ENPM673 Homework1 Report

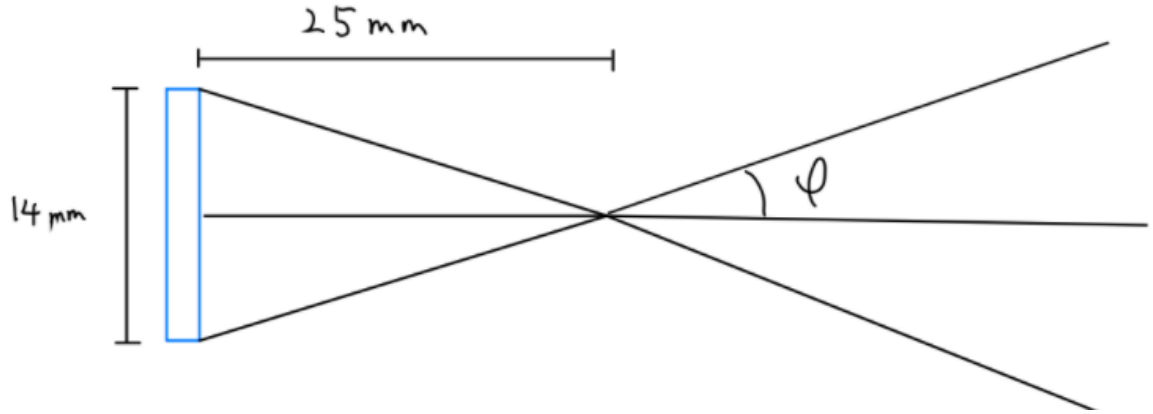
Problem 1:

Camera resolution: 5MP

Camera sensor is a square with width of 14mm and focal length of 25 mm

1. Calculate FOV

Horizontal FOV



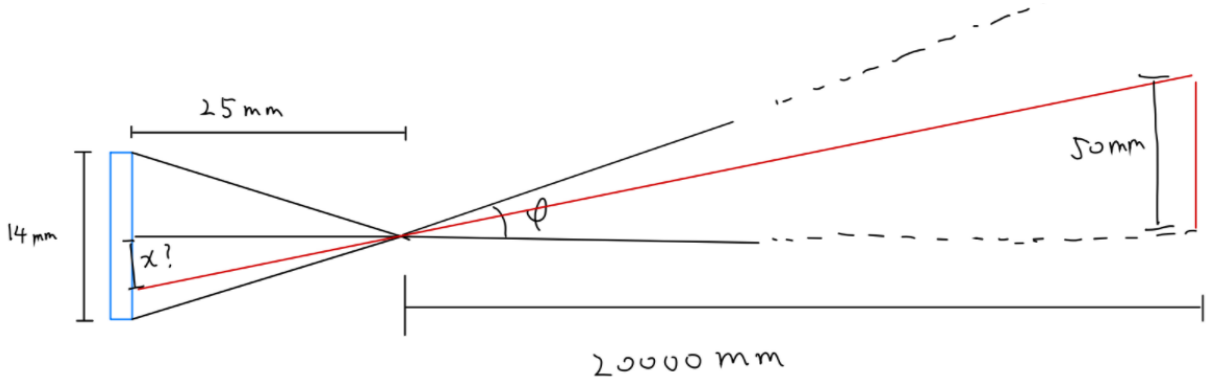
$$\varphi = \tan^{-1}(\frac{14/2}{25}) = 0.273\ rad$$

$$FOV = 2\varphi = 0.545\ rad$$

Because the camera sensor is a square the vertical FOV and the horizontal FOV are the same.

2. Calculate Occupied Pixels

First consider the height of the sqaured shape object



$$\frac{50}{20000} = \frac{x}{25}$$

The height projected on to the camera sensor $x = 0.0625\ mm$.
Because the object is a sqaure and the camera sensor is also a square, the projected width is the same as the height.
The total projected image occupies area of $0.0625 \times 0.0625\ mm^2$

The occupied pixels p is calculated by:

$$\frac{p}{5M} = \frac{0.625 * 0.625}{14 * 25}$$

$$p = 5580.357$$
$$\min\ p = 5580$$

Problem 2:

Fit the ball trajectory in two videos

1. Video1 (no noise)

Step 1. filter out the red ball.

