

Relatório PDF

Dicionario de dados ■

a: Identificador BRR
b: Plaqueta
c: Código PPA
d: Código licitação
e: Conta contábil
f: Serviço
g: Município
h: Descrição
i: Quantitativo
j: C_principal
k: C_EA
l: C_EA%
m: C_CA
n: C_CA%
o: C_imob
p: C_JOA
q: C_JOA%
r: WACC
s: Períodos_JOA [meses]
t: C_JOAR%
u: C_JOAR
v: CH
w: Data imob
x: Data oper
y: Data-base laudo
z: Data-base BRR
aa: IPCA_imob
ab: Data_IPCA_final
ac: IPCA_final
ad: Delta_IPCA
ae: CHC
af: Código contrato
ag: Prazo contrato [anos]
ah: Vida útil física [anos]
ai: Vida útil regulatória [critério]
aj: Vida útil regulatória [anos]
ak: TDR [anual]
al: TDR [mensal]
am: DeltaT_oper [meses]
an: DRA
ao: CHC_liquido
ap: IA
aq: Baixa
ar: Qtde_baixa
as: Data_baixa
at: Delta_baixa
au: Elegível QRR
av: Elegível juros
aw: BRR_bruta
ax: BRR_liquida
ay: QRR [anual]
az: Juros [anual]

Formulas:
Coluna b (Plaqueta):
b=Plaqueta+1

Coluna k (C_EA):
k=C_principal*C_EA%

Coluna m (C_CA):
m=C_principal*C_CA%

Coluna o (C_imob):
o=C_principal+C_EA+C_CA

Coluna q (C_JOA%):
q=C_JOA/C_imob

Coluna r (WACC):
r=WACC

Coluna u (C_JOAR):
u=C_imob*C_JOAR%

Coluna v (CH):
v=C_imob+C_JOAR

Coluna z (Data-base BRR):
z=Data-base BRR

Coluna ab (Data_IPCA_final):
ab=Data-base laudo

Coluna ad (Delta_IPCA):
ad=ROUND(IPCA_final/IPCA_imob-1;4)

Coluna ae (CHC):
ae=CH*(1+Delta_IPCA)

Coluna aj (Vida útil regulatória [anos]):
aj=IF(Vida útil regulatória [critério]="Contrato";Prazo contrato [anos];IF(Vida útil regulatória [critério]="Nãoodeprecia";0;Vida útil física [anos]))

Coluna ak (TDR [anual]):
ak=IF(Vida útil regulatória [anos]<>0;1/Vida útil regulatória [anos];0)

Coluna al (TDR [mensal]):
al=TDR [anual]/12

Coluna am (DeltaT_oper [meses]):
am=IF((Data-base BRR-Data oper)/365*12>Vida útil regulatória [anos]*12,Vida útil regulatória [anos]*12,(Data-base BRR-Data oper)/365*12)

Coluna an (DRA):
an=TDR [mensal]*DeltaT_oper [meses]*CHC

Coluna ao (CHC_liquido):
ao=CHC-DRA

Coluna at (Delta_baixa):

Coluna au (Elegível QRR):
au=IF(Vida útil regulatória [critério]="Nãoodeprecia";"Não";"Sim")

Coluna aw (BRR_bruta):
aw=IF(Data-base BRR>Data oper;IF(Elegível QRR="Sim";IF(CHC_liquido>0;IF(Baixa="Sim";IF(Qtde_baixa>Quantitativo;0;CHC*IA*(Delta_baixa)+CHC*IA*(1-Qtde_baixa/Quantitativo)*(1-Delta_baixa)),CHC*IA),0),0),0)

Coluna ax (BRR_liquida):
ax=IF(Data-base BRR>Data oper;IF(Elegível juros="Sim";IF(CHC_liquido>0;IF(Baixa="Sim";IF(Qtde_baixa>Quantitativo;0;CHC_liq
uido*IA*(Delta_baixa)+CHC_liquido*IA*(1-Qtde_baixa/Quantitativo)*(1-Delta_baixa)),CHC_liquido*IA),0),0),0)

Coluna ay (QRR [anual]):
ay=IF(Elegível QRR="Sim";BRR_bruta*TDR [anual];0)

Coluna az (Juros [anual]):
az=IF(Elegível juros="Sim";BRR_liquida*WACC;0)

leitura_s1.py ■

Código que faz a leitura da planilha e gera os prompts.

