**[Dynamics of HCI] Homework 3**

20204571

Sihun Cha

**(1) Create a task application that can measure your own 1D point-and-click performance. The application must be able to measure the trial completion time by varying the target distance and target size. You can use any computing language to build the application. Please submit your source code. (50 pts)**

I designed task condition with three distinguished target width and four different target distance described as below.

**IV1**: target width (unit: *Unity unit*) = 15, 30, 45

**IV2**: target distance (unit: *Unity unit*) = 150, 300, 450, 600

The task application and the task results are referred as below. The Task is constructed in 1D point-and-click performance discrete type.

**Task app**: ‘Dynamics-HCI\_HW3.exe’ file in ‘Dynamics-HCI\_HW3\_taskapplication.zip’

**Main App source code**: ‘TargetClick.cs’, ‘ExperimentManager.cs’

*For more details about source code:* [*https://github.com/chacorp/Fitts\_Law\_test*](https://github.com/chacorp/Fitts_Law_test)

**Task result**: ’FittsLaw\_taskResult.csv’.

**(2) Measure your point-and-click performance in at least 12 different ID (index of difficulty) conditions. For example, you can design 12 different ID conditions through a 3 (target width) by 4 (target distance) factorial design. All IDs must be set to a value between 2 bits and 6 bits. Perform at least 30 trials for each condition (12\*30). (50 pts)**

The ID conditions are constructed as below. Task trials were performed 30 times. For more details, please refer to the attached file: ’FittsLaw\_taskResult.csv’.

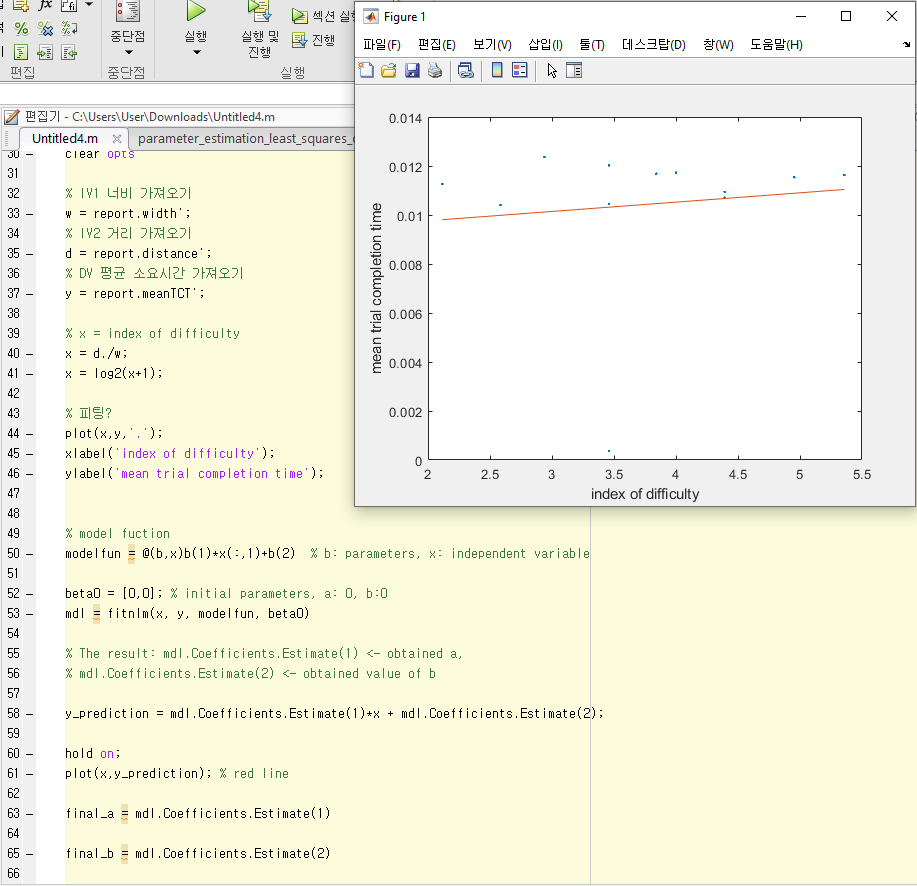
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Distance** | | | | |
| **Width** | ***Index of difficulty(bits)*** | **150** | **300** | **450** | **600** |
| **15** | *3.4594* | *4.3923* | *4.9541* | *5.3575* |
| **30** | *2.5849* | *3.4594* | *4* | *4.3923* |
| **45** | *2.1154* | *2.9385* | *3.4594* | *3.7369* |

