1. Let us consider the problem of estimating a probability density \( f \) of observations \( y_1, \ldots, y_n \). The most simple approach would be to use histogram which count the number of observations that belong to \( m \) disjoint bins. But it gives a discrete estimation. Suppose \( S = \bigcup_{i=1}^{m} A_i \) is the partition of \( S \).

\[
\hat{f}(x) = \frac{\# \text{ in the } i\text{-th interval}}{n}.
\]

2. But we know the true density to be continuous. Using the Kernel density estimator of Rosenblatt (1956) and Parzen (1962),

\[
\hat{f}_\sigma(x) = \frac{1}{n} \sum_{i=1}^{n} K_\sigma(x - x_i)
\]

where the proper choice of kernel \( K_\sigma \) would be that it should be symmetric probability density. Some other boring conditions are assumed from time to time. It can be shown that \( \hat{f}_\sigma \) is a probability density as well - it is related with a mixture model. We will use the discrete version of \( K_\sigma \sim N(0, \sigma^2) \) (see lecture 28-30).

3. How good is our density estimator and how we choose \( \sigma \)? We can try to minimize the mean integrated squared error (\( \sigma \)):

\[
MISE(\sigma) = E \int [\hat{f}_\sigma - f(x)]^2 \, dx.
\]

See Ruppert, Wand and Carroll’s semiparametric regression.

4. Bivariate kernel density estimate of observations \( z_i = (x_i, y_i) \)' is given by

\[
\hat{f}_\sigma(x, y) = \frac{1}{n} \sum_{i=1}^{n} K_\sigma(x - x_i)K_\sigma(y - y_i)
= \frac{1}{n} \sum_{i=1}^{n} K_\sigma(z - z_i)
\]

where \( K_\sigma \) is the discrete version of bivariate normal \( N(0, \sigma^2 I) \). For strength.data,

\[
[x, y] = \text{meshgrid}(0:4:200, 0:4:140);
\]

\[
\text{kernel} = \text{inline}(\exp(-x.^2/\sigma^2)/\text{sum(sum(exp(-x.^2/\sigma^2))))}', 'x', 'y', 'sigma');
\]

\[
\text{sigma} = 10, \%10, 15, 20
\]

\[
\text{fhat} = \text{zeros(size(x))};
\]

\[
\text{for } i = 1:147 \text{ number of sample}
\]

\[
\text{fhat} = \text{fhat} + \text{kernel(x-grip(i), y-arm(i),sigma)}/147;
\]

\[
\text{end;}
\]

\[
\text{hold on; plot(fhat)};
\]

\[
\text{ans} = 1.0000
\]

\[
\text{surf(x,y,fhat); colorbar}
\]

Our discrete version gives a probability function rather than a probability density such that \( \hat{f}_\sigma \) sum over all possible grid points should be 1, i.e.

\[
\sum_{i,j} \hat{f}_\sigma(x_i, y_j) = 1.
\]