Inputs:

filename (em várias funções):

Representa o caminho do arquivo Excel sendo processado.

target_line (em remove_lines):

Indica a linha específica a ser removida do arquivo de texto.

texto (em verifica_formula, substituir_sheet, substituir_virgula_por_ponto_virgula, substituir_ponto_por_virgula, remove_space, remove_num):

Representa uma string de entrada para diversas manipulações.

lista (em sao_todas_strings):

Lista de elementos para verificar se todos são strings.

n (em int_to_col_letter):

Número a ser convertido para uma letra de coluna do Excel.

val (em trocar_por_chave):

Valor a ser trocado por sua chave correspondente em um dicionário.

f (em gerar_prompt):

Fórmula do Excel usada como base para gerar o prompt.

- regex_var:

str (objeto = "") -> str

str (bytes_or_buffer [, codificação [, erros]]) -> str

Crie um novo objeto String a partir do objeto especificado. Se codificar ou

erros são especificados, então o objeto deve expor um buffer de dados

Isso será decodificado usando a codificação e manipulador de erros fornecidos.

Caso contrário, retorna o resultado do objeto .__str__() (se definido)

ou repr (objeto).

codificando padrões para sys.getdefaultencoding().

erros padronizam para 'rigoroso'.

- a:

Dados tabulares bidimensionais, de tamanho grande e potencialmente heterogêneo.

A estrutura de dados também contém eixos marcados (linhas e colunas).

As operações aritméticas estão alinhadas nos rótulos das linhas e colunas. Pode ser

pensado como um contêiner do tipo dict para objetos de série. O primário

Estrutura de dados de pandas.

- C:

str (objeto = "") -> str

str (bytes_or_buffer [, codificação [, erros]]) -> str

Crie um novo objeto String a partir do objeto especificado. Se codificar ou

erros são especificados, então o objeto deve expor um buffer de dados

Isso será decodificado usando a codificação e manipulador de erros fornecidos.

Caso contrário, retorna o resultado do objeto .__str__() (se definido)

ou repr (objeto).

codificando padrões para sys.getdefaultencoding().

erros padronizam para 'rigoroso'.

- Cell_value:

- col_name:

str (objeto = "") -> str

str (bytes_or_buffer [, codificação [, erros]]) -> str

Crie um novo objeto String a partir do objeto especificado. Se codificar ou

erros são especificados, então o objeto deve expor um buffer de dados

Isso será decodificado usando a codificação e manipulador de erros fornecidos.

Caso contrário, retorna o resultado do objeto .__str__() (se definido)

ou repr (objeto).

codificando padrões para sys.getdefaultencoding().

erros padronizam para 'rigoroso'.

- col_name_print:

str (objeto = "") -> str

str (bytes_or_buffer [, codificação [, erros]]) -> str

Crie um novo objeto String a partir do objeto especificado. Se codificar ou erros são especificados, então o objeto deve expor um buffer de dados. Isso será decodificado usando a codificação e manipulador de erros fornecidos. Caso contrário, retorna o resultado do objeto `.__str__()` (se definido) ou `repr()` (objeto).

codificando padrões para `sys.getDefaultEncoding()`.

erros padronizam para 'rigoroso'.

- `col_values`:

NDarray unidimensional com rótulos de eixo (incluindo séries temporais).

Os rótulos não precisam ser únicos, mas devem ser do tipo hashable. O objeto suporta indexação baseada em inteiro e rótulo e fornece uma série de

Métodos para executar operações envolvendo o índice. Estatística

Os métodos da NDARRAY foram substituídos para excluir automaticamente

Dados ausentes (atualmente representados como NAN).

- Coluna:

`str(objeto = "") -> str`

`str(bytes_or_buffer[, codificação[, erros]]) -> str`

Crie um novo objeto String a partir do objeto especificado. Se codificar ou

erros são especificados, então o objeto deve expor um buffer de dados

Isso será decodificado usando a codificação e manipulador de erros fornecidos.

Caso contrário, retorna o resultado do objeto `.__str__()` (se definido)

ou `repr()` (objeto).

codificando padrões para `sys.getDefaultEncoding()`.

erros padronizam para 'rigoroso'.

- `colunas_i`:

Sequência mutável embutida.

Se nenhum argumento for fornecido, o construtor cria uma nova lista vazia.

