


```

1 import pandas as pd
2
3 # Carregar o arquivo completo
4 arquivo = 'rideaddress_v1_tratado_sem_desconhecido.csv'
5 df = pd.read_csv(arquivo, delimiter=',', dtype={'Lat': str, 'Lng': str}, on_bad_lines='skip')
6
7 #Limpar os nomes das colunas para evitar espaços ou caracteres ocultos
8 df.columns = df.columns.str.strip()
9
10 #Limpar valores incorretos e converter para float
11 df['Lat'] = pd.to_numeric(df['Lat'].str.replace(',','.'), errors='coerce')
12 df['Lng'] = pd.to_numeric(df['Lng'].str.replace(',','.'), errors='coerce')
13
14 #Remover linhas com valores nulos em Lat e Lng
15 df = df.dropna(subset=['Lat', 'Lng'])
16
17 #Criar uma amostra aleatória de 1.000 linhas
18 amostra = df.sample(n=1000, random_state=42)
19
20 #Salvar o novo arquivo com a amostra
21 amostra.to_csv('rideaddress_v1_sample.csv', index=False)
22 print("Arquivo 'rideaddress_v1_sample.csv' salvo com sucesso!")
23

```

 Arquivo 'rideaddress_v1_sample.csv' salvo com sucesso!

```

1 import pandas as pd
2 import numpy as np
3
4 # Carregar a amostra com 1.000 linhas
5 arquivo = 'rideaddress_v1_sample.csv'
6 df = pd.read_csv(arquivo)
7
8 #Garantir que Lat e Lng estejam como float
9 df['Lat'] = pd.to_numeric(df['Lat'], errors='coerce')
10 df['Lng'] = pd.to_numeric(df['Lng'], errors='coerce')
11
12 #Remover linhas com valores nulos em Lat e Lng após conversão
13 df = df.dropna(subset=['Lat', 'Lng'])
14
15 # Ponto inicial (exemplo) – São Paulo
16 start_lat, start_lng = -23.550520, -46.633308
17
18 #Função para calcular a distância euclidiana entre dois pontos (aproximação)
19 def calcular_distancia(lat1, lng1, lat2, lng2):
20     return np.sqrt((lat1 - lat2) ** 2 + (lng1 - lng2) ** 2)
21
22 #Algoritmo de busca gulosa
23 def busca_gulosa(lat_inicio, lng_inicio):
24     df_copy = df.copy()
25     caminho = []
26
27     while not df_copy.empty:
28         #Calcula a distância para cada ponto
29         df_copy['Distancia'] = calcular_distancia(lat_inicio, lng_inicio, df_copy['Lat'], df_copy['Lng'])
30
31         #Seleciona o ponto mais próximo (menor distância)
32         ponto_mais_proximo = df_copy.loc[df_copy['Distancia'].idxmin()]
33
34         #Adiciona ao caminho
35         caminho.append(ponto_mais_proximo)
36
37         # Atualiza o ponto inicial para o novo ponto escolhido
38         lat_inicio, lng_inicio = ponto_mais_proximo['Lat'], ponto_mais_proximo['Lng']
39
40         # Remove o ponto escolhido da lista
41         df_copy = df_copy.drop(ponto_mais_proximo.name)
42
43     # Retorna apenas colunas relevantes
44     return pd.DataFrame(caminho)[['Neighborhood', 'City', 'State', 'Lat', 'Lng']]
45
46 #Executar a busca gulosa
47 resultado_busca = busca_gulosa(start_lat, start_lng)

```

```
48
49 #Mostrar o resultado
50 print("\nResultado da Busca Gulosa:")
51 print(resultado_busca.head(20))
52
53 #Salvar o resultado em um novo CSV para análise futura (opcional)
54 resultado_busca.to_csv('resultado_busca_gulosa.csv', index=False)
55 print("\nArquivo 'resultado_busca_gulosa.csv' salvo com sucesso!")
```



Resultado da Busca Gulosa:

	Neighborhood	City \
598	Sé	São Paulo
371	R. Marquês de Itu, 308 - Vila Buarque, São Pau...	Vila Buarque
910	Consolação	São Paulo
75	Bela Vista	São Paulo
493	Bela Vista	São Paulo
647	Av. Nove de Julho, 1954 - Bela Vista, São Paul...	Bela Vista
885	Rua Dona Adma Jafet, 91 - Bela Vista, São Paul...	Bela Vista
916	Rua Dona Adma Jafet, 91 - Bela Vista, São Paul...	Bela Vista
248	Bela Vista	São Paulo
297	Bela Vista	São Paulo
23	Bela Vista	São Paulo
772	Bela Vista	São Paulo
386	P?oximo ao Número 661 - Alameda Jaú, S/N - Jar...	Jardim Paulista
821	Jardim Paulista	São Paulo
602	Jardim Paulista	São Paulo
553	Jardim Paulista	São Paulo
238	Jardim Paulista	São Paulo
674	R. Manuel da Nóbrega, 1015 - Paraíso, São Paul...	São Paulo
38	Paraíso	São Paulo
187	Vila Mariana	São Paulo

	State	Lat	Lng
598	São Paulo	-23.546532	-46.632222
371	SP	-23.543605	-46.647340
910	São Paulo	-23.551049	-46.649755
75	SP	-23.557085	-46.653191
493	SP	-23.557085	-46.653191
647	SP	-23.557392	-46.652837
885	SP	-23.557802	-46.654223
916	SP	-23.557802	-46.654223
248	São Paulo	-23.560450	-46.653649
297	SP	-23.560631	-46.653750
23	SP	-23.560657	-46.653823
772	SP	-23.560657	-46.653823
386	São Paulo	-23.565672	-46.655558
821	SP	-23.568658	-46.657602
602	São Paulo	-23.570415	-46.660850
553	São Paulo	-23.570395	-46.661174
238	São Paulo	-23.574227	-46.656037
674	SP	-23.574800	-46.653767
38	São Paulo	-23.578926	-46.652721
187	São Paulo	-23.580464	-46.647457

Arquivo 'resultado_busca_gulosa.csv' salvo com sucesso!