2D Moffat

2024-03-25

library(ggplot2)  
library(plotly)

##   
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':  
##   
## last\_plot

## The following object is masked from 'package:stats':  
##   
## filter

## The following object is masked from 'package:graphics':  
##   
## layout

library(ggpointdensity)  
library(viridis)

## Loading required package: viridisLite

library(tidyr)  
library(ggExtra)  
library(ggpubr)  
library(cowplot)

##   
## Attaching package: 'cowplot'

## The following object is masked from 'package:ggpubr':  
##   
## get\_legend

library(pracma)  
  
## Color pallete  
cbPalette <- c("#56B4E9", "#E69F00", "#009E73", "#F0E442", "#0072B2", "#D55E00", "#CC79A7")

Importing the data and transforming for stages required at this analysis step.

library(here)

## here() starts at C:/Users/Blake/Dropbox/My PC (LAPTOP-R53ILDBG)/Documents/GitHub/Terzan-5

i\_am("2D Moffat.Rmd")

## here() starts at C:/Users/Blake/Dropbox/My PC (LAPTOP-R53ILDBG)/Documents/GitHub/Terzan-5

# laptop data from csv  
Terzan5 <- readr::read\_csv(here("Data", "Terzan 5 X-ray events.csv"), col\_types = list(.default = readr::col\_guess()), )  
# head(Terzan5)  
  
#fishbowl data from ODS  
# library(readODS)  
# Terzan5 <- read\_ods(here("Raw Data.ods"), col\_types = list(.default = readr::col\_guess()), )  
#removing extraneous columns  
T5 <- data.frame(Terzan5$x, Terzan5$y)  
colnames(T5) <- c('x','y')  
  
#filtering into two Points of Interest  
POI1 <- T5 %>%  
 filter(x>4160 & x<4180 & y>4180 & y<4200)  
POI2 <- T5 %>%  
 filter(x>4105 & x< 4125 & y>4040 & y<4060)

The 2D Moffat Model has this form:

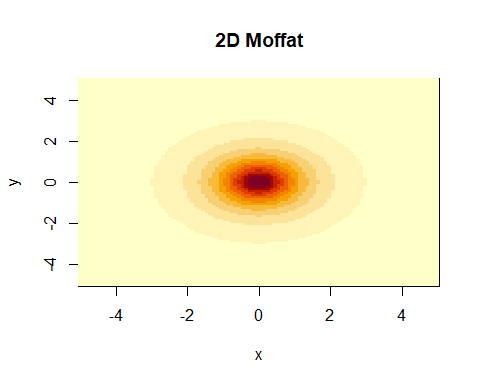
Where A = amplitude, = x-position of the maximum, = y-position of the maximum, = core width, = power index.

Symmetric uncorr function - same influence

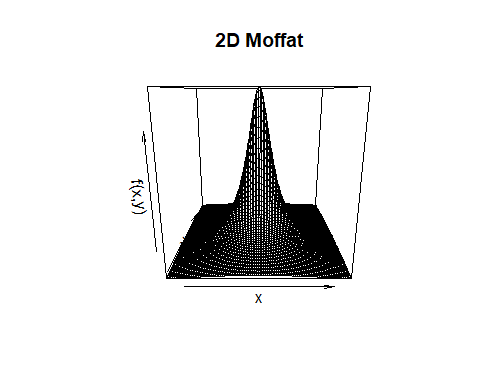
Similarly to the 1D Moffat, the amplitude must be greater than 0. Alpha can’t be equal to zero and Beta must be greater than 0.

Visualising the 2D Moffat

Moffat2D <- function(grid, parameters){  
 x <- grid[[1]]  
 y <- grid[[2]]  
   
 amplitude <- parameters[[1]]  
 x0 <- parameters[[2]]  
 y0 <- parameters[[3]]  
 alpha <- parameters[[4]]  
 beta <- parameters[[5]]  
   
 amplitude\*(1+((x - x0)^2 + (y - y0)^2)/alpha^2)^(-beta)  
}  
  
#grid of variables  
xRange <- seq(-5, 5, length = 100)  
yRange <- seq(-5, 5, length = 100)  
grid <- expand.grid(x = xRange, y = yRange)  
  
#Dummy parameters  
dummyParams <- c(amplitude = 1, x0 = 0, y0 = 0, alpha = 1, beta = 1)  
  
z <- Moffat2D(grid, dummyParams)  
zMatrix <- matrix(z, nrow = length(xRange), ncol = length(yRange))  
  
image(xRange, yRange, zMatrix, xlab = "x", ylab = "y", main = "2D Moffat")

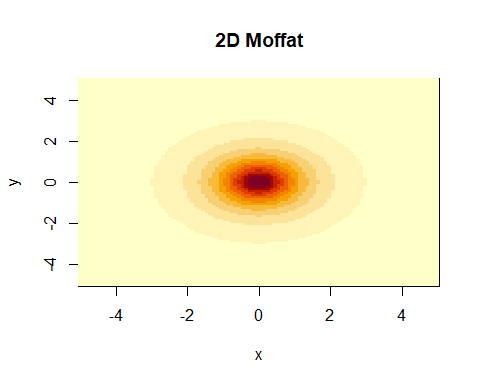


persp(xRange, yRange, zMatrix, xlab = "x", ylab = "y", zlab = "f(x,y)", main = "2D Moffat")