O argumento deve ser um iterável se especificado.

- correspondências:

Sequência mutável embutida.

Se nenhum argumento for fornecido, o construtor cria uma nova lista vazia.

O argumento deve ser um iterável se especificado.

- `D_SHEETS`:

`dict()` -> novo dicionário vazio

`dict(mapeamento)` -> Novo dicionário inicializado a partir de um objeto de mapeamento

(chave, valor) pares

`dict(iterable)` -> Novo dicionário inicializado como se via:

`d = {}`

Para K, V em Iterable:

`d[k] = V`

`dict(**kwargs)` -> Novo dicionário inicializado com o nome = pares de valor

na lista de argumentos de palavras-chave. Por exemplo: `dict(um = 1, dois = 2)`

- `df`:

Dados tabulares bidimensionais, de tamanho grande e potencialmente heterogêneo.

A estrutura de dados também contém eixos marcados (linhas e colunas).

As operações aritméticas estão alinhadas nos rótulos das linhas e colunas. Pode ser

pensado como um contêiner do tipo dicto para objetos de série. O primário

Estrutura de dados de pandas.

- `IDX`:

`int([x])` -> inteiro

`int(x, base = 10)` -> Inteiro

Converter um número ou string em um número inteiro ou retornar 0 se não houver argumentos

são dados. Se x for um número, retorne `x.__int__()`. Para ponto flutuante

Números, isso trunca em direção a zero.

Se x não for um número ou se a base for fornecida, então x deve ser uma string,

bytes, ou instância de bytearray representando um inteiro literal no

dada base. O literal pode ser precedido por '+' ou '-' e ser cercado

por espaço em branco. Os padrões de base para 10. As bases válidas são 0 e 2-36.

A base 0 significa interpretar a base da string como um número inteiro literal.

```
>>> int('0b100', base = 0)
```

```
4
```

- `int_to_col_letter`:

Converte um número para uma letra de coluna do Excel.

- `k`:

```
str (objeto = "") -> str
```

```
str (bytes_or_buffer [, codificação [, erros]]) -> str
```

Crie um novo objeto String a partir do objeto especificado. Se codificar ou

erros são especificados, então o objeto deve expor um buffer de dados

Isso será decodificado usando a codificação e manipulador de erros fornecidos.

Caso contrário, retorna o resultado do objeto `.__str__()` (se definido)

ou `repr(objeto)`.

codificando padrões para `sys.getDefaultEncoding()`.

erros padronizam para 'rigoroso'.

- `eu`:

```
str (objeto = "") -> str
```

```
str (bytes_or_buffer [, codificação [, erros]]) -> str
```

Crie um novo objeto String a partir do objeto especificado. Se codificar ou

erros são especificados, então o objeto deve expor um buffer de dados

Isso será decodificado usando a codificação e manipulador de erros fornecidos.

Caso contrário, retorna o resultado do objeto `.__str__()` (se definido)

ou `repr(objeto)`.

codificando padrões para `sys.getDefaultEncoding()`.

erros padronizam para 'rigoroso'.

- `letra`:

```
str (objeto = "") -> str
```

```
str (bytes_or_buffer [, codificação [, erros]]) -> str
```

Crie um novo objeto String a partir do objeto especificado. Se codificar ou

erros são especificados, então o objeto deve expor um buffer de dados

Isso será decodificado usando a codificação e manipulador de erros fornecidos.

Caso contrário, retorna o resultado do objeto `.__str__()` (se definido)

ou `repr(objeto)`.

codificando padrões para `sys.getDefaultEncoding()`.

erros padronizam para 'rigoroso'.

- `nome`:

```
str (objeto = "") -> str
```

```
str (bytes_or_buffer [, codificação [, erros]]) -> str
```

Crie um novo objeto String a partir do objeto especificado. Se codificar ou

erros são especificados, então o objeto deve expor um buffer de dados

Isso será decodificado usando a codificação e manipulador de erros fornecidos.

Caso contrário, retorna o resultado do objeto `.__str__()` (se definido)

ou `repr(objeto)`.

codificando padrões para `sys.getDefaultEncoding()`.

erros padronizam para 'rigoroso'.

