Final_Report_MA677

Shuting

5/11/2022

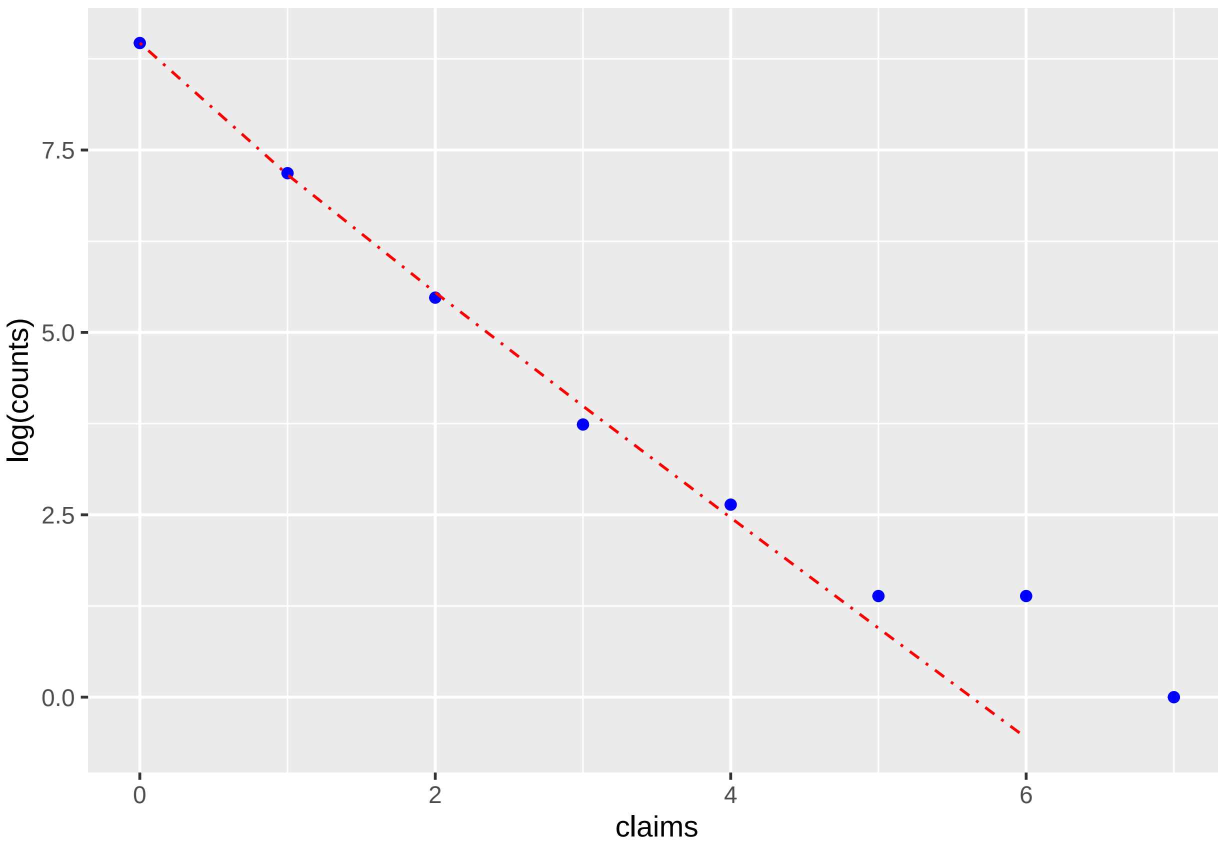
Introduction to Empirical Bayes

Insurance Claims

```
##import data
claims = seq(0,7)
counts = c(7840,1317,239,42,14,4,4,1)
dta <- data.frame(claims, counts)
##Robbins' formula
RobbinFormula <- NULL
for (i in 1:length(counts)){
  RobbinFormula[i] <- round(claims[i+1]*(counts[i+1]/counts[i]),3)
}
dta <- cbind(dta, RobbinFormula)
##gamma MLE
f <- function(x,nu,sigma){
  gamma = sigma / (1 + sigma)
  numer = gamma ^ (nu + x) * gamma(nu + x)
  denom = sigma ^ nu * gamma(nu) * factorial(x)
  return(numer/denom)
}
negloglikelihood <- function(params){
  nu = params[1]
  sigma = params[2]
  out = -sum(counts*log(f(claims,nu=nu,sigma=sigma)))
  return(out)
}

p <- matrix(c(0.5, 1),2,1)
ans_auto <- nlm(f = negloglikelihood,p,hessian=T)
nu = ans_auto$estimate[1]
sigma = ans_auto$estimate[2]

gamma_mle <- NULL
for (i in 0:6){
  gamma_mle[i+1] <- round((i+1)*f((i+1), nu, sigma)/f(i, nu, sigma),3)
}

##combination
dta <- cbind(dta, gamma_mle = c(gamma_mle, NA))
t(dta)
```

```
##                     [,1]      [,2]    [,3]   [,4]    [,5]  [,6]  [,7] [,8]
## claims             0.000     1.000   2.000  3.000   4.000 5.000 6.000    7
## counts          7840.000  1317.000 239.000 42.000  14.000 4.000 4.000    1
## RobbinFormula      0.168     0.363   0.527  1.333   1.429 6.000 1.750   NA
## gamma_mle          0.164     0.398   0.632  0.866   1.100 1.334 1.568   NA
```