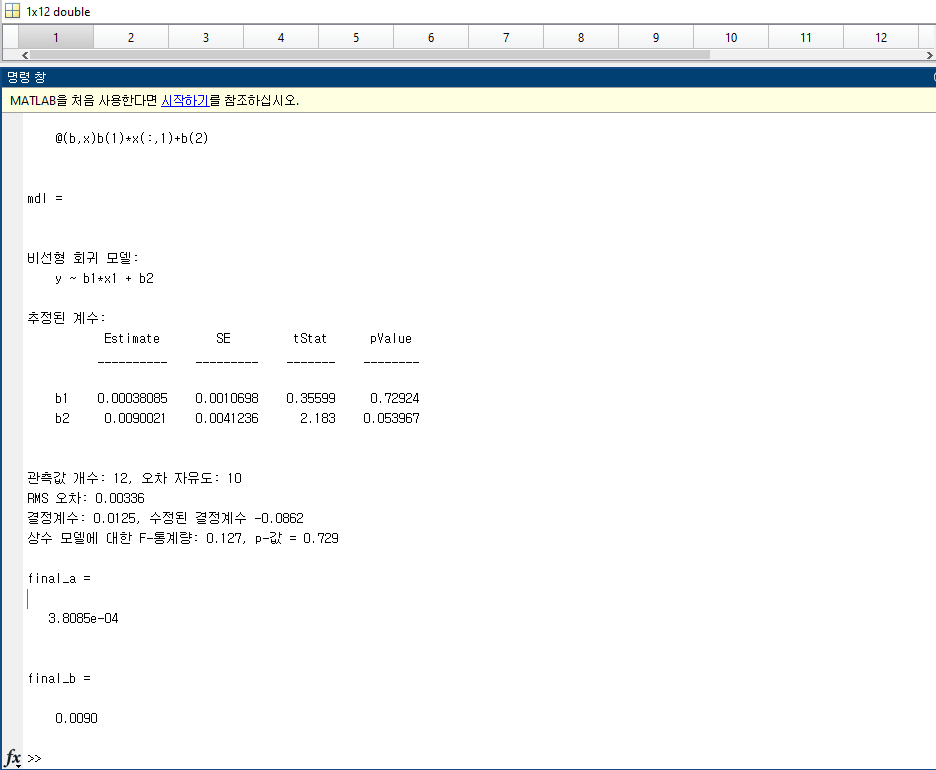
**(3) Fit your data to Fitts' law via the least squares method (you will get a and b). (50 pts)**

The mean trial completion time was calculated at *sheet2* in ’FittsLaw\_taskResult.csv’ and it was imported to *Matlab* for least squares method adaptation. The result of least squares method is referred to attached file: *‘FittsLaw\_task\_least\_squares.m’*.

**Final a = 3.8085e-04**

**Final b = 0.0090**



****

**(4) Present the performance of your parameter estimation in terms of the following aspects: R2, adjusted-R2, k-fold cross validation, residual plot, confidence interval via bootstrapping (100 pts)**

The results of the R2, adjusted-R2, k-fold cross validation, residual plot, confidence interval via bootstrapping are referred to attached files:

(It is recommended to open the file *’FittsLaw\_task\_least\_squares.m’* first.)

**R2 and Adjusted R2**: ‘FittsLaw\_task\_least\_squares\_r2.m’

**k-fold cross validation**: ‘FittsLaw\_task\_least\_squares\_cross\_validation.m’

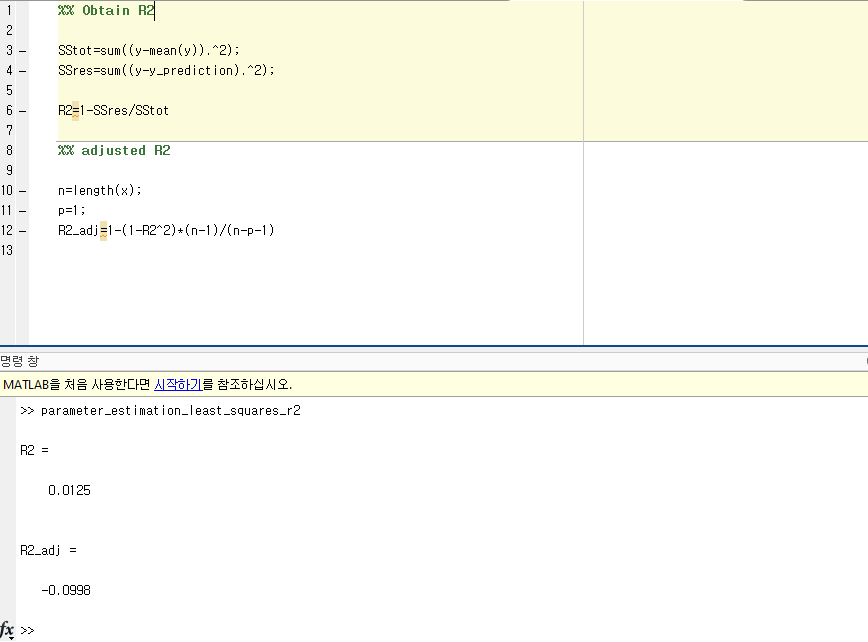
**residual plot**: ‘FittsLaw\_task\_least\_squares\_residual\_plot.m’

