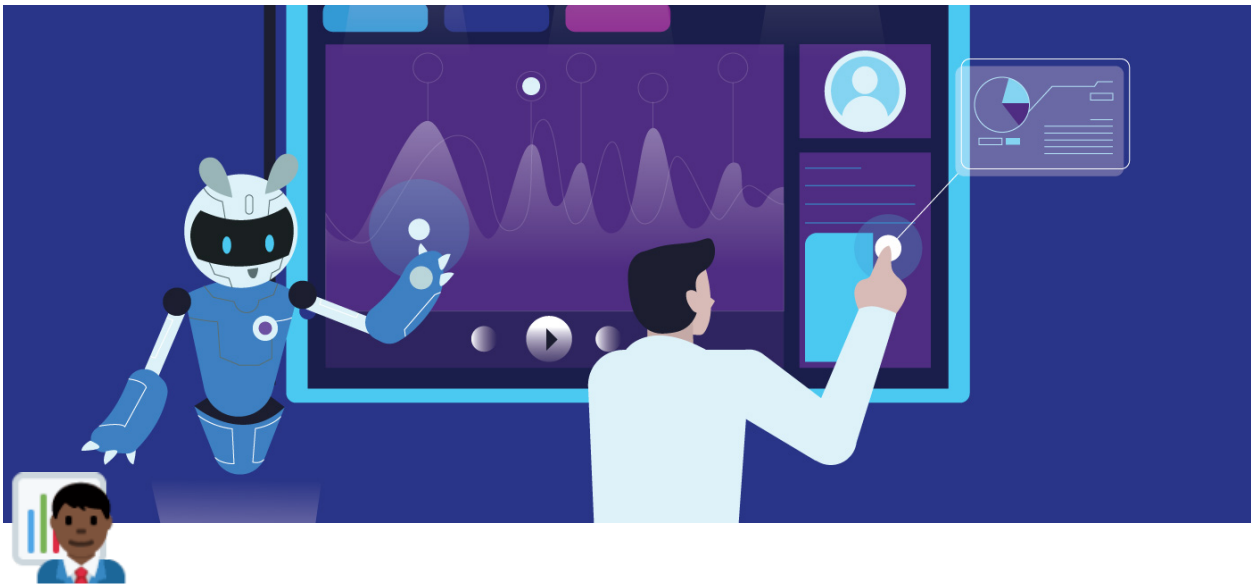


# TRATAMENTO DE DADOS

## Expressões Regulares com Python

No mês de novembro de 2019, a comunidade Data Hackers fez uma pesquisa sobre o mercado de Data Science no Brasil.

E quero tratar de uma das coluna que me chamou a atenção.

A coluna (P16) retorna esses valores

```
In [8]: df["('P16', 'salary_range')"].head()

Out[8]: 0    de R$ 1.001/mês a R$ 2.000/mês
        1     de R$ 2.001/mês a R$ 3000/mês
        2    de R$ 4.001/mês a R$ 6.000/mês
        3    de R$ 1.001/mês a R$ 2.000/mês
        4    de R$ 6.001/mês a R$ 8.000/mês
        Name: ('P16', 'salary_range'), dtype: object
```

Fui abordado com a seguinte pergunta : Qual a faixa salarial por idade ?

Para responder essa pergunta fui analisar o dataset e separar apenas as colunas de interesse e até adicionei a coluna que da situação de contrato (CLT, Freelancer, Empreendedor, etc).

```
In [6]: # faixa de salário
        faixa_de_salario = df["('P16', 'salary_range')"]
        # idade
        idade = df["('P1', 'age')"]
        # tipo de trabalho
        tipo_de_trabalho = df["('P10', 'job_situation')"]
```

```
In [21]: lista_dados = {
                        'idade': idade,
                        'sitação de trabalho': tipo_de_trabalho,
                        'salario': faixa_de_salario
                       }

         df_dados = pd.DataFrame(data = lista_dados)
         df_dados.head()
```