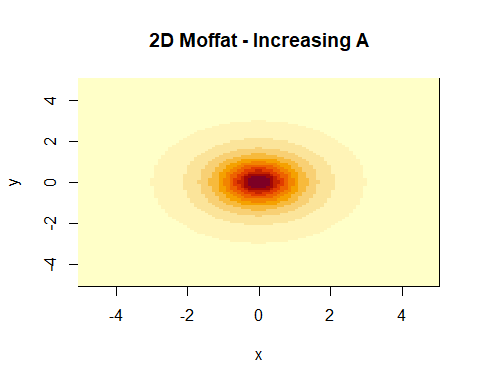


Visualising the effects of changing parameters

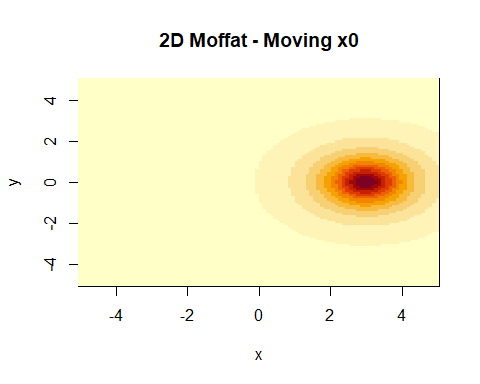
#baseline of 1,0,0,1,1  
dummyParams <- c(amplitude = 1, x0 = 0, y0 = 0, alpha = 1, beta = 1)  
dummyParams1 <- c(amplitude = 4, x0 = 0, y0 = 0, alpha = 1, beta = 1)  
dummyParams2 <- c(amplitude = 1, x0 = 3, y0 = 0, alpha = 1, beta = 1)  
dummyParams3 <- c(amplitude = 1, x0 = 0, y0 = -2, alpha = 1, beta = 1)  
dummyParams4 <- c(amplitude = 1, x0 = 0, y0 = 0, alpha = 3, beta = 1)  
dummyParams5 <- c(amplitude = 1, x0 = 0, y0 = 0, alpha = 1, beta = 2.6)  
  
#baseline plot  
z <- Moffat2D(grid, dummyParams)  
zMatrix <- matrix(z, nrow = length(xRange), ncol = length(yRange))  
image(xRange, yRange, zMatrix, xlab = "x", ylab = "y", main = "2D Moffat")



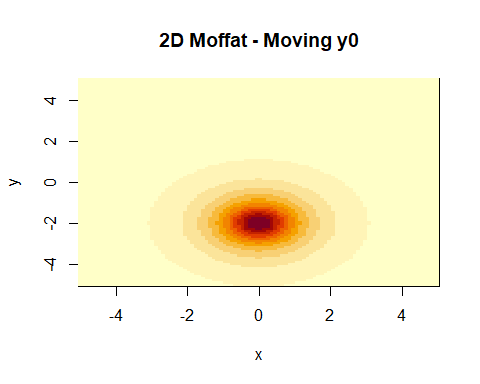
#increasing amplitude  
z1 <- Moffat2D(grid, dummyParams1)  
z1Matrix <- matrix(z, nrow = length(xRange), ncol = length(yRange))  
image(xRange, yRange, z1Matrix, xlab = "x", ylab = "y", main = "2D Moffat - Increasing A")



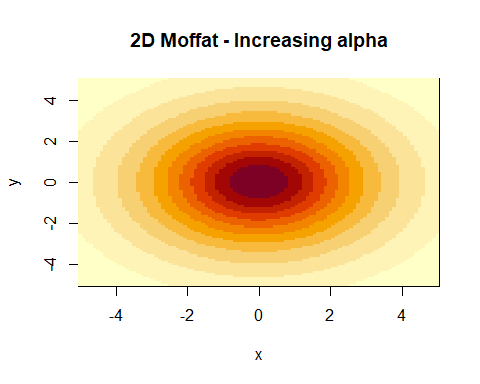
#moving x0  
z2 <- Moffat2D(grid, dummyParams2)  
z2Matrix <- matrix(z2, nrow = length(xRange), ncol = length(yRange))  
image(xRange, yRange, z2Matrix, xlab = "x", ylab = "y", main = "2D Moffat - Moving x0")



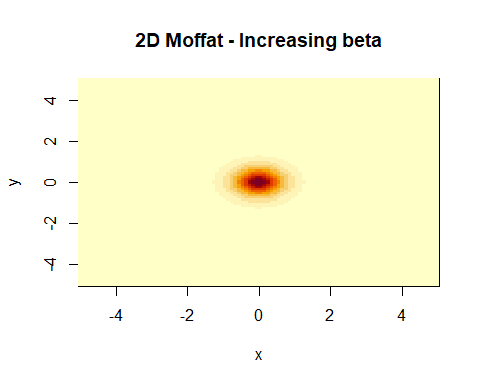
#moving y0  
z3 <- Moffat2D(grid, dummyParams3)  
z3Matrix <- matrix(z3, nrow = length(xRange), ncol = length(yRange))  
image(xRange, yRange, z3Matrix, xlab = "x", ylab = "y", main = "2D Moffat - Moving y0")