The image is first transformed into gray scale.
Because the background is white, the red ball can be filtered out by eliminating pixels with intensity of 255. In practice, pixels that have value above 200 are elimnated because of some small noise.

Step 2. Calculate the standard least square by fitting the data to a parabola curve

The curve function is $ax^2 + bx + c = y$

The least square for k data points is $\min \sum_{i=1}^{k} (y_i - ax_i^2 - bx_i - c)^2$

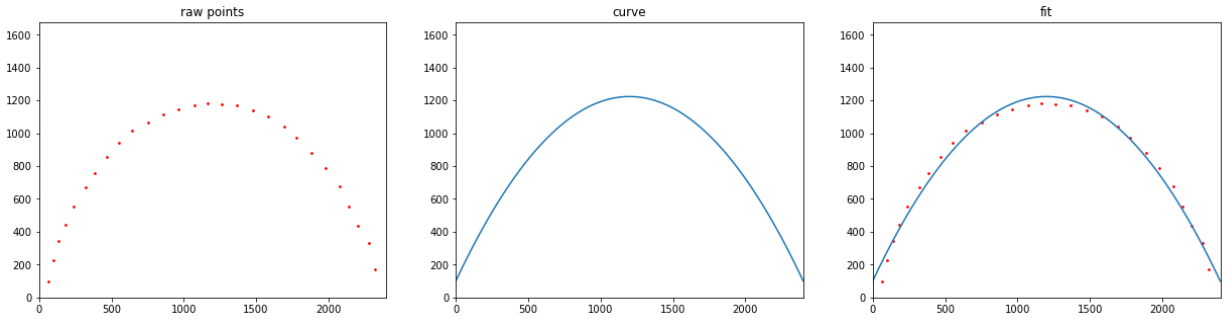
Set the y, x, and other parameters as follow matrices

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_k \end{bmatrix}, \; X = \begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ & . & \\ & . & \\ & . & \\ x_k^2 & x_k & 1 \end{bmatrix}, \; B = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

The least square function becomes $E = ||Y - XB||$

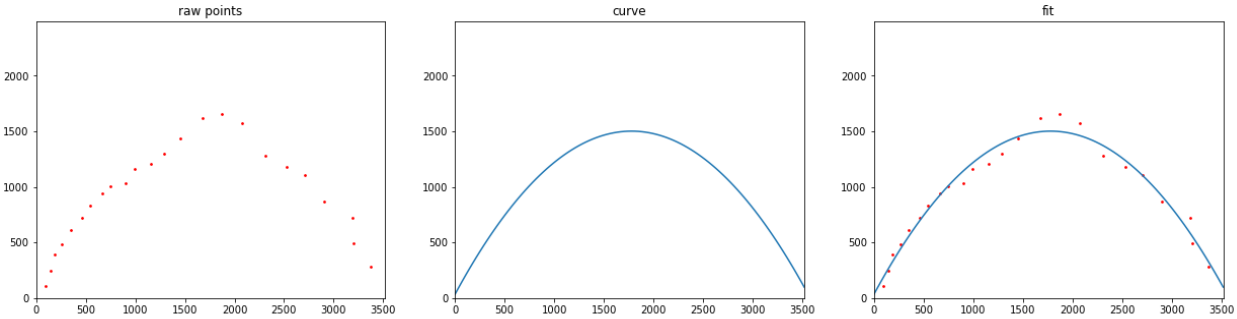
Set the derivative to 0 to find minimum $\frac{dE}{dB} = 2X^T XB - 2X^T Y = 0$

