MA677 Final - Introduction to Empirical Bayes

Ranfei Xu

2022/5/11

Insurance Claims

- Create Auto accident data

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

- Based on Robbins' formula, calculate the expectation of the number of claims for a single customer

```
n <- 8
robbin1<-round(((auto$Claims+1)[1:7]*auto$Counts[2:8]/auto$Counts[1:7]),3)
```

- calculate the parametrically 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,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(robbin1,re)
```

- Create the plot that compare the raw counts y_x with their parametric cousins $\hat{y}_x$ of Auto accident data. The dashed line is a gamma MLE fit.

```
auto$pred=c(f(seq(0,6),mu,sigma)*9461,NA)
auto %>% ggplot() + geom_point(aes(x=Claims,y=log(Counts)),color='blue') +geom_line(aes(x=Claims,y=log(
```

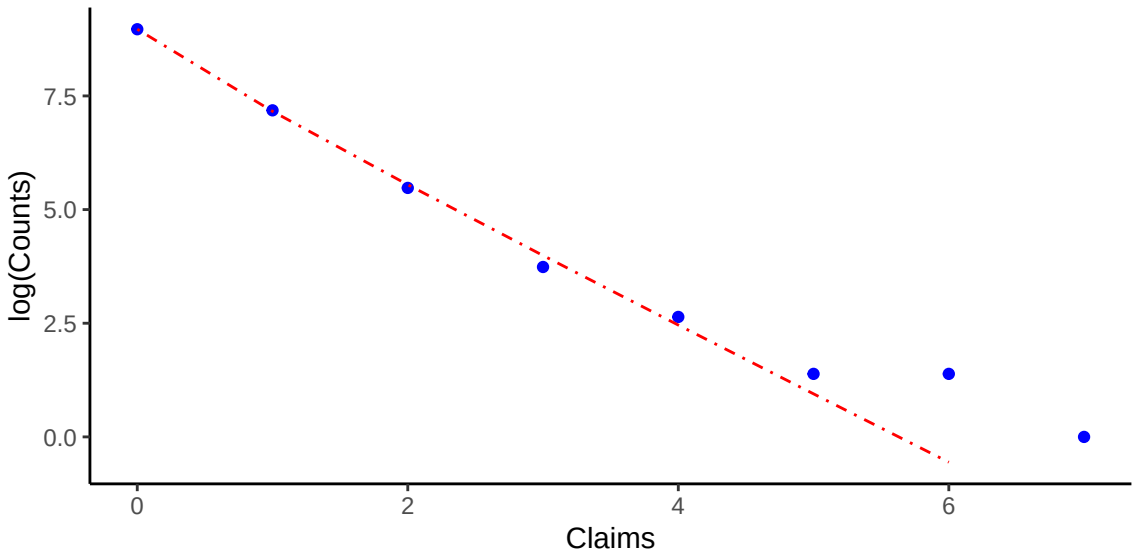


Figure 1: log(counts) vs claims for 9461 auto insurance policies

Species Discovery

- Create butterfly data

```
butterfly <- data.frame(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))
```

- estimate the expected number of new species seen in the new trapping period $E(t)$ with Robbins' formula

```
Fisher1<-function(t){
  re<-round(butterfly$y * t^(butterfly$x)* (-1)^(butterfly$x-1),2)
  sd<-round((sum(butterfly$y * (t)^(2)))^(1/2),2)
  return(list('est'=sum(re),'sd'=sd))
}

