R Lab. - Exercise 6

Michele Guadagnini - Mt. 1230663

May 27, 2020

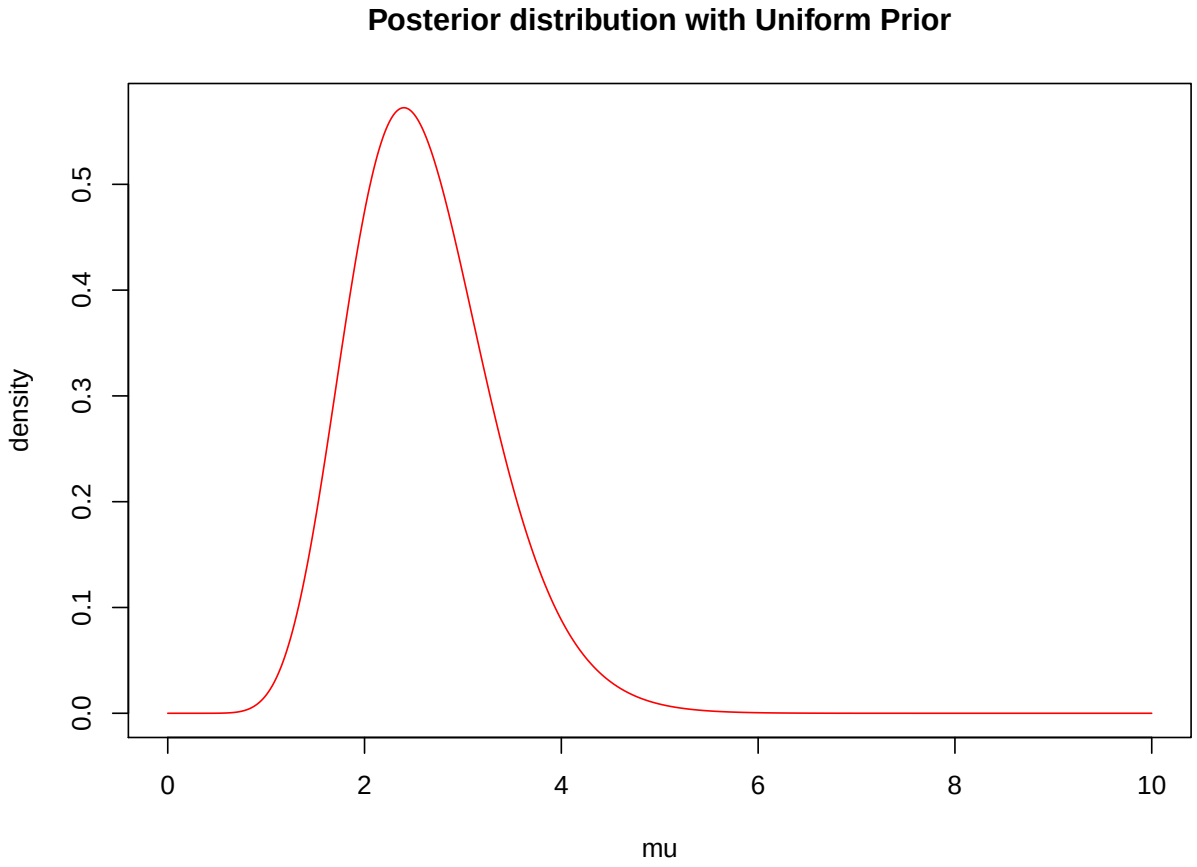
Exercise 1 - Radioactive source

The number of particles emitted by a radioactive source during a fixed interval of time ($\Delta t = 10$ s) follows a Poisson distribution on the parameter μ. The number of particles observed during consecutive time intervals is: 4, 1, 3, 1 and 3.

A) suppose a uniform prior distribution for the parameter μ:

- determine and draw the posterior distribution for μ, given the data
- evaluate mean, median and variance, both analytically and numerically in R

```
dt <- 10 #seconds
outcomes <- c(4, 1, 3, 1, 3)

## A ##
mu <- seq(0, 10, len=1000)
prior <- 1
likelihood <- function(lam) { # likelihood as product of 5 independent measures
    k <- dpois(outcomes[1], lam)
    for (i in 2:length(outcomes)) {
        k <- k*dpois(outcomes[i], lam)
    }
    return (k)
}
post.unif <- prior*likelihood(mu)
post.unif <- post.unif/sum(post.unif)

# numeric results
mean.num <- sum(mu*post.unif)
median.num <- mu[min( which(cumsum(post.unif) >= 0.5) )]
var.num  <- sum(mu*mu*post.unif) - mean.num**2
# analytic results
# posterior can be calculated also as: dgamma(mu, shape=13, rate=5)
shape <- sum(outcomes)+1
rate <- length(outcomes)
mean.th <- shape/rate
var.th  <- shape/(rate*rate)

# plotting the posterior distribution
plot(mu, post.unif/0.01, type='l', col='red', ylab="density",
     main="Posterior distribution with Uniform Prior")
```



The numerical results obtained in R are:

- mean: 2.6
- variance: 0.52
- median: 2.533

The analytical results obtained considering the posterior as a gamma distribution with proper parameters shape and rate are:

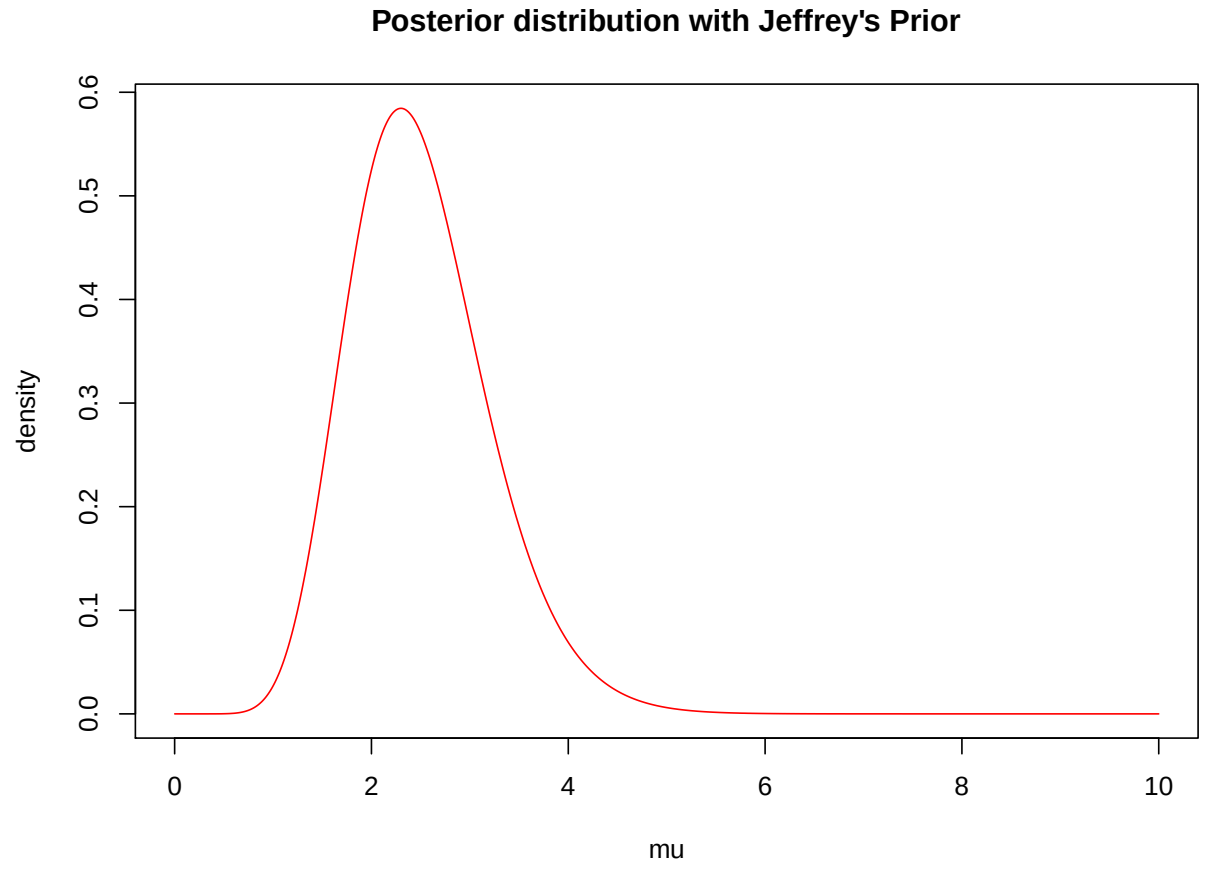
- mean: 2.6
- variance: 0.52

B) suppose a Jeffrey's prior for the parameter μ:

- determine and draw the posterior distribution for μ, given the data
- evaluate mean, median and variance, both analytically and numerically in R

```
## B ##
mu <- seq(0.001, 10, len=1000)
prior <- function(x) { 1/sqrt(x) }
likelihood <- function(lam) { # likelihood as product of 5 independent measures
    k <- dpois(outcomes[1], lam)
    for (i in 2:length(outcomes)) {
        k <- k*dpois(outcomes[i], lam)
    }
    return (k)
}
```

```r
post.jeff <- prior(mu)*likelihood(mu)
post.jeff <- post.jeff/sum(post.jeff)

# numeric results
mean.num <- sum(mu*post.jeff)
median.num <- mu[min( which(cumsum(post.jeff) >= 0.5) )]
var.num  <- sum(mu*mu*post.jeff) - mean.num**2
# analytic results
# posterior can be calculated also as: dgamma(mu, shape=12.5, rate=5)
shape <- sum(outcomes)+1/2
rate <- length(outcomes)
mean.th <- shape/rate
var.th  <- shape/(rate*rate)

#plotting the posterior distribution
plot(mu, post.jeff/0.01, type='l', col='red', ylab="density",
     main="Posterior distribution with Jeffrey's Prior")
```



The numerical results obtained in R are:

- mean: 2.5
- variance: 0.5
- median: 2.433

The analytical results obtained considering the posterior as a gamma distribution with proper parameters shape and rate are:

- mean: 2.5
- variance: 0.5

C) evaluate a 95% credibility interval for the results obtained with both priors. Compare the result with that obtained using a normal approximation for the posterior distribution, with the same mean and standard deviation

```
## C ##
low.unif <- mu[max(which(cumsum(post.unif)<=0.025))]
upp.unif <- mu[min(which(cumsum(post.unif)>=0.975))]

low.jeff <- mu[max(which(cumsum(post.jeff)<=0.025))]
upp.jeff <- mu[min(which(cumsum(post.jeff)>=0.975))]

normapprox <- pnorm(mu, mean.th, sqrt(var.th))
low.norm <- mu[max(which(normapprox<=0.025))]
upp.norm <- mu[min(which(normapprox>=0.975))]
```