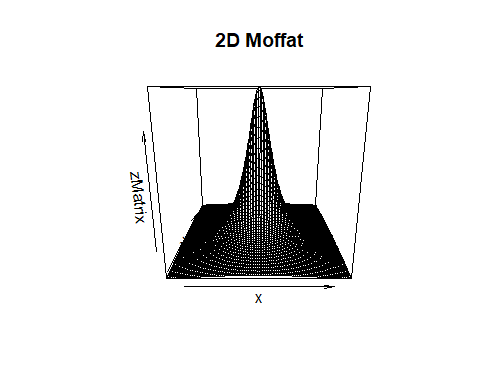
#Increasing alpha  
z4 <- Moffat2D(grid, dummyParams4)  
z4Matrix <- matrix(z4, nrow = length(xRange), ncol = length(yRange))  
image(xRange, yRange, z4Matrix, xlab = "x", ylab = "y", main = "2D Moffat - Increasing alpha")



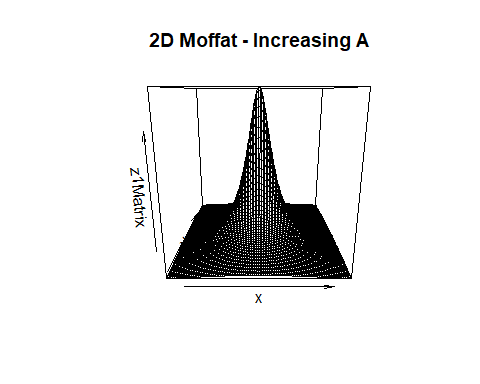
#Increasing beta  
z5<- Moffat2D(grid, dummyParams5)  
z5Matrix <- matrix(z5, nrow = length(xRange), ncol = length(yRange))  
image(xRange, yRange, z5Matrix, xlab = "x", ylab = "y", main = "2D Moffat - Increasing beta")



persp(xRange, yRange, zMatrix, xlab = "x", ylab = "y", main = "2D Moffat")



persp(xRange, yRange, z1Matrix, xlab = "x", ylab = "y", main = "2D Moffat - Increasing A")

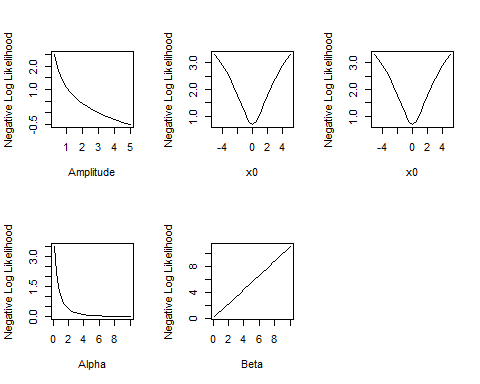


The Negative Log Likelihood of the 2D Moffat Model:

NLL <- function(grid, parameters){  
 predictedDens <- Moffat2D(grid, parameters)  
 return(-sum(log(predictedDens)))  
}

Investigating how the parameters effect the negative log likelihood function

#changing the value of the amplitude  
amplitude <- seq(0.25, 5, by = 0.25)  
  
i <- 1  
changingA <- vector("numeric", length = 20)  
  
while(i<=20){  
 parameters <- c(amplitude[[i]], 1, 1, 1, 1)  
 grid <- c(0,0)  
 changingA[i] <- (NLL(grid, parameters))  
 i <- i+1  
}  
  
#changing the value of x0  
x0 <- seq(-5, 5, by = 0.5)  
  
i <- 1  
changingx0 <- vector("numeric", length = 21)  
  
while(i<=21){  
 parameters <- c(1, x0[[i]], 1, 1, 1)  
 grid <- c(0,0)  
 changingx0[i] <- (NLL(grid, parameters))  
 i <- i+1  
}  
  
#changing the value of y0  
y0 <- seq(-5, 5, by = 0.5)  
  
i <- 1  
changingy0 <- vector("numeric", length = 21)  
  
while(i<=21){  
 parameters <- c(1, 1, y0[[i]], 1, 1)  
 grid <- c(0,0)  
 changingy0[i] <- (NLL(grid, parameters))  
 i <- i+1  
}  
  
#changing the value of alpha  
alpha <- seq(0.25, 10, by = 0.25)  
  
i <- 1  
changingAlpha <- vector("numeric", length = 40)  
  
while(i<=40){  
 parameters <- c(1, 1, 1, alpha[[i]], 1)  
 grid <- c(0,0)  
 changingAlpha[i] <- (NLL(grid, parameters))  
 i <- i+1  
}  
  
