MA 677 final project

Peng Liu

2022-05-12

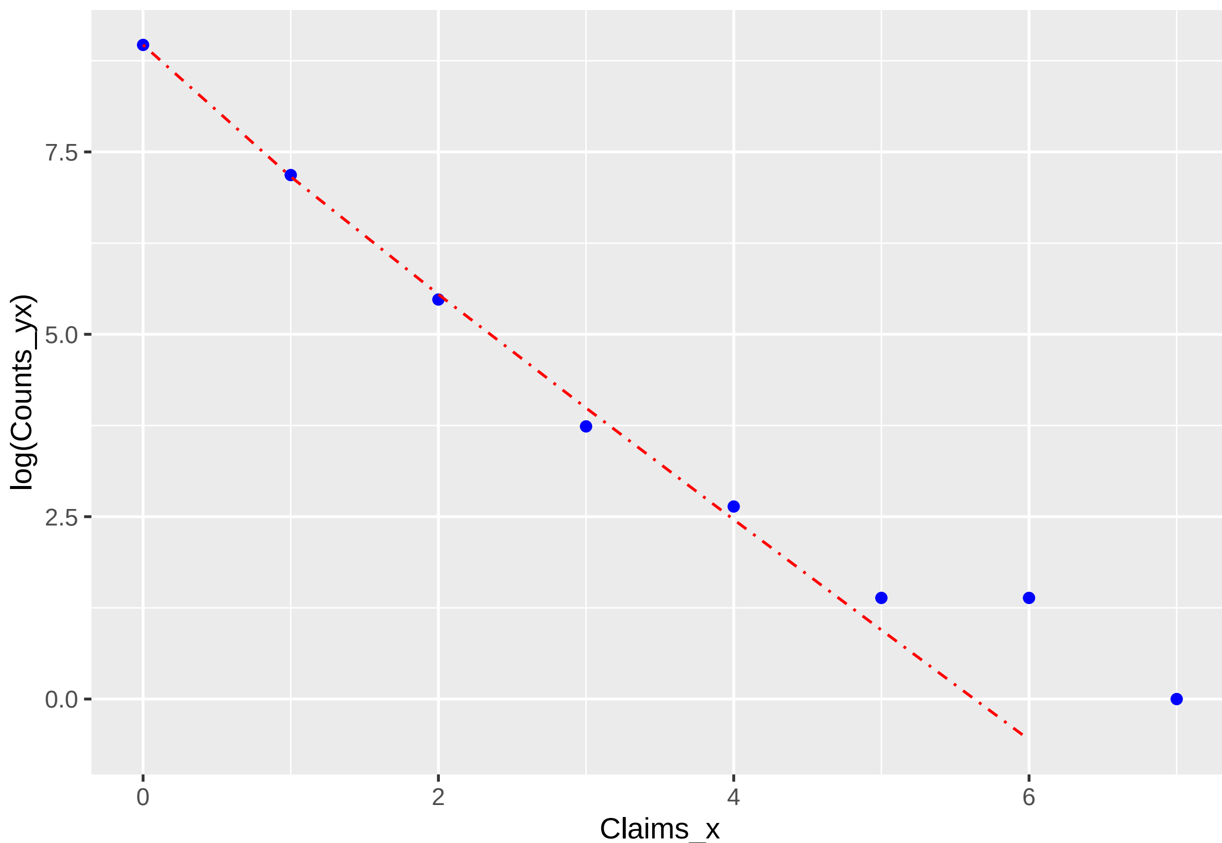
insurance claims

```
auto <- data.frame(Claims_x=seq(0,7),
           Counts_yx=c(7840,1317,239,42,14,4,4,1))
#auto
# Calculate the expectation of the number of claims for a single customer
n <- 8
robbin<-round(((auto$Claims_x+1)[1:7]*auto$Counts_yx[2:8]/auto$Counts_yx[1:7]),3)
# Calculate the parametric estimated marginal density and
# then get the maximum likelihood fitting to the counts y_x
f <- function(x,mu,sigma){
  gamma = sigma / (1 + sigma)
  numer = gamma ^ (mu + x) * gamma(mu + x)
  denom = sigma ^ mu * gamma(mu) * factorial(x)
  return(numer/denom)
}
neg_like <-function(param){
  mu=param[1]
  sigma=param[2]
  tmp=-sum(auto$Counts*log(f(auto$Claims_x,mu=mu,sigma=sigma)))
  return(tmp)
}
p <- array(c(0.5, 1), dim = c(2, 1))
ans_auto <- nlm(f = neg_like,p,hessian=T)
mu=ans_auto$estimate[1]
sigma=ans_auto$estimate[2]
re <- round((seq(0,6)+1)*f(seq(0,6)+1,mu,sigma)/f(seq(0,6),mu,sigma),3)
rbind(robbin,re)
```

```
##         [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]
## robbin 0.168 0.363 0.527 1.333 1.429 6.000 1.750
## re     0.164 0.398 0.632 0.866 1.100 1.334 1.568
```

```
auto$pred=c(f(seq(0,6),mu,sigma)*9461,NA)

ggplot(data=auto) +
  geom_point(aes(x=Claims_x,y=log(Counts_yx)),color='blue')+
  geom_line(aes(x=Claims_x,y=log(pred)),color='red',lty=4)
```



