**Trabalho analítico de Jogos da Mega Senna.**

Autor - *João Luis Calmon Florentino*

|  |
| --- |
| LogoJoao-2019 |

1. Descrição das atividades desenvolvidas no Jupyter Notebook.

Comandos referentes ao notebook ***megaPrimeirosPassos.ipynb***:

**list(df)** - Comando que resulta em listar o nome de todas as colunas presentes no dataset pandas.

**df.info()** - Comando para obter os tipos ( Variáveis, int, float, string, etc.) do dataset pandas.

**new\_df = df.iloc[:, [0,2,3,4,5,6,7,9,12]]** - Comando que separa, no caso em um novo dataframe(new\_df) as colunas de números (0,2,3, ... etc) do dataframe original, através do comando **iloc**, onde o i no comando informa ao pandas que as colunas a serem procuradas serão passadas em ordem numérica, evitando desta forma que tenha que se digitar o nome completo da coluna.

Como este comando pode também localizar linhas, no inicio da lista de linhas e colunas vemos o : (dois pontos sem nenhum numero) que significa que deve ser procurar em todas as linhas do dataframe.

**new\_df.head(10) -** Mostra as dez primeiras linhas do dataframe ( número dentro do parênteses).

**new\_df.tail(5)** - Mostra as 5 últimas linhas do dataframe.

**new\_df.to\_csv("data/MegaPrimeiroFiltro.csv")** - Salva o dataframe indicado em um arquivo tipo csv.

**new\_df.isnull().sum()** - Exibe no dataframe indicado todos os valores contendo NaN ou sem informações (vazio, sem dados na célula) e em conjunto ao método sum() soma todos que forem encontrados e apresenta a listagem exibindo a quandtidade de NaN presentes no dataframe.

**new\_df.dropna(inplace=True)** - o comando dopna elimina todas as linhas que contem NaN do dataframe selecionado, e o **inplace=True** faz com que o resultado seja armazenado no próprio dataframe.

**indice = new\_df.loc[(new\_df['Ganhadores\_Sena'] == 0)]** - Neste comando, localizo no dataframe os jogos onde NÃO houveram ganhadores, ou seja a célula Ganhadores\_sena o valor é igual a zero, pois nas células onde algum jogo foi sorteado aparecerá o valor 1 ou mais. O resultado é armazenado na variável indice.

**df2 = new\_df.drop(indice.index)** - Neste comando retiro do dataframe new\_df os jogos perdedores e crio um dataframe contendo somente os jogos ganhadores.

**from IPython.core.display import HTML** - Este comando importa suporte ao python de linguagem HTML para que o resultado seja apresentado tanto em python quanto em HTML.

**display(HTML ('<h3 style= "color:blue"> Análise:</h3>'))** - Apresenta o código HTML dentro do parênteses.

**len(df)** - Informa a quantidade de linhas de um dataframe.

Comandos referentes ao notebook ***Analise-N1 Ganhadores***

**df.shape** - Este comando informa a quantidade de linhas e colunas de um dataframe.

**df2.size** - Informa a quantidade de células preenchidas (contendo informações) em um dataframe

**df3 = df2.apply(pd.Series.value\_counts)** - MUITO IMPORTANTE - Este comando avalia em cada coluna o numero de vezes que uma informação aparece. No caso, utilizei para me informar quantas vezes um numero foi sorteado em cada coluna de dezena para que pudesse ter a frequência de cada um dos 60 números da mega sena.

**df3.count()** - Este comando mostra a quantidade de valores diferentes que aparecem em um dataframe. Coluna por Coluna.

**df3["Frequencia"] = df3[list(df3.columns)].sum(axis=1)** - este comando esta criando uma coluna chamada frequencia no dataframe df3 e como dados ele soma todas as colunas (axis=1) do dataframe e deposita o resultado linha a linha nesta coluna.

**df3\_b = df3.sort\_values(["Frequencia"], ascending=False)** - Este comando ordena a coluna frequencia em determinada ordem, no caso, do maior para o menor (ascending=False).

**num = []**

**for i in range(1,61,1):**

**num.append(i)** - Esta sequencia de comandos está criando uma lista [] contento a sequencia de números de 1 até 60 de 1 em 1 valores.