#changing the value of beta  
beta <- seq(0.25, 10, by = 0.25)  
  
i <- 1  
changingBeta <- vector("numeric", length = 40)  
  
while(i<=40){  
 parameters <- c(1, 1, 1, 1, beta[[i]])  
 grid <- c(0,0)  
 changingBeta[i] <- (NLL(grid, parameters))  
 i <- i+1  
}  
  
par(mfrow = c(2,3))  
plot(amplitude, changingA, type = "l", xlab = "Amplitude", ylab = "Negative Log Likelihood")  
plot(x0, changingx0, type = "l", xlab = "x0", ylab = "Negative Log Likelihood")  
plot(y0, changingy0, type = "l", xlab = "x0", ylab = "Negative Log Likelihood")  
plot(alpha, changingAlpha, type = "l", xlab = "Alpha", ylab = "Negative Log Likelihood")  
plot(beta, changingBeta, type = "l", xlab = "Beta", ylab = "Negative Log Likelihood")

 This is similar to the plots of the effects the parameters of the 1D Moffat Function have on the Negative Log Likelihood.

Amplitude needs to be expressed in terms of alpha and beta.

Validating the 2D Moffat PDF

Moffat2DPDF <- function(parameters, x, y){  
 x0 <- parameters[[1]]  
 y0 <- parameters[[2]]  
 alpha <- parameters[[3]]  
 beta <- parameters[[4]]  
   
 alpha <- pmax(alpha, 1e-6)  
 beta <- pmax(beta, 1 + 1e-6)  
   
 return(((beta-1)/(pi\*alpha^2))\*(1+((x - x0)^2 + (y - y0)^2)/alpha^2)^(-beta))  
}  
  
library(pracma)  
#Test 1  
T1Params <- c(x0 = 1, y0 = 4, alpha = 0.4, beta = 2)  
MoffatIntegral2D1 <- integral2(Moffat2DPDF, xmin = -100, xmax = 100, ymin = -100, ymax = 100, parameters = T1Params)[[1]]  
MoffatIntegral2D1

## [1] 0.9999706

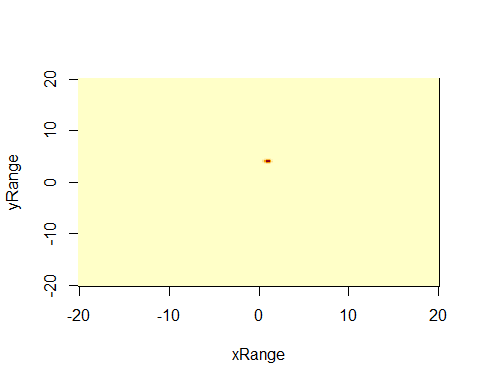
#Test 2  
T2Params <- c(x0 = -3, y0 = 7, alpha = 1.2, beta = 1.3)  
MoffatIntegral2D2 <- integral2(Moffat2DPDF, xmin = -100, xmax = 100, ymin = -100, ymax = 100, parameters = T2Params)[[1]]  
MoffatIntegral2D2

## [1] 0.9379584

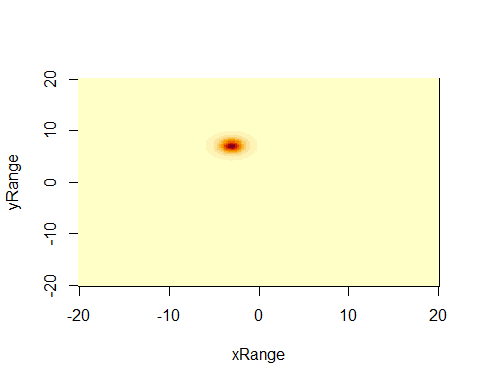
#Test 3  
T3Params <- c(x0 = 4, y0 = 0, alpha = 3, beta = 2.4)  
MoffatIntegral2D3 <- integral2(Moffat2DPDF, xmin = -100, xmax = 100, ymin = -100, ymax = 100, parameters = T3Params)[[1]]  
MoffatIntegral2D3

## [1] 1.047134

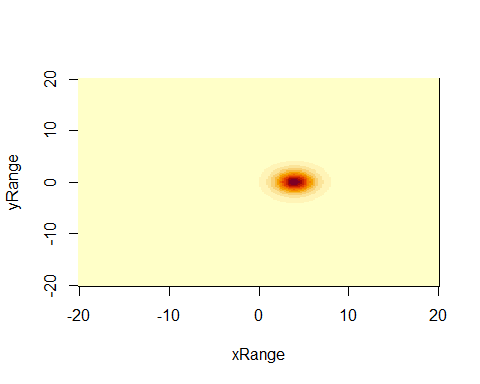
xRange <- seq(-20, 20, length = 200)  
yRange <- seq(-20, 20, length = 200)  
grid <- expand.grid(xRange, yRange)  
  
A1 <- (T1Params[[4]]-1)/(pi \* T1Params[[3]])  
zT1 <- Moffat2D(grid, parameters = c(A1, T1Params))  
zT1Matrix <- matrix(zT1, nrow = length(xRange), ncol = length(yRange))  
image(xRange, yRange, zT1Matrix)



A2 <- (T2Params[[4]]-1)/(pi \* T2Params[[3]])  
zT2 <- Moffat2D(grid, parameters = c(A2, T2Params))  
zT2Matrix <- matrix(zT2, nrow = length(xRange), ncol = length(yRange))  
image(xRange, yRange, zT2Matrix)