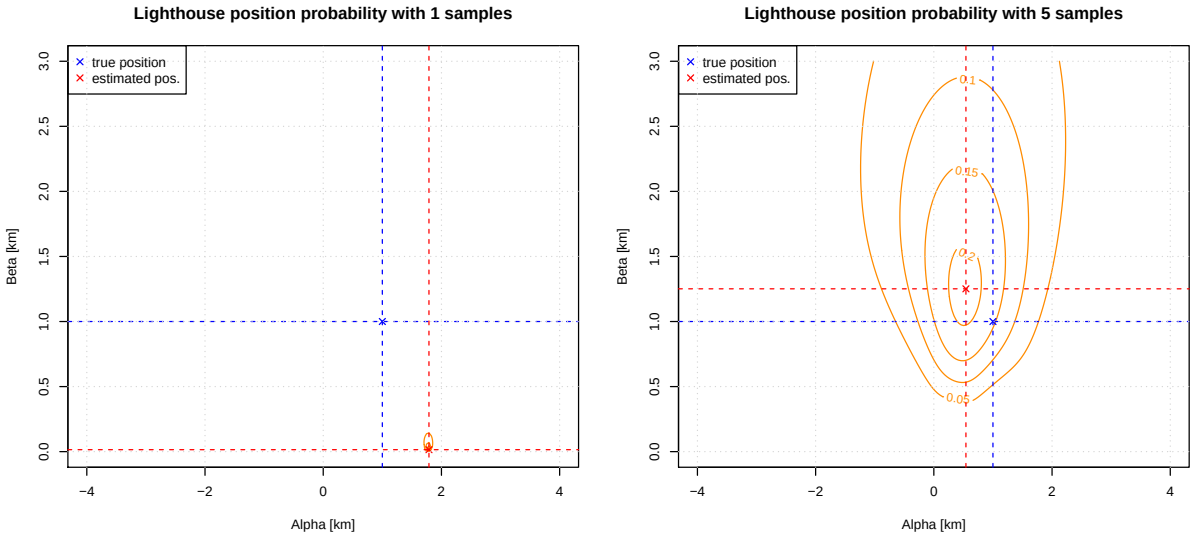
The obtained 95% credibility intervals are:

- with uniform prior: [1.372, 4.195]
- with Jeffrey's prior: [1.302, 4.065]
- for the normal approximation: [1.112, 3.895]

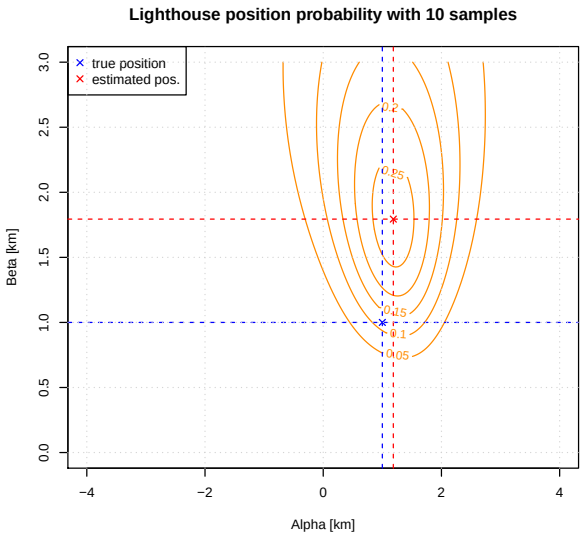
Exercise 2 - Lighthouse problem

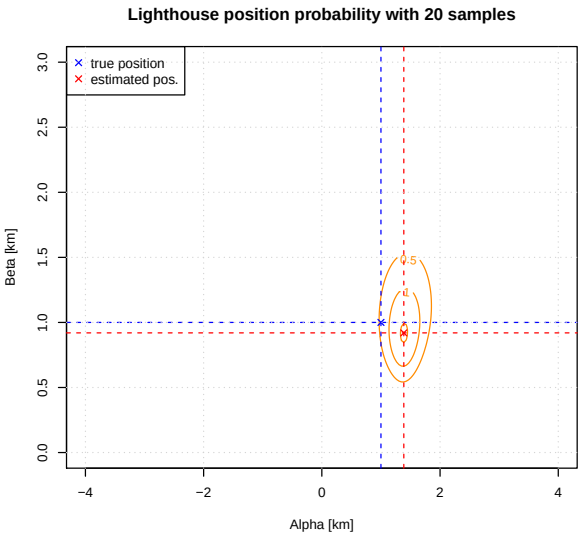
Given the problem of the lighthouse discussed last week, study the case in which both the position along the shore (α) and the distance out at sea (β) are unknown.

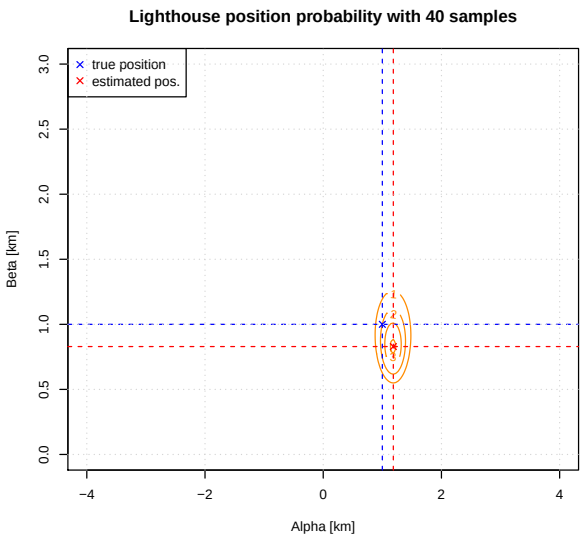
```
# Generation of the data (flashes)
rx <- function(n, a, b) {
    set.seed(13)
    x <- a + b*tan(runif(n,-pi/2,pi/2))
    return(x)
}

nsamples <- 100
alpha.true <- 1 #km
beta.true  <- 1 #km
xflashes <- rx(nsamples, alpha.true, beta.true)

# log of the posterior distribution
log.post <- function(flashes,a,b){
  logL <- 0.0
  for(x in flashes){
      logL <- logL + log(b/(pi*((x-a)**2 + b**2)))
  }
  return(logL)
}

amin <- -4; amax <- 4; bmax <- 3
np <- 200
alphas <- seq(amin,amax, len=np)
betas  <- seq(   0,bmax, len=np)

ndata <- c(1,5,10,20,40,100)

par(mfrow=c(3,2))
for (i in ndata) {
    ff <- xflashes[1:i]
    func <- function(a,b) { log.post(ff, a, b) }

    log_values <- outer(alphas, betas, Vectorize(func))
    values <- exp(log_values)/( ((amax-amin)/np)*(bmax/np)*sum(exp(log_values)) )
    grid_values <- matrix(values, nrow=length(alphas), ncol=length(betas))

    contour(alphas, betas, grid_values, xlab="Alpha [km]", ylab="Beta [km]",
            main=paste("Lighthouse position probability with",i,"samples"),
            col='darkorange', nlevels=5 )
    points(alpha.true, beta.true, pch=4, col="blue")
    max_ids <- c( which(grid_values==max(grid_values), arr.ind=TRUE) )
    points(alphas[max_ids[1]], betas[max_ids[2]], pch=4, col="red")
    legend("topleft", c("true position", "estimated pos."), col=c("blue","red"), pch=4)
    abline(v=alpha.true, h=beta.true, col="blue", lty=2)
    abline(v=alphas[max_ids[1]], h=betas[max_ids[2]], col="red", lty=2)
    grid()
}
```