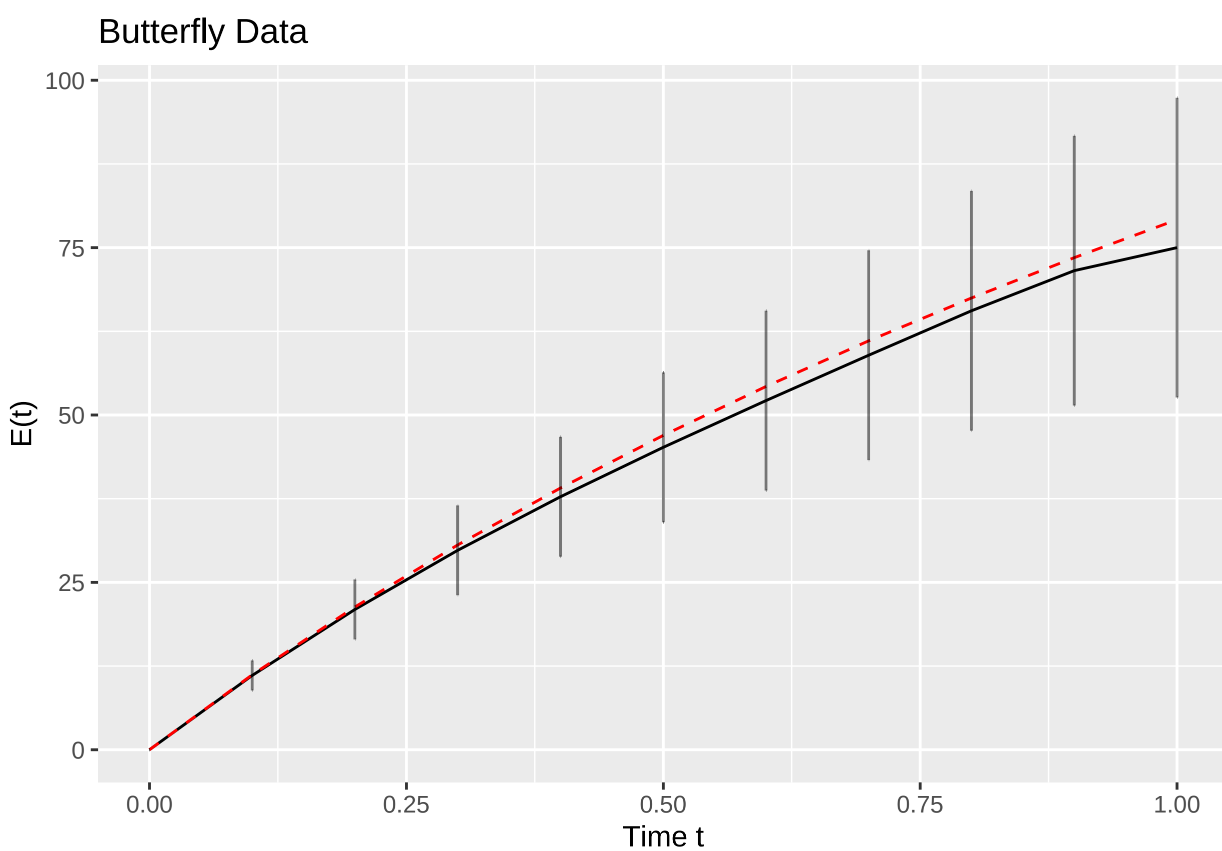
The Missing-Species Problem

```
x<- seq(1,24)
y <- c(118, 74, 44, 24, 29, 22, 20, 19, 20, 15, 12, 14, 6, 12, 6, 9, 9, 6, 10, 10, 11, 5, 3, 3)
butterfly <- data.frame(x, y)
t= seq(0, 1, 0.1)
exp <- NULL
sd <- NULL
for (i in 1:length(t)){
  exp[i] <- round(sum(y*(t[i]^x)*(-1)^(x-1)),2)
  sd[i] <- round(sqrt(sum(y*t[i]^(2))),2)
}
Fisher<- data.frame(t, exp, sd)
```

```
v <- 0.104
sigma <-  89.79
gamma <- sigma / (1 + sigma)
E_1 <- y[1]
gamma_est <- NULL
for (i in 1:length(t)){
  gamma_est[i] <- round(E_1*((1 - (1+gamma*t[i])^(-v)) / (gamma * v)),2)
}
E_1 <- y[1]
```

```
gamma_est <- NULL
for (i in 1:length(t)){
  gamma_est[i] <- round(E_1*((1 - (1+gamma*t[i])^(-v)) / (gamma * v)),2)
}
```