Out[21]:

| | idade | sitação de trabalho | salario |
|---|---|---|---|
| 0 | 37.0 | Empregado (CTL) | de R1.001/mêsaR 2.000/mês |
| 1 | 24.0 | Empregado (CTL) | de R2.001/mêsaR 3000/mês |
| 2 | 26.0 | Empregado (CTL) | de R4.001/mêsaR 6.000/mês |
| 3 | 21.0 | Estagiário | de R1.001/mêsaR 2.000/mês |
| 4 | 27.0 | Freelancer | de R6.001/mêsaR 8.000/mês |

Ao bater o olho no dataset, me pergunto: E agora, como tratar os dados da coluna salário para criar um gráfico ?

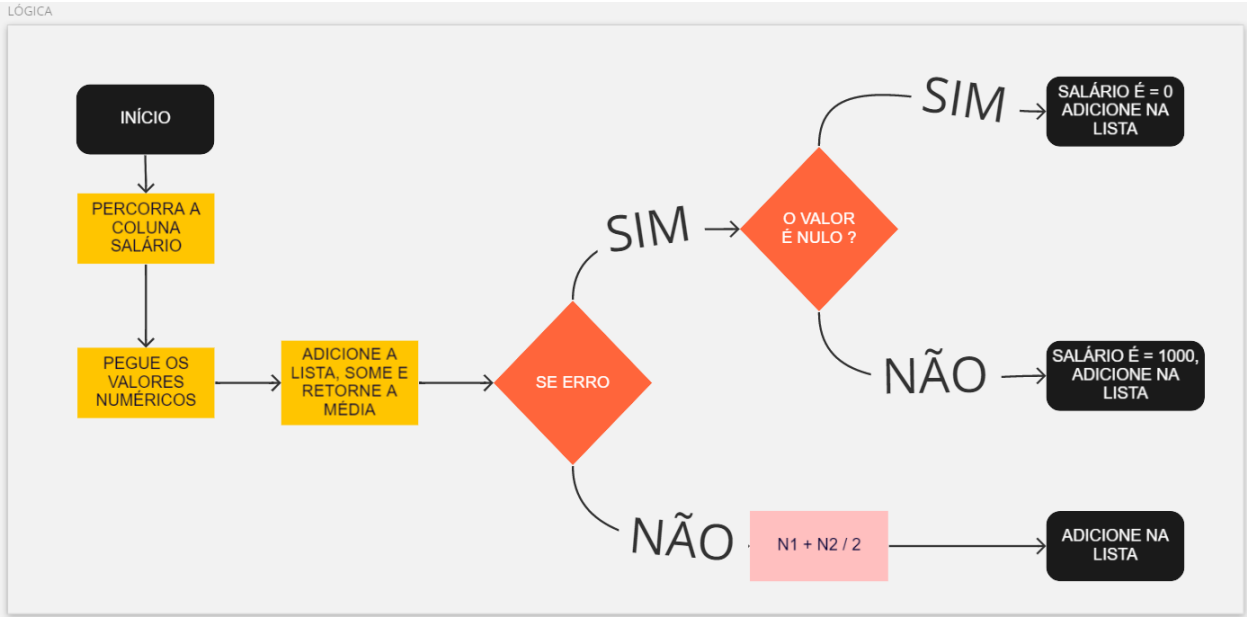
Então é nesse momento que lhe apresento o módulo re + a função findall() que já está incluída dentro do Python. Com essa linguagem é possível especificar regras no nosso caso utilizaremos a coluna "salario" para pegar apenas os números e continuar a solução.

Como a função findall() vai criar uma lista com os valores encontrados, pensei em criar uma coluna com o salário médio. Faz sentido para você ?

