1 Data Normalization

We have some source data $X = \{x_1, x_2, \ldots, x_N\}$. Normalization of X is mapping the data into a suitable range while preserving relations between the components.

1.1 Min-max normalization onto an interval $< A, B >$

Design a function `mmscale(x)` which maps an arbitrary input vector `x` linearly onto the interval $< A, B >$. Usually we use min-max normalization onto the interval $< 0, 1 >$ or $< -1, 1 >$. For example

```
>> mmscale([0.12 3 -123],-1,1)
ans =

   0.95429   1.00000  -1.00000

>> mmscale([0.12 3 -123],0,1)
ans =

  0.97714  1.00000  0.00000
```

Using the command `plot` depict in a single graph `x` and `mmscale(x,A,B)` - e.g. with `x` on the horizontal axis and `mmscale(x,A,B)` on the vertical axis.

1.2 Normalization by the standard deviation

Standard deviation is the statistical function

$$sd(X) = \sigma_X = \sqrt{\frac{\sum(X - \bar{X})^2}{N - 1}},$$

where $\bar{X} = \frac{1}{N} \cdot \sum_{i=1}^{N} x_i$ is the mean value of X.

This normalization transforms an input vector linearly in such a way that the mapped data will have mean 0 and deviation 1.

Write a function `sdscale(x)` transforming `x` linearly to a vector with the mean 0 and the standard deviation 1.

Example:

```
>> sdscale([1 2 3])
ans =

       -1          0          1

>> sdscale([-2 2 7])
ans =

  -0.96099  -0.07392   1.03491
```

1.3 Sigmoid normalization

The sigmoid function (or logistic function), is the real function

$$f(x) = \frac{1}{1+e^{-\lambda x}}, \text{ where the real constant } \lambda \text{ is called } \textit{slope}.$$

The domain of sigmoid is $(-\infty, +\infty)$ and its range is $(0, 1)$.

We can graph the sigmoid function for $\lambda = 1$ e.g. in the following way

```
>> x=-10:0.2:10;
>> plot(x,1./(1+exp(-x)))
```

Write a function `sigmscale(x,l)` transforming entries of the vector `x` by the sigmoid function with slope `l`.

Example:

```
>> sigmscale(-3:3,2) ans =

   0.00247   0.01799   0.11920   0.50000   0.88080   0.98201   0.99753
```

Implement also a transformation `sigmscale_inv(x,l)` inverse to the transformation `sigmscale(x,l)`.