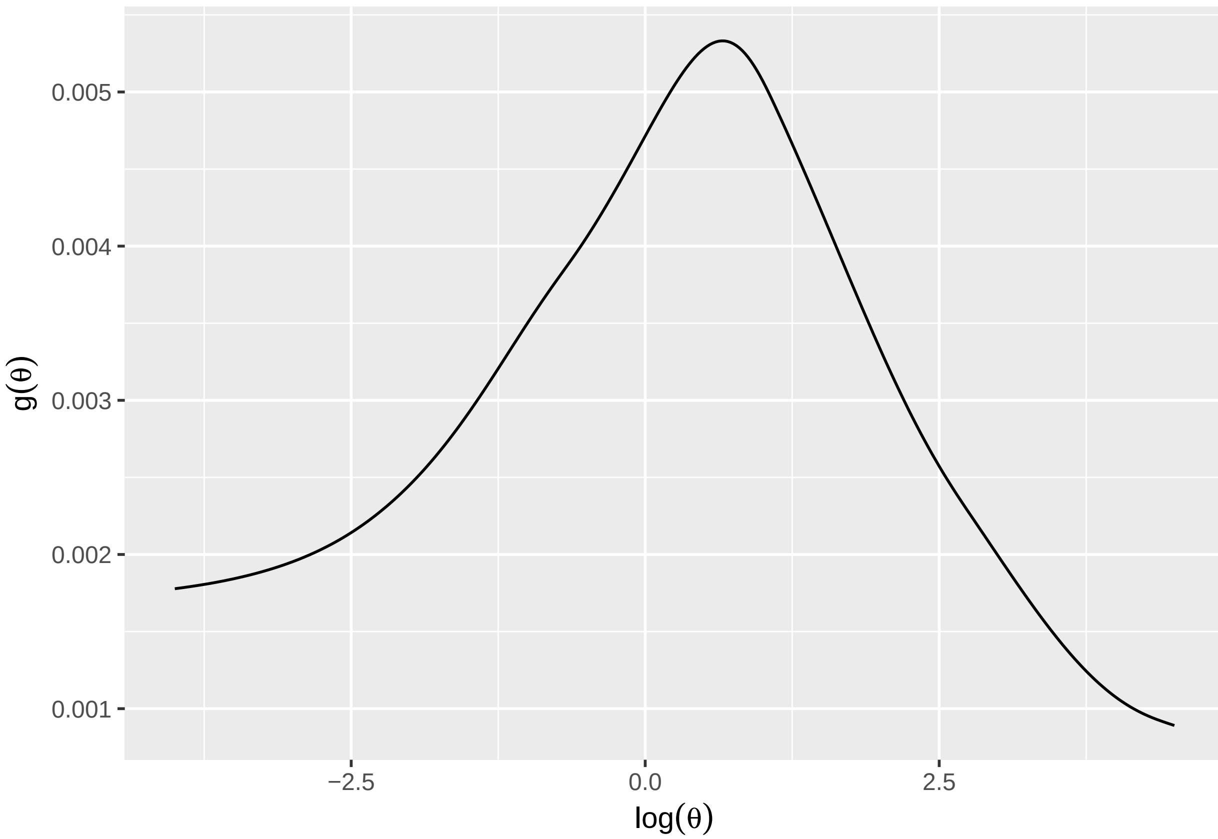
```
# Nonparametric fit (solid) +/- 1 standard deviation; gamma model (dashed).
ggplot(data=Fisher, aes(x=t))+
  geom_line(aes(y=exp))+
  geom_line(aes(y=gamma_est), col="red", linetype="dashed")+
  geom_errorbar(aes(ymin=(exp-sd), ymax=(exp+sd)), width=0, alpha=0.5)+
  ggtitle("Butterfly Data")+ylab("E(t)")+xlab("Time t") +
  theme(legend.position="topleft")
```



