

Imports

In []:

```
1 from pulp import *
2 import googlemaps
3 import pandas as pd
4 from IPython.display import Image
```

In []:

```
1 df_unidades = pd.read_csv(r'..\2. Obtenção da matriz de distâncias\Matriz de
distâncias final - 131 unidades.csv').drop(columns=['Unnamed: 0'])
2 df_centros = pd.read_csv(r'..\2. Obtenção da matriz de distâncias\Centros.csv')
```

In []:

```
1 df_unidades_old = pd.read_csv(r'..\2. Obtenção da matriz de distâncias\Matriz de
distâncias final.csv')
```

In []:

```
1 unidades = df_unidades['Coordenadas'].to_list()
2 CDs = df_centros['centros'].to_list()
3
4 demanda = df_unidades[['Coordenadas', 'Demanda']].set_index('Coordenadas').to_dict()
['Demanda']
5 oferta = df_centros[['centros', 'oferta']].set_index('centros').to_dict()['oferta']
6
7 df = df_unidades.drop(columns=['Demanda', 'UBS ou ponto de vacinação', 'Nome Google
Maps']).set_index('Coordenadas').to_dict()
```

Modeling

In []:

```
1 prob = LpProblem("THE_P_MEDIAN_PROBLEM", LpMinimize)
```

In []:

```
1 p = 8
```

In []:

```
1 tarifa = 0.01 # R$/KM*1u
```

In []:

```
1 x = LpVariable.dicts('X_%s_to_%s', (CDs,unidades),
2     cat = 'Binary',
3     lowBound = 0,
4     upBound = 1)
```

In []:

```
1 y = LpVariable.dicts('Y_%s', (CDs),
2     cat = 'Binary',
3     lowBound = 0,
4     upBound = 1)
```

In []:

```
1 Image(filename='Modelo.jpg')
```

Objective function

In []:

```
1 prob += sum(x[i][j]*df[i][j]*demanda[j]*tarifa for i in CDs for j in unidades),
    "Custo total estimado"
```

Constraints

In []:

```
1 # cada unidade deve receber vacinas de um, e apenas um, CD
2 for j in unidades:
3     prob += sum(x[i][j] for i in CDs) == 1
```

In []:

```
1 # a quantidade de CDs abertos é igual a p
2 prob += lpSum(y[j] for j in CDs) == p
```

In []:

```
1 # apenas CDs abertos podem entregar vacinas
2 for i in CDs:
3     for j in unidades:
4         prob += x[i][j] <= y[i]
```

In []:

```
1 # cada CD só pode atender no máximo x unidades
2 for i in CDs:
3     prob += lpSum(x[i][j] for j in unidades) <= 16
```

In []:

```
1 # a soma das demandas das unidades para as quais determinado CD irá fazer entregas é
    menor que (ou igual) à oferta do CD
2 for i in CDs:
3     prob += lpSum(demanda[j]*x[i][j] for j in unidades) <= oferta[i]*y[i]
```

In []:

```
1 prob
```

Solving

In []:

```
1 prob.solve()
```

In []:

```
1 print(f'Total: R$ {prob.objective.value()/1000}')
```

In []:

```
1 for v in prob.variables():
2     print(v.name , '=', v.varValue)
3     #if v.varValue == 1:
4         #print(v.name , '=', v.varValue)
```

In []:

```
1 df = pd.DataFrame(columns=['origem', 'destino'])
2 for v in prob.variables():
3     #print(v.name , '=', v.varValue)
4     if v.varValue == 1:
5         #print(v.name , '=', v.varValue)
6         split = v.name.split('_to_')
7         split[0] = split[0].replace('X_', '')
8         try:
9             df = df.append({'origem': split[0], 'destino': split[1]},
10                             ignore_index=True)
11         except:
12             pass
```

Result

In []:

```
1 df['origem'] = df['origem'].replace('_', ' ', regex=True)
2 df['destino'] = df['destino'].replace('__', '-', regex=True)
3 df['destino'] = df['destino'].replace('_', '-', regex=True)
```

In []:

```
1 df.style.set_sticky()
```

Get informations to output dataframe

In []:

```
1 df['destino'] = df['destino'].str.replace('-', ', -')
```

In []:

```
1 df = df.rename(columns={'destino': 'Coordenadas'})
```

In []:

```
1 df_inner = df.merge(df_unidades, on='Coordenadas',how='inner')
```

In []:

```
1 df_inner.to_csv('Optimized.csv')
```