```
dta$gamma_counts <- c(f(seq(0,6), nu, sigma)*sum(counts), NA)

ggplot(dta) +
  geom_point(aes(x=claims,y=log(counts)),color='blue')+
  geom_line(aes(x=claims,y=log(gamma_counts)),color='red',lty=4)
```



So, without prior distribution of $g(\theta)$, we can also get the expectation of number of claims for single customer.

Missing Species

```
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)
##exp&sd
t <- seq(0,1,by=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)
}
dta <- data.frame(t=t, exp=exp, sd=sd)
```

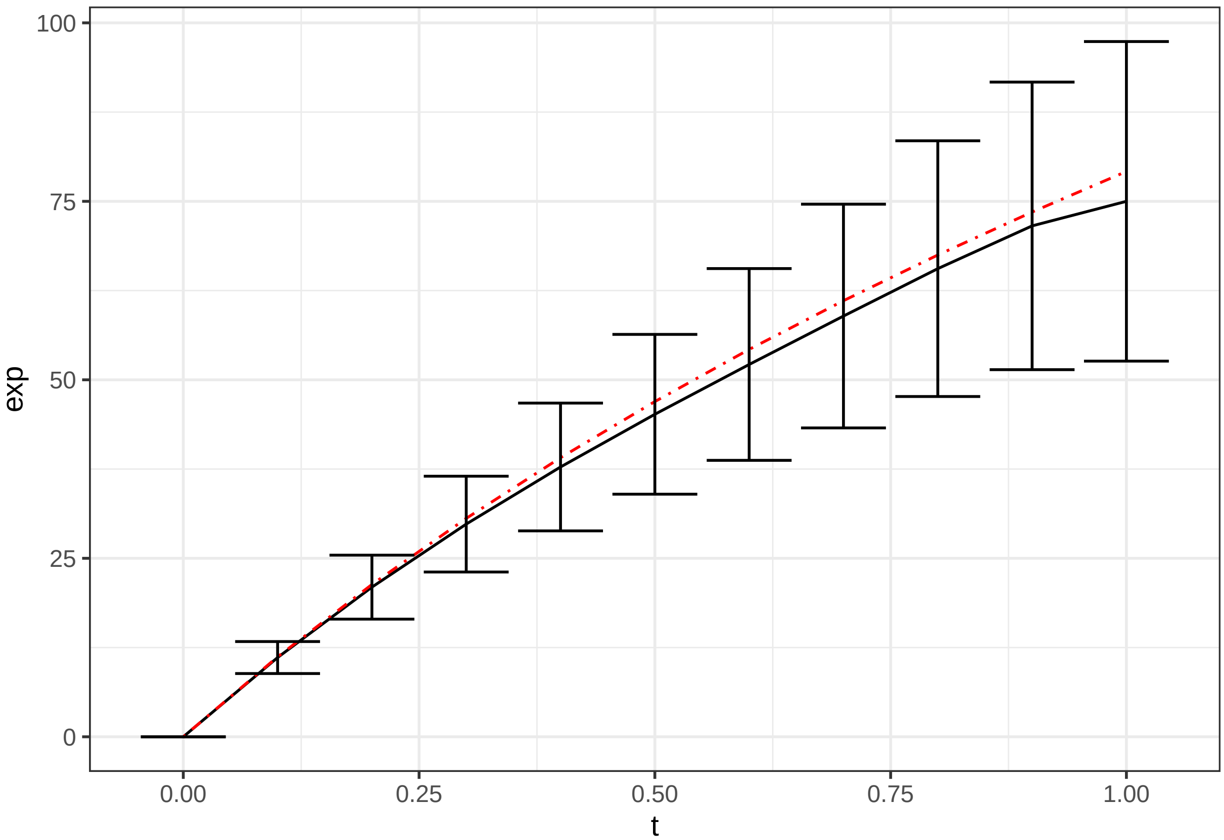
```
dta
```

```
##       t   exp    sd
## 1  0.0  0.00  0.00
## 2  0.1 11.10  2.24
## 3  0.2 20.96  4.48
## 4  0.3 29.79  6.71
## 5  0.4 37.79  8.95
## 6  0.5 45.17 11.19
## 7  0.6 52.15 13.43
## 8  0.7 58.93 15.67
## 9  0.8 65.57 17.91
## 10 0.9 71.56 20.14
## 11 1.0 75.00 22.38
```

```
##gamma estimate
v <- 0.104
sigma <-  89.79
gamma <- sigma / (1 + sigma)
e1 <- y[1]
gamma_esti <- NULL
for (i in 1:length(t)){
  gamma_esti[i] <- round(e1*((1 - (1+gamma*t[i])^(-v)) / (gamma * v)),2)
}
gamma_esti
```

```
##  [1]  0.00 11.20 21.33 30.59 39.09 46.95 54.26 61.08 67.48 73.50 79.18
```

```
##vasualization
ggplot(dta)+
  geom_line(aes(x=t,y=exp))+
  geom_line(aes(x=t,y=gamma_esti),lty=4,color='red')+
  geom_errorbar(aes(x=t,ymin=(exp-sd),ymax=(exp+sd)))+theme_bw()
```

