# Handout #3

**Lesson 2 (cont): Looking at data, univariate analysis**

Density Estimation
Although a relative frequency histogram provides a sort of estimate of the density function for a dataset, it only places mass on the observed values (in other words, the "probability" of any unobserved value is always set to 0). Because a data vector represents a sample from an unknown distribution (usually continuous) it is often desirable to attempt to infer the probability of unobserved values.

A parametric approach. The MASS function `fitdistr` will find the maximum likelihood estimates for a specified distribution.

Syntax: fitdistr(x, densfun, start, …)

densfun options: "beta", "cauchy", "chi-squared", "exponential", "f", "gamma", "log-normal", "lognormal", "logistic", "negative binomial", "normal", "t", "uniform", "weibull"
start: initial starting values for parameters (you could use empirical estimates here)

Example:
```
library(MASS)
normsamp=rnorm(100,0,2)
hist(normsamp,prob=T)
mean(normsamp)
sd(normsamp)
fit.norm=fitdistr(normsamp, "normal")
```

*Why doesn't the estimate for the standard deviation agree with our sample estimate?*

Now let's superimpose it over our histogram to see how good it looks:

```
xvals=seq(-10,10,.01)
lines(xvals,dnorm(xvals,fit.norm$estimate[1],fit.norm$estimate[2]),col= "red")
```

Kernel density estimation. This is a nonparametric technique that is useful for estimating densities with multiple modes and/or atypical shapes.

The idea: Using a specified "kernel" function *K*, the density for a sample $x_1, x_2, ..., x_n$ is fit by

$\hat{f}(x) = \frac{1}{nb}\sum_{j=1}^{n} K\left(\frac{x - x_j}{b}\right)$ for a specified bandwidth *b*.

Typical kernel functions:
Gaussian (standard normal)

Rectangular (uniform)
Triangular: $f(x) = 1 - |x|, \quad -1 \le x \le 1$
The "smoothness" of the density estimate is determined by the bandwidth *b*.

In R, the function `density` fits a kernel density estimate to a numerical vector.

Syntax: density(x, bw = "nrd0", adjust = 1, kernel = c("gaussian", "epanechnikov", "rectangular", "triangular", "biweight", "cosine", "optcosine"), window = kernel, width, give.Rkern = FALSE, n = 512, from, to, cut = 3, na.rm = FALSE)

Most of these options can be ignored for now: We're mainly interested in the choice of kernel and the binwidth, which can be specified using either the "bw" or "width" options.

Let's look at the different kernel shapes for a single point:

```
x=0
par(mfrow=c(3,3))
plot(density(x, kernel= "g", width=1),main= "Gaussian")
plot(density(x, kernel= "e", width=1),main= "Epanechnikov")
plot(density(x, kernel= "r", width=1),main= "Rectangular")
plot(density(x, kernel= "b", width=1),main= "Biweight")
plot(density(x, kernel= "c", width=1),main= "Cosine")
plot(density(x, kernel= "o", width=1),main= "Optcosine")
plot(density(x,kernel="t",width=1), main= "Triangular")
plot(density(x,kernel="t",width=2), main= "Triangular, width=2")
plot(density(x,kernel="t",width=4), main= "Triangular, width=4")
```

Add a second point:

```
x=c(0,2)
par(mfrow=c(2,2))
plot(density(x,kernel="t",width=1))
plot(density(x,kernel="t",width=2))
plot(density(x,kernel="t",width=4))
plot(density(x,kernel="t",width=8))
```

Now let's try it for a bimodal random sample.

```
par(mfrow=c(1,1)
x=c(rnorm(10,0,1),rnorm(10,4,1))
hist(x,prob=T,xlim=c(-3,7))
lines(seq(-3,7,.1),0.5*dnorm(seq(-3,7,.1),0,1)+0.5*dnorm(seq(-3,7,.1),4,1), type="l")
lines(density(x,kernel="g"),lty=2)
```

*How did it do? Try changing the binwidth and the kernel type…*

There are several choices of calculated bandwidths available in R:

"nrd0" (the default): $0.9\min(s, IQR/1.34)n^{-1/5}$, where $s$ is the sample standard deviation, $IQR$ is the inter-quartile range, and $n$ is the sample size (this is the Silverman formula)

"nrd": $1.06\min(s, IQR/1.34)n^{-1/5}$ (Scott's variation)

"ucv" and "bcv": cross-validation based methods to find the optimal width

"SJ-ste" and "SJ-dpi": methods designed to minimize the estimated mean integrated squared error (see p.129 for more details)

Here's an example (stolen from the help files) comparing the fit generated by different bin widths for the precipitation data that we looked at last time:

```
data(precip)
plot(density(precip, n = 1000),xlab= "Rainfall in Inches", main = "Density Estimates")
rug(precip)
lines(density(precip, bw="nrd"), col = 2)
lines(density(precip, bw="ucv"), col = 3)
lines(density(precip, bw="bcv"), col = 4)
lines(density(precip, bw="SJ-ste"), col = 5)
lines(density(precip, bw="SJ-dpi"), col = 6)
legend(55, 0.035, legend = c("nrd0", "nrd", "ucv", "bcv", "SJ-ste", "SJ-dpi"), col = 1:6, lty = 1)
```

Let's wrap up by looking at the "geyser" dataset, which is also discussed in V&R. It contains estimates for both eruption lengths and waiting times between eruptions for the geyser Old Faithful. Try out different kernels and binwidths – how does the estimate vary with your choices?