- `Padrao`:

```
str (objeto = "") -> str
```

```
str (bytes_or_buffer [, codificação [, erros]]) -> str
```

Crie um novo objeto String a partir do objeto especificado. Se codificar ou

erros são especificados, então o objeto deve expor um buffer de dados

Isso será decodificado usando a codificação e manipulador de erros fornecidos.

Caso contrário, retorna o resultado do objeto `.__str__()` (se definido)

ou `repr(objeto)`.

codificando padrões para `sys.getDefaultEncoding()`.

erros padronizam para 'rigoroso'.

- `caminho`:

```
str (objeto = "") -> str
```

```
str (bytes_or_buffer [, codificação [, erros]]) -> str
```

Crie um novo objeto String a partir do objeto especificado. Se codificar ou erros são especificados, então o objeto deve expor um buffer de dados. Isso será decodificado usando a codificação e manipulador de erros fornecidos. Caso contrário, retorna o resultado do objeto `.__str__()` (se definido) ou `repr()` (objeto).
codificando padrões para `sys.getDefaultEncoding()`.
erros padronizam para 'rigoroso'.

- `path_txt`:

`str(objeto = "") -> str`

`str(bytes_or_buffer[, codificação[, erros]]) -> str`

Crie um novo objeto String a partir do objeto especificado. Se codificar ou erros são especificados, então o objeto deve expor um buffer de dados. Isso será decodificado usando a codificação e manipulador de erros fornecidos. Caso contrário, retorna o resultado do objeto `.__str__()` (se definido) ou `repr()` (objeto).
codificando padrões para `sys.getDefaultEncoding()`.
erros padronizam para 'rigoroso'.

- `PD`:

PANDAS - Um pacote Python que fornece dados rápidos, flexíveis e expressivos estruturas projetadas para tornar o trabalho com dados "relacionais" ou "rotulados" ambos fácil e intuitivo. Ele pretende ser o bloco de construção fundamental de alto nível para Fazendo prática, **Análise de dados do mundo real** em Python. Além disso, tem O objetivo mais amplo de se tornar **os dados de código aberto mais poderosos e flexíveis** Ferramenta de análise / manipulação disponível em qualquer idioma. Já está bem em seu caminho para esse objetivo.

- `Re`:

Suporte para expressões regulares (RE).

Este módulo fornece operações de correspondência de expressão regulares semelhantes a aqueles encontrados em Perl. Ele suporta seqüências de 8 bits e unicode; ambos O padrão e as cordas que estão sendo processadas podem conter bytes nulos e Personagens fora da faixa ASCII dos EUA.

- `Sheet_names`:

Sequência mutável embutida.

Se nenhum argumento for fornecido, o construtor cria uma nova lista vazia.

O argumento deve ser um iterável se especificado.

- `Sheet_anges`:

Representa uma planilha.

- `v`:

Cria um novo objeto string a partir do objeto fornecido. Se codificação ou erros forem especificados, o objeto deve expor um buffer de dados que será decodificado usando a codificação e o manipulador de erros fornecidos. Caso contrário, retorna o resultado de `object.str()` (se definido) ou `repr(object)`.

- `var`:

Sequência mutável embutida.

Se nenhum argumento for fornecido, o construtor cria uma nova lista vazia. O argumento deve ser um iterável, se especificado.

- `verifica_formula`:

Verifica se uma string representa uma fórmula do Excel.

- `wb`:

A Pasta de trabalho é o contêiner para todas as outras partes do documento.

- `written`:

Cria uma coleção não ordenada de elementos únicos.

Funções Principais:

`sao_todas_strings(lista)`:

Verifica se todos os elementos em uma lista são strings.

`remove_lines(filename, target_line)`:

Remove linhas específicas de um arquivo de texto.

`verifica_formula(texto)`:

Verifica se uma string representa uma fórmula do Excel.

`substituir_sheet(texto)`:

Substitui referências a planilhas por identificadores específicos.

`substituir_virgula_por_ponto_virgula(texto):`

Substitui vírgulas por ponto e vírgula em uma string.

`substituir_ponto_por_virgula(texto):`

Substitui pontos por vírgulas em números decimais.

`remove_space(string):`

Remove espaços de uma string.

`remove_num(string):`

Remove números de uma string.

`int_to_col_letter(n):`

Converte um número para uma letra de coluna do Excel.

`trocar_por_chave(val, dct):`

Troca um valor em um dicionário por sua chave correspondente.