Solve the equation to find all the curve parameter a, b, c



2. Video2 (with noise)

Video2 is calculated using the same methods as video1

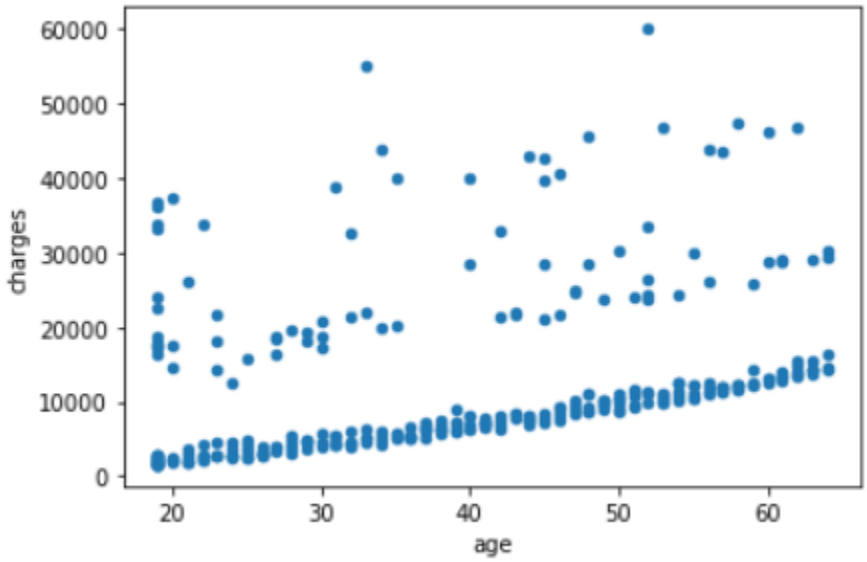


Problem 3:

Analyze age insurance relationship data using LS, TLS, RANSAC

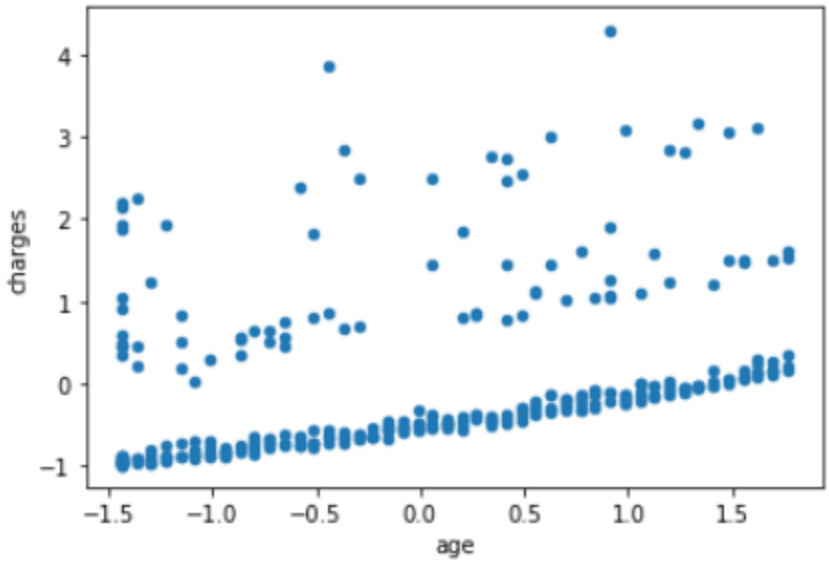
1. Compute the covariance matrix and draw the eigenvector along with the data

