**Estatística Com Python**

**Parte I – Frequências e Medidas:**

Todos os arquivos serão baixados, mas deixo o link do drive de qualquer forma: <https://drive.google.com/drive/folders/1MoAf3dLinpNBXVGIIyetaY5qfSwZ31Gt>.

1. **Aula 1 – Qual Seu Tipo de Dado:**
   1. Dependendo do tipo de dados análises diferentes serão feitas. Importante saber quais são os tipos.
      1. Quantitativos: Expressa quantidade de dados. Contagem, mensurações. Ex.:
         1. Idade;
         2. Renda;
         3. Altura.
      2. Qualitativos: Expressa qualidade dos dados. Ex.:
         1. UF;
         2. Sexo;
         3. Cor;
         4. Anos de Estudo.
      3. Podemos dividir os qualitativos em ordinais e nominais, e os quantitativos em discretos e contínuos.
   2. O que aprendemos:
      1. A identificar a versão de uma biblioteca do Python;
      2. A ler um dataset no formato CSV e criar um DataFrame pandas, com o conteúdo deste dataset;
      3. A identificar e classificar as variáveis de um dataset como quantitativas ou qualitativas;
      4. Que as variáveis qualitativas se dividem em ordinais e nominais;
      5. Que as variáveis quantitativas se dividem em discretas e contínuas.
2. **Aula 2 – Distribuição de Frequência:**
   1. S.value\_counts(): conta a quantidade de vezes que uma variável aparece na series. Podemos passar o parâmetro normalize = True, onde ele devolve a porcentagem de cada variável, onde ele soma elas e divide cada um pela soma:

dados['Sexo'].value\_counts(normalize=True) \* 100



* + 1. Podemos usar a porcentagem e a frequência para criar um df de distribuição e frequência qualitativas:

dist\_freq\_qualitativas = pd.DataFrame({'Frequência': frequencia, 'Porcentagem (%)': percentual})



* + 1. Df.rename\_axis(‘nome\_index’, axis=’columns’/’rows’, inplace=True): Renomeia a coluna do index:

dist\_freq\_qualitativas.rename(index= {0: 'Masculino', 1: 'Feminino'}, inplace=True)

dist\_freq\_qualitativas.rename\_axis('Sexo', axis='columns', inplace=True)

dist\_freq\_qualitativas



* 1. Pd.crosstab(variável\_linha, variável\_coluna): Cria um df onde faz o cruzamento dos dados da linha e da coluna passadas. No caso utilizamos cor e sexo, então ele irá mostrar na tabela quantas pessoas do sexo masculino tiveram de cada cor, bem como os femininos:

sexo = {0: 'Masculino',

        1: 'Feminino'}

cor = {0: 'Indígena',

        2: 'Branca',

        4: 'Preta',

        6: 'Amarela',

        8: 'Parda',

        9: 'Sem declaração'}

frequencia = pd.crosstab(dados.Sexo, dados.Cor)

frequencia.rename(index=sexo, columns=cor, inplace=True)

frequencia



* + 1. Podemos passar o mesmo parâmetro (normalize=True) que para value\_counts() e já receber o valor das porcentagens:

frequencia = pd.crosstab(dados.Sexo, dados.Cor, normalize=True) \* 100

frequencia



* + 1. Podemos ainda fazer algo bem mais legal. Podemos passar o valor de outra variável do nosso dataset para ser analisado nesses cruzamentos, ou seja, se escolhermos média da renda, por exemplo, ao invés de mostrar quantos homens e mulheres existem de cada cor, ele passa a exibir, em média, quanto os homens e mulheres ganham de renda. Para isso utilizaremos os parâmetros aggfunc=análise\_estat’ e values = dados[‘index’]:

frequencia = pd.crosstab(dados.Sexo, dados.Cor, aggfunc = 'mean', values = dados.Renda)

frequencia