`extract_lines(filename, target_variable):`

Extrai linhas de um arquivo de texto até encontrar um delimitador.

`gerar_prompt(f):`

Gera um prompt para documentação de função Python com base em uma fórmula do Excel.

Prompt Gerado:

faça uma função em python para a formula de excel: [fórmula]

insira uma documentação da função indicando a correspondência de cada variável, preservando os nomes originais escritos na tabela abaixo e usando as letras como atributos, seguindo dicionário de dados abaixo:

[nome do atributo: significado presente APENAS na documentação]

...

Exemplo:

```
def função(inputs):
```

```
# Função que calcula a variável x
```

```
# Inputs
```

```
# x: coluna x
```

```
# Outputs
```

```
# a: coluna a
```

```
# b: coluna b
```

```
x = a + b
```

```
return x
```

Fluxo Geral:

O código carrega um arquivo Excel, obtém os nomes das planilhas e cria um dicionário de correspondência entre identificadores e nomes de planilhas.

Gera um DataFrame a partir das células da planilha.

Cria um arquivo de texto contendo um dicionário de dados e extrai fórmulas presentes nas células, gerando prompts para documentação.

Algumas funções auxiliares são utilizadas para manipular strings e realizar substituições.

O script remove linhas específicas do arquivo de texto.

O script gera prompts para cada fórmula encontrada e imprime no console.

O código finaliza informando que o processamento foi concluído, e o resultado foi salvo em um arquivo chamado 'Teste.txt'.

Funções geradas pelos prompts.

1. ``calcular_b(B)``
 - Inputs:
 - ``B`` (float): Valor da coluna B.
 - Outputs:
 - ``plaqueta`` (float): Valor calculado da Plaqueta (b).
 - Descrição:
 - Calcula a Plaqueta (b) com base na fórmula $b = B + 1$.
2. ``calcular_k(J, L)``
 - Inputs:
 - ``J`` (float): Valor da coluna C_principal.
 - ``L`` (float): Valor da coluna L.
 - Outputs:
 - ``C_EA`` (float): Resultado do produto entre C_principal e L.
 - Descrição:
 - Calcula a variável C_EA.
3. ``calcular_m(J, N)``
 - Inputs:
 - ``J`` (float): Valor da coluna C_principal.
 - ``N`` (float): Valor da coluna N.
 - Outputs:
 - ``C_CA`` (float): Resultado do cálculo da fórmula $m = J * N$.
 - Descrição:
 - Calcula a variável C_CA usando a fórmula $m = J * N$.
4. ``calcular_o(J, K, M)``
 - Inputs:
 - ``J`` (float): Valor da coluna J (correspondente a C_principal).
 - ``K`` (float): Valor da coluna K (correspondente a C_EA).
 - ``M`` (float): Valor da coluna M.
 - Outputs:
 - ``C_imob`` (float): Valor calculado da variável C_imob.
 - Descrição:
 - Calcula a variável C_imob com base na fórmula: $C_{imob} = J + K + M$.
5. ``calcular_q(P, O)``
 - Inputs:
 - ``P`` (float): Valor da coluna C_JOA (numerador).
 - ``O`` (float): Valor da coluna C_JOA (denominador).
 - Outputs:
 - ``q`` (float): Valor da coluna C_JOA%, calculado como P/O .
 - Descrição:
 - Calcula o percentual q com base na fórmula $q = P/O$.
6. ``calcular_r(R)``
 - Inputs:
 - ``R`` (float): Representa o custo médio ponderado de capital (WACC).
 - Outputs:
 - ``r`` (float): O WACC calculado.
 - Descrição:
 - Calcula o Weighted Average Cost of Capital (WACC).
7. ``calcular_u(o, t)``
 - Inputs:
 - ``o`` (float): Valor correspondente a C_imob.
 - ``t`` (float): Valor correspondente a C_JOAR%.
 - Outputs:
 - ``u`` (float): Valor correspondente a C_JOAR.
 - Descrição:
 - Calcula a variável u com base na fórmula do Excel: $u = O * T$.
8. ``calcular_v(O, U)``