The raw data distribution is as follow

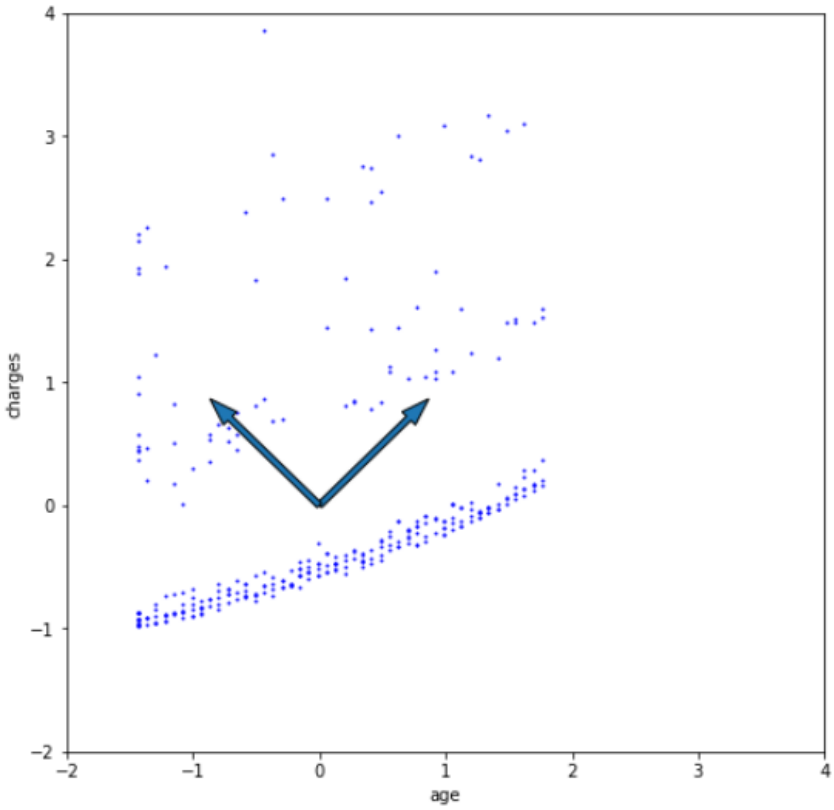


Because the magnitude difference between the age data and the charges data is large, the data is first normalized before any further analysis.

