677

Ruining Jia

5/11/2022

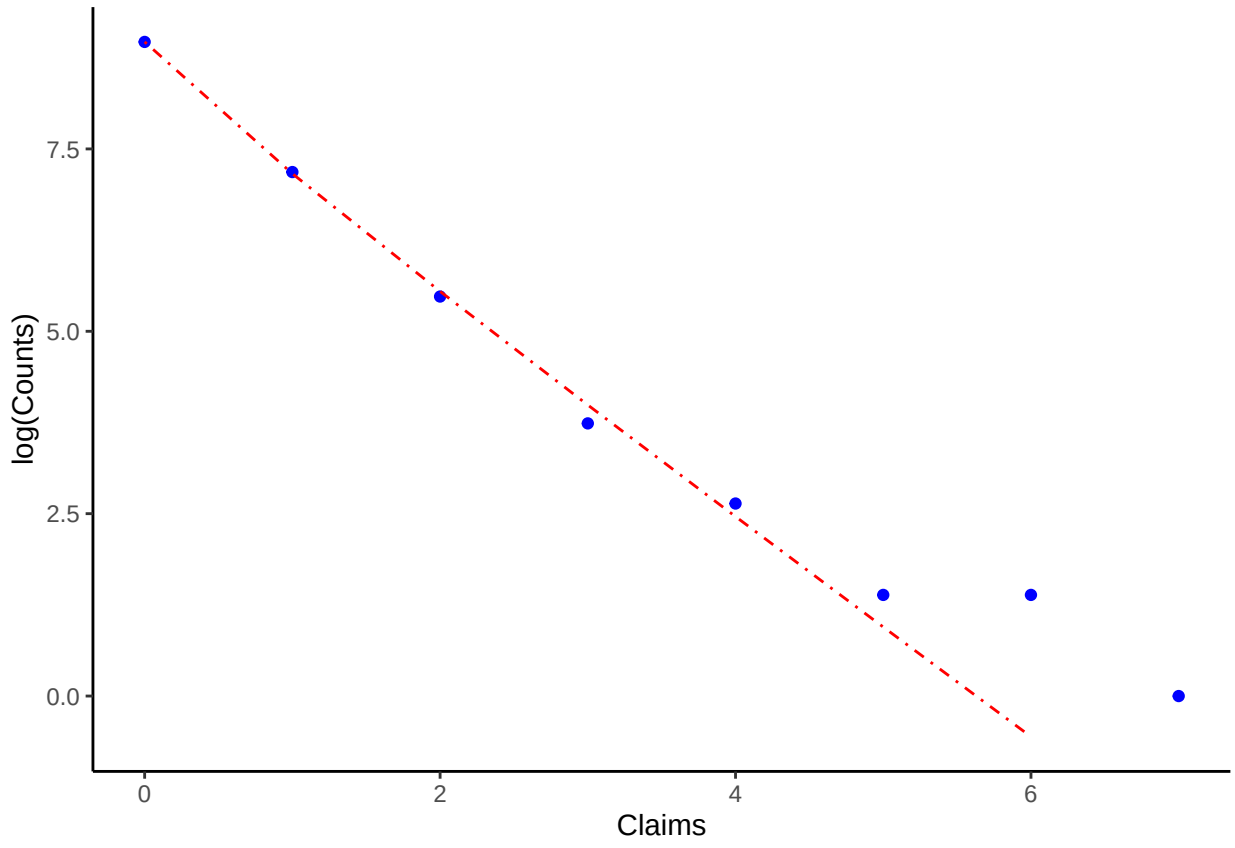
insurance claims

```
# counts = dict(enumerate([7840, 1317, 239, 42, 14, 4, 4, 1]))
# counts
# formula6p7 = [(i + 1) * counts[i + 1] / counts[i] for i in range(len(counts) - 1)]
# def f(x, nu, sigma):
#     gamma = sigma / (1 + sigma)
#     numer = gamma ** (nu + x) * special.gamma(nu + x)
#     denom = sigma ** nu * special.gamma(nu) * factorial(x)
#     return numer / denom
#
# def negloglikelihood(params):
#     nu, sigma = params
#     return -sum(counts[x] * log(f(x, nu, sigma)) for x in range(8))
# Create Data Frame
auto <- data.frame(Claims=seq(0,7),
           Counts=c(7840,1317,239,42,14,4,4,1))
# Gamma-prior version
n <- 8
robbin<-round(((auto$Claims+1)[1:7]*auto$Counts[2:8]/auto$Counts[1:7]),3)
f <- function(x,mu,sigma){
  gamma = sigma / (1 + sigma)
  numer = gamma ^ (mu + x) * gamma(mu + x)
  denom = sigma ^ mu * gamma(mu) * factorial(x)
  return(numer/denom)
}
neg_like <-function(param){
  mu=param[1]
  sigma=param[2]
  tmp=-sum(auto$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(robbin,re)
```

```
##         [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]
```