F1 <- sapply(seq(0,1,0.1),Fisher1)
F1
```

```
##     [,1] [,2] [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9]  [,10] [,11]
## est 0    11.1 20.96 29.79 37.81 45.2  52.14 58.94 65.55 71.57 75
## sd  0    2.24 4.48  6.71  8.95  11.19 13.43 15.67 17.91 20.14 22.38
```

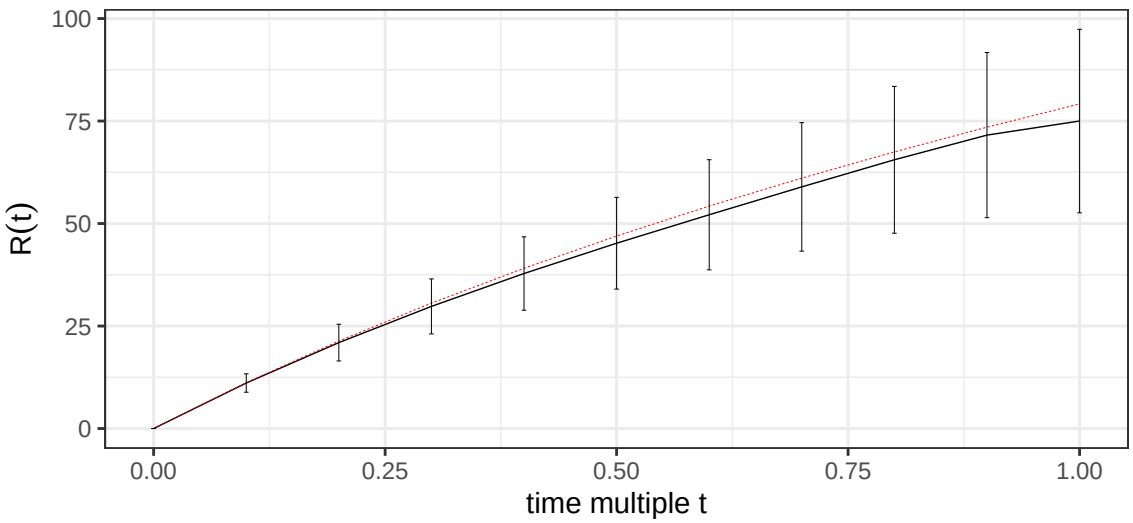
- calculate the parametric estimate of $E(t)$ using $\hat{e}_1, \hat{v}, \hat{\sigma}$

```
v <- 0.104
sigma <-  89.79
gamma <- sigma / (1 + sigma)
e1 <- 118
fisherFn <- function(t){
  re<-e1*((1 - (1+gamma*t)^(-v)) / (gamma * v))
  return(re)
}

EST2<-sapply(seq(0,1,0.1),fisherFn)
EST2
```

```
##  [1]  0.00000 11.19732 21.33347 30.58504 39.08842 46.95109 54.25922 61.08287
##  [9] 67.47981 73.49817 79.17850
```

- plot the expected number of new species in t units of additional trapping time, with nonparametric fit (solid) ±1 standard deviation; gamma model (dashed).

```
df<-data.frame(time=seq(0,1,0.1),est1=unlist(F1[1,]),sd=unlist(F1[2,]),est2=EST2)
df %>% ggplot() +
geom_line(mapping = aes(x = time, y = est1), size = 0.25) +
geom_line(mapping = aes(x = time, y = est2), color = "red", size = 0.1, linetype = "dashed") +
## geom_hline(yintercept = 0.0, color = "blue", linetype="dotted") +
## geom_vline(xintercept = 0.0, color = "blue", linetype="dotted") +
geom_errorbar(mapping = aes(x = time, ymin = (est1 - sd),
ymax = (est1 + sd)),
width=0.005, color="black", size = 0.001) +
labs(x = "time multiple t", y = expression(R(t)), caption = "Figure")+theme_bw()
```



Figure

Shakespeare's Vocabulary

- Refer to Haviland's code

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

d <- data.frame(lambda = lambda, g = stats[, "g"], tg = stats[, "tg"],
                SE.g = stats[, "SE.g"])
indices <- seq(1, length(lambda), 5)
print(
    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 = "green") +
    labs(x = expression(log(theta)), y = expression(g(theta))) +
    ##ylim(-0.001, 0.006) +
    xlim(-4, 4) +
    geom_vline(xintercept = 0.0, linetype = "dotted", color = "blue") +
    geom_hline(yintercept = 0.0, linetype = "dotted", color = "blue") +
    geom_line(mapping = aes(x = lambda, y = tg),
              linetype = "dashed", color = "red") +
    annotate("text", x = c(-4, -3, -2, -1, 0, 1, 2, 3, 4),
             y = rep(-0.0005, 9),
             label = c("0.02", "0.05", "0.14", "0.37", "1.00", "2.72", "7.39", "20.09", "90.02"), size =
    scale_y_continuous(breaks = c(-0.0005, 0.0, 0.002, 0.004, 0.006),
                       labels = c(expression(theta), "0.000", "0.002", "0.004", "0.006"),
                       limits = c(-0.0005, 0.006)) +
    labs(caption="Figure 1")
)
```

