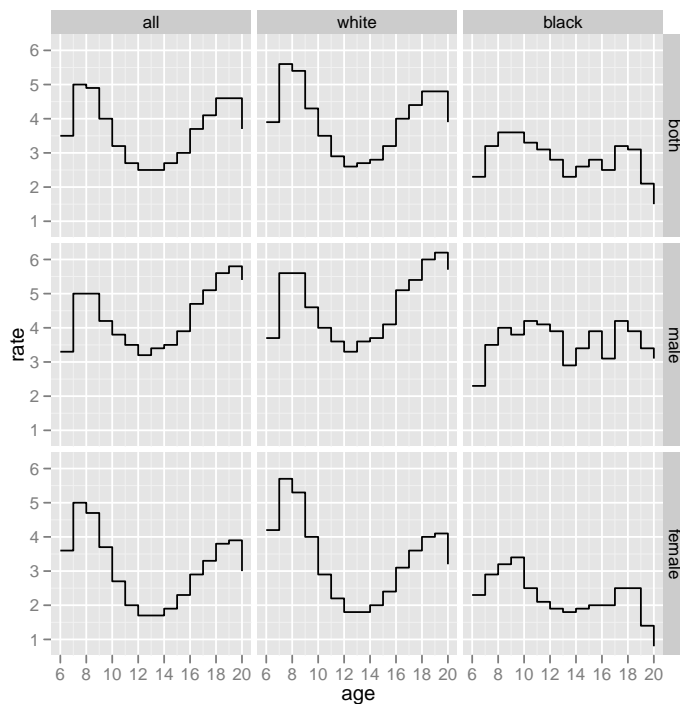


Read in data and add labels:

```
> cdat <- read.csv("cancer.dat", comment = "#", header = FALSE)
> names(cdat) <- c("age", "race", "sex", "rate")
> cdat <- transform(cdat, race = factor(race, labels = c("all",
  "white", "black")), sex = factor(sex, labels = c("both",
  "male", "female")))
```

Basic plot:

```
> library(ggplot2)
> g1 <- ggplot(cdat, aes(x = age, y = rate)) + geom_step() + facet_grid(sex ~
  race)
> print(g1)
```



Function for 2-component normal mixture prediction:

```
> binormfun <- function(p, age) {
  with(as.list(p), r * (p * dnorm(age, mean = age1, sd = exp(logsd1)) +
    (1 - p) * dnorm(age, mean = age2, sd = exp(logsd2))))
}
```

Objective function for sum of squares:

```
> ssqfun <- function(p, age, obsrate) {
  sum((obsrate - binormfun(p, age))^2)
}
```

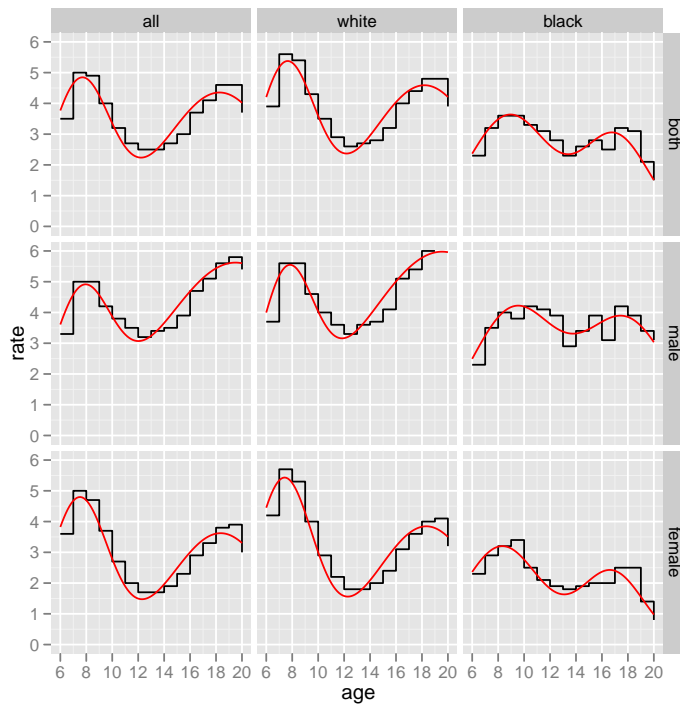
Try it out on the “white male” data:

```
> cdat.whitemale <- subset(cdat, sex == "male" & race == "white")
> opt1 <- optim(fn = ssqfun, par = c(age1 = 8, age2 = 18, p = 0.4,
  r = 3, logsd1 = 0, logsd2 = 0), age = cdat.whitemale$age,
  obsrate = cdat.whitemale$rate, method = "L-BFGS-B", lower = c(age1 = 5,
    age2 = 5, p = 0.01, r = 0.1, logsd1 = -2, logsd2 = -2),
  upper = c(age1 = 21, age2 = 21, p = 0.99, r = 1000, logsd1 = 2,
    logsd2 = 2))
> with(cdat.whitemale, {
  plot(age, rate, type = "s")
})
> agevec <- seq(6, 20, length = 101)
> lines(agevec, binormfun(opt1$par, agevec), col = 2)

> fitvals <- ddply(cdat, c("race", "sex"), function(d) {
  optd <- optim(fn = ssqfun, par = c(age1 = 8, age2 = 18, p = 0.4,
    r = 3, logsd1 = 0, logsd2 = 0), age = d$age, obsrate = d$rate,
    method = "L-BFGS-B", lower = c(age1 = 5, age2 = 5, p = 0.01,
      r = 0.1, logsd1 = -1, logsd2 = -1), upper = c(age1 = 21,
        age2 = 21, p = 0.99, r = 1000, logsd1 = 2, logsd2 = 2))
  data.frame(matrix(optd$par, nrow = 1))
})
> names(fitvals) <- c("race", "sex", names(opt1$par))

> predvals <- ddply(fitvals, c("race", "sex"), function(d) {
  data.frame(race = d$race, sex = d$sex, age = agevec, rate = binormfun(unlist(d)[-(1:2)]
    agevec))
})

> print(g1 + geom_line(data = predvals, colour = "red") + ylim(0,
  6))
```



(Black/both, male/all, and white/female were bogus. with lower bound of -2 on logsd; it all looks pretty good with lower bound of -1 on logsd)

Note that there is some ambiguity in the definitions: in particular, the order of the components is reversed in the black/both case (we should swap age1/age2, logsd1/logsd2, make  $p = 1 - p$ )

```
> fitvals
```

	race	sex	age1	age2	p	r	logsd1	logsd2
1	all	both	7.561503	18.271615	0.3631651	73.96802	0.8371808	1.4623602
2	all	male	7.622782	19.497017	0.2254730	105.27504	0.8150130	1.7555535
3	all	female	7.462559	18.354971	0.4297468	61.06485	0.7937995	1.3442167
4	white	both	7.518527	18.194329	0.3705929	79.05653	0.8190903	1.4643271
5	white	male	7.569445	19.547095	0.2154434	113.87088	0.7292788	1.7856019
6	white	female	7.369442	18.337691	0.4455884	66.70584	0.7928463	1.3444088
7	black	both	17.037426	8.885149	0.3984541	47.16762	0.9423928	1.1403505
8	black	male	9.213164	17.798298	0.5133961	64.31690	1.1739763	1.2004874
9	black	female	8.305421	16.752966	0.6211937	38.19505	1.0856280	0.8905987

```
> print(ggplot(melt(fitvals, id.var = 1:2), aes(x = race, y = value,
  colour = sex)) + geom_point() + facet_wrap(~variable, scale = "free"))
```