A3 <- (T3Params[[4]]-1)/(pi \* T3Params[[3]])  
zT3 <- Moffat2D(grid, parameters = c(A3, T3Params))  
zT3Matrix <- matrix(zT3, nrow = length(xRange), ncol = length(yRange))  
image(xRange, yRange, zT3Matrix)



Gradient of the negative log likelihood with respect to each parameter

NLL2D <- function(parameters, x, y){  
 predictedDens <- Moffat2DPDF(parameters, x, y)  
 return(-sum(log(predictedDens)))  
}

deriv2DNLL <- function(parameters, x, y){  
 x0 <- parameters[[1]]  
 y0 <- parameters[[2]]  
 alpha <- parameters[[3]]  
 beta <- parameters[[4]]  
   
 dx0 <- -sum((2\*beta\*(x-x0))/(alpha^2 + (x-x0)^2 + (y-y0)^2))  
 dy0 <- -sum((2\*beta\*(y-y0))/(alpha^2 + (x-x0)^2 + (y-y0)^2))  
 dAlpha <- 2\*sum(1/alpha + (beta\*((x-x0)^2 + (y-y0)^2))/(alpha^3 + alpha\*((x-x0)^2 + (y-y0)^2)))  
 dBeta <- -sum(1/beta - log(1+((x-x0)^2 + (y-y0)^2)/alpha^2))  
   
 return(c(dx0, dy0, dAlpha, dBeta))  
}

mean(POI1$x)

## [1] 4171.755

mean(POI1$y)

## [1] 4187.421

initGuess <- c(x0 = mean(POI1$x), y0 = mean(POI1$y), alpha = 1, beta = 2)  
lowerBounds <- c(0, 0, 1e-6, 1+1e-6)  
  
resultPOI1 <- optim(par = initGuess, fn = NLL2D, gr = deriv2DNLL, x = POI1$x, y = POI1$y, method = "L-BFGS-B", lower = lowerBounds)  
resultPOI1

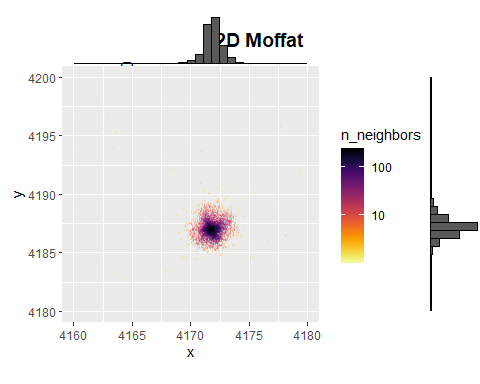
## $par  
## x0 y0 alpha beta   
## 4171.8503337 4187.1942118 0.7774318 1.9415123   
##   
## $value  
## [1] 14282.57  
##   
## $counts  
## function gradient   
## 45 45   
##   
## $convergence  
## [1] 0  
##   
## $message  
## [1] "CONVERGENCE: REL\_REDUCTION\_OF\_F <= FACTR\*EPSMCH"

optimPOI1Par <- resultPOI1$par  
optimPOI1Par

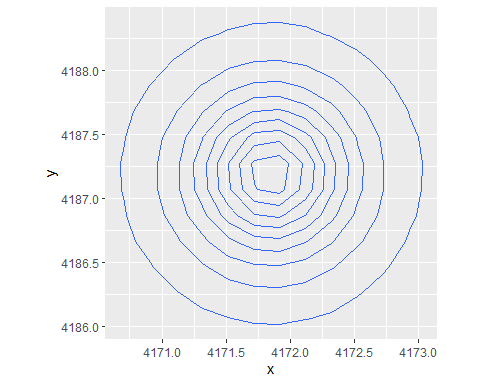
## x0 y0 alpha beta   
## 4171.8503337 4187.1942118 0.7774318 1.9415123

Viewing the optimised function and viewing the data.

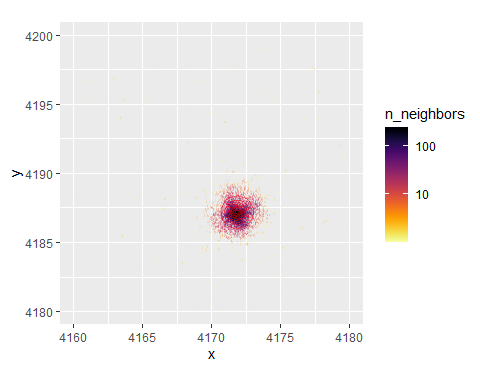
xRange <- seq(4160, 4180, length = 100)  
yRange <- seq(4180, 4200, length = 100)  
grid <- expand.grid(x = xRange, y = yRange)  
  
amplitude <- (optimPOI1Par[[4]]-1)/(pi\*optimPOI1Par[[3]]^2)  
parameters <- c(amplitude, optimPOI1Par)  
  
zPOI1<- Moffat2D(grid, parameters)  
zPOI1Matrix <- matrix(zPOI1, nrow = length(xRange), ncol = length(yRange))  
par(pty = "s")  
image(xRange, yRange, zPOI1Matrix, xlab = "x", ylab = "y", main = "2D Moffat")  
  
model <- ggplot(POI1, aes(x,y)) +  
 geom\_pointdensity(adjust = 0.05, size = 0.1, shape = "1") +   
 scale\_color\_viridis(direction = -1, option = "B", trans = "log", breaks = c(10, 100, 1000)) +   
 coord\_fixed()  
  
contours <- ggplot(grid, aes(x = x, y = y)) +   
 stat\_contour(aes(z = zPOI1), bins = 10) +  
 coord\_fixed()  
  
ggMarginal(model, type = "histogram")