Shakespeare's word counts

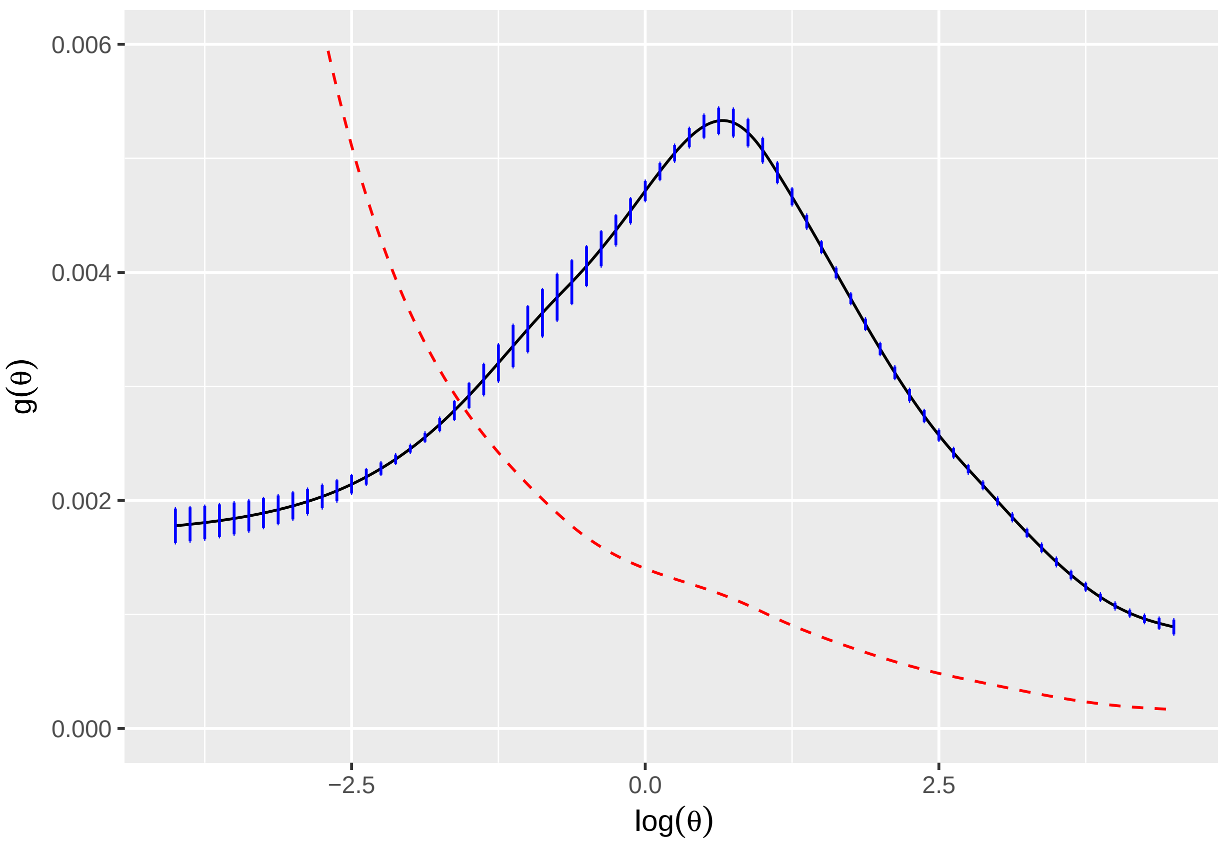
```
data(bardWordCount)
# str(bardWordCount)
lambda <- seq(-4, 4.5, .025)
tau <- exp(lambda)
result <- deconv(tau = tau, y = bardWordCount, n = 100, c0=2)
stats <- result$stats
# Empirical Bayes deconvoluation estimates
```

```r
ggplot() +
    geom_line(mapping = aes(x = lambda, y = stats[, "g"])) +
    labs(x = expression(log(theta)), y = expression(g(theta)))
```