- Inputs:
 - `O` (float): Valor da coluna O, representando a variável C_imob.
 - `U` (float): Valor da coluna U, representando a variável C_JOAR.
 - Outputs:
 - `CH` (float): Valor calculado da variável CH usando a fórmula $CH = O + U$.
 - Descrição:
 - Calcula a variável V (CH) com base nas entradas O (C_imob) e U (C_JOAR).
9. `calcular_z(z)`
- Inputs:
 - `z` (float): Valor da variável Z.
 - Outputs:
 - `z` (float): Valor calculado da variável z (Data-base BRR).
 - Descrição:
 - Calcula a variável z (Data-base BRR) com base na fórmula $z = Z$.
10. `calcular_ab(Y)`
- Inputs:
 - `Y` (float): Valor da variável Y.
 - Outputs:
 - `ab` (float): Resultado do cálculo da variável AB.
 - Descrição:
 - Calcula a variável AB a partir da variável Y.
11. `calcular_ad(ac, aa)`
- Inputs:
 - `ac` (float): Valor da coluna 'IPCA_final' na tabela.
 - `aa` (float): Valor da coluna 'IPCA_imob' na tabela.
 - Outputs:
 - `ad` (float): Resultado do cálculo do Delta IPCA arredondado para 4 casas decimais.
 - Descrição:
 - Calcula o Delta IPCA de acordo com a fórmula do Excel: $ad = \text{ROUND}(ac / aa - 1, 4)$.
12. `calcular_ae(V, AD)`
- Inputs:
 - `V` (float): Valor da variável V (CH na fórmula original).
 - `AD` (float): Valor da variável AD (Delta_IPCA na fórmula original).
 - Outputs:
 - `CHC` (float): Valor calculado da variável CHC usando a fórmula $CHC = V * (1 + AD)$.
 - Descrição:
 - Calcula a variável CHC usando a fórmula: $CHC = V * (1 + AD)$.
13. `calcular_aj(ag, ah, ai)`
- Inputs:
 - `ag` (float): Prazo do contrato em anos.
 - `ah` (float): Vida útil física em anos.
 - `ai` (str): Vida útil regulatória (critério), deve ser "Contrato" ou "Nãoodeprecia".
 - Outputs:
 - `aj` (float): Vida útil regulatória em anos, calculada com base na condição especificada na fórmula.
 - Descrição:
 - Calcula a vida útil regulatória com base em uma condição.
14. `calcular_ak(aj)`
- Inputs:
 - `aj` (float): Vida útil regulatória em anos.
 - Outputs:
 - `ak` (float): TDR anual calculada.
 - Descrição:
 - Calcula a Taxa de Depreciação Regulatória (TDR) com base na fórmula do Excel.
15. `calcular_al(AK)`
- Inputs:
 - `AK` (float): Taxa de Desconto Racional (TDR) anual.
 - Outputs:
 - `AL` (float): Taxa de Desconto Racional (TDR) mensal calculada.

- Descrição:
- Calcula a Taxa de Desconto Racional (TDR) mensal a partir da TDR anual.

16. `calcular_am(x, z, aj)`

- Inputs:
- `x` (datetime): Data operacional.
- `z` (datetime): Data-base BRR.
- `aj` (float): Vida útil regulatória em anos.
- Outputs:
- `am` (float): DeltaT_oper em meses.

- Descrição:
- Calcula a variável AM (DeltaT_oper) com base na fórmula do Excel.

17. `calcular_an(al, am, ae)`

- Inputs:
- `al` (float): TDR (Taxa de Depreciação Residual) mensal.
- `am` (float): DeltaT_oper (Delta de Tempo Operacional) em meses.
- `ae` (float): Valor correspondente a AE (ou outra variável com significado semelhante).
- Outputs:
- `dra` (float): Resultado do cálculo da variável DRA.

- Descrição:
- Calcula a variável DRA (an) com base na fórmula: $DRA = AL * AM * AE$.

18. `calcular_ao(AE, AN)`

- Inputs:
- `AE` (float): Valor de CHC (colesterol HDL) na amostra.
- `AN` (float): Valor de DRA.
- Outputs:
- `CHC_liquido` (float): Valor do CHC líquido calculado ($CHC - AN$).

- Descrição:
- Calcula o CHC líquido com base nos valores de CHC e AN.

19. `calcular_at(z, AS)`

- Inputs:
- `z` (datetime): Data-base BRR.
- `AS` (datetime): Data_baixa.
- Outputs:
- `at` (float): Delta_baixa calculado.

- Descrição:
- Calcula a variável Delta_baixa com base na fórmula do Excel.

