Advanced Statistical Methods Hw7

Do Hyup Shin

2021-11-08

Problem 9.3

Redraw Figure 9.2, changing the "knot" location from 11 to 12.

Solution

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
NCOG_data <- read.csv("https://web.stanford.edu/~hastie/CASI_files/DATA/ncog.txt", sep = " ")


Arm_A <- subset(NCOG_data,subset = arm == "A")
Arm_B <- subset(NCOG_data,subset = arm == "B")


## harzard function of Arm_A
num_A = nrow(Arm_A)

Km_A <- matrix(0, 47, 4)

copy_A = Arm_A

for(i in 1:47){
  Km_A[i,1] = nrow(copy_A)
  inter_A = copy_A %>% filter(t > 30.4*(i-1) & t <= 30.4*i)
  Km_A[i, 2] <- length(which(inter_A$d == 1))
  Km_A[i, 3] <- length(which(inter_A$d == 0))
  Km_A[i, 4] <- Km_A[i, 2] / Km_A[i,1]
  copy_A = copy_A %>% filter(t > 30.4*i)
}

x_a <- matrix(0, 47, 4)
```

```r
for(i in 1:47){
  if(i <= 11){
    x_a[i,] <- c(1, i, (i-12)^2, (i-12)^3)
  }
  else{
    x_a[i,1:2] <- c(1, i)
  }
}

n_a = Km_A[,1]
y_a = Km_A[,2]


## Iteratively Rewighted least square(IRLS)
# initial value of alpha
alpha_0 = c(-1, -0.01, 0.1, 0.01)

alpha = matrix(0, 1001, 4)
alpha[1,] = alpha_0


for(i in 1:1000){
  lambda = x_a %*% alpha[i,]
  # mu_k = n * exp(lambda_k)/ (1 + exp(lambda_k))
  mu = n_a * exp(lambda) / (1 + exp(lambda))

  # z_k = lambda_k + n / (mu_k * (n - mu_k)) * (y - mu_k)
  z = lambda  + (n_a / (mu * (n_a - mu)) ) * (y_a - mu)


  # D_k = diag(mu_k(n - mu_k) / n)
  D = diag(as.numeric(mu*(n_a - mu)/n_a))


  # V_k = diag(mu_k (1 - mu_k/n))
  V = diag(as.numeric(mu* (1 - mu/n_a)))
  # W_k = D %*% solve(V) %*% D

  W = D

  # alpha_(k+1) = (X^t W_k X)^(-1) X^t W_k Z_k
  alpha[i+1, ] = solve((t(x_a) %*% W %*% x_a)) %*% t(x_a) %*% W %*% z
}
alpha[1:5,]
```

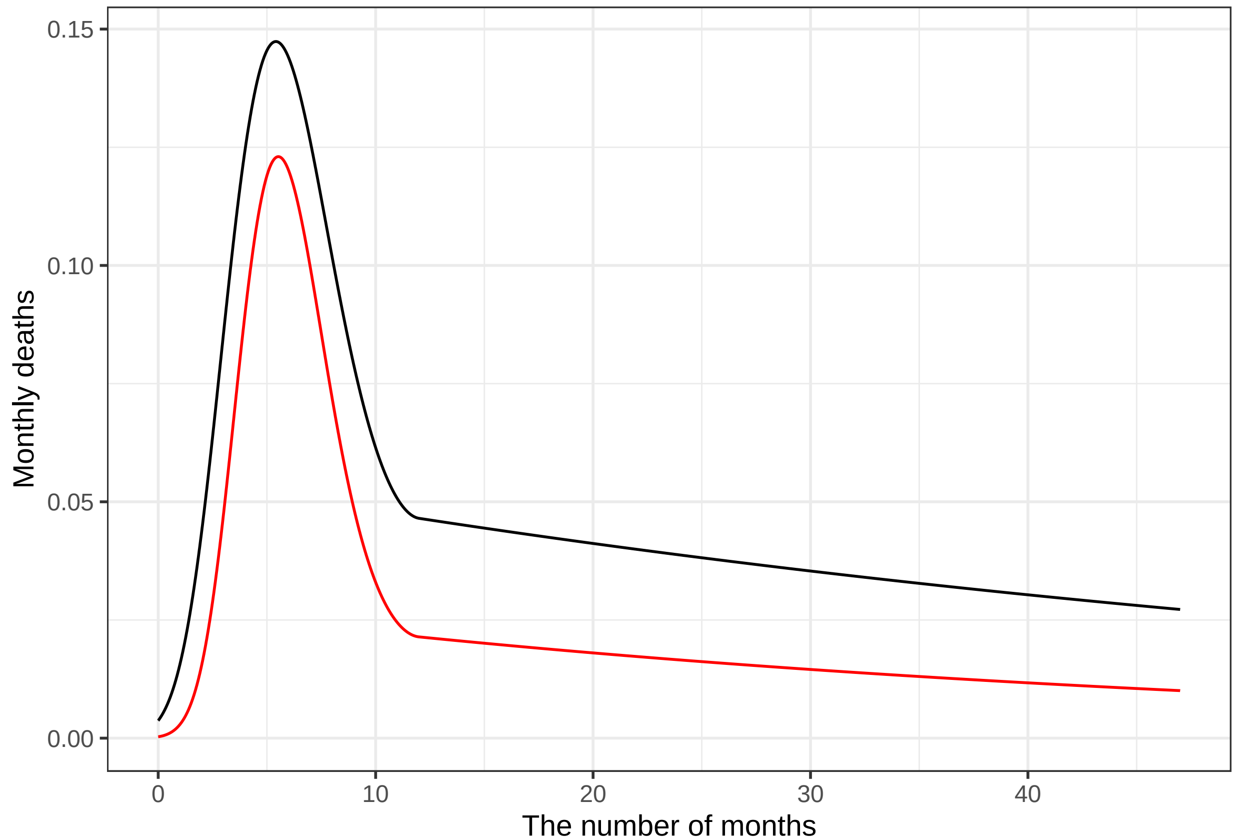
```
##           [,1]         [,2]       [,3]        [,4]
## [1,] -1.000000 -0.010000000 0.10000000 0.010000000
## [2,] -2.072016 -0.009647184 0.05328930 0.005486873
## [3,] -2.639540 -0.012215924 0.06904242 0.007067727
## [4,] -2.816163 -0.015069066 0.08079383 0.008283587
## [5,] -2.830310 -0.015844475 0.08264287 0.008485633
```

