Empirical_Bayes

Mi Zhang

5/09/2022

European automobile insurance

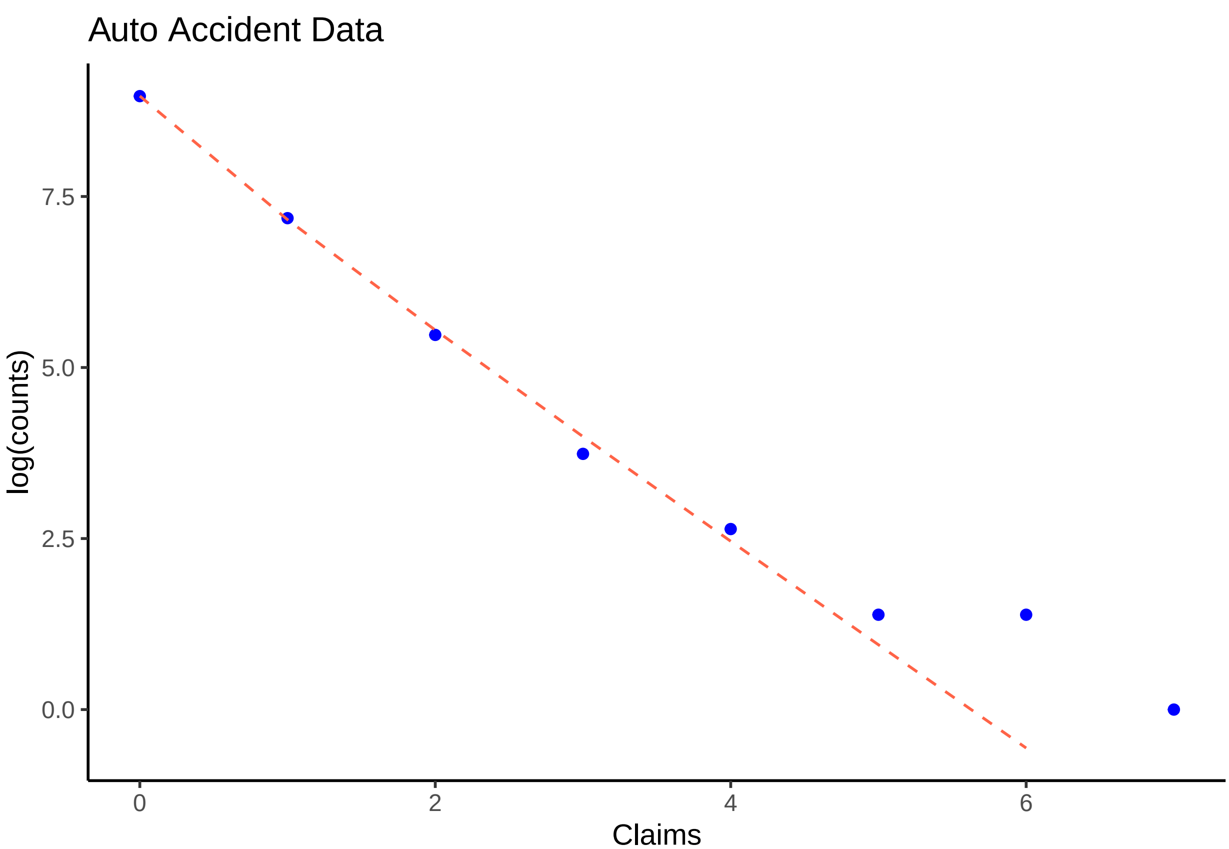
```
Claims <- seq(0, 7)
counts <- c(7840, 1317, 239, 42, 14, 4, 4, 1)
ins <- data.frame(Claims, counts)

#Robinson's formula
#empirical bayes
empirical <- NULL
for (i in 1:length(counts)){
  empirical[i] <- round(Claims[i+1]*(counts[i+1]/counts[i]),2)
}

empirical<- empirical[1:7]
#gamma-prior

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]
  eqt <- -sum(counts*log(f(Claims, nu=nu, sigma=sigma)))
  return(eqt)
}

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

gamma_mle <- round((seq(0,6)+1)*f(seq(0,6)+1,nu,sigma)/f(seq(0,6),nu,sigma), 2)

table1<- cbind(Claims, counts, empirical, gamma_mle) %>% as.data.frame()

table1$gamma_mle=round(c(f(seq(0,6),nu,sigma)*9461,NA),2)
```

```r
kable(rbind(Claims, counts, empirical, gamma_mle), "pipe", caption = "European Automobile Insurance")
```