20. `calcular_au(ai)`

- Inputs:
- `ai` (str): Vida útil regulatória. Deve ser uma string indicando se o ativo deprecia.
- Outputs:
- `au` (str): Elegibilidade QRR. Retorna "Não" se a vida útil regulatória for "Não deprecia", caso contrário, retorna "Sim".

- Descrição:
- Calcula a elegibilidade QRR com base na vida útil regulatória.

21. `calcular_aw(i, x, z, ae, ao, ap, aq, ar, at, au)`

- Inputs:
- `i` (quantitativo): Quantitativo.
- `x` (float): Data oper.
- `z` (float): Data-base BRR.
- `ae` (float): CHC.
- `ao` (float): CHC_liquido.
- `ap` (float): IA.
- `aq` (str): Baixa.
- `ar` (float): Qtde_baixa.
- `at` (float): Delta_baixa.
- `au` (str): Elegível QRR.
- Outputs:
- `aw` (float): BRR_bruta.

- Descrição:

- Calcula a variável `aw` com base na fórmula do Excel fornecida.

22. ``calcular_ax(i, x, z, ao, ap, aq, ar, at, av)``

- Inputs:

- ``i`` (int): Quantitativo.

- ``x`` (float): Data oper.

- ``z`` (float): Data-base BRR.

- ``ao`` (float): CHC_liquido.

- ``ap`` (float): IA.

- ``aq`` (str): Baixa.

- ``ar`` (float): Qtde_baixa.

- ``at`` (float): Delta_baixa.

- ``av`` (str): Elegível juros.

- Outputs:

- ``ax`` (float): BRR_liquida.

- Descrição:

- Calcula a variável `ax` usando a fórmula do Excel fornecida.

23. ``calcular_ay(ak, au, aw)``

- Inputs:

- ``ak`` (float): TDR (Taxa de Desconto Racional) anual.

- ``au`` (str): Elegível QRR (indicador de elegibilidade).

- ``aw`` (float): BRR_bruta (receita bruta base).

- Outputs:

- ``ay`` (float): QRR (anual) calculada.

- Descrição:

- Calcula a variável QRR (anual) com base na fórmula do Excel.

24. ``calcular_az(av, ax, r)``

- Inputs:

- ``av`` (str): Elegibilidade de juros. Deve ser "Sim" para ativar o cálculo de juros.

- ``ax`` (float): BRR_liquida.

- ``r`` (float): WACC (Custo Médio Ponderado de Capital).

- Outputs:

- ``az`` (float): Juros anuais calculados de acordo com a fórmula, ou 0 se `av` não for "Sim".

- Descrição:

- Calcula os juros anuais com base nos parâmetros fornecidos.

code_s1.py ■

Utiliza as funções geradas, cria uma planilha adicional, faz a comparação com a referência e exibe o relatório comparativo no arquivo Relatorio_S1.txt.

Módulos Importados:

inspect: Permite extrair informações sobre objetos, como assinaturas de funções.

re: Fornece operações de expressões regulares.

traceback: Fornecer informações detalhadas sobre exceções.

openpyxl: Biblioteca para manipulação de arquivos Excel.

funcs_s1: Módulo contendo funções utilizadas no script.

pandas: Biblioteca para manipulação e análise de dados em Python.

datetime: Fornece classes para trabalhar com datas e horas.

numpy: Biblioteca para operações numéricas eficientes.

Variáveis:

1. Variáveis relacionadas a caminhos de arquivos:

input_excel_path: Contém o caminho do arquivo Excel de entrada. Este arquivo é utilizado para extrair dados das planilhas.

DictDados0.txt: Este é um arquivo de texto contendo informações específicas que são extraídas e utilizadas no código.

2. Variáveis relacionadas a dados:

xl: Objeto da classe pd.ExcelFile que representa o arquivo Excel de entrada. É utilizado para ler planilhas.

nomes_planilhas: Lista que armazena os nomes das planilhas presentes no arquivo Excel de entrada.

d_sheets: Dicionário que armazena os DataFrames das planilhas do arquivo Excel. As chaves são strings no formato 'S{i}', onde i é o índice da planilha.

dict_dados: Dicionário que é preenchido a partir do arquivo DictDados0.txt e contém informações relevantes para o código.