```r
alpha_hat = alpha[1001,]
```

```r
harzard_func <- function(x){
  h = 1/(1 + exp(-x%*%alpha_hat))
  return(h)
}


spline_func <- function(x){
  if(x<= 12){
    return(c(1, x, (x-12)^2,(x-12)^3))
  }
  else{
    return(c(1, x, 0, 0))
  }
}


range_xa = seq(0, 47, 0.01)


h_a = rep(0, 4701)

for(i in 1:4701){
  v = spline_func(range_xa[i])
  h_a[i] = harzard_func(v)
}

harzard_ratio_a = as.data.frame(cbind(range_xa, h_a))

#ggplot(data=harzard_ratio_a, aes(x=range_xa, y=h_a)) + geom_line(color='black', lwd=0.5)



## harzard function of Arm_B
num_B = nrow(Arm_B)

Km_B <- matrix(0, 76, 4)

copy_B = Arm_B


for(i in 1:76){
  Km_B[i,1] = nrow(copy_B)
  inter_B = copy_B %>% filter(t > 30.4*(i-1) & t <= 30.4*i)
  Km_B[i, 2] <- length(which(inter_B$d == 1))
  Km_B[i, 3] <- length(which(inter_B$d == 0))
  Km_B[i, 4] <- Km_B[i, 2] / Km_B[i,1]
  copy_B = copy_B %>% filter(t > 30.4*i)
}



x_b <- matrix(0, 76, 4)
for(i in 1:76){
```

```r
  if(i <= 11){
    x_b[i,] <- c(1, i, (i-12)^2, (i-12)^3)
  }
  else{
    x_b[i,1:2] <- c(1, i)
  }
}

n_b = Km_B[,1]
y_b = Km_B[,2]


## Iteratively Rewighted least square(IRLS)

# initial value of alpha
alpha_0 = c(1, 0.01, 0.01, 0.01)
alpha = matrix(0, 1001, 4)
alpha[1,] = alpha_0


for(i in 1:1000){
  lambda = x_b %*% alpha[i,]

  # mu_k = n * exp(lambda_k)/ (1 + exp(lambda_k))
  mu = n_b * exp(lambda) / (1 + exp(lambda))

  # z_k = lambda_k + n / (mu_k * (n - mu_k)) * (y - mu_k)
  z = lambda  + (n_b/ (mu * (n_b - mu)) ) * (y_b - mu)


  # D_k = diag(mu_k(n - mu_k) / n)
  D = diag(as.numeric(mu*(n_b-mu)/n_b))


  # V_k = diag(mu_k (1 - mu_k/n))
  V = diag(as.numeric(mu* (1 - mu/n_b)))

  # W_k = D %*% solve(V) %*% D

  W = D

  # alpha_(k+1) = (X^t W_k X)^(-1) X^t W_k Z_k
  alpha[i+1, ] = solve((t(x_b) %*% W %*% x_b)) %*% t(x_b) %*% W %*% z
}


alpha_hat_b = alpha[1001,]

harzard_func <- function(x){
  h = 1/(1 + exp(-x%*%alpha_hat_b))
  return(h)
}
```

```r
spline_func <- function(x){
  if(x<= 12){
    return(c(1, x, (x-12)^2,(x-12)^3))
  }
  else{
    return(c(1, x, 0, 0))
  }
}


range_xb = seq(0, 47, 0.01)


h_b = rep(0, 4701)

for(i in 1:4701){
  v = spline_func(range_xb[i])
  h_b[i] = harzard_func(v)
}



harzard_ratio_b = as.data.frame(cbind(range_xb, h_b))

#ggplot(data=harzard_ratio_b, aes(x=range_xb, y=h_b)) + geom_line(color='red', lwd=0.5)

ggplot() + geom_line(data = harzard_ratio_a, aes(x=range_xa, y=h_a), color = "black", lwd = 0.5) + labs
  theme(legend.position = c(0.9,0.7)) +
  geom_line(data = harzard_ratio_b, aes(x = range_xb, y = h_b), color = 'red', lwd = 0.5) +
  xlab("The number of months") + ylab("Monthly deaths") +
  theme_bw()
```



Problem 9.5

Why does the hypergeometric distribution enter into formula (9.24)?

Solution