

Question 3 implementation option:

a) Formalisation of least-squares problem.

we have $n = 10000$ triplets and the problem is minimizing total vertical distance of these data points to the surface and to do so we should fit a quadratic surface of the form :

$$z = g(x, y) = a_1 x^2 + a_2 y^2 + a_3 xy + a_4 x + a_5 y + a_6$$

therefore the problem is :

$$a_* = \underset{a \in \mathbb{R}^6}{\operatorname{argmin}} \|Z - g_a(X, Y)\|_2^2$$

where

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad X^2 = \begin{bmatrix} x_1^2 \\ \vdots \\ x_n^2 \end{bmatrix}, \quad Y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, \quad Y^2 = \begin{bmatrix} y_1^2 \\ \vdots \\ y_n^2 \end{bmatrix}, \quad XY = \begin{bmatrix} x_1 y_1 \\ \vdots \\ x_n y_n \end{bmatrix}, \quad Z = \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix}, \quad \mathbb{1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

which is the same as :

$$a_* = \underset{a \in \mathbb{R}^6}{\operatorname{argmin}} \|Z - \mathcal{X}a\|_2^2$$

where :

$$\mathcal{X} = [X^2, Y^2, XY, X, Y, \mathbb{1}] = \begin{bmatrix} x_1^2 & y_1^2 & x_1 y_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^2 & y_n^2 & x_n y_n & x_n & y_n & 1 \end{bmatrix}$$

So the solution for this problem is :

$$a_* = (\mathcal{X}\mathcal{X}^t)^{-1} \mathcal{X}^t Z$$

b) Implementation of least-squares algorithm.

```
In [1]: import numpy as np
import pandas as pd

# Loading data set
df=pd.read_csv('rls_data.csv', sep=',',header=None)

#def data_transformation
m=6 #number of columns
M = np.zeros([df.shape[0],m])
M[:,0] = df[0]**2
M[:,1] = df[1]**2
M[:,2] = df[0]*df[1]
M[:,3] = df[0]
M[:,4] = df[1]
M[:,5] = np.ones(df.shape[0])

# Ordinary least square function
def Ols(X, y):
    X_transpose = X.T
    param = np.linalg.inv(X_transpose.dot(X)).dot(X_transpose).dot(y)
    return param

par = Ols(M, df[2])
par
```

```
Out[1]: array([ 2.00001558,  2.9999851 ,  5.00000588,  6.99933268, 10.9999725 ,
                13.5192896 ])
```

c) testing our function with OLS function from statsmodels.api library

```
In [2]: import statsmodels.api as sm
model = sm.OLS(df[2], M)
results = model.fit()
print(results.summary())
```

```
/Users/mac/anaconda/lib/python3.5/site-packages/statsmodels/compat/pandas.py:56: FutureWarning: The pandas.core.datetools module is deprecated and will be removed in a future version. Please use the pandas.tseries module instead.
```

```
from pandas.core import datetools
```