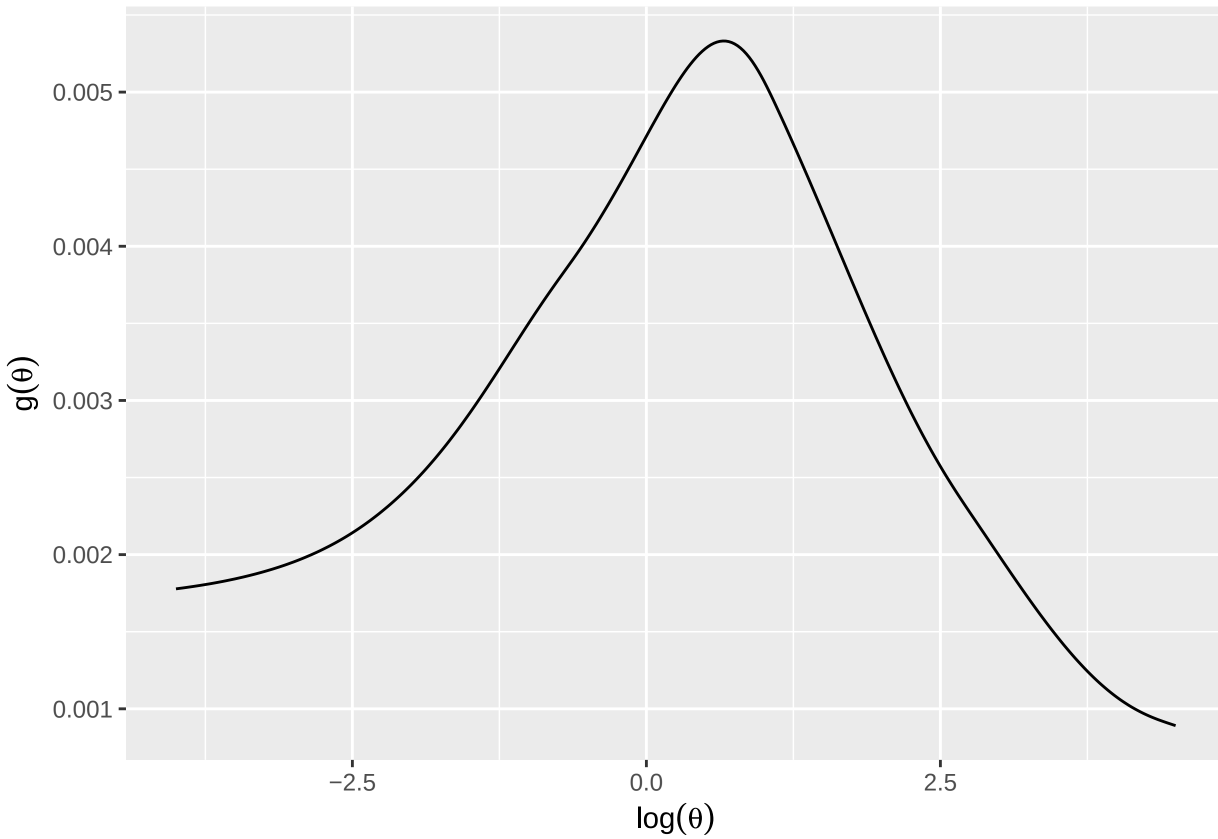


Shakespeare's Vocabulary

```
#Reference: https://github.com/bnaras/deconvolveR/blob/master/vignettes/deconvolution.Rmd
data("bardWordCount", package = "deconvolveR")
str(bardWordCount)
```

