1. No class Fri May 28 or Mon May 31.
2. Kernel regression and kernel smoothing point processes.
3. Intensities and conditional intensities for point processes.
4. Poisson processes.

2) Kernel regression and kernel smoothing point processes.
Kernel density estimation.
$\hat{f}(x) = 1/(n\,h) \sum_{i=1}^{n} K[(x - x_i)/h]$.
Kernel regression.
$m^0(x) = 1/n \sum_{i=1}^{n} w_i Y_i / \sum_{i=1}^{n} w_i$, where $w_i = K[(x_i-x)/h]$.

```
## a) kernel density estimation or kernel smoothing
x1 = sort(c(runif(1000)*30,rnorm(300)*2+10))
hist(x1,nclass=50,prob=T,main="kernel density estimates of
x1")
lines(x1,dunif(x1,min=0,max=30)*1000/1300+dnorm(x1,
mean=10,sd=2)*300/1300,lty=1,col=1)
lines(density(x1,bw=.3),lty=1,col=2)
z1 = density(x1,bw=bw.nrd0(x1))
lines(z1,lty=1,col=3)
z2 = density(x1,bw=bw.nrd(x1))
lines(z2,lty=1,col=4)
lines(density(x1,bw=10),lty=1,col=5)
legend(20,.08,c("true density", "h=0.3", "bw.nrd0",
"bw.nrd", "h=10"),
lty=rep(1,5),col=1:5)
```

Kernel smoothing for point processes is the same as kernel density estimation, except that the integral over all times of your estimate does not have to be 1 with kernel smoothing.

Start with
hist(x1,nclass=50,prob=F,main="kernel density estimates of x1")

b) kernel regression
y = x1^3/2000 - x1^2/100 + x1/2 - sin(x1) + 1.2*rnorm(1300)
plot(x1,y,pch=".",main="kernel regression")
lines(x1,x1^3/2000 - x1^2/100 + x1/2 - sin(x1),col=2)
z2 = ksmooth(x1,y, bandwidth = bw.nrd(x1))
lines(z2,col=3)
lines(ksmooth(x1,y,bandwidth=30),col=4)
legend(10,20,c("true mean of Y given X", "bw.nrd","h=30"),lty=c(1,1,1),col=2:4)

c) generalized additive models.
Fit a model $Y = f_1(x_1) + f_2(x_2) + ... + f_p(x_p) + \varepsilon$, where $f_1,...,f_p$ are continuous functions, and ε_i are iid with mean 0.
Estimate f_1, ..., f_p nonparametrically.
gam() does this with splines, but you could do this with kernels.
x1 = runif(100)*10
x2 = runif(100)*20
x3 = runif(100)*30
y = x1^3 - 10*x1^2 + 5*x1 - .1*x2^3 + 2*x2^2 + 4*x2 + 12 + 7 * x3 + 14*rnorm(100)
plot(x1,y)
c1 = ksmooth(x1,y, bandwidth = bw.nrd(x1),x.points=x1)
lines(c1)
fit1 = rep(0,100)

```
fit1[order(x1)] = c1$y
y1 = y - fit1 ## these are the residuals from this first fit
## note that x1 is not ordered, so if you do lines(x1,y1), it
won't look good
## because lines joins the first point to the second point to
the third, etc.
## but you can do points(x1,fit1,pch=2) to check that things
are going ok.
plot(x2,y1,ylab="residuals after first kernel smoothing")
c2 = ksmooth(x2,y1, bandwidth = bw.nrd(x2),x.points=x2)
lines(c2)
fit2 = fit1
fit2[order(x2)] = c2$y
## to check, try points(x2,fit2,pch=2)
y2 = y1 - fit2 ## residuals after 2nd fit
plot(x3, y2,xlab="x3",ylab="residuals after 2 kernel
smoothings")
c3 = lsfit(x3, y2)
abline(c3)
c3$coef ## compare with 7
plot(x3,c3$resid,ylab="residuals after 2 kernels and a line")
par(mfrow=c(1,2))
hist(y2,nclass=20,xlab="residuals after 2 kernels",main="",
xlim=range(c(y2,c3$resid)))
hist(c3$resid, nclass=20, xlab="residuals after 2 kernels
and a line",main="", xlim=range(c(y2,c3$resid)))
par(mfrow=c(1,1))
plot(x3, y2,xlab="x3",ylab="residuals after 2 kernel
smoothings")
c4 = ksmooth(x3, y2, bandwidth = bw.nrd(x3),x.points=x3)
lines(c4)
```

```
fit3 = fit1
fit3[order(x3)] = c4$y
y3 = y2 - fit3
plot(x3, y3,ylab="residuals after 3 kernel smoothings")
par(mfrow=c(1,2))
x5 = hist(c3$resid, nclass=20, xlab="residuals after 2
kernels and a line",main="", xlim=range(c(y3,c3$resid)))
hist(y3, breaks=20, xlab="residuals after 3
kernels",main="", xlim=range(c(y3,c3$resid)))
```

3) Intensity and conditional intensity.
Intensity = rate.
$E\lambda(t) = \lim_{\Delta t \to 0} E[N(t, t+\Delta_t)]/\Delta_t =$ rate at which we expect points to accumulate around time t.
$E\lambda$ is the overall intensity. The conditional intensity λ (sometimes called $\lambda(t|H_t)$ is especially relevant for modeling and prediction. It is often interesting to look at the expected number of points around time t, given the whole history H_t consisting of all information on what points have occurred previously.

$\lambda(t) = \lim_{\Delta t \to 0} E[N(t, t + \Delta_t) \mid H_t]/\Delta_t =$ rate at which we expect points to accumulate around time t, given info on what points have occurred prior to t.

If, for disjoint sets $B_1, B_2, ..., B_j$, the number of points in these sets are independent of each other, then $E\lambda = \lambda$. If in addition N is simple and orderly, then N is called a Poisson process.

4) Poisson processes.

Suppose N is simple and orderly, and $N(B_1)$, $N(B_2)$, ..., $N(B_j)$, are independent random variables, provided B_1, B_2, ..., B_j are disjoint sets. Then N is a Poisson process. The name comes from the fact that it follows that for any set B, N(B) has a Poisson distribution with some mean $\mu(B)$. That is:

$$P[N(B) = k] = \mu(B)^k e^{-\mu(B)} / k!, \quad \text{for } k = 0,1,2, \ldots$$

For a Poisson process, $E\lambda = \lambda$, so we can call them both the "intensity".

How does the intensity λ relate to the mean $\mu(B)$?

When λ exists and is finite, the mean $\mu(B)$ is simply the integral of the overall intensity:

$$\mu(B) = \int_B \lambda(t)dt.$$

$\lambda(t)$ is the expected number of points per unit time and $\mu(B)$ is the expected number of points in the time interval B.

If N is a Poisson process, does that mean that $\lambda(t)$ is constant for all t?

No. It just means that $\lambda(t)$ doesn't depend on what other points have occurred. The rate could be high at some times

and low at other times, regardless of what other points have happened.

Can a point process not be a Poisson process?

Yes. We will discuss examples next time.