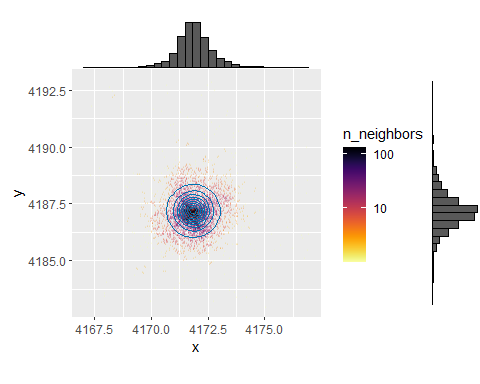
contours



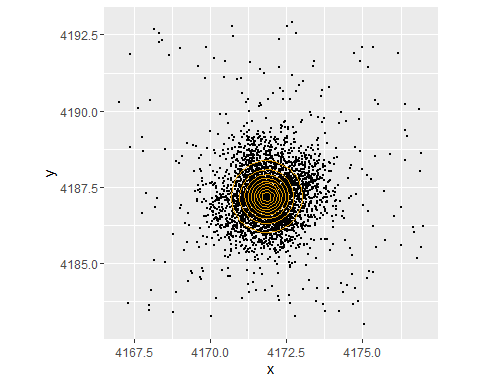
contourData <- ggplot\_build(contours)$data[[1]] #extracting the contours (from the optimised function)  
combined <- model + geom\_path(data = contourData, aes(x = x, y = y, group = group), color = "red", alpha = 0.5)  
combined

 Zooming in on this point of interest to try to get a clearer picture.

library(ggExtra)  
#refiltering the POI1  
POI1ZOOM <- POI1 %>%  
 filter(x>4167 & x<4177 & y>4183 & y<4193)  
  
#creating a point density plot  
modelZoomedInPOI1 <- ggplot(POI1ZOOM, aes(x,y)) +  
 geom\_pointdensity(adjust = 0.05, size = 0.1, shape = "1") +   
 scale\_color\_viridis(direction = -1, option = "B", trans = "log", breaks = c(10, 100, 1000)) +   
 coord\_fixed()  
  
contourData <- ggplot\_build(contours)$data[[1]]  
  
#combining the contour plot and the point density plot  
combined <- modelZoomedInPOI1 + geom\_path(data = contourData, aes(x = x, y = y, group = group), color = cbPalette[5], alpha = 1, linewidth= 0.3)  
  
#viewing  
ggMarginal(combined, type = "histogram")



#plotting contours on the raw data  
ggplot(POI1ZOOM, aes(x,y)) +  
 geom\_point(size = 0.2) +   
 geom\_path(data = contourData, aes(x = x, y = y, group = group), color = cbPalette[2], alpha = 1) +  
 coord\_fixed()

 Add the marginal PDF of the model on top of the histogram

mean(POI2$x)

## [1] 4115.227

mean(POI2$y)

## [1] 4051.706

initGuess <- c(x0 = 4115.227, y0 = 4052.706, alpha = 1, beta = 2)  
  
result2D <- optim(par = initGuess, fn = NLL2D, gr = deriv2DNLL, x = POI2$x, y = POI2$y, method = "L-BFGS-B", lower = lowerBounds)  
result2D

## $par  
## x0 y0 alpha beta   
## 4115.2611263 4051.9035135 0.6832075 1.9000752   
##   
## $value  
## [1] 14069.95  
##   
## $counts  
## function gradient   
## 26 26   
##   
## $convergence  
## [1] 0  
##   
## $message  
## [1] "CONVERGENCE: REL\_REDUCTION\_OF\_F <= FACTR\*EPSMCH"

optimPOI2Par <- result2D$par  
optimPOI2Par

## x0 y0 alpha beta   
## 4115.2611263 4051.9035135 0.6832075 1.9000752

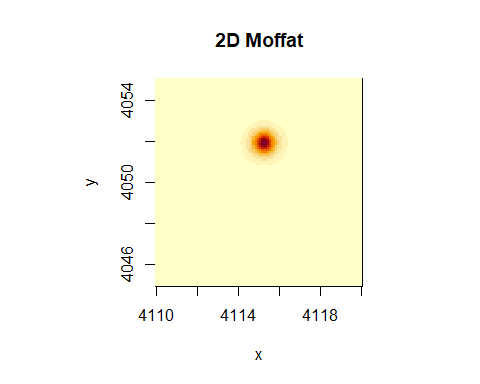
#check, the area under the curve should be equal to 1.  
integral2(Moffat2DPDF, xmin = 3000, xmax = 5000, ymin = 3000, ymax = 5000, parameters = optimPOI2Par)[[1]]

