**README - Processamento de Dados em DataFrame**

Este código em Python tem como objetivo realizar operações de processamento em um arquivo Excel e extrair dados específicos em diferentes formatos. Abaixo, descrevemos as etapas, o uso e o propósito de cada seção do código.

**Pré-requisitos**

Antes de executar este código, certifique-se de ter o Python instalado no seu sistema. Além disso, você deve ter as seguintes bibliotecas Python instaladas:

* pandas: Para manipulação de dados tabulares.
* numpy: Para operações numéricas.
* re: Para trabalhar com expressões regulares.

Você pode instalar essas bibliotecas usando o pip:

bash

pip install pandas numpy

**Uso**

**1. Leitura do Arquivo Excel**

O código começa lendo um arquivo Excel específico (no exemplo, "bmg2 (2).xlsx") usando o pandas. Ele pula as duas primeiras linhas e redefine os índices do DataFrame. As colunas do DataFrame são nomeadas a partir da primeira linha do arquivo.

**2. Remoção de Colunas Indesejadas**

O código remove a coluna "Nova comissão diferida" do DataFrame, que pode não ser necessária para o processamento. Certifique-se de ajustar essa etapa de acordo com o seu caso de uso.

**3. Concatenação de Colunas**

O código cria uma nova coluna chamada "Tabela" combinando as colunas "Logo" e "Descrição Entidade". Os dados são renomeados, e a coluna "Antecipação Equivalente Liquida " é renomeada para "Saque".

**4. Extração de Dados e Salvamento em Arquivos**

O código realiza as seguintes operações para criar e salvar arquivos específicos:

**a. Saque Sem Seguro**

* Cria um DataFrame saque\_sem\_seg com a coluna "Tabela" contendo informações de "Seguro Prestamista" e a coluna "Saque".
* Salva o DataFrame em um arquivo Excel chamado "Sem Seguro.xlsx".
* Fornece um botão de download para o arquivo gerado.

**b. Saque com Seguro Master**

* Cria um DataFrame saque\_segmaster com a coluna "Tabela" contendo informações de "Seguro Master" e a coluna "Saque".
* Salva o DataFrame em um arquivo Excel chamado "Seguro Master.xlsx".
* Fornece um botão de download para o arquivo gerado.

**c. Saque com Seguro Super**

* Cria um DataFrame saque\_segsuper com a coluna "Tabela" contendo informações de "Seguro Super" e a coluna "Saque".
* Salva o DataFrame em um arquivo Excel chamado "Seguro Super.xlsx".
* Fornece um botão de download para o arquivo gerado.

**d. Venda**

* Cria um DataFrame Venda com a coluna "Tabela" contendo informações de "VENDA" e a coluna "Comissão Venda".
* Salva o DataFrame em um arquivo Excel chamado "Venda.xlsx".
* Fornece um botão de download para o arquivo gerado.

**e. Ativação**

* Cria um DataFrame ativação com a coluna "Tabela" contendo informações de "ATIVAÇÃO" e a coluna "Comissão Ativação".
* Salva o DataFrame em um arquivo Excel chamado "ativação.xlsx".
* Fornece um botão de download para o arquivo gerado.

**5. Exportação do DataFrame Original**

* O DataFrame original com todas as modificações e colunas removidas é salvo em um arquivo Excel chamado "df.xlsx".

**Personalização**

Este código é um exemplo genérico e deve ser personalizado para atender às necessidades específicas do seu conjunto de dados. Certifique-se de ajustar as etapas de processamento, renomear colunas e criar novos DataFrames de acordo com os dados do seu arquivo Excel.

**Nota**

Este código faz uso da biblioteca Jupyter Notebook para exibir botões de download, portanto, se você não estiver usando um ambiente Jupyter, talvez não obtenha a mesma funcionalidade. Se você deseja executar o código fora de um ambiente Jupyter, você pode remover as partes relacionadas aos botões de download e, em vez disso, usar métodos de salvamento padrão oferecidos pelo pandas.

## Autor

Este código foi desenvolvido por [Luan Valente].