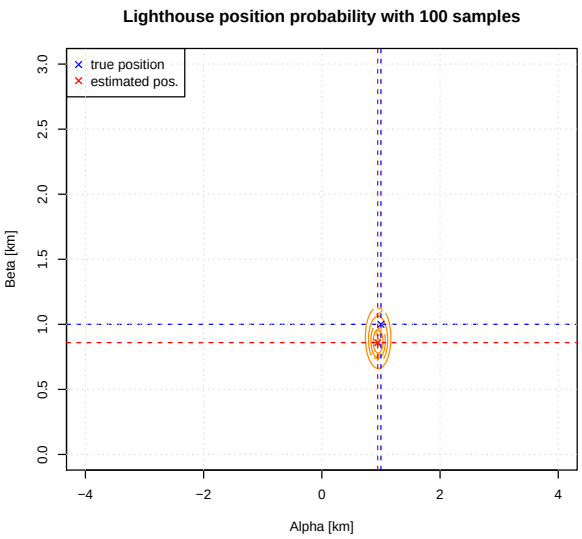

Lighthouse position probability with 1 samples
true position
estimated pos.
Beta [km]
Alpha [km]

Lighthouse position probability with 5 samples
true position
estimated pos.
0.1
0.15
0.2
0.05
Beta [km]
Alpha [km]


Lighthouse position probability with 10 samples
true position
estimated pos.
0.2
0.25
0.15
0.1
0.05
Beta [km]
Alpha [km]


Lighthouse position probability with 20 samples
true position
estimated pos.
0.5
1
Beta [km]
Alpha [km]


Lighthouse position probability with 40 samples
true position
estimated pos.
1
2
3
Beta [km]
Alpha [km]


Lighthouse position probability with 100 samples
true position
estimated pos.
1
2
3
Beta [km]
Alpha [km]

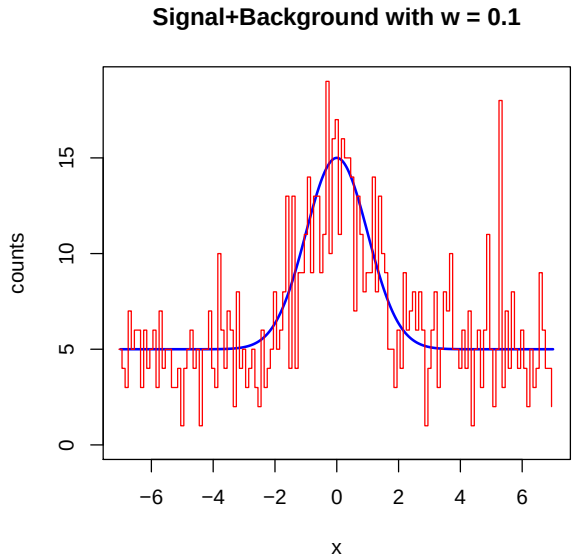
Exercise 3 - Signal over Background

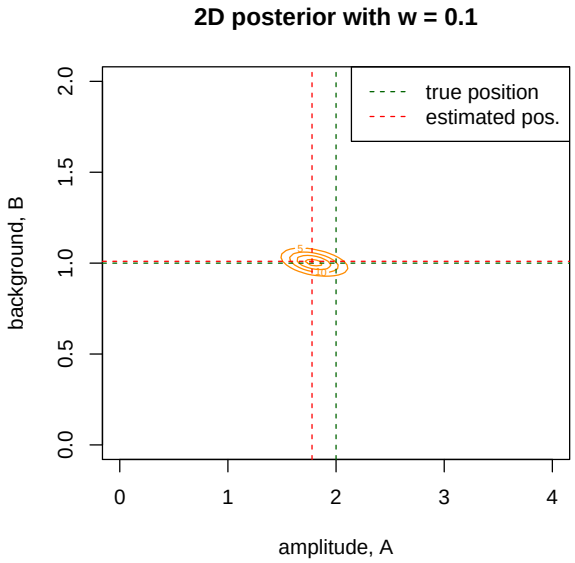
Given the Signal over Background discussed last week, analyze and discuss the following cases:

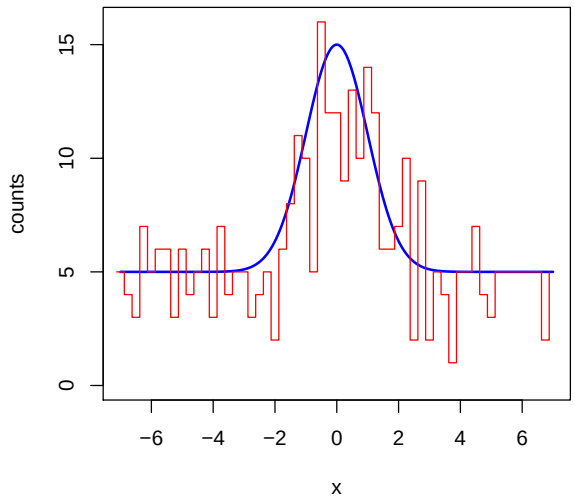
A) vary the sampling resolution used to generate the data, keeping the same sampling range