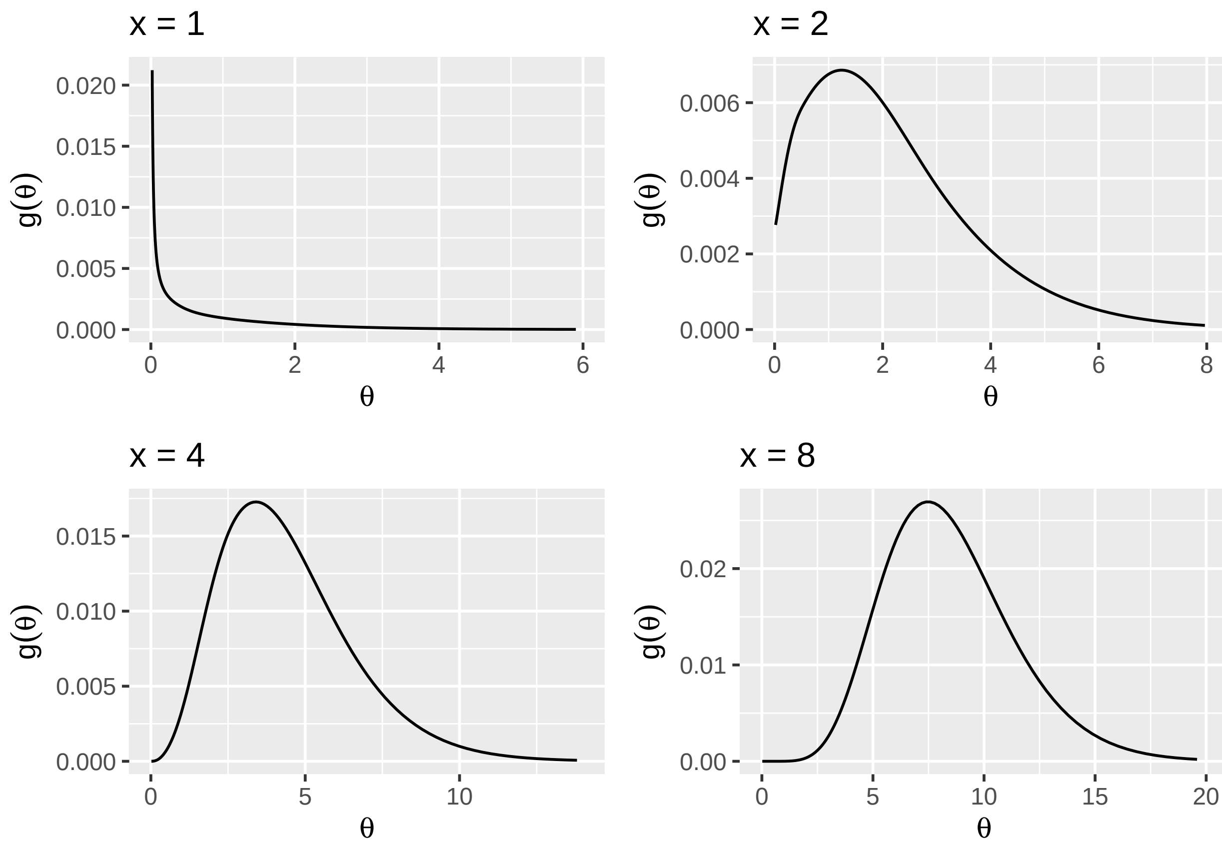



```r
d <- data.frame(lambda = lambda, g = stats[, "g"], tg = stats[, "tg"], SE.g = stats[, "SE.g"])

indices <- seq(1, length(lambda), 5)
ggplot(data = d) +
    geom_line(mapping = aes(x = lambda, y = g)) +
    geom_errorbar(data = d[indices, ],
                  mapping = aes(x = lambda, ymin = g - SE.g, ymax = g + SE.g),
                  width = .01, color = "blue") +
    labs(x = expression(log(theta)), y = expression(g(theta))) +
    ylim(0, 0.006) +
    geom_line(mapping = aes(x = lambda, y = tg), linetype = "dashed", color = "red")
```