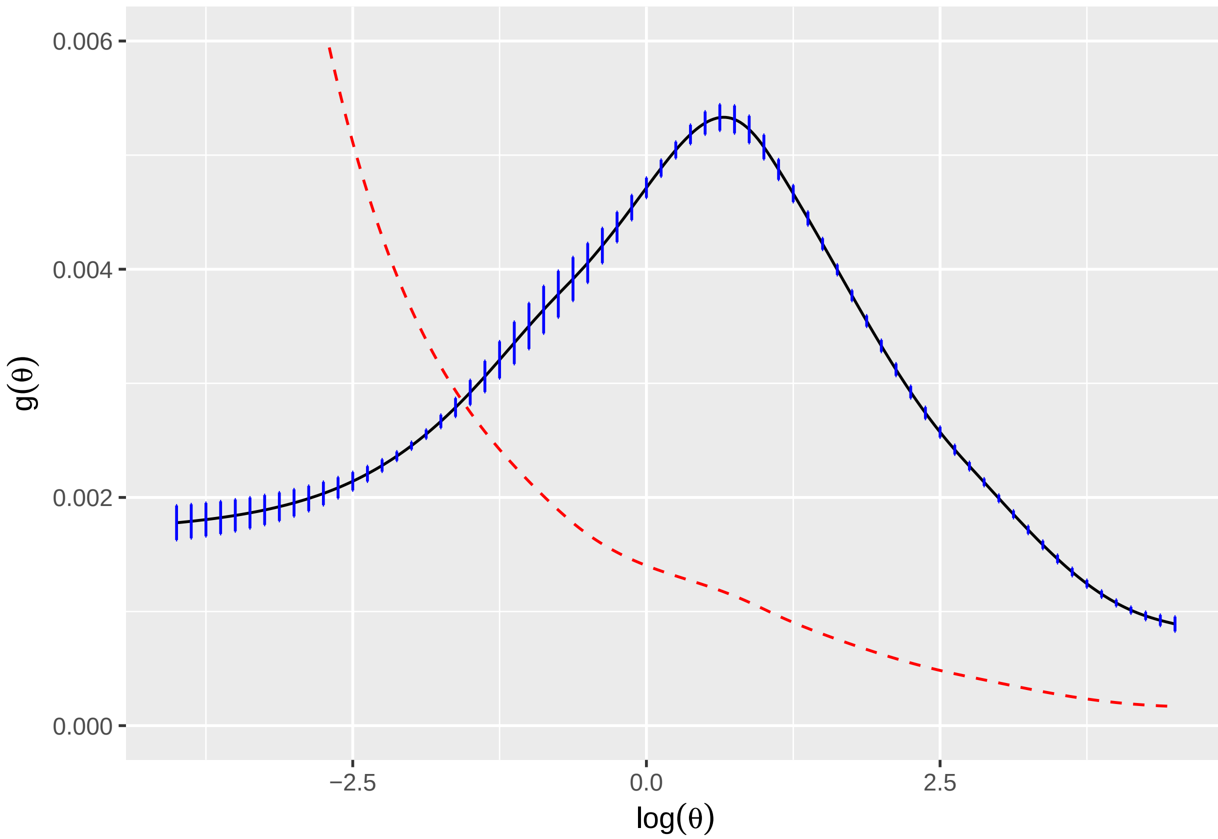
```
##  num [1:100] 14376 4343 2292 1463 1043 ...
```

```
lambda <- seq(-4, 4.5, .025)
tau <- exp(lambda)

result <- deconv(tau = tau, y = bardWordCount, n = 100, c0=2)