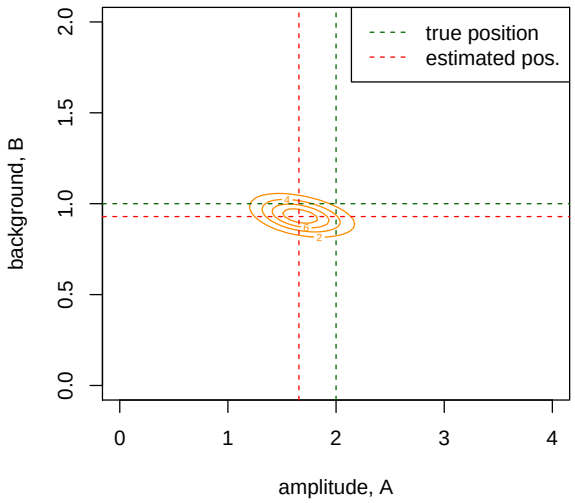
```
# signal function
signal <- function (x, a, b, x0, w, t) {
     t * (a*exp (-(x-x0)**2/(2*w**2)) + b)
}
# Log posterior
log.post <- function (d, x, a, b, x0, w, t) {
            if(a<0 || b <0) { return (-Inf )} # the effect of the prior
            sum( dpois(d, lambda = signal (x, a, b, x0, w, t), log=TRUE ))
}
# Parameters
x0 <- 0
dt <- 5 # exposure time in seconds
a.true <- 2
b.true <- 1
width <- 1

## A ##
ws <- c(0.1, 0.25, 1, 2, 3)

par(mfrow=c(1,2))
for (w in ws) {
    set.seed(241)
    xdat <- seq(from=-7*width, to=7*width, by=w*width)
    s.true <- signal(xdat, a.true, b.true, x0, width, dt)
    s.counts <- rpois(length(s.true), s.true)

    xps <- seq(min(xdat), max(xdat), len=1000)
    sps <- signal(xps, a.true, b.true, x0, width, dt)
    ymax <- max( c(max(sps), max(s.counts)) )

    plot(xps, sps, type='l', col='blue', ylim=c(0,ymax), xlab="x", ylab="counts",
         main=paste("Signal+Background with w =",w), xlim=c(-7,7), lwd=2)
    lines(xdat-w/2, s.counts, type='s', col='red' )

    a <- seq(0,4,len=100)
    b <- seq(0,2,len=100)
    delta_a <- 4/100
    delta_b <- 2/100
    # Computing log unnormalized posterior on a regular grid
    z <- matrix(data=NA , nrow= length(a), ncol= length(b))
    for (j in 1: length(a)) {
        for (k in 1: length(b)) {
            z[j,k] <- log.post(s.counts, xdat, a[j], b[k], x0, width, dt)
        }
    }
    # Normalizing posterior
    z.norm <- exp(z)/(delta_a*delta_b*sum(exp(z)))
```

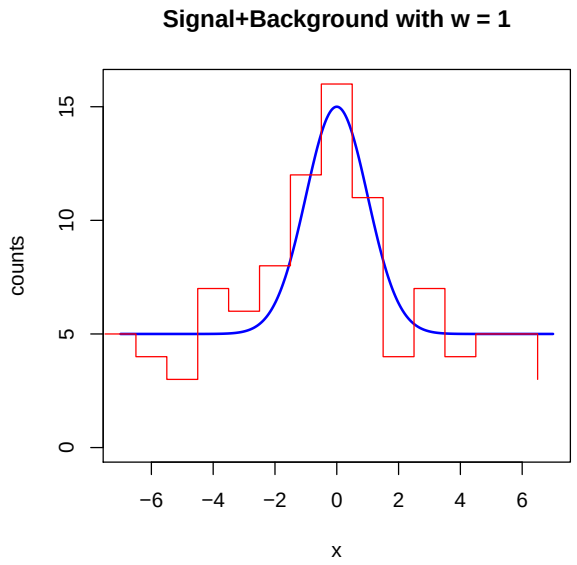
```
    # Plot normalized 2D posterior as contours
    contour(a, b, z.norm, nlevels = 4, labcex = 0.5,, xlab="amplitude, A",
            ylab="background, B", col="darkorange", main=paste("2D posterior with w =",w))
    abline(v=a.true, h=b.true, col="darkgreen", lty=2)
    max_ids <- c( which(z.norm==max(z.norm), arr.ind=TRUE) )
    abline(v=a[max_ids[1]], h=b[max_ids[2]], col="red", lty=2)
    legend("topright", c("true position", "estimated pos."),
           col=c("darkgreen","red"), lty=2)
}
```