Documentação disponível em: https://docs.python.org/pt-br/3.8/howto/regex.html

## Tratando dados

A lógica que pensei para resolver o problema foi a seguinte :

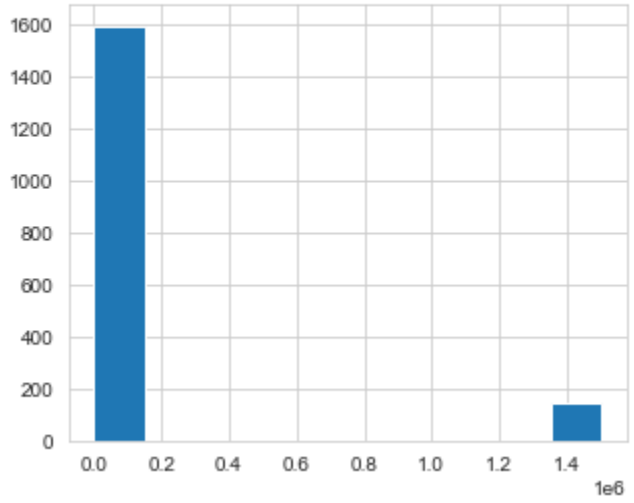



Pensando assim, desenvolvi o seguinte código

In [9]:
```
import re

x = 0
lista_media_de_salario = []
while x < len(faixa_de_salario):
    try:
        salario = faixa_de_salario[x]
        n = [float(s) for s in re.findall(r'-?\d+\.?\d*', salario)]
        n1 = n[0] * 1000
        n2 = n[1] * 1000
        salario_medio = (n1 + n2) / 2
        lista_media_de_salario.append(salario_medio)
    except:
        if pd.isna(faixa_de_salario[x]) == True:
            salario_medio = 0
            lista_media_de_salario.append(salario_medio)
        else:
            salario_medio = 1000
            lista_media_de_salario.append(salario_medio)

    x += 1

new_dataset = {'idade': idade.values,
               'tipo_de_trabalho': tipo_de_trabalho.values,
               'salario_medio': lista_media_de_salario}

df2 = pd.DataFrame(data=new_dataset)

df2.loc[df2['salario_medio'] == 1501000.5, 'salario_medio'] = 1500.5
```

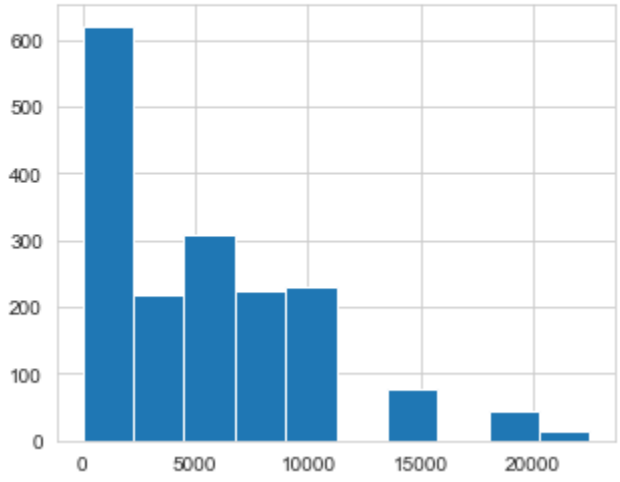
Repare que no final, adicionei uma outra verificação e o motivo dessa verificação é que os salários entre 2.001 á 3.000, está com um pequeno erro ou simplesmente faltando um ponto ( . ), sendo assim, acabava adicionando outliers indesejadas para o dataset e isso foi percebido ao imprimir um histograma, veja só o resultado antes da solução :

```
In [31]: plt.hist(df2['salario_medio'])
         plt.show()
```




Agora veja o resultado, depois da solução :

```
In [40]: plt.hist(df2['salario_medio'])
         plt.show()
```




Bem melhor, concorda ou não ?

