Project 3: Solar Equity

Day 3 Handout

In this exercise we will explore LOWESS regressions. You will submit two files:

- The code generated here. You may submit either your full script with one or several sections (%% headers) relating to today's exercises, or you may submit a "partial script" that contains today's code only—the graders will run the Day 1 code to initialize.
- A Word or PDF document with answers to the questions.

Task 3: LOWESS regressions

Sometimes a polynomial function will not adequately fit the data, especially in the case of outliers. LOWESS (Locally Weighted Scatterplot Smoothing) or LOWESS (locally estimated scatterplot smoothing) are techniques that will smooth your data and exclude outliers. You can learn about these techniques in the YouTube video: https://youtu.be/Vf7oJ6z2LCc.

In this exercise we will generate LOWESS traces for each of the five groups: Asian Majority, Black Majority, Hispanic Majority, White Majority and No Majority.

Task 3.1 Order your data.

The first task is to sort your data tables in order of increasing median income. (This step is necessary to properly plot the data later). You can sort the `AsianMajority` table with the following code:

```
AsianMajority = sortrows(AsianMajority, 'Med_HHD_Inc_ACS_09_13');
```

Sort the tables for all five groups using similar code.

Question 3.1: Either by visually inspecting tables or by using min(__) and max(__) functions, find the largest and smallest median income for each group. Report these values in your answer.

Task 3.2 Perform LOWESS Smoothing on Black Majority Data

For simplicity, let's first abstract the data that we will be analyzing. Copy median incomes and normalized existing installations into vectors:

```
Xblack = BlackMajority.Med_HHD_Inc_ACS_09_13;
Yblack = BlackMajority.normalized_existing_installation;
```

Now apply the LOWESS smoothing algorithm

```
Fraction = 0.4
YblackLOWESS = LowessSmoothing(Yblack, Fraction);
```

Note here that `YblackLOWESS` and `Yblack` should have the same number of points, both functions of the data in `Xblack`.

Now plot your original data (circles) and smoothed data (line) on the same plot:

```
figure
hold on
plot(Xblack, Yblack, 'o')
plot(Xblack, YblackLOWESS, '-', 'LineWidth', 3)
```

Be sure to add titles, axis labels and legend.

Question 3.2: Remember that `Fraction` controls the size of the moving window used by the LOWESS function. Try different fractions and see how the MSE changes. Report your values. Describe in words how the LOWESS fit changes with fraction. What might be the best fraction to use for these plots?

Question 3.3: Select the "best" fraction. 0.4 seems to work well but there is no right or wrong answer here. Compare your LOWESS plot with the best polynomial fit that you obtained in the last handout. Copy paste these plots into your Word document and provide a brief comparison. Which fit do you think is better?

Task 3.3. Generate LOWESS traces for all groups.

Now expand your code to generate plots for each of the five groups. Be sure to include proper titles on each graph. Be sure to use `figure` so that you get five separate plots. Use the same fraction for all of these.

Task 3.4. Generate overlays

Now that we have trends we will want to directly compare them. Plot the LOWESS fits only for each of the five groups on the same graph. Since these are fits you will want to connect your points with lines rather than using makers. Be sure to include a legend as well as title and axis labels. Matlab should generate colors automatically, but feel free to assign your own colors if you'd like.

Question 3.4. What general insights can you gather from these overlaid traces? Note that at this point all we know is that the trends are different—we do not yet know if they are statistically different!

Challenge (optional)

1. Thus far we have been generating plots on linear scales. These data may be easier to compare using a semilog plot (x axis $\log_{10}$, y axis linear). Experiment with and adapt the following code to achieve semilog plots like those reported in the paper:

```
hold on
semilogx(X, Y,'o');
semilogx(Xfit, Yfit, 'LineWidth', 3)

xticks( [ 25000 50000 75000 100000 150000 200000 250000 ] );
xlim([ 23835 250000 ])
```

2. If you open the function LowessSmoothing you will find that it is essentially encapsulating the smooth(___) function. We are using 'rlowess' which uses a linear least squares fit within each window and assigns 0 weight to points that are outside six mean deviations. Experiment with the other methods for smoothing: 'moving', 'lowess', 'loess', 'sgolay', 'rloess'. You can find elaboration about these options in the help file.

TO SUBMIT:

- Your code should generate six plots, five corresponding to Task 3.3 and one corresponding to Task 3.4. If you wish to include additional graphs that is fine, but be sure that they are clearly labeled so that the grader knows which plot goes with which Task. Be sure that all graphs are labeled properly.