Normalized data



Plot the eigenvector (only plot the direction of the eigenvector)



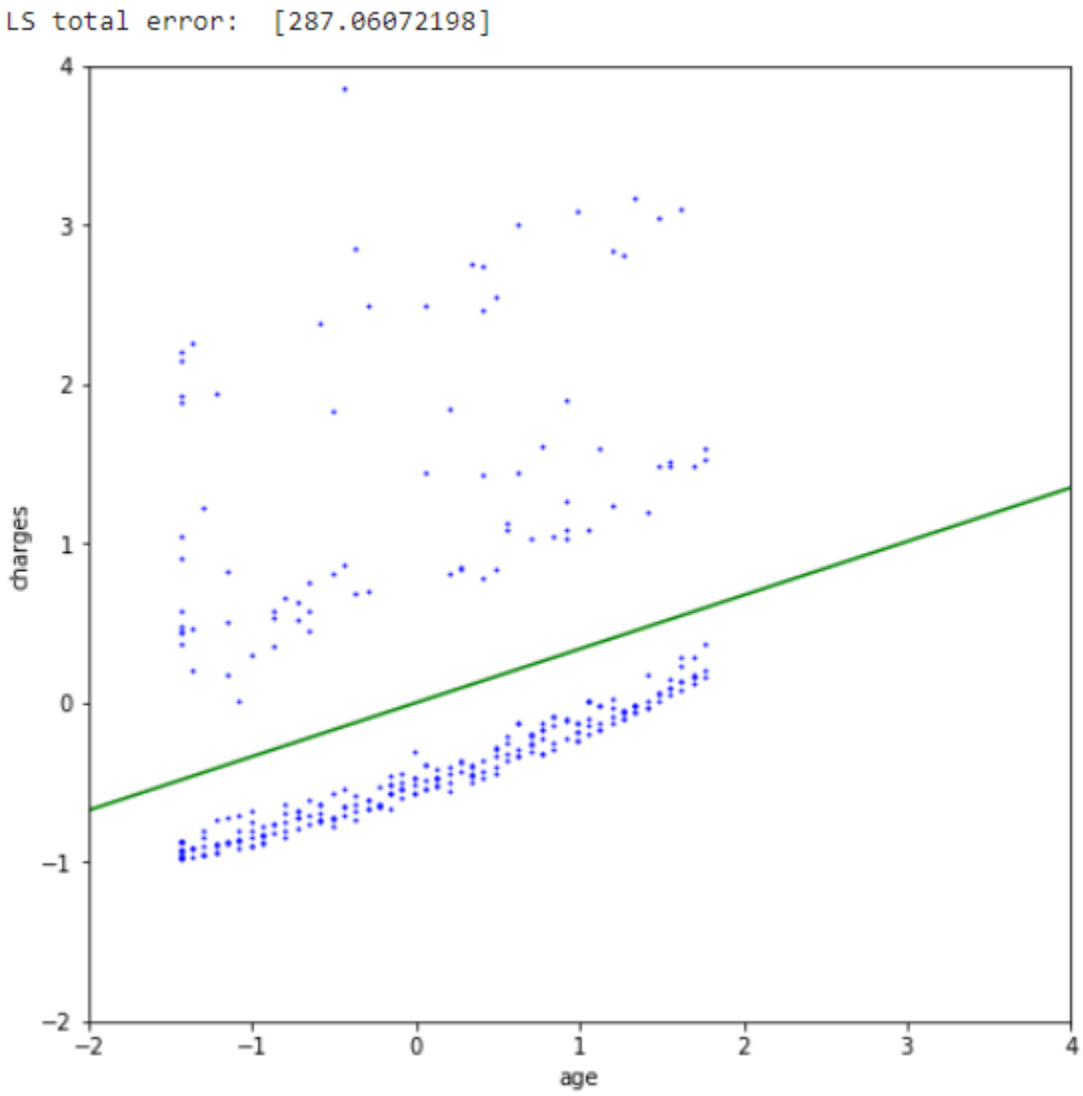
2. Fit the data using Least Square, Total Least Square, and RANSAC