## [1] 0.9994666

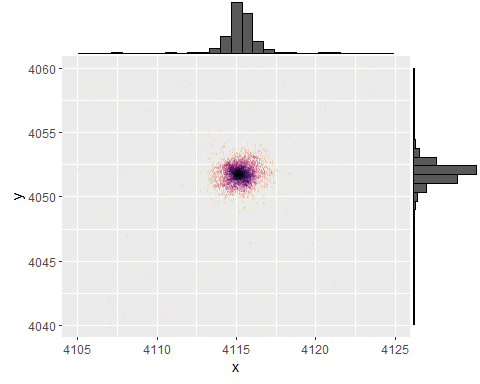
#maximum amplitude  
Moffat2DPDF(optimPOI2Par, 4115.26, 4051.90)

## [1] 0.6137615

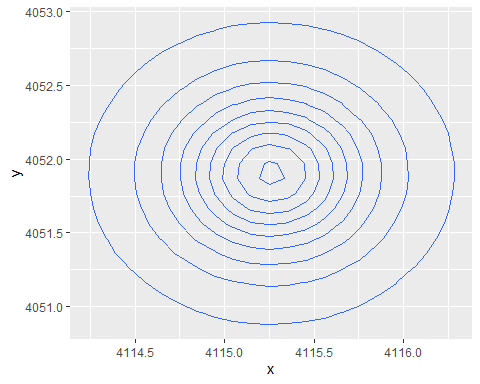
xRange <- seq(4110, 4120, length = 100)  
yRange <- seq(4045, 4055, length = 100)  
grid <- expand.grid(x = xRange, y = yRange)  
  
amplitude <- (optimPOI2Par[[4]]-1)/(pi\*optimPOI2Par[[3]]^2)  
#amplitude  
parameters <- c(amplitude, optimPOI2Par)  
  
zPOI2<- Moffat2D(grid, parameters)  
zPOI2Matrix <- matrix(zPOI2, nrow = length(xRange), ncol = length(yRange))  
par(pty = "s")  
image(xRange, yRange, zPOI2Matrix, xlab = "x", ylab = "y", main = "2D Moffat")



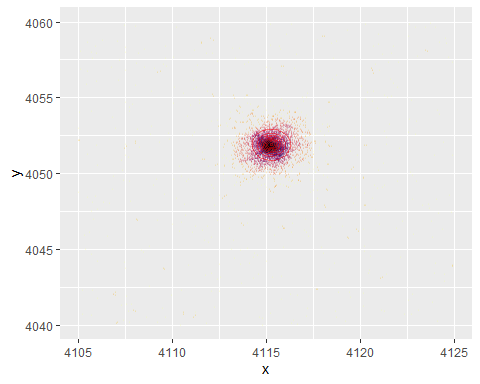
xRange <- seq(4110, 4120, length = 100)  
yRange <- seq(4045, 4055, length = 100)  
data <- expand.grid(x = xRange, y = yRange)  
  
model <- ggplot(POI2, aes(x,y)) +  
 geom\_pointdensity(adjust = 0.05, size = 0.1, shape = "1") +   
 scale\_color\_viridis(direction = -1, option = "B", trans = "log", breaks = c(10, 100, 1000)) +  
 theme(legend.position = "none")  
  
contours <- ggplot(data, aes(x = x, y = y)) +   
 stat\_contour(aes(z = zPOI2), bins = 10)   
  
ggMarginal(model, type = "histogram")



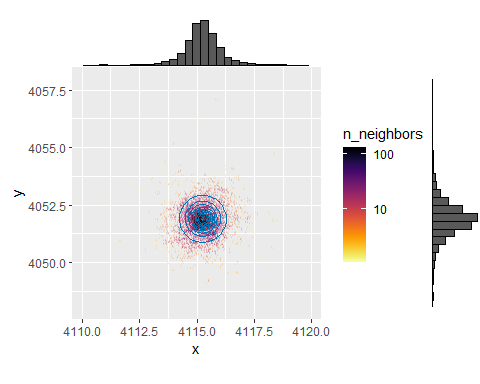
contours



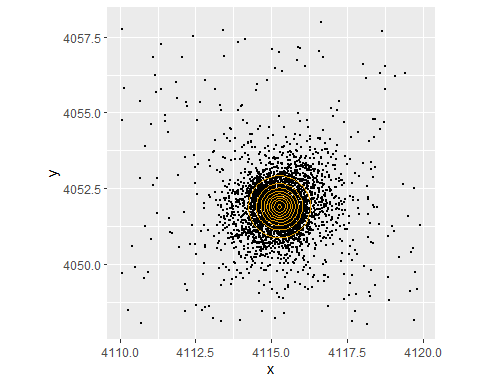
contourData <- ggplot\_build(contours)$data[[1]]  
  
combined <- model + geom\_path(data = contourData, aes(x = x, y = y, group = group), color = "red", alpha = 0.5)  
  
combined

 Zooming in on this point of interest to try to get a clearer picture.