Table 1: European Automobile Insurance

Claims	0.00	1.00	2.00	3.00	4.00	5.00	6.00	7.00
counts	7840.00	1317.00	239.00	42.00	14.00	4.00	4.00	1.00
empirical	0.17	0.36	0.53	1.33	1.43	6.00	1.75	0.17
gamma_mle	0.16	0.40	0.63	0.87	1.10	1.33	1.57	0.16

```r
# log(counts) vs claims for 9461 auto insurance policies. The dashed line is a gamma MLE fit.
ggplot(data=table1)+
  geom_point(aes(x=Claims, y=log(counts)), col="blue")+
  geom_line(aes(x=Claims, y= log(gamma_mle)), col="tomato", linetype = "dashed")+ theme_classic()+ ggti
```



The Missing-Species Problem

```r
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)
```

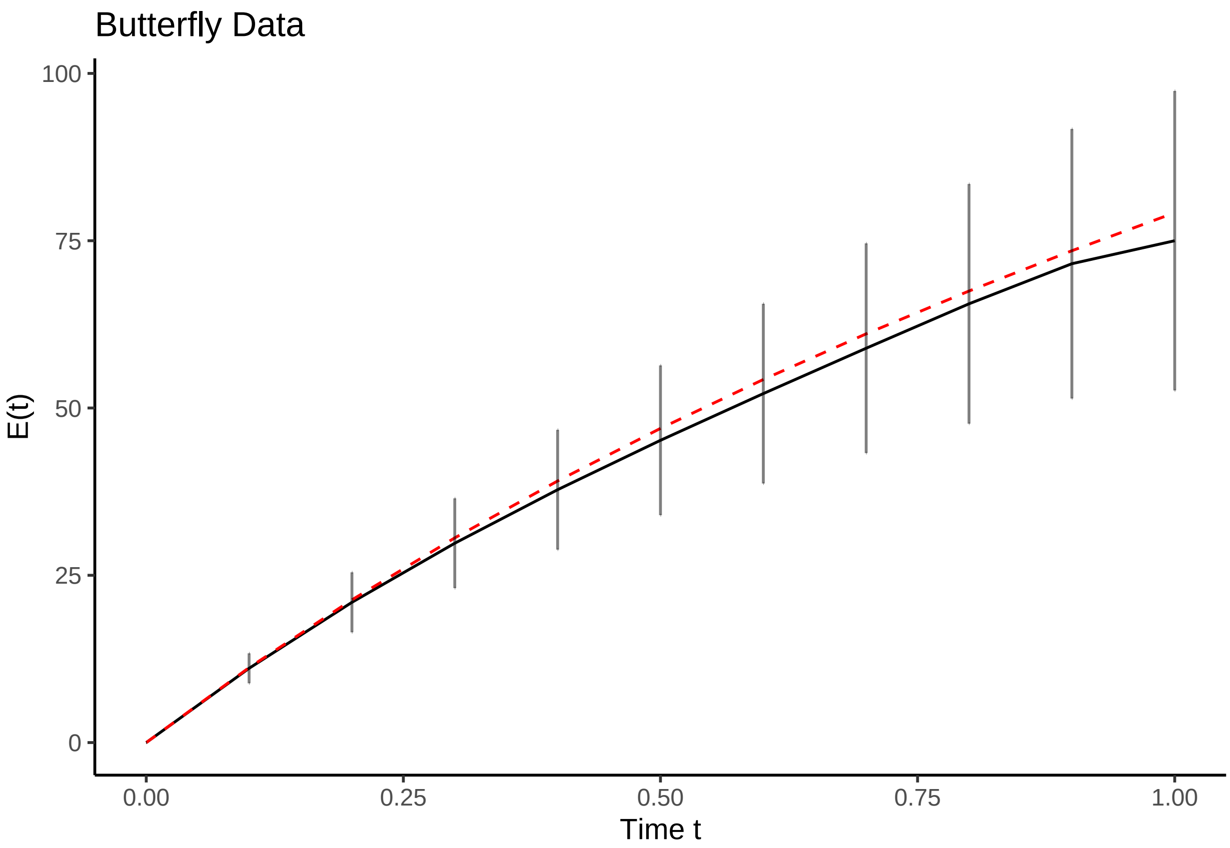
```r
t= seq(0, 1, 0.1)

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)
}

table2<- data.frame(t, exp, sd)
```

```r
# gamma estimate
# 0 =< t =<1
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)
}

kable(rbind(t, exp, sd, gamma_est), "pipe", caption ="The Missing-Species" )
```