Least Square:

Fit the model with line function $y = ax + b$

Find the parameters by solving linear equation as problem 2

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_k \end{bmatrix}, \; X = \begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ & . & \\ & . & \\ x_k^2 & x_k & 1 \end{bmatrix}, \; B = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$



Total Least Square:

Fit the model with line function $ax + by - d = 0$

Minimize the error function $E = \sum_{i=1}^{n}(ax_i + by_i - d)^2$

d is actually related to the average of data points.

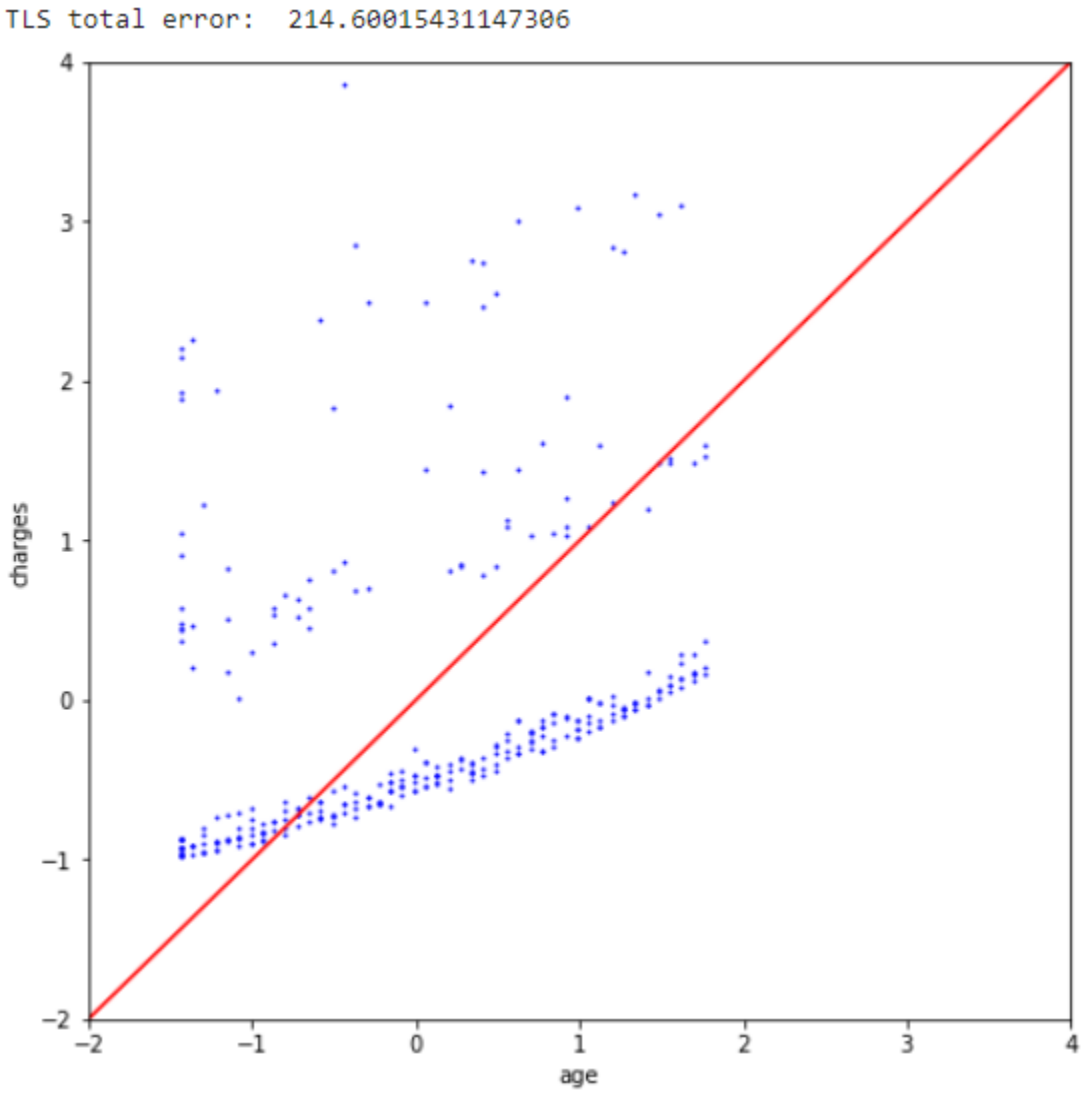
d=0 in this case because of the normalization

$$d = \frac{a}{n}\sum_{i=1}^{n} x_i + \frac{b}{n}\sum_{i=1}^{n} y_i = a\bar{x} + b\bar{y}$$

set $N = \begin{bmatrix} a \\ b \end{bmatrix}$

$$E = \| \begin{bmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ \vdots & \vdots \\ x_n - \bar{x} & y_n - \bar{y} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \| = (UN)^T (UN)$$

The solution that solve this minimization of linear system is to find the eigenvector of $U^T U$ associated with the smallest eigenvalue.



RANSAC:

Steps in RANSAC

Step 1. Randomly select points based on minmal amount of parameter that the model needs

Step 2. Form a model using sampled points

The model uses linear equation of $y = ax + b$

This model has some minor problem when the sampled points form a vertical line

The program skip the model and continue on next sampled points when this happens.

Step 3. Examine the amount of points within an error range of the model

The error range is set to be 5% of the total error in Total Least Square

Step 4. Continue the above steps N time and select the best model based on Step 3

Parameters in RANSAC

s: amount of sampled points to form the model
N: number of iteration to run RANSAC
w: amount of inlier/total data
p: probability to have good model with in iteration N

The probability to have at least one point in Step 1 to be categorized as outlier is $1 - w^s$