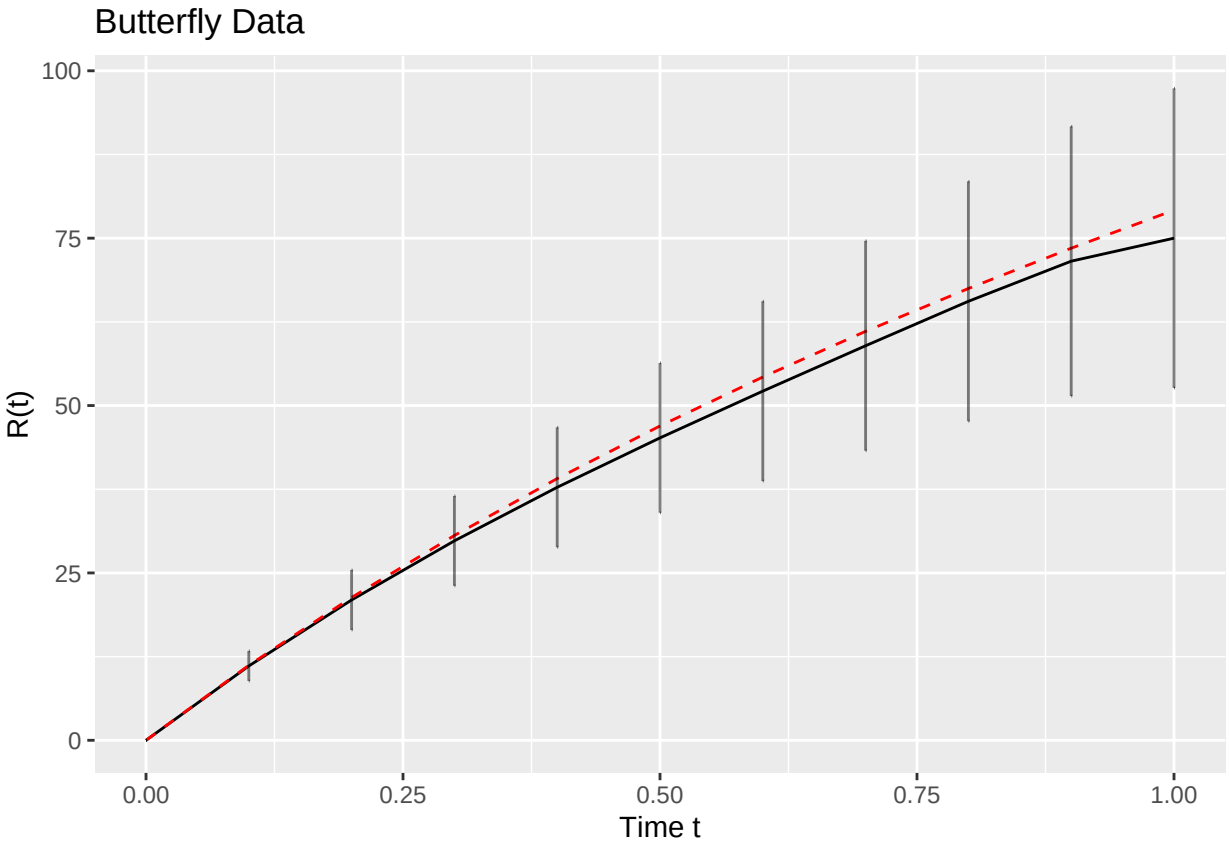
```
## robbin 0.168 0.363 0.527 1.333 1.429 6.000 1.750
## re     0.164 0.398 0.632 0.866 1.100 1.334 1.568
```

```
auto$pred=c(f(seq(0,6),mu,sigma)*9461,NA)
p1 <- ggplot(data=auto) +
  geom_point(aes(x=Claims,y=log(Counts)),color='blue')+
  geom_line(aes(x=Claims,y=log(pred)),color='red',lty=4)+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(), panel.background = eleme
p1
```