3. Variáveis relacionadas a avaliação de fórmulas:

dict_func: Dicionário que armazena informações sobre as funções a serem aplicadas aos dados. As chaves são obtidas do arquivo DictDados0.txt e os valores são listas de argumentos formatados.

attr_str: String que representa a expressão a ser avaliada dinamicamente. É construída com base nas informações do dict_func e planilhas presentes no arquivo Excel.

4. Variáveis relacionadas à criação do arquivo de saída:

workbook: Objeto que representa o arquivo Excel de saída onde os resultados serão armazenados.

sheet: Representa a planilha ativa dentro do arquivo Excel de saída.

5. Variáveis relacionadas ao estilo da planilha de saída:

Patternfill: Padrões de preenchimento de área para uso em estilos.

Alignment: Opções de alinhamento para uso em estilos.

font: Define o estilo da fonte para a primeira linha (caracteres brancos).

border: Define o estilo das bordas para a primeira linha (bordas pretas).

date_style: Define um estilo para formatar datas.

Funções Auxiliares:

sub_n(dfxc)

Remove espaços desnecessários e substitui '\n' por espaços nas colunas de um DataFrame.

sub_ci(ci)

Remove quebras de linha e reduz múltiplos espaços consecutivos a um único espaço em uma string.

letras_para_numeros(texto)

Converte uma representação de coluna de letras para números. Por exemplo, "AB" seria convertido para 26.

identifica_numero(coluna)

Verifica se uma coluna contém apenas números.

listar_argumentos(funcao)

Obtém os argumentos necessários para chamar uma função.

has_formula(x)

Verifica se uma determinada coluna tem uma fórmula associada em um arquivo de dados.

extract_lines(tdict)

Extrai linhas de um arquivo e cria um dicionário com os dados.

Leitura e Manipulação de Dados:

Abre um arquivo de texto (DictDados0.txt) para extrair dados e cria um dicionário (dict_dados).

Abre um arquivo Excel (Simulação_calculos_BRR_v6.xlsx), lê as planilhas e armazena em um dicionário (d_sheets).

Inicializa um DataFrame (df_aux) com os dados da primeira planilha.

Converte colunas para tipo float e preenche valores nulos com 0.

Avaliação de Fórmulas:

Avalia as fórmulas para cada coluna extraída e compara com os dados originais.

Escreve um relatório indicando se as fórmulas são compatíveis ou incompatíveis com os dados originais.

Atualiza o DataFrame (df_aux) com os resultados das fórmulas.

Geração de Relatório em Excel:

Cria um novo arquivo Excel (OUTPUT_S1.xlsx) com os resultados do processamento.

Formata as datas e adiciona um estilo para as células da primeira linha.

Salva o arquivo Excel.

Verificação ■

Verificação da compatibilidade da planilha gerada com a de input:

a : Compatível com a referencia
b : Compatível com a referencia
c : Compatível com a referencia
d : Compatível com a referencia
e : Compatível com a referencia
f : Compatível com a referencia
g : Compatível com a referencia
h : Compatível com a referencia
i : Compatível com a referencia
j : Compatível com a referencia
k : Compatível com a referencia
l : Compatível com a referencia
m : Compatível com a referencia
n : Compatível com a referencia
o : Compatível com a referencia
p : Compatível com a referencia
q : Compatível com a referencia
r : Compatível com a referencia
s : Compatível com a referencia
t : Compatível com a referencia
u : Compatível com a referencia
v : Compatível com a referencia
w : Compatível com a referencia
x : Compatível com a referencia
y : Compatível com a referencia
z : Compatível com a referencia
aa : Compatível com a referencia
ab : Compatível com a referencia
ac : Compatível com a referencia
ad : Compatível com a referencia
ae : Compatível com a referencia
af : Compatível com a referencia
ag : Compatível com a referencia
ah : Compatível com a referencia
ai : Compatível com a referencia
aj : Compatível com a referencia
ak : Compatível com a referencia
al : Compatível com a referencia
am : Compatível com a referencia
an : Compatível com a referencia
ao : Compatível com a referencia
ap : Compatível com a referencia
aq : Compatível com a referencia
ar : Compatível com a referencia
as : Compatível com a referencia
at : Compatível com a referencia
au : Compatível com a referencia
av : Compatível com a referencia
aw : Compatível com a referencia
ax : Compatível com a referencia
ay : Compatível com a referencia
az : Compatível com a referencia