

```

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

import statsmodels.api as sm
import scipy.stats as stats

from sklearn.metrics import r2_score, mean_absolute_percentage_error
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

sns.set()
pd.options.display.max_columns = None

df = pd.read_csv("fountains.csv", sep=";", decimal=",")
df.head()

```

|   | x        | y        | location     |
|---|----------|----------|--------------|
| 0 | 2.397371 | 1.889389 | Левобережный |
| 1 | 2.518151 | 3.218424 | Левобережный |
| 2 | 1.812677 | 1.812690 | Левобережный |
| 3 | 3.263370 | 2.613948 | Левобережный |
| 4 | 1.624420 | 2.434048 | Левобережный |

```

coords = df[["x", "y"]]

```

```

# 1

```

```

def evclid(p1, p2):
    return ((p1[0] - p2[0])**2 + (p1[1] - p2[1])**2)**0.5

```

```

fonts = [(5, 3.5)]
l = []
for s in coords.iterrows():
    x = s[1]["x"]
    y = s[1]["y"]
    pol = max([10 / (2 + evclid((x, y), f)) for f in fonts])
    l.append(pol)

```

```

print(sum(l) / len(l))

```

```

2.064461732964563

```

```

# 2

```

```

def evclid(p1, p2):
    return ((p1[0] - p2[0])**2 + (p1[1] - p2[1])**2)**0.5

```

```

fonts = [(2, 2), (8, 2)]

```

```

l = []
for s in coords.iterrows():
    x = s[1]["x"]
    y = s[1]["y"]
    pol = max([10 / (2 + evclld((x, y), f)) for f in fonts])
    l.append(pol)

```

```
print(sum(l) / len(l))
```

3.089928468250307

# 3

```

def evclld(p1, p2):
    return ((p1[0] - p2[0])**2 + (p1[1] - p2[1])**2)**0.5

```

```
fonts = [(2, 2), (8, 2), (5, 5)]
```

```

l = []
for s in coords.iterrows():
    x = s[1]["x"]
    y = s[1]["y"]
    pol = max([10 / (2 + evclld((x, y), f)) for f in fonts])
    l.append(pol)

```

```
print(sum(l) / len(l))
```

3.5033464606501767      1. Ответ: Третья схема

```
gol = []
```

```

for s in coords.iterrows():
    x = s[1]["x"]
    y = s[1]["y"]

    a = evclld((x, y), (5, 3.5))
    b = min([evclld((x, y), f) for f in [(2, 2), (8, 2)]]))
    c = min([evclld((x, y), f) for f in [(2, 2), (8, 2), (5, 5)]]))

    if [a, b, c].count(min([a, b, c])) >= 2:
        if a <= b and a <= c:
            gol.append(1)
        elif b <= c and b <= a:
            gol.append(2)
        elif c <= b and c <= a:
            gol.append(3)

    else:
        if min([a, b, c]) == a:
            gol.append(1)

```

```

        elif min([a, b, c]) == b:
            gol.append(2)
        elif min([a, b, c]) == c:
            gol.append(3)
        else:
            assert 1

len(gol)
500
pd.Series(gol).value_counts()
2      392
3       93
1       15
Name: count, dtype: int64

```

! 2 Победит второй вариант

```

summ = 0
fonts = [(5, 3.5)]

for s in coords.iterrows():
    x = s[1]["x"]
    y = s[1]["y"]

    plat = min([20 / (2 + evclid((x, y), font)) for font in fonts])

    summ += plat

print(summ - 1500)
564.461732964563

summ = 0
fonts = [(2, 2), (8, 2)]

for s in coords.iterrows():
    x = s[1]["x"]
    y = s[1]["y"]

    plat = min([20 / (2 + evclid((x, y), font)) for font in fonts])

    summ += plat

print(summ - 3000)
-1690.6894277484744

summ = 0
fonts = [(2, 2), (8, 2), (5, 5)]

```

```
for s in coords.iterrows():
    x = s[1]["x"]
    y = s[1]["y"]

    plat = min([20 / (2 + evclid((x, y), font)) for font in fonts])

    summ += plat

print(summ - 4500)
-3190.875184355853
```

### 3. Ответ первый вариант