stats <- result$stats

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

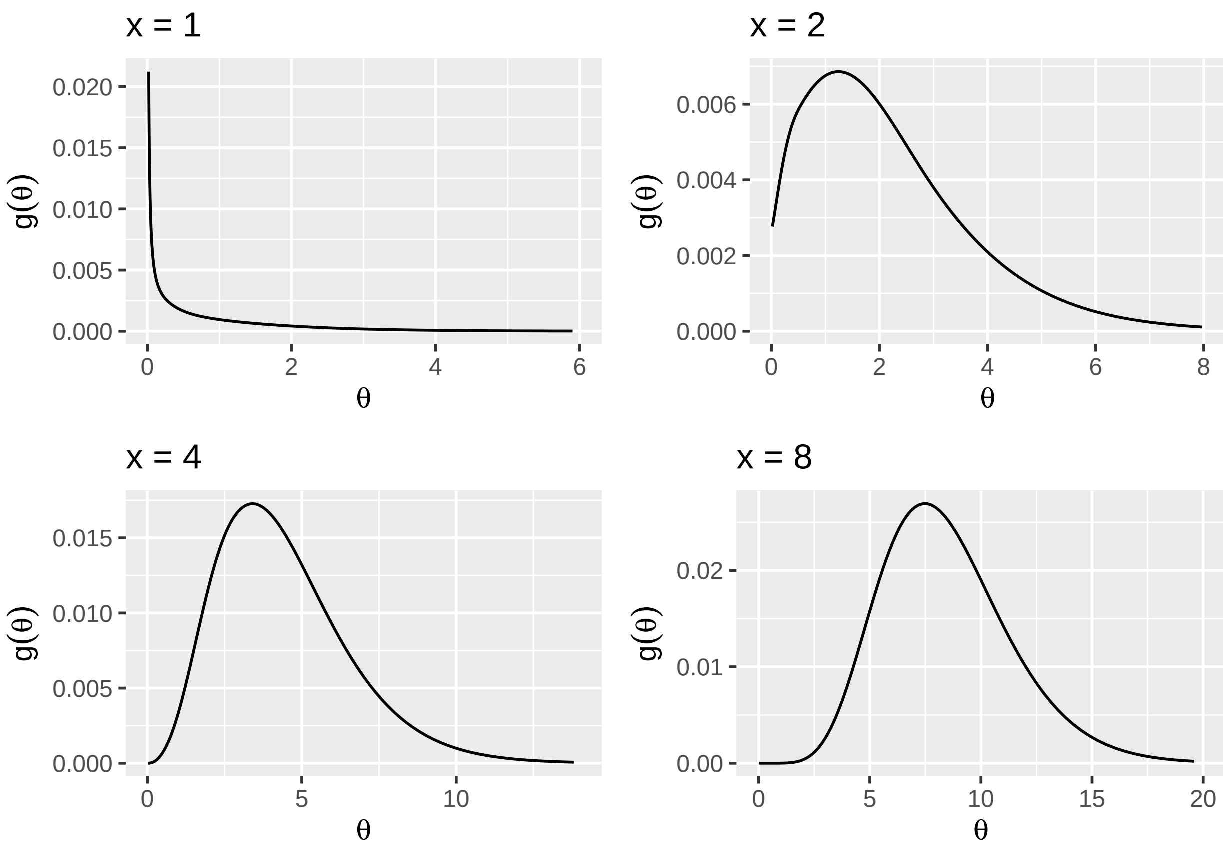


The plot below shows the Empirical Bayes deconvoluation estimates for the Shakespeare word counts.