Table 2: The Missing-Species

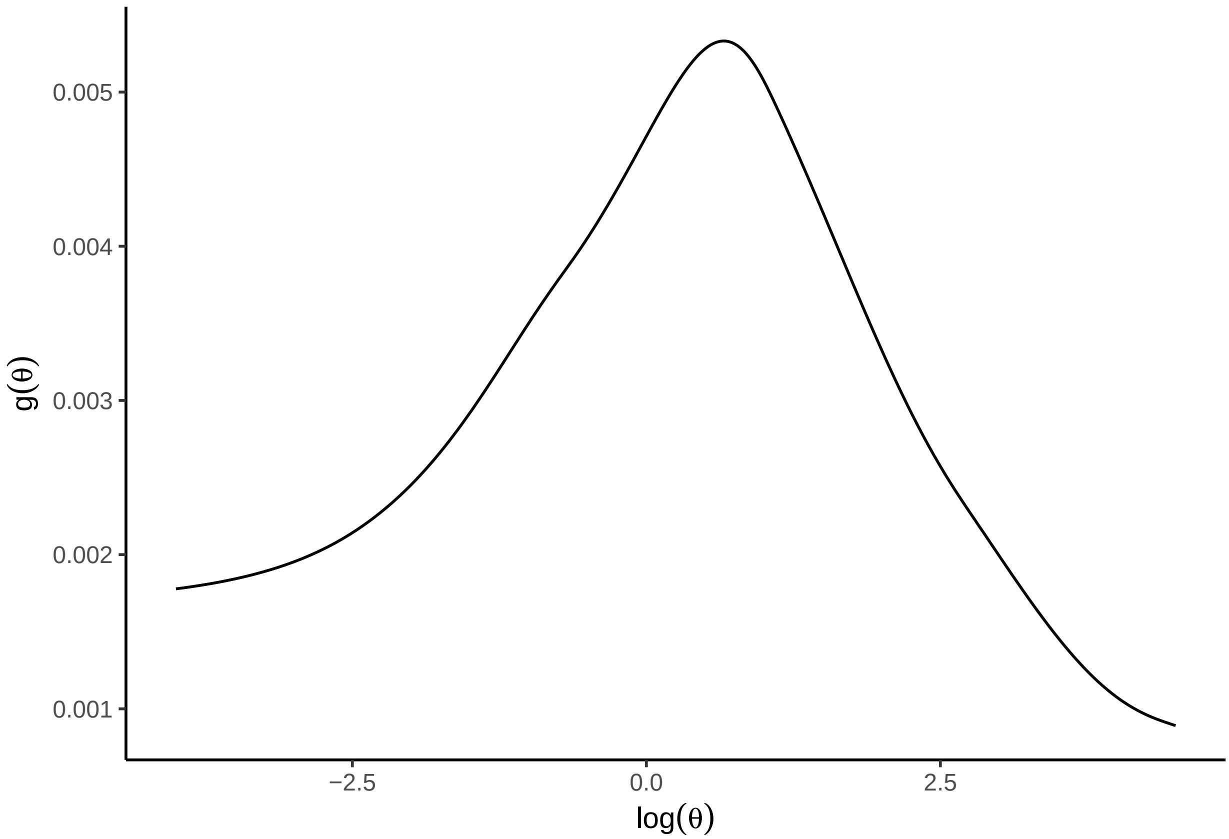
t	0	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90	1.00
exp	0	11.10	20.96	29.79	37.79	45.17	52.15	58.93	65.57	71.56	75.00
sd	0	2.24	4.48	6.71	8.95	11.19	13.43	15.67	17.91	20.14	22.38
gamma_est	0	11.20	21.33	30.59	39.09	46.95	54.26	61.08	67.48	73.50	79.18

```r
# Nonparametric fit (solid) +/- 1 standard deviation; gamma model (dashed).
ggplot(data=table2, 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_classic()+
  theme(legend.position="topleft")
```