species discovery

```
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 <- 118
gamma_est <- NULL
for (i in 1:length(t)){
  gamma_est[i] <- round(E_1*((1 - (1+gamma*t[i])^(-v)) / (gamma * v)),2)
}
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("R(t)")+xlab("Time t")
```


Butterfly Data
100
75
50
25
0
R(t)
0.00
0.25
0.50
0.75
1.00
Time t

Shakespeare's word counts

```
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")
)
```

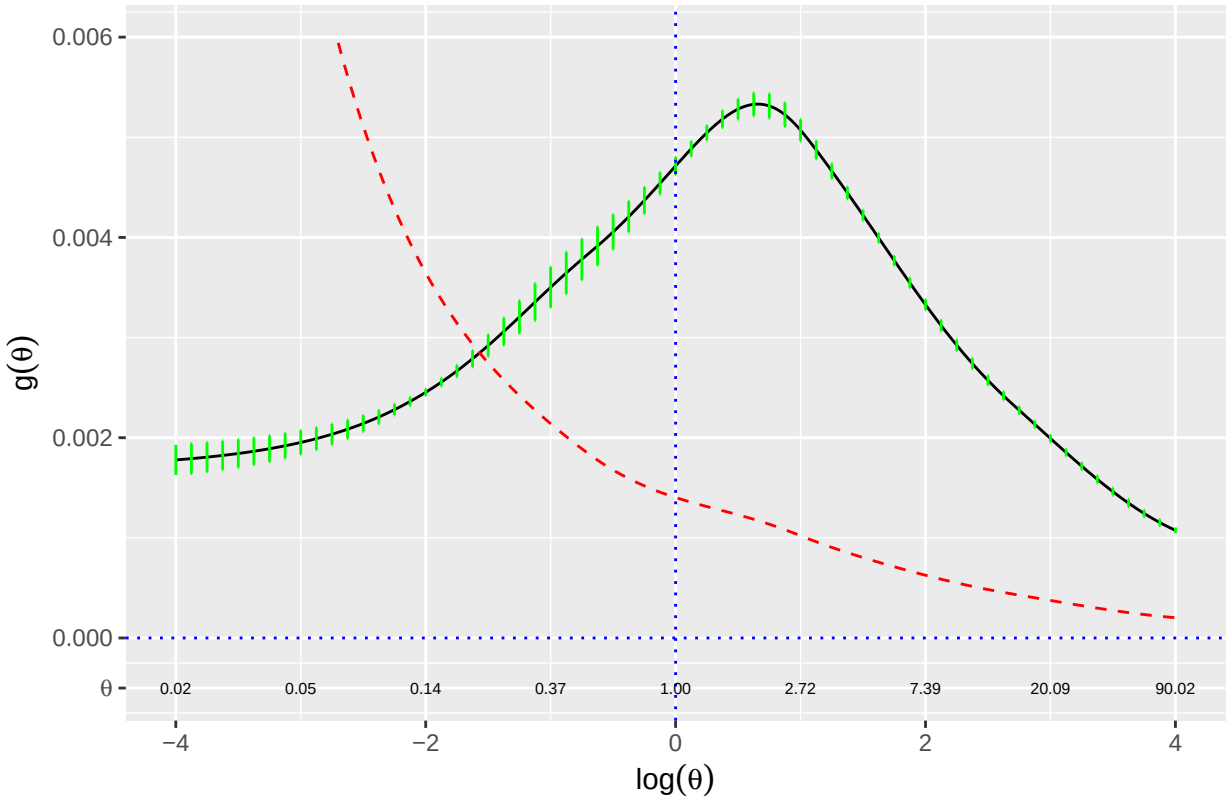
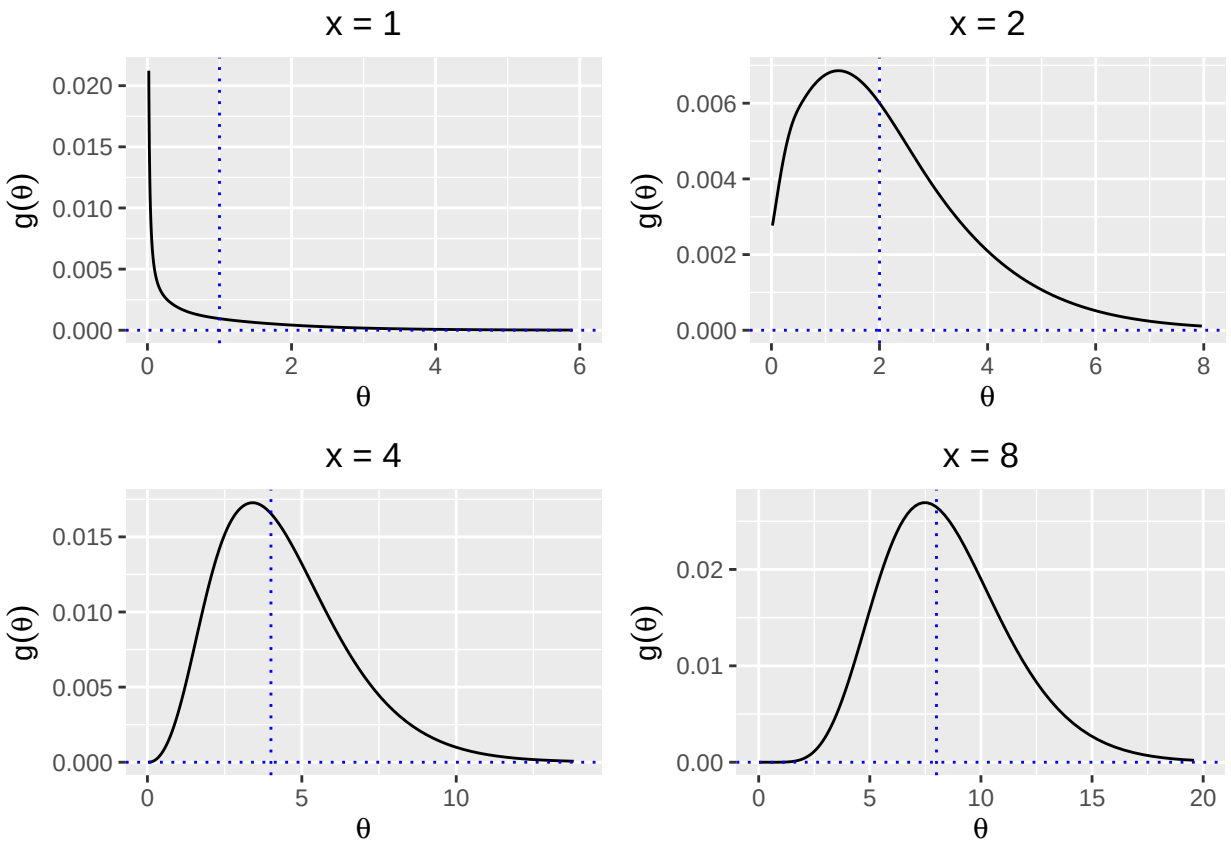