```
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")
```

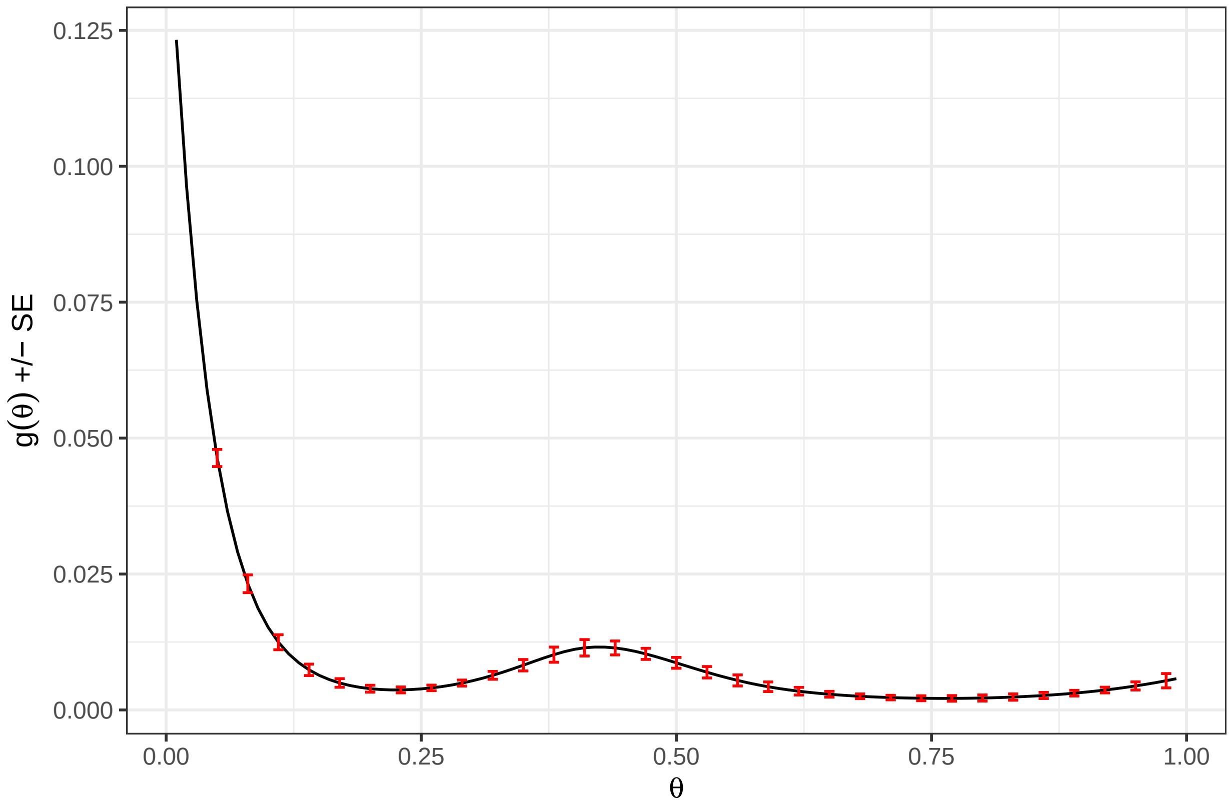



```
gPost <- sapply(seq_len(100), function(i) local({tg <- d$tg * result$P[i, ]; tg / sum(tg)}))
plots <- lapply(c(1, 2, 4, 8), function(i) {
    ggplot() +
        geom_line(mapping = aes(x = tau, y = gPost[, i])) +
        labs(x = expression(theta), y = expression(g(theta)),
             title = sprintf("x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