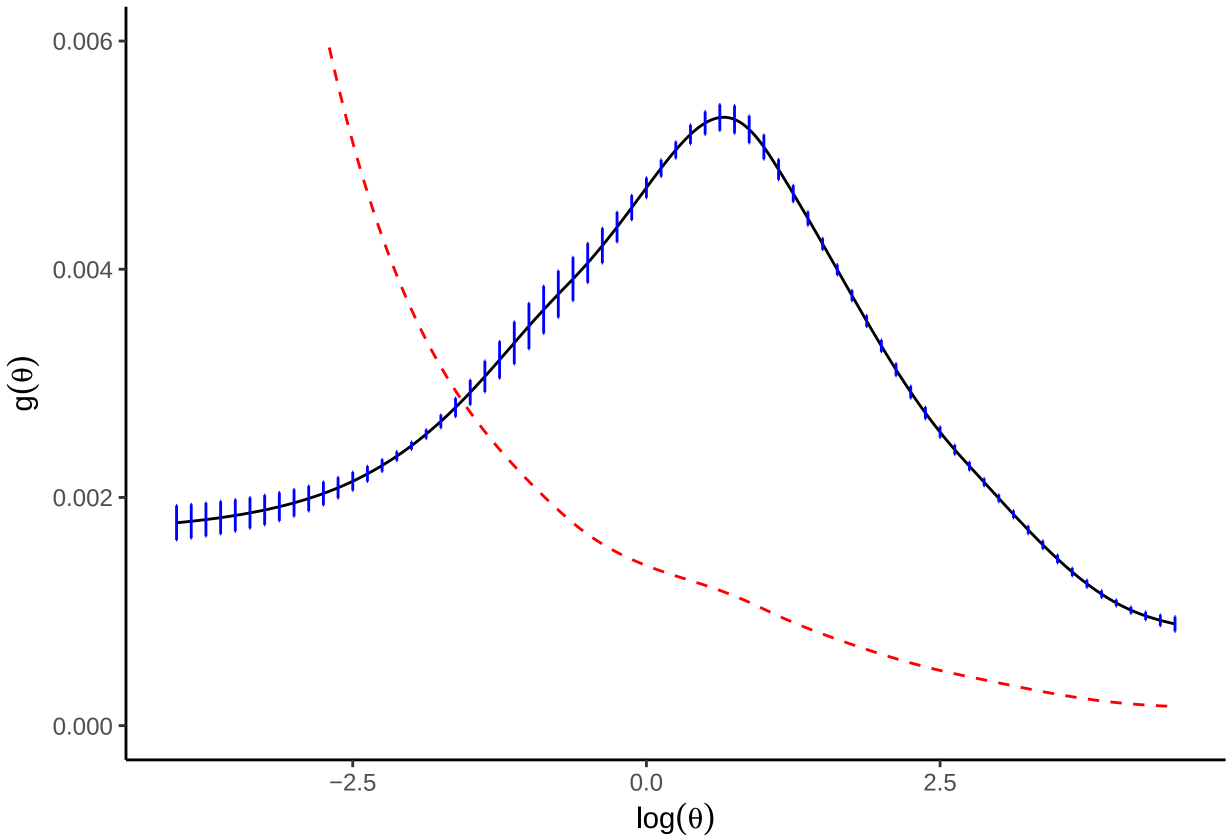


Shakespeare's word counts

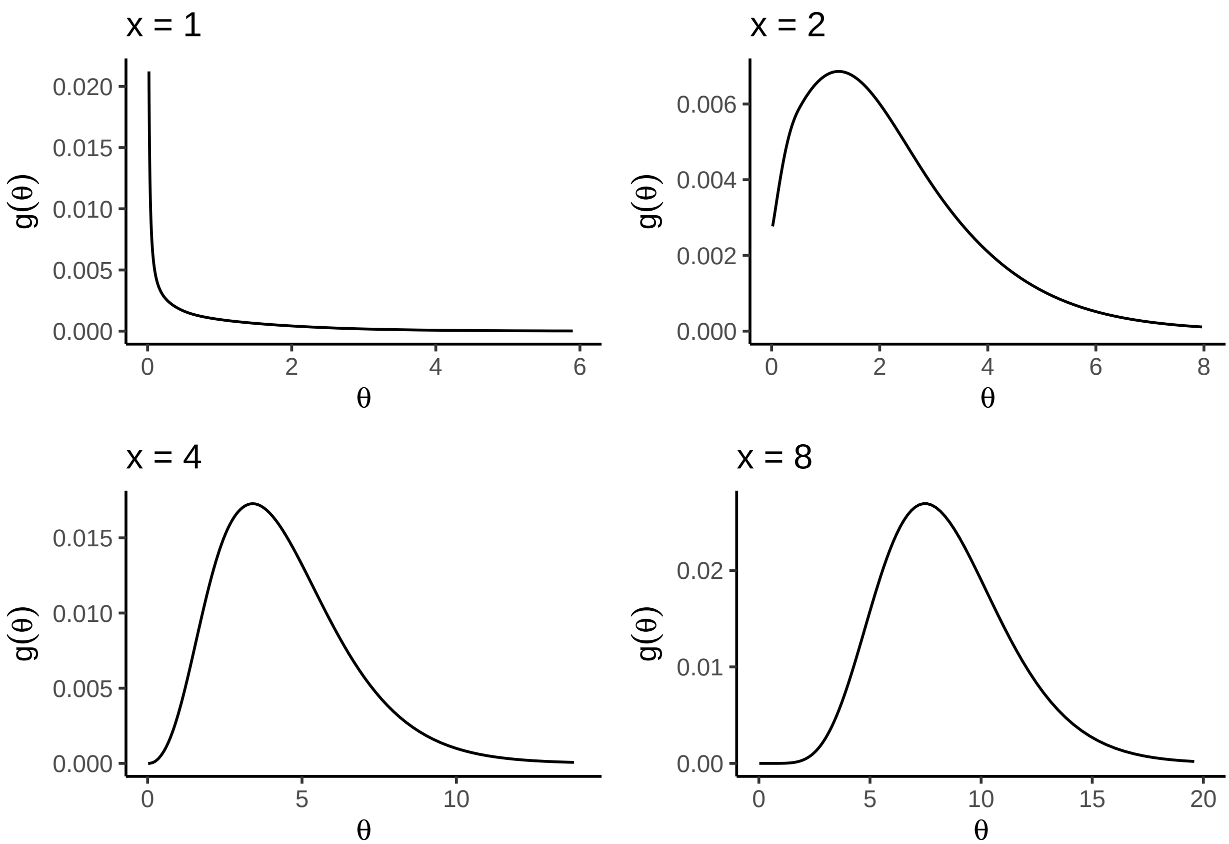
```
data(bardWordCount)
# str(bardWordCount)
lambda <- seq(-4, 4.5, .025)
tau <- exp(lambda)
result <- deconv(tau = tau, y = bardWordCount, n = 100, c0=2)
stats <- result$stats

# Empirical Bayes deconvoluation estimates
ggplot() +
    geom_line(mapping = aes(x = lambda, y = stats[, "g"])) +
    labs(x = expression(log(theta)), y = expression(g(theta)))+theme_classic()
```




```
d <- data.frame(lambda = lambda, g = stats[, "g"], tg = stats[, "tg"], SE.g = stats[, "SE.g"]) #tg=corr
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")+theme_classic()
```

