**Relatório Explicativo sobre o Cálculo de Custo por Unidade para Clientes e Fábricas**

Este documento tem o objetivo de descrever o procedimento utilizado para calcular o custo por unidade de transporte para cada cliente de cada uma das três fábricas de forma didática e acessível. Vamos detalhar cada etapa e explicar o funcionamento do código desenvolvido, de maneira que não é necessário nenhum conhecimento prévio de programação.

**Introdução**

Nosso trabalho consiste em calcular o custo médio de transporte por unidade (como quilos ou caixas, dependendo do tipo de produto) que cada cliente recebe de cada fábrica. Para isso, precisamos somar todos os valores transportados e todos os valores de frete para cada cliente e fábrica, e depois dividir o total do frete pelo total transportado.

**Ferramenta Utilizada**

Para realizar este cálculo, usamos o Google Colab, uma plataforma gratuita que permite escrever e executar códigos na linguagem de programação Python. O Python é uma linguagem popular e útil para análise de dados, como no caso deste projeto. Para quem não está familiarizado com programação, basta pensar no Google Colab como uma “planilha de cálculos” avançada.

**Explicação do Código**

**Passo 1: Carregar os Dados**

Primeiro, precisamos carregar o arquivo de dados (que está no formato CSV, semelhante a uma planilha). Esse arquivo contém as informações das fábricas, clientes, quantidade transportada e valores de frete. Aqui está o código para carregar esses dados no Colab:

import pandas as pd

# Carregar a tabela CSV

df = pd.read\_csv('seu\_arquivo.csv', encoding='ISO-8859-1', delimiter=';')

* **import pandas as pd**: “Pandas” é uma ferramenta que nos ajuda a organizar e manipular dados em Python.
* **pd.read\_csv('seu\_arquivo.csv', encoding='ISO-8859-1', delimiter=';')**: Este comando lê o arquivo e organiza as informações para podermos trabalhar com elas. O encoding e o delimiter são detalhes que ajudam a ler corretamente o arquivo.

**Passo 2: Converter Colunas para Números**

Às vezes, os dados numéricos podem vir com problemas (por exemplo, se um número estiver formatado como texto). Portanto, precisamos garantir que as colunas de quantidade e frete estejam em formato numérico. Para isso, usamos o seguinte código:

df['Qtd.Transp'] = pd.to\_numeric(df['Qtd.Transp'], errors='coerce')

df['Vlr.Frete'] = pd.to\_numeric(df['Vlr.Frete'], errors='coerce')

df.dropna(subset=['Qtd.Transp', 'Vlr.Frete'], inplace=True)

* **pd.to\_numeric()**: Essa função converte o conteúdo das colunas “Qtd.Transp” e “Vlr.Frete” para números.
* **errors='coerce'**: Se algum valor não puder ser convertido para número, ele é substituído por um valor vazio.
* **df.dropna()**: Remove as linhas que tenham valores vazios. Assim, trabalhamos somente com dados válidos.

**Passo 3: Filtrar Fábricas e Clientes Específicos**

Aqui, selecionamos apenas as três fábricas e os clientes específicos para nossa análise:

fabricas = [3424402, 3403208, 3423909]

clientes = range(2301, 2352)

df = df[df['CO.Fabrica'].isin(fabricas) & df['CO.Cliente'].isin(clientes)]

* **fabricas e clientes**: Definimos as fábricas e clientes específicos que queremos analisar.
* **.isin()**: Filtra os dados para incluir apenas as fábricas e clientes escolhidos.

**Passo 4: Agrupar e Somar Quantidades e Fretes**

Agora somamos os valores de quantidade transportada e valor do frete para cada cliente e fábrica. Isso é feito com o seguinte código:

resultados = df.groupby(['CO.Fabrica', 'CO.Cliente']).agg({

'Qtd.Transp': 'sum',

'Vlr.Frete': 'sum'

}).reset\_index()

* **groupby()**: Agrupa os dados por fábrica e cliente.
* **.agg({'Qtd.Transp': 'sum', 'Vlr.Frete': 'sum'})**: Realiza a soma de todas as quantidades e todos os fretes de cada grupo (ou seja, de cada combinação de fábrica e cliente).
* **.reset\_index()**: Organiza os dados em uma tabela fácil de entender.

**Passo 5: Calcular o Custo por Unidade**

Com as somas de quantidade e frete prontas, calculamos o custo por unidade. O custo por unidade é obtido dividindo o total de frete pelo total transportado. Para evitar o arredondamento dos valores e permitir que as casas decimais sejam exibidas, vamos apenas limitar o valor a duas casas decimais.

resultados['Custo\_por\_Unidade'] = (resultados['Vlr.Frete'] / resultados['Qtd.Transp']).round(2)

* **(resultados['Vlr.Frete'] / resultados['Qtd.Transp']):** Divide o total de frete pelo total transportado.
* **.round(2):** Limita o valor a duas casas decimais, mantendo as frações para maior precisão.

**Passo 6: Salvar e Baixar o Arquivo de Resultados**

Por fim, salvamos os resultados em um novo arquivo CSV, que pode ser baixado para análise.

resultados[['CO.Fabrica', 'CO.Cliente', 'Custo\_por\_Unidade']].to\_csv('custo\_por\_unidade.csv', index=False, sep=';')

from google.colab import files

files.download('custo\_por\_unidade.csv')

* **.to\_csv()**: Salva os resultados em um arquivo CSV com as colunas que queremos.
* **files.download()**: Abre uma janela de download no Colab para que o arquivo possa ser baixado.

**Conclusão**

Este código é uma ferramenta prática para calcular automaticamente o custo de transporte por unidade para cada cliente e fábrica. Assim, obtemos uma visão mais precisa dos custos e das quantidades movimentadas, ajudando na análise financeira e logística.