binom.pmf(número que se quer calcular a porcentagem, quantidade de amostras, taxa de erro ou acerto) -> mesmos parâmetros da função cdf, mas calcula a taxa naquele ponto específico que foi descrito no primeiro parâmetro  
binom.var(quantidade de amostras, taxa de erro ou acerto) -> variância entre a quantidade de amostras e a taxa de erro ou acerto  
binom.sf(x, y, z) ou 1 - binom.cdf(x, y, z) dá o mesmo resultado
```

### -Distribuição de Poisson

```
poisson.cdf(total que você procura, média da distribuição) - Cumulative distribution function - igual ou menos que  
poisson.pmf(total que você procura, média da distribuição) - Probability mass function - para o valor exato  
poisson.sf(total que você procura, média da distribuição) - Para mais que (similar a 1 - cdf)  
poisson.mean(total que você procura, média da distribuição) - Para a média da distribuição  
poisson.var(total que você procura, média da distribuição) - Para a variância da distribuição  
poisson.std(total que você procura, média da distribuição) - Para o desvio padrão da distribuição
```

### -Distribuição Normal(Usada quando se tem o desvio padrão)

```
np.random.normal(normalmente é o tamanho de algo que você tem, desvio padrão, número de vezes que você quer gerar)  
Funções:  
norm.cdf(número que você quer, número que você possui, desvio padrão) - Cumulative distribution function - igual ou menos que  
norm.pdf(número que você quer, número que você possui, desvio padrão) - Probability density function - para o valor exato  
norm.sf(número que você quer, número que você possui, desvio padrão) - Para mais que (similar a 1 - cdf)  
norm.mean(número que você possui) - Para a média da distribuição  
norm.var(número que você possui) - Para a variância da distribuição  
norm.std(número que você possui) - Para o desvio padrão da distribuição
```