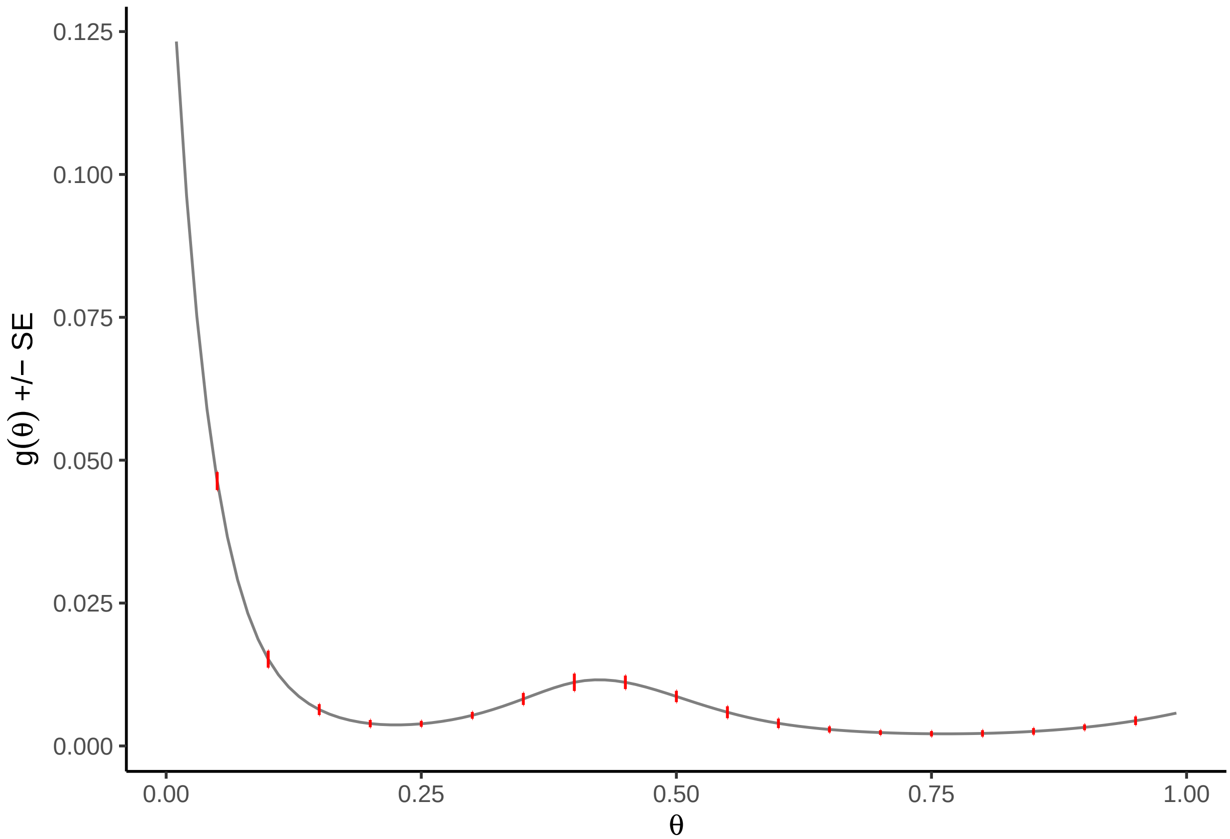



```
# Posterior estimates
gPost <- sapply(seq_len(100), function(i) local({tg <- d$tg * result$P[i, ]; tg / sum(tg)}))
plots <- lapply(c(1, 2, 4, 8), function(i) {
    ggplot() +
        geom_line(mapping = aes(x = tau, y = gPost[, i])) +
        labs(x = expression(theta), y = expression(g(theta)),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

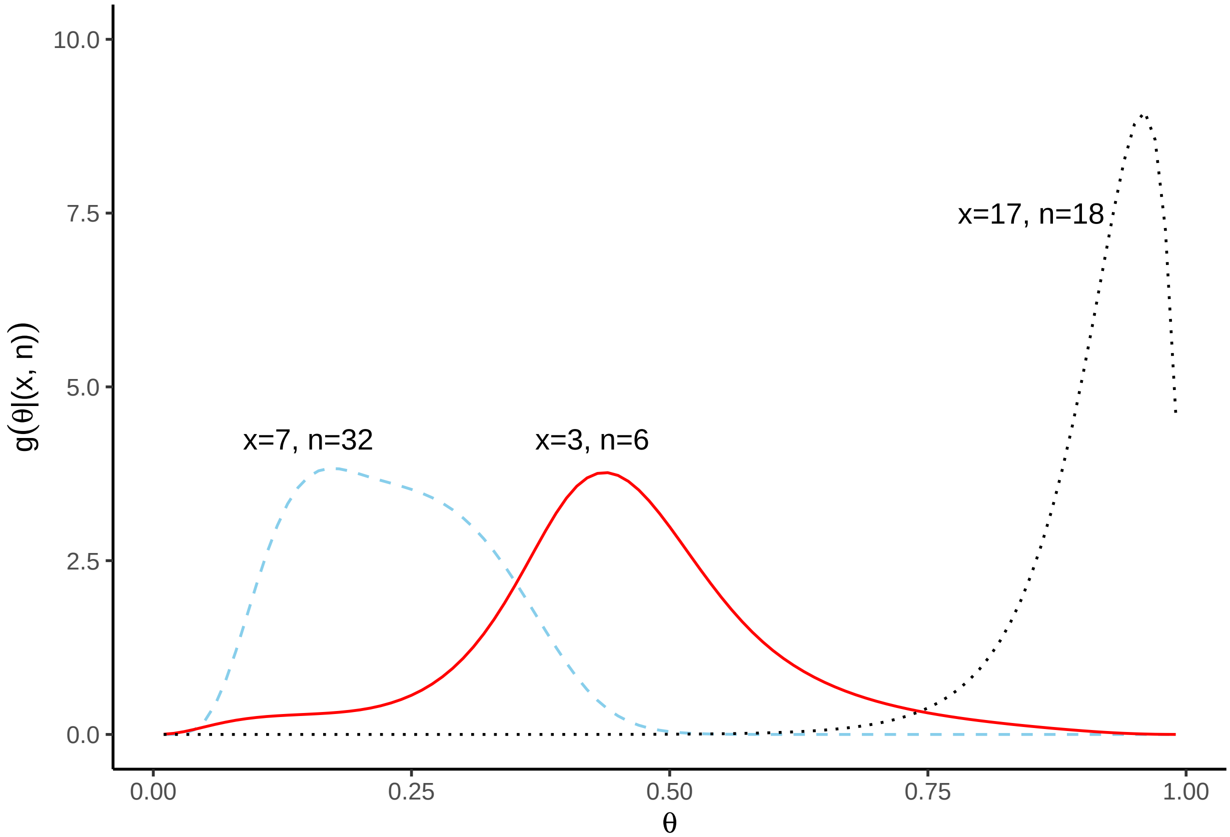
```
data(surg)
tau <- seq(from = 0.01, to = 0.99, by = 0.01)
result <- deconv(tau = tau, X = surg, family = "Binomial", c0 = 1)
d <- data.frame(result$stats)
indices <- seq(5, 99, 5)
errorX <- tau[indices]

# estimated prior density of $g(\theta)$
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")))+theme_classic()
```




```
# kable(d[indices, ], row.names = FALSE)
```

```
# 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")+
```

```
  theme_classic()
```



Main takeaway from this project

In this project, I learned how to apply empirical Bayes methods to real-world data. However, I struggled a lot because I don't have strong statistical background knowledge. For this semester, I feel like we were doing something different from last semester in which we intended to combine the theoretical part with the empirical part. Over the summer, I plan to continue reading those textbooks and hope I can gain more statistical knowledge and improve my coding skill.

Acknowledgement

I would like to thank my classmates, Zening Ye and Shuting Li, who explained the project to me. A special thanks to Yuli Jin who helped me with coding.

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