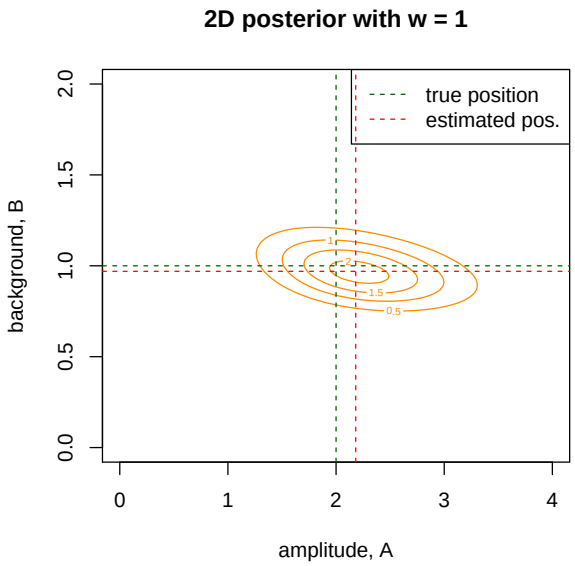


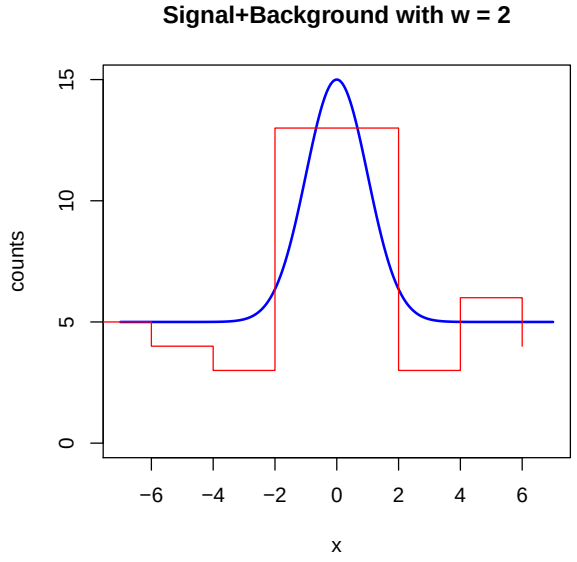


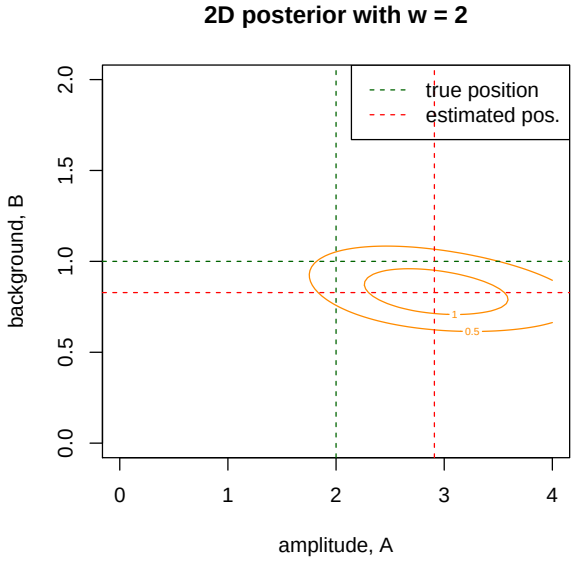


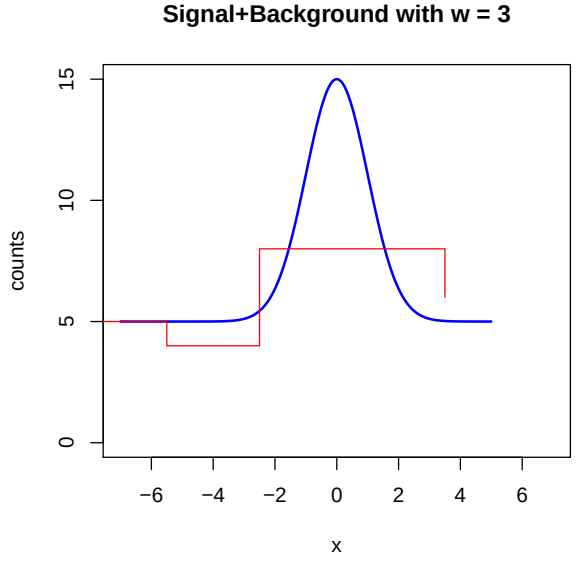


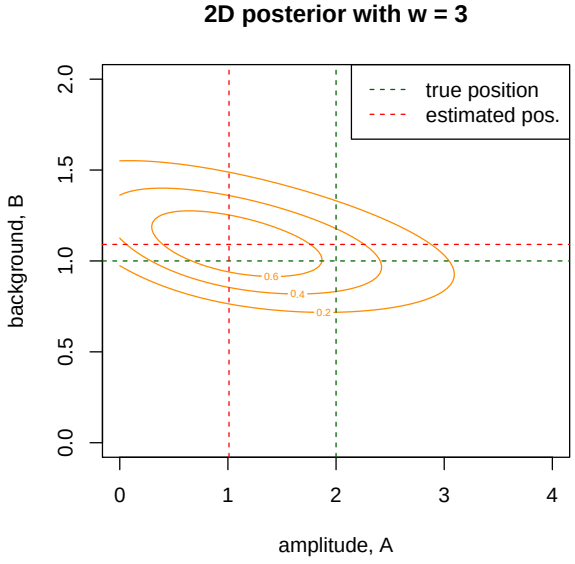

Signal+Background with w = 1
counts
x


2D posterior with w = 1
true position
estimated pos.
background, B
amplitude, A


Signal+Background with w = 2
counts
x

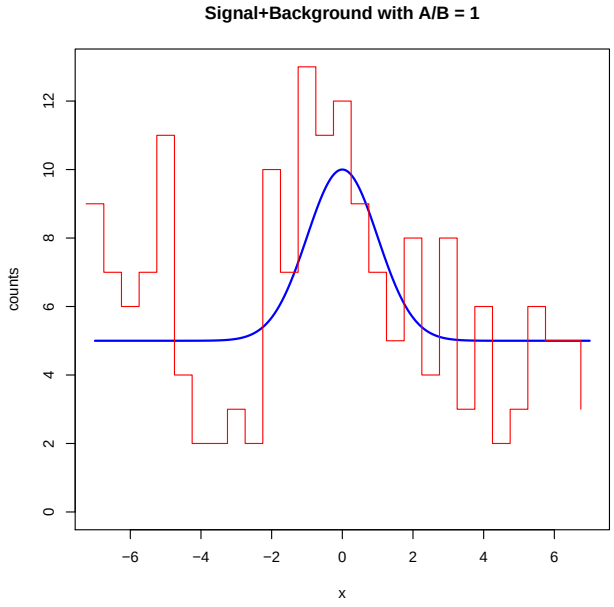

2D posterior with w = 2
true position
estimated pos.
background, B
amplitude, A

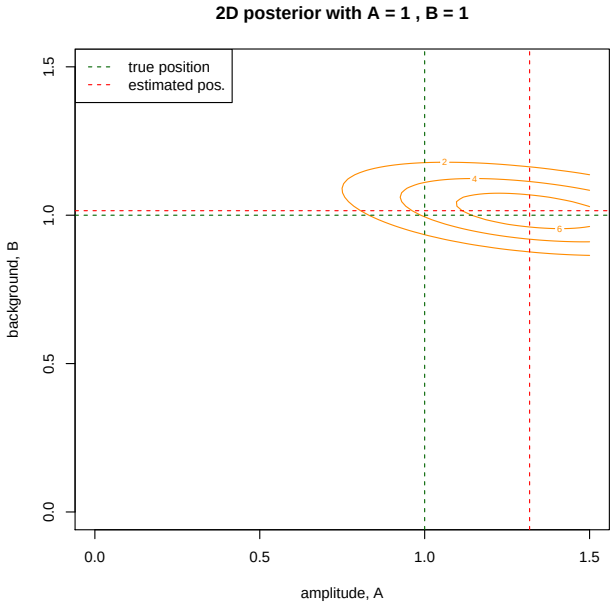

Signal+Background with w = 3
counts
x

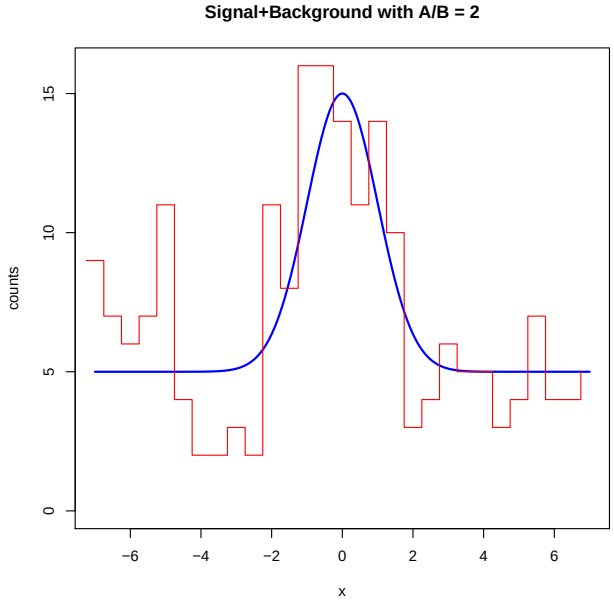

2D posterior with w = 3
true position
estimated pos.
background, B
amplitude, A

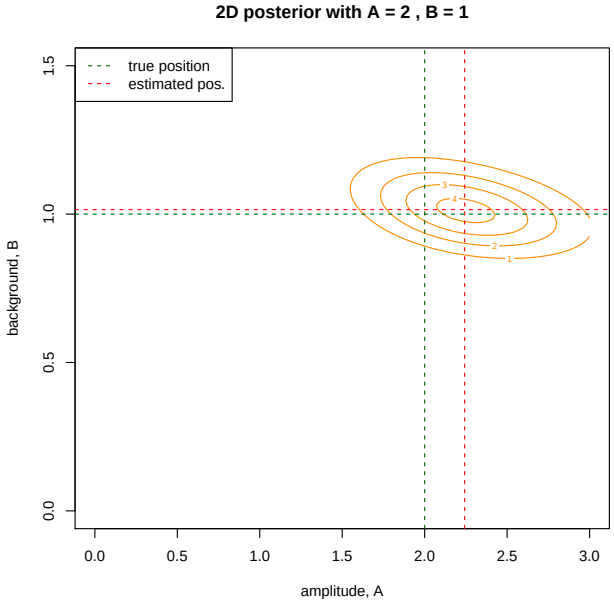
B) change the ratio A/B used to simulate the data (keeping both positive in accordance with the prior)

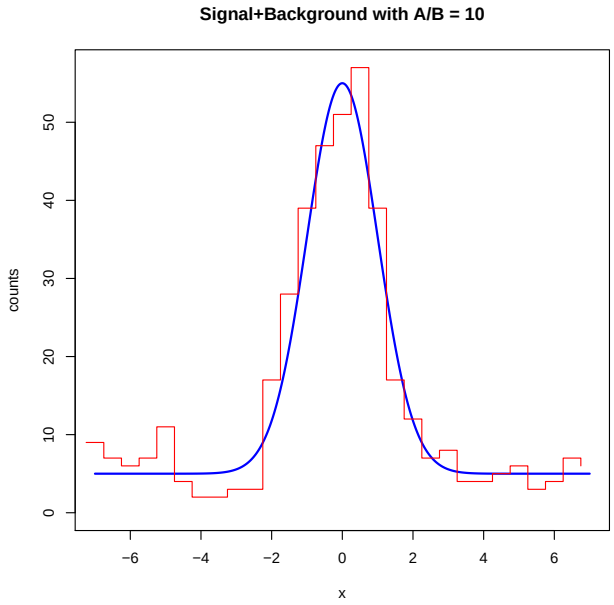
```
## B ##
# Parameters
x0 <- 0
dt <- 5     #exposure time in seconds
w  <- 0.5   #sampling resolution
width <- 1

as.true <- c(1,2,10,20)
bs.true <- c(1,1,1,1)

par(mfrow=c(2,2))
for (i in 1:4) {
    set.seed(2381)
    xdat <- seq(from=-7*width, to=7*width, by=w*width)
    s.true <- signal(xdat, as.true[i], bs.true[i], x0, width, dt)
    s.counts <- rpois(length(s.true), s.true)

    xps <- seq(min(xdat), max(xdat), len=1000)
    sps <- signal(xps, as.true[i], bs.true[i], x0, width, dt)