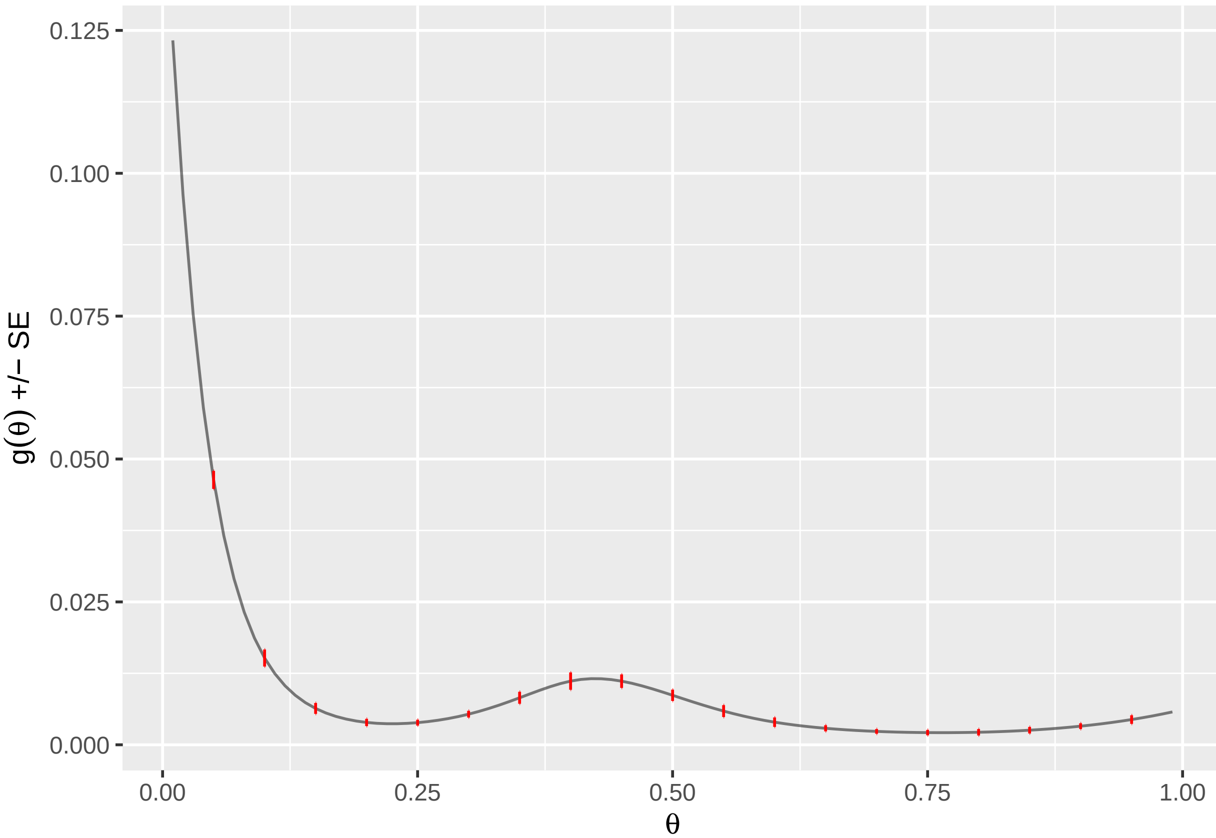

```
# Posterior estimates
gPost <- sapply(seq_len(100), function(i) local({tg <- d$tg * result$P[i, ]; tg / sum(tg)}))
plots <- lapply(c(1, 2, 4, 8), function(i) {
    ggplot() +
        geom_line(mapping = aes(x = tau, y = gPost[, i])) +
        labs(x = expression(theta), y = expression(g(theta)),
             title = sprintf("x = %d", i))
})
plots <- Map(f = function(p, xlim) p + xlim(0, xlim), plots, list(6, 8, 14, 20))
plot_grid(plotlist = plots, ncol = 2)
```