**bootstrapping**: ‘FittsLaw\_task\_least\_squares\_bootstrapping.m’

The result of R2 and Adjusted R2:

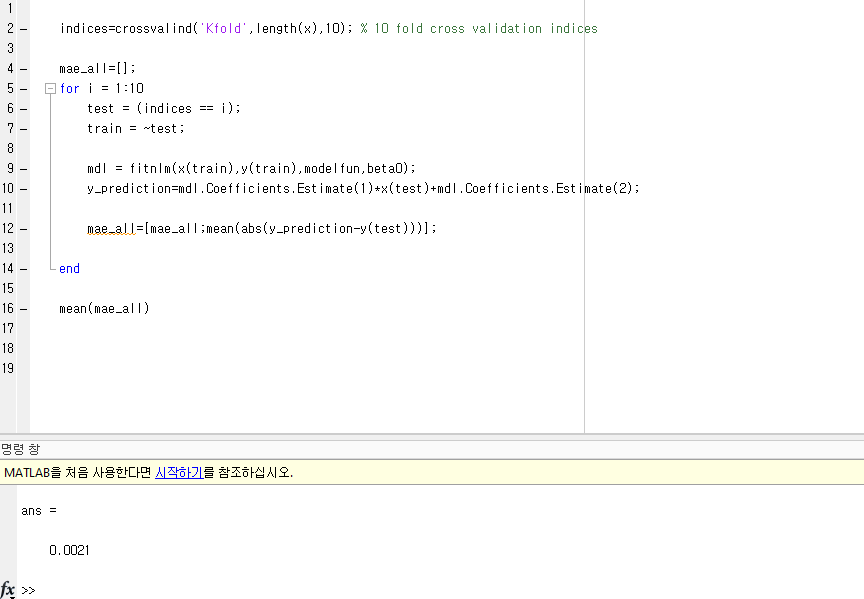
R2 = 0.0125

Adjusted R2 = -0.0998

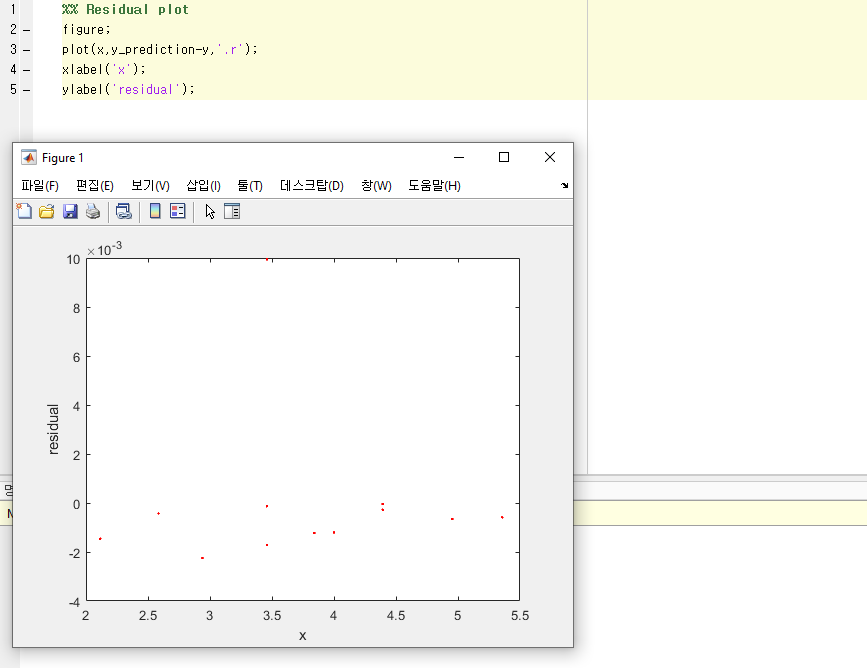


The result of k-fold cross validation:

ans = 0.0021



residual plot:



Bootstrapping:

a = 3.8085e-04

