Topic 2: Noise Reduction

For problems 1 through 4, any code referenced is in "noise_ex1.m" on the website.

1. In the section of the code with a header of "Line: Multiple Observations", we have make *k* observations of *y*, which is defined to be equal to *x* with some noise added in. Each observation of *y* occurs for a number of *x* values, each spaced 0.5 units apart. We then average the *k* observations in an attempt to reduce the effect of the noise.

   Let's explore the effect of changing the number of observations *k* and how close together the *x* values are.

   a. Try using the following numbers for *k*: 2, 5, 10, 50, 100.
      Try using the following values to define how close the *x* are: .01, .1, .5, 1, 5.
      For each combination of these numbers, compare the amount of noise present in the average to that from the default values of 10 and .5.

   Now, let's try changing the amount of noise present. In the line of code "y(ind,jnd)=x(ind)+normrnd(0,.2)", the command "normrnd" adds noise. The first input to the command describes the magnitude of noise provided by the command. The second input is called the standard deviation and, in this context, describes the amount of noise. We will focus on the second term.

   b. Repeat part a, but for the following standard deviations: .1, .5, 1, 2.

2. Repeat problem 1, but for the section of the code with the header "Sin Example: Linear and Median Filters".

3. In the two above problems, have you found any patterns between these three quantities that we have looked at that help describe the performance of the noise reduction? If so, how are they related?

4. Refer to the code in the section with a header of "Sin Example: Linear and Median Filters". Similarly to problems 1 and 2, let's explore the effect of changing how close together the *x* values are and the standard deviation of the noise.

   Try using the following values to define how close the *x* are: .01, .1, .5, 1, 5.
   Try using the following values for the standard deviation: .02, .1, .5, 1, 2.
   For each combination of these numbers, compare the amount of noise present after applying the linear filter and (separately) the median filter to each other and to the default values of .1 and .05.

For problems 5 through 9, any code referenced is in "noise_images.m" on the website.

5. Refer to the code in the section with a header of "Noise reduction via averaging for images". In the command "imnoise", the third input, which is a number between 0 and 1, describes the amount of noise present in the image.

    a. Similarly to problem 1, let's try changing some of the inputs.
    Try using the following number of observations *m*: 2, 5, 10, 50, 100.
    Try the following amounts of **salt & pepper** noise: 0.1, 0.2, 0.5, 0.8, 1.
    For each combination of these numbers, compare the effects on the average image to those of the default values of 100 and 0.5.
    b. Repeat part a, but for **Gaussian** noise.
    NOTE: the default value for the amount of this type of noise is 0.2.
    c. Can you see any pattern between the values of these two quantities and their effect on each type of noise? If so, what is the pattern?
    d. Does this method work better for one of the two types of noise? If so, which?

6. Refer to the code in the section with a header of "Linear Filter for images". The quantity *c* defines the group of pixels that we consider to be nearby a given pixel. For example, if *c*=3, then this group of nearby pixels consists of the 3 by 3 square of pixels centered at our selected pixel. Let's try to look at the relationship of the size of this region and the amount of noise present in the image.

    a. Try using the following values for *c*: 3, 5, 7, 9.
    Try the following amounts of **salt & pepper** noise: 0.1, 0.2, 0.5, 0.8, 1.
    For each combination of these numbers, compare the effects on the Linear Filter to those of the default values of 3 and 0.5.
    b. Repeat part a, but for **Gaussian** noise.
    NOTE: the default value for the amount of this type of noise is 0.2.
    c. Can you see any pattern between the values of these two quantities and their effect on each type of noise? If so, what is the pattern?
    d. Does this method work better for one of the two types of noise? If so, which?

7. Repeat problem 6, but for the Median Filter, referencing the code in the section with a header of "Median Filter for images".

8. Repeat problem 5 for the RGB image "football.jpg".

9. Repeat problem 6 for the RGB image "football.jpg".