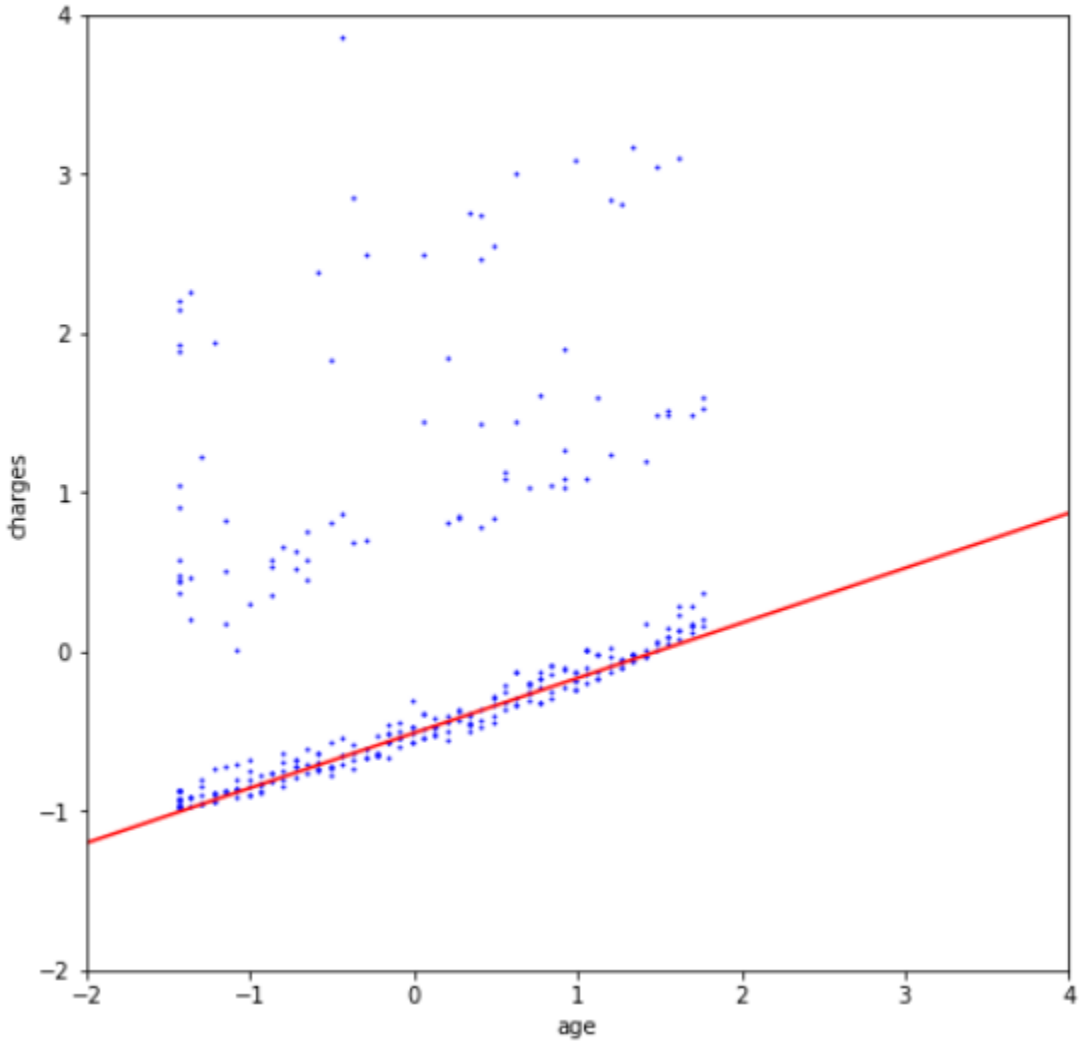
When outlier is picked as sample point, the model can be considered as bad model

The probability to have bad model through all the iteration is $(1 - w^s)^N$

The following equation is then form.

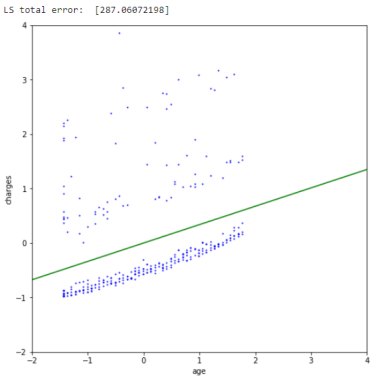
$$1 - p = (1 - w^s)^N$$
$$N = \frac{\log 1 - p}{\log (1 - w^s)}$$

```
Iterations:  4
Max inlier:  237
total error:  [371.74599903]
```