    ymax <- max( c(max(sps), max(s.counts)) )

    plot(xps, sps, type='l', col='blue', ylim=c(0,ymax), xlab="x", ylab="counts",
         main=paste("Signal+Background with A/B =",as.true[i]/bs.true[i]),
         xlim=c(-7,7), lwd=2)
    lines(xdat-w/2, s.counts, type='s', col='red' )

    max.a <- as.true[i]+as.true[i]/2; max.b <- bs.true[i]+bs.true[i]/2
    a <- seq(0,max.a,len=100)
    b <- seq(0,max.b,len=100)
    delta_a <- max.a/100
    delta_b <- max.b/100
    # Computing log unnormalized posterior on a regular grid
    z <- matrix(data=NA , nrow= length(a), ncol= length(b))
    for (j in 1: length(a)) {
        for (k in 1: length(b)) {
            z[j,k] <- log.post(s.counts, xdat, a[j], b[k], x0, width, dt)
        }
    }
    # Normalizing posterior
    z.norm <- exp(z)/(delta_a*delta_b*sum(exp(z)))

    # Plot normalized 2D posterior as contours
    contour(a, b, z.norm, nlevels = 4, labcex = 0.5,, xlab="amplitude, A",
             ylab="background, B", col="darkorange",
             main=paste("2D posterior with A =",as.true[i],", B =",bs.true[i]))
    abline(v=as.true[i], h=bs.true[i], col="darkgreen", lty=2)
    max_ids <- c( which(z.norm==max(z.norm), arr.ind=TRUE) )
    abline(v=a[max_ids[1]], h=b[max_ids[2]], col="red", lty=2)
    legend("topleft", c("true position", "estimated pos."),
           col=c("darkgreen","red"), lty=2)
}
```