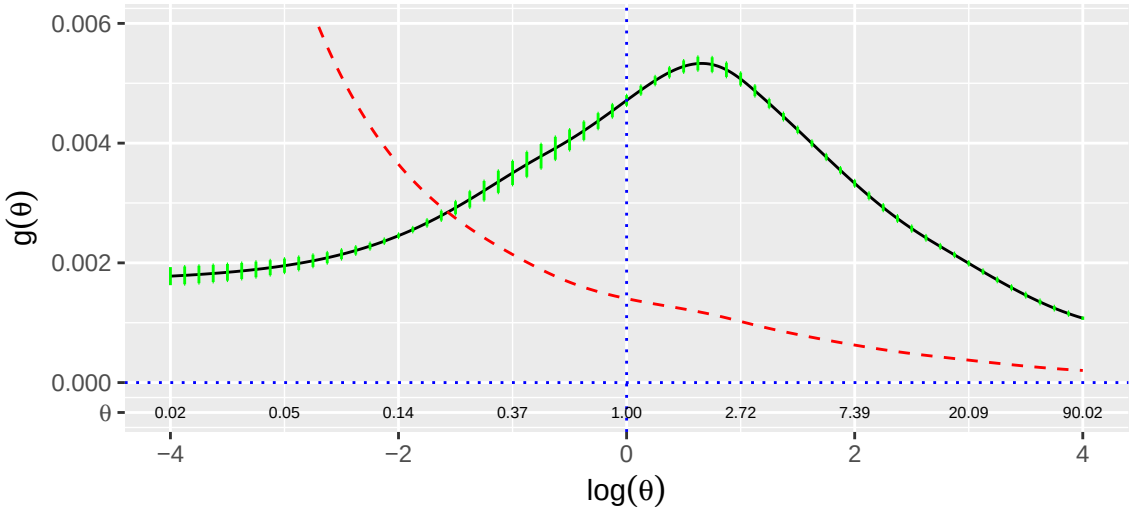
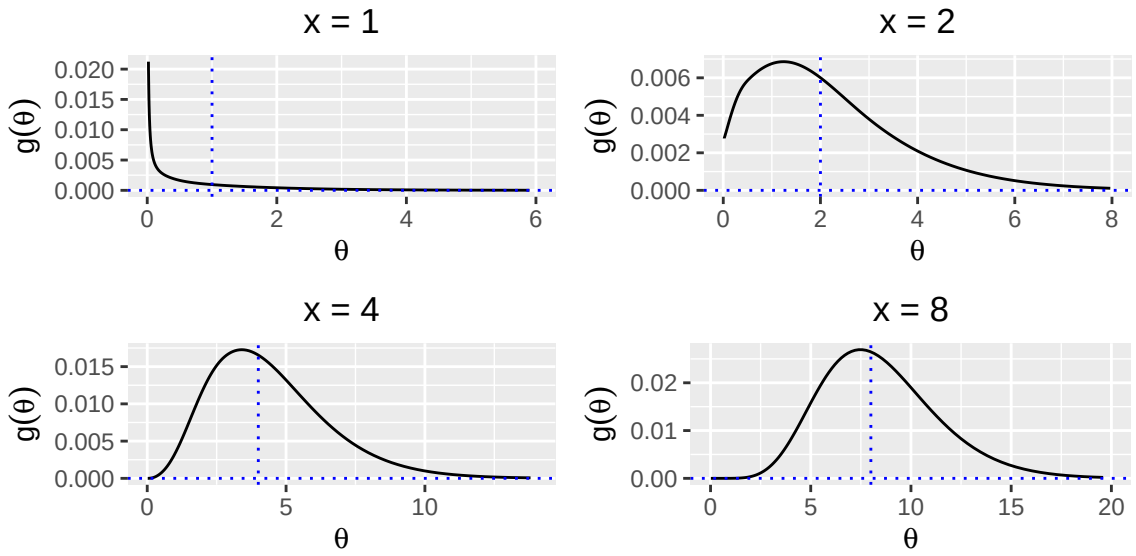