3. Compare the result of LS, TLS, and RANSAC

LS **TLS** **RANSAC**



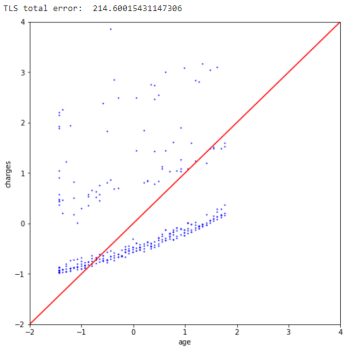
Total error: 287

Pros:

- Easy to implement
- Very Intuitive

Cons:

- Affected by the outliers
- Larger error compared with TLS because it only considered the vertical error of the data points.



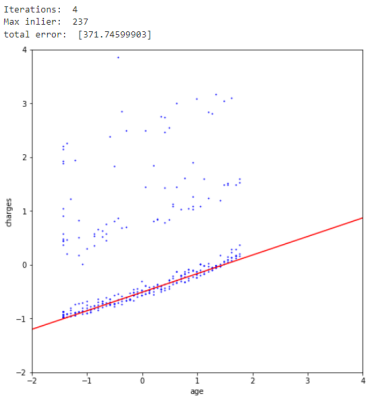
Total error: 214

Pros:

- Lowest total error

Cons:

- Affected by the outliers
- The small total error is because of considering the outliers



Total error: 371

Pros:

- Best model for outlier rejection among the three

Cons:

- Requires many parameters, some of which are obtained through observaton and guesses.
- Requires large computation time if the probability of inlier is low

Problem 4:

Calculate SVD

Homography matrix

Solve $Ax = 0$

The solution is the eigenvector associate with the smalles eigenvalue of $A^T A$ (The last row of V^T in $U\Sigma V^T$)

```
homography matrix
[[ 5.31056351e-02 -4.91718843e-03  6.14648552e-01]
 [ 1.77018784e-02 -3.93375075e-03  7.86750146e-01]
 [ 2.36025045e-04 -4.91718843e-05  7.62164205e-03]]
```

1. Compute SVD with math

From the following relationship, U and V can actually be calculated through finding the eigenvector of AA^T or A^TA, and the square root of the eigenvalue become the sigular value in Σ

$$AA^T = U\Sigma V^T V\Sigma^T U^T = U\Sigma\Sigma^T U^T$$
$$A^TA = V\Sigma^T U^T U\Sigma V^T = V\Sigma^T\Sigma V^T$$

However, becuase U and V are related and the above relationship is not enough to determine the sign of U and V. One of them should be found first, and the other one can be drived through the below equations.

$$U = AV\Sigma^{-1}$$
$$V = A^TU{\Sigma^T}^{-1}$$

Because U and V has to be orthogonal matrix and the eigen decomposition of AA^T or A^TA can only determine a maximum number of orthognal eigenvector regarding to the rank of A, the rest of the column of U and V can be determined through finding column vector that is orthogonal to the existed eigenvector and together form the basis vector of the dimiension of U and V.

The rest of the column vector can also be found through finding the nullspace of A that satisfied the orthogonal properties of U and V. This is shown in the following equation. The undertermined vector in U is associated with 0 in Σ which is also the vector in the nullspace of A^T.

$$A^TU = V\Sigma^T$$

Take a 3 by 2 matrix B for example, the maximum orthognal eigenvector BB^T or B^TB can get is 2. So first find the V matrix from finding the eigenvector of B^TB.

$$B^TB = V\Sigma^T U^T U\Sigma V^T = V\Sigma^T\Sigma V^T$$

Then calculate the U matrix through the column vector of V, NA^T stands for the nullspace of A^T

$$U = \begin{bmatrix} \frac{1}{\sigma_1}Av_1 & \frac{1}{\sigma_2}Av_2 & \frac{NA^T}{\|NA^T\|} \end{bmatrix}$$

2. Write python code to compute the SVD

See attachment problem4.ipynb