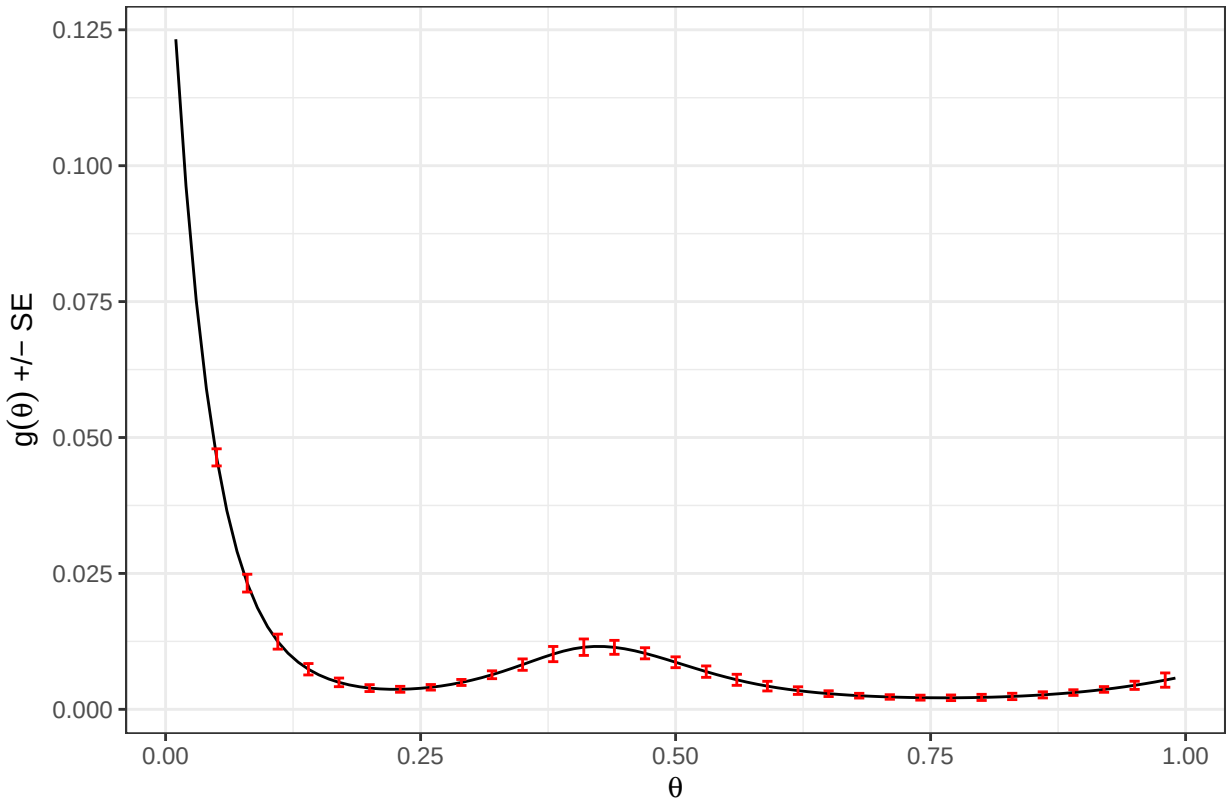
Figure 1

```
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))
```


x = 1
x = 2
x = 4
x = 8
g(θ)
θ
0.020
0.015
0.010
0.005
0.000
0.006
0.004
0.002
0.02
0.01
0.00
0
2
4
5
6
8
10
15
20

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/jrfiedler/CASI_Python/blob/master/chapter06/ch06s01.ipynb http://www.columbia.edu/~mh2078/MachineLearningORFE/MCMC_MasterSlides.pdf class note: "File deconvolveR hw.R" https://github.com/bnaras/deconvolveR/blob/master/vignettes/deconvolution.Rmd https://stackoverflow.com/questions/59435824/nlm-with-multiple-variables-in-r