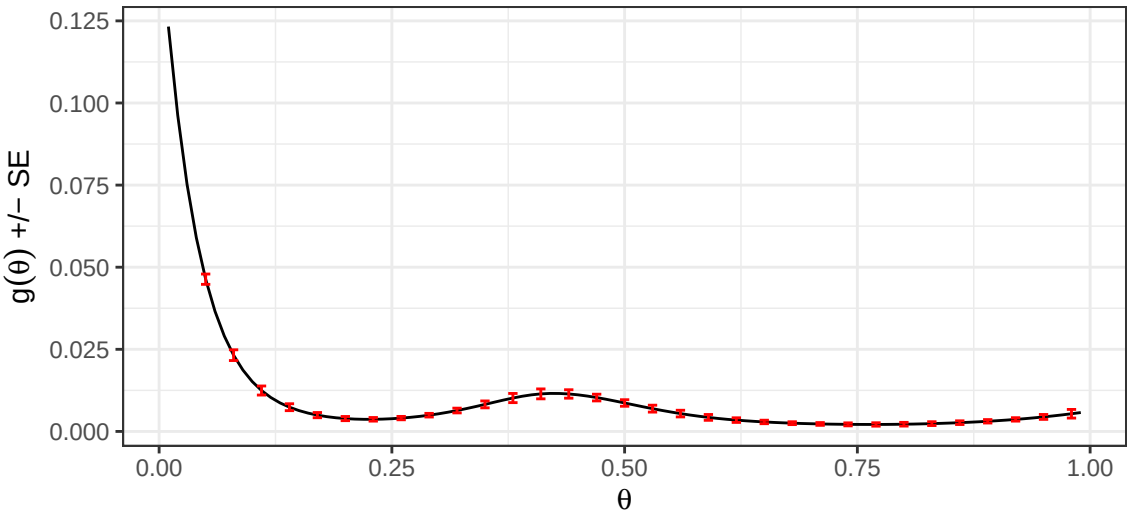
Figure 1

```
library("cowplot")
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])) +
        geom_vline(xintercept = i, linetype = "dotted", color = "blue") +
        geom_hline(yintercept = 0.0, linetype = "dotted", color = "blue") +
        labs(x = expression(theta), y = expression(g(theta)),
             title = sprintf("x = %d", i))
})
plots <- Map(f = function(p, xlim) p + xlim(0, xlim) + theme(plot.title=element_text(hjust=0.5)),
             plots, list(6, 8, 14, 20))

print(plot_grid(plotlist = plots, ncol = 2))
```



Lymph Node Counts

```
library(tidyverse)
data(surg)
p <- surg$x/surg$n
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

Reference

https://github.com/MA615-Yuli

https://github.com/jrfiedler/CASI_Python/blob/master/chapter06/ch06s01.ipynb

Haviland's class note: "File deconvolveR hw.R"

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