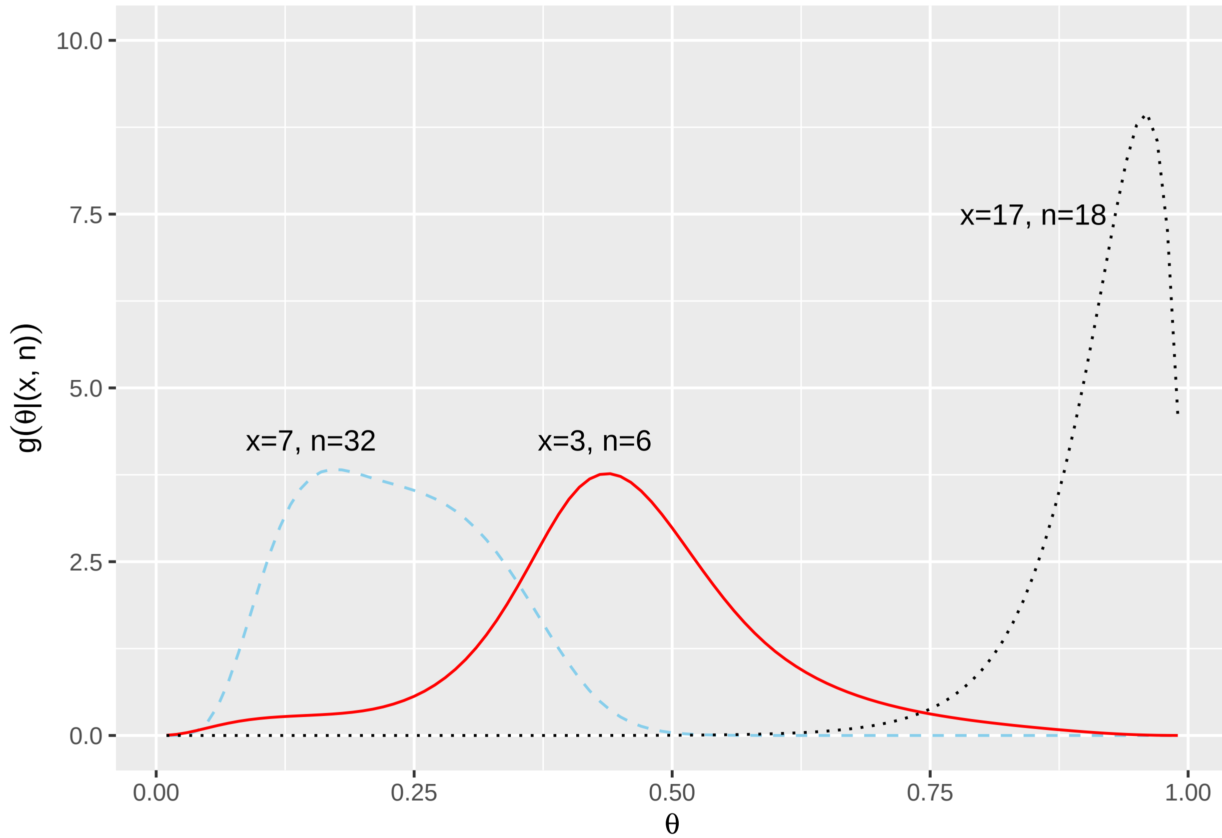
A Medical Example

```
data(surg)
tau <- seq(from = 0.01, to = 0.99, by = 0.01)
result <- deconv(tau = tau, X = surg, family = "Binomial", c0 = 1)
d <- data.frame(result$stats)
indices <- seq(5, 99, 5)
errorX <- tau[indices]

ggplot() +
    geom_line(data = d, mapping = aes(x = tau, y = g), alpha=0.5) +
    geom_errorbar(data = d[indices, ], mapping = aes(x = theta, ymin = g - SE.g, ymax = g + SE.g),width
    labs(x = expression(theta), y = expression(paste(g(theta), " +/- SE")))
```




```r
# Posterior Estimates
theta <- result$stats[, 'theta']
gTheta <- result$stats[, 'g']
f_alpha <- function(n_k, x_k) {
    ## .01 is the delta_theta in the Riemann sum
    sum(dbinom(x = x_k, size = n_k, prob = theta) * gTheta) * .01
}
g_theta_hat <- function(n_k, x_k) {
    gTheta * dbinom(x = x_k, size = n_k, prob = theta) / f_alpha(n_k, x_k)
}
# Empirical Bayes posterior densities of $\theta$ for three patients,
# given x= number of positive nodes,n= number of nodes.
g1 <- g_theta_hat(x_k = 7, n_k = 32)
g2 <- g_theta_hat(x_k = 3, n_k = 6)
g3 <- g_theta_hat(x_k = 17, n_k = 18)
ggplot() +
    geom_line(mapping = aes(x = theta, y = g1), col = "skyblue", linetype="dashed") +
    ylim(0, 10) +
    geom_line(mapping = aes(x = theta, y = g2), col = "red") +
    geom_line(mapping = aes(x = theta, y = g3), col = "black", linetype="dotted") +
    labs(x = expression(theta), y = expression(g(paste(theta, "|(x, n)")))) +
    annotate("text", x = 0.15, y = 4.25, label = "x=7, n=32") +
    annotate("text", x = 0.425, y = 4.25, label = "x=3, n=6") +
    annotate("text", x = 0.85, y = 7.5, label = "x=17, n=18")
```



References

[1] Haviland's lecture notes

[2] https://github.com/jrfiedler/CASI_Python/tree/master/chapter06

[3] https://github.com/bnaras/deconvolveR/blob/master/vignettes/deconvolution.Rmd

[4] https://github.com/MA615-Yuli