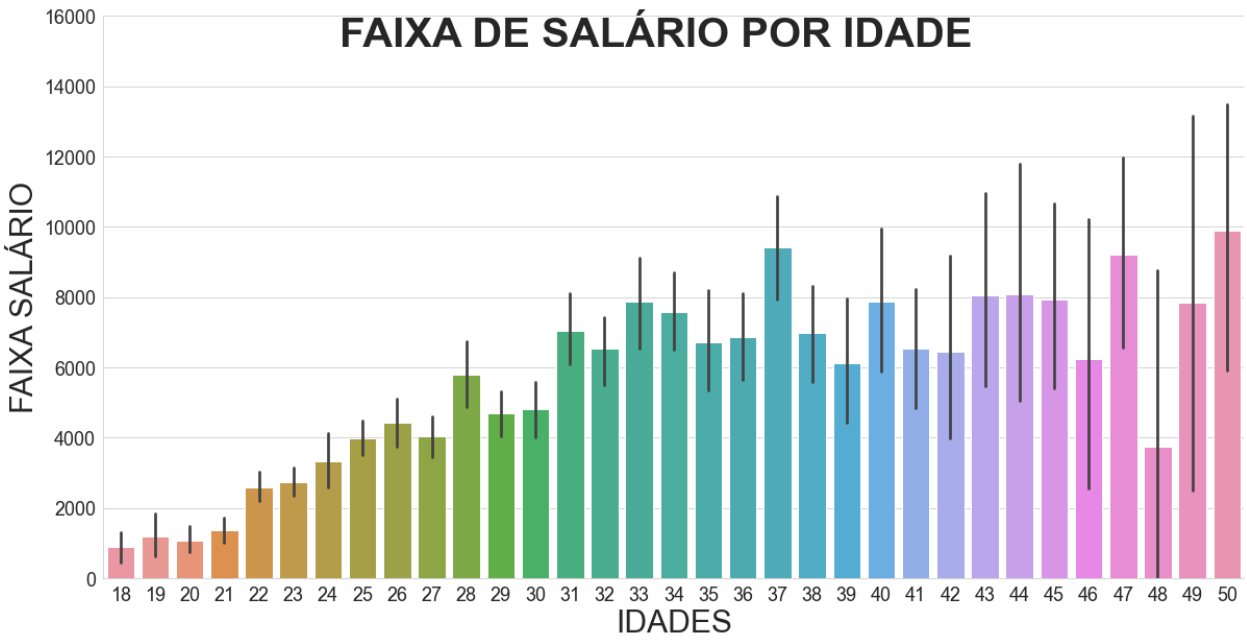
Já estamos perto de finalizar, vamos só verificar como ficou o dataset depois do código implementado :

```
In [37]: df2.head()
```

Out[37]:

| | idade | tipo_de_trabalho | salario_medio |
|---|---|---|---|
| **0** | 37 | Empregado (CTL) | 1500.5 |
| **1** | 24 | Empregado (CTL) | 1500.5 |
| **2** | 26 | Empregado (CTL) | 5000.5 |
| **3** | 21 | Estagiário | 1500.5 |
| **4** | 27 | Freelancer | 7000.5 |

Agora sim, senti confiança em plotar um gráfico e decidi que fosse um gráfico de barras, o resultado final foi o seguinte :




Finalizamos essa breve solução, com um gráfico bem bonito :)
Estarei disponibilizando este notebook no meu github, assim como este material.

Fique a vontade para comentar e colocar perguntas, críticas positivas e elogios também são bem vindos, o meu foco é evoluir e poder agregar valor a quem precisa.

Me siga nas redes sociais e saiba que gostaria muito de ter você como conexão no LinkedIn.

Forte abraço!

dataset : https://www.kaggle.com/datahackers/pesquisa-data-hackers-2019

Instagram : https://www.instagram.com/andrade_rafa93/

LinkedIn : https://www.linkedin.com/in/andraderafa/

Github : https://github.com/rafaelandradeslv