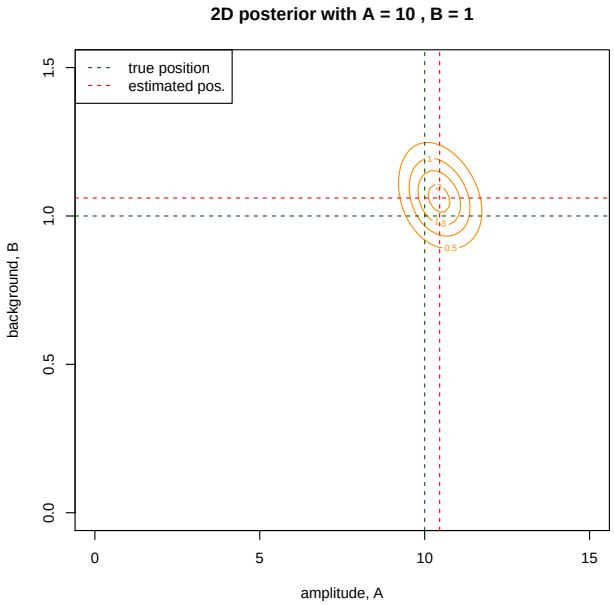

Signal+Background with A/B = 1
counts
x

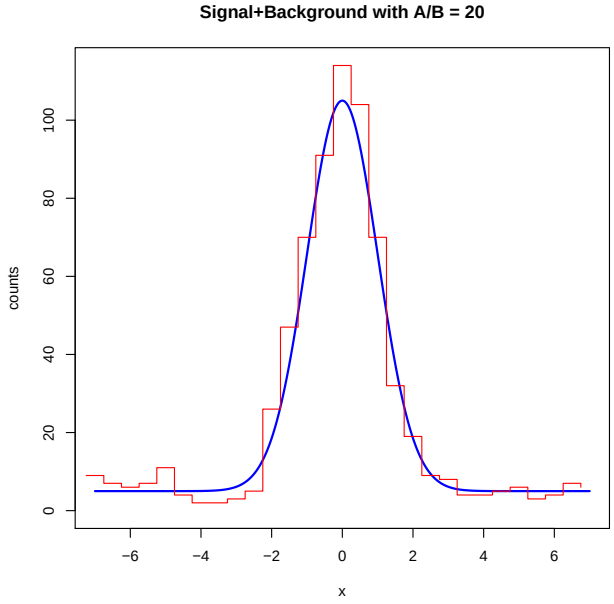

2D posterior with A = 1 , B = 1
true position
estimated pos.
background, B
amplitude, A

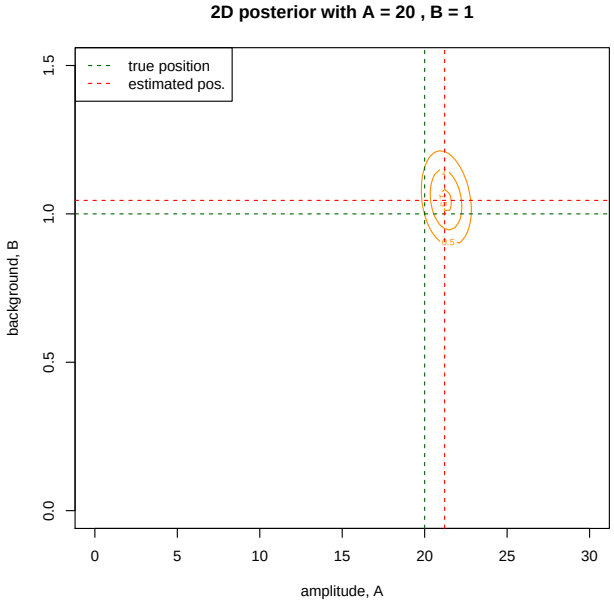

Signal+Background with A/B = 2
counts
x


2D posterior with A = 2 , B = 1
true position
estimated pos.
background, B
amplitude, A


Signal+Background with A/B = 10
counts
x
0
10
20
30
40
50
−6
−4
−2
0
2
4
6


2D posterior with A = 10 , B = 1
true position
estimated pos.
background, B
amplitude, A
0.0
0.5
1.0
1.5
0
5
10
15


Signal+Background with A/B = 20
counts
x
0
20
40
60
80
100
−6
−4
−2
0
2
4
6


2D posterior with A = 20 , B = 1
true position
estimated pos.
background, B
amplitude, A
0.0
0.5
1.0
1.5
0
5
10
15
20
25
30