lymph node counts

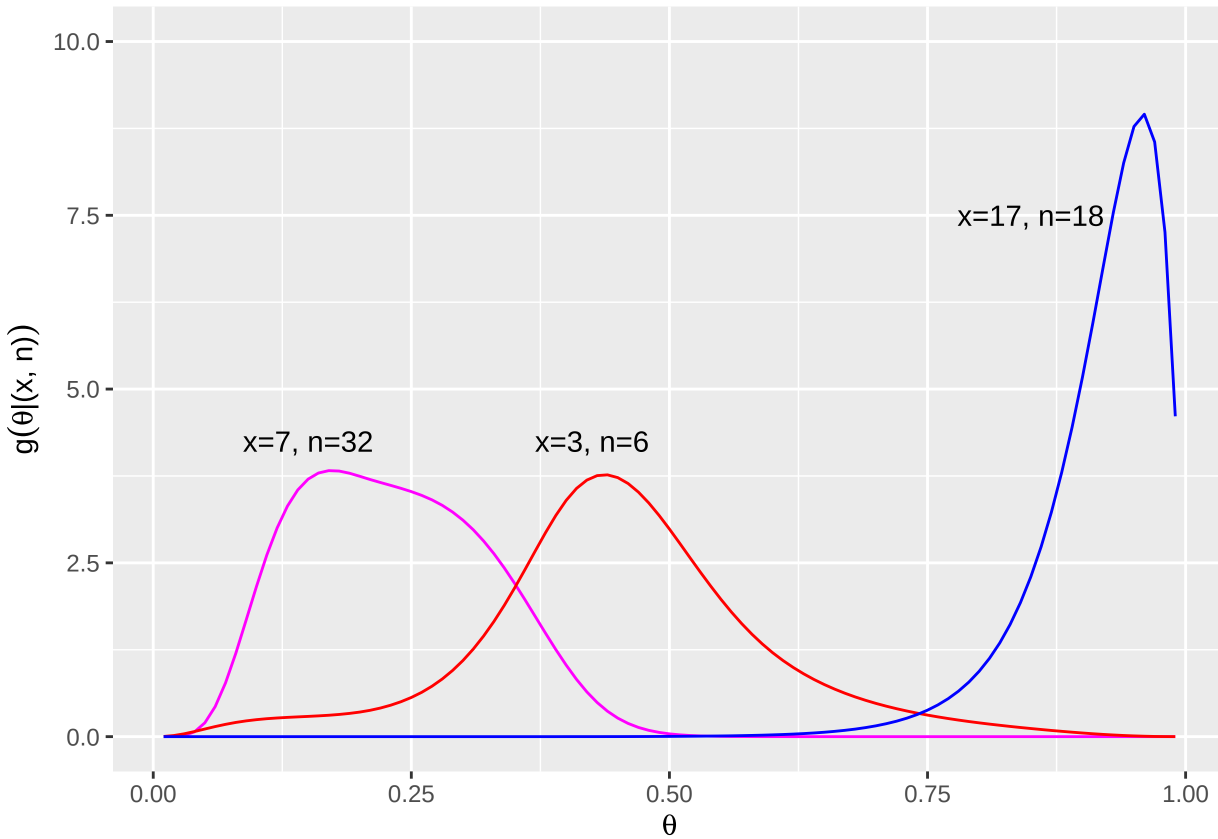
```
#Reference: https://github.com/bnaras/deconvolveR/blob/master/vignettes/deconvolution.Rmd
data(surg)
tau <- seq(from = 0.01, to = 0.99, by = 0.01)
result <- deconv(tau = tau, X = surg, family = "Binomial")
d <- data.frame(result$stats)
indices <- seq(5, 99, 3)
errorX <- tau[indices]
ggplot() +
  geom_line(data = d, mapping = aes(x = tau, y = g)) +
  geom_errorbar(data = d[indices, ],
                mapping = aes(x = theta, ymin = g - SE.g, ymax = g + SE.g),
                width = .01, color = "red") +
  labs(x = expression(theta), y = expression(paste(g(theta), " +/- SE")), caption = "Figure")+theme_bw(
```



Figure

Estimated prior density $g(\theta)$ for the nodes study.

```
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)
}

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 = "magenta") +
    ylim(0, 10) +
    geom_line(mapping = aes(x = theta, y = g2), col = "red") +
    geom_line(mapping = aes(x = theta, y = g3), col = "blue") +
    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")
```



Empirical Bayes posterior densities of θ for three patients, x is number of positive nodes, n is number of nodes.