#refiltering the POI2  
POI2ZOOM <- POI2 %>%  
 filter(x>4110 & x<4120 & y>4048 & y<4058)  
  
#creating a point density plot  
modelZoomedInPOI2 <- ggplot(POI2ZOOM, aes(x,y)) +  
 geom\_pointdensity(adjust = 0.05, size = 0.1, shape = "1") +   
 scale\_color\_viridis(direction = -1, option = "B", trans = "log", breaks = c(10, 100, 1000)) +   
 coord\_fixed()  
  
contourData <- ggplot\_build(contours)$data[[1]]  
  
#combining the contour plot and the point density plot  
combined <- modelZoomedInPOI2 + geom\_path(data = contourData, aes(x = x, y = y, group = group), color = cbPalette[5], alpha = 1, linewidth= 0.3)  
  
#viewing  
ggMarginal(combined, type = "histogram")

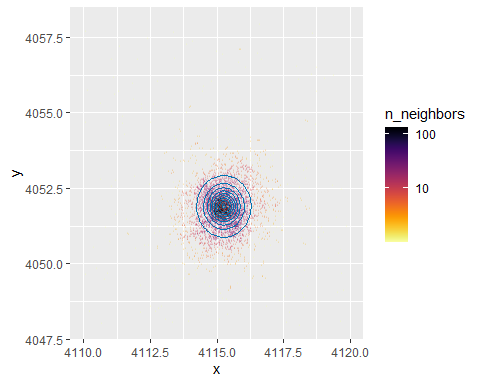


#plotting contours on the raw data  
ggplot(POI2ZOOM, aes(x,y)) +  
 geom\_point(size = 0.2) +   
 geom\_path(data = contourData, aes(x = x, y = y, group = group), color = cbPalette[2], alpha = 1) +  
 coord\_fixed()

 Alternate zoom method

combined + coord\_cartesian(xlim = c(4110, 4120), ylim = c(4048, 4058))

## Coordinate system already present. Adding new coordinate system, which will  
## replace the existing one.



I would like to display the Marginal PDF’s of the model.

Obtaining Marginal PDF’s - Technique 1

xRange <- seq(4105, 4125, length = 200)  
yRange <- seq(4040, 4060, length = 200)  
data <- expand.grid(x = xRange, y = yRange)  
  
#for all values within the above grid, apply the 2D Moffat PDF function with the optimised parameters. This will generate a grid of values. Where X is rows and Y are columns  
jointPDFValues <- outer(X = xRange, Y = yRange, Moffat2DPDF, parameters = optimPOI2Par)  
  
#check  
tot <- sum(jointPDFValues)  
  
#To obtain the marginal PDFS, I need to integrate the joint function with respect to one of the variables.   
#Instead I sum up all the values in each row. This has the effect of integrating with respect to y. As I take the value from each position over the range.   
marginalX <- apply(jointPDFValues, 1, sum)  
marginalY <- apply(jointPDFValues, 2, sum)  
  
data.frame(xRange, marginalX)

## xRange marginalX  
## 1 4105.000 0.002689192  
## 2 4105.101 0.002775461  
## 3 4105.201 0.002865247  
## 4 4105.302 0.002958727  
## 5 4105.402 0.003056085  
## 6 4105.503 0.003157521  
## 7 4105.603 0.003263243  
## 8 4105.704 0.003373474  
## 9 4105.804 0.003488450  
## 10 4105.905 0.003608423  
## 11 4106.005 0.003733658  
## 12 4106.106 0.003864440  
## 13 4106.206 0.004001073  
## 14 4106.307 0.004143877  
## 15 4106.407 0.004293196  
## 16 4106.508 0.004449398  
## 17 4106.608 0.004612873  
## 18 4106.709 0.004784041  
## 19 4106.809 0.004963346  
## 20 4106.910 0.005151270  
## 21 4107.010 0.005348322  
## 22 4107.111 0.005555053  
## 23 4107.211 0.005772051  
## 24 4107.312 0.005999948  
## 25 4107.412 0.006239422  
## 26 4107.513 0.006491205  
## 27 4107.613 0.006756081  
## 28 4107.714 0.007034897  
## 29 4107.814 0.007328564  
## 30 4107.915 0.007638068  
## 31 4108.015 0.007964472  
## 32 4108.116 0.008308926  
## 33 4108.216 0.008672674  
## 34 4108.317 0.009057067  
## 35 4108.417 0.009463567  
## 36 4108.518 0.009893764  
## 37 4108.618 0.010349385  
## 38 4108.719 0.010832312  
## 39 4108.819 0.011344596  
## 40 4108.920 0.011888474  
## 41 4109.020 0.012466391  
## 42 4109.121 0.013081023  
## 43 4109.221 0.013735302  
## 44 4109.322 0.014432443  
## 45 4109.422 0.015175983  
## 46 4109.523 0.015969812  
## 47 4109.623 0.016818218  
## 48 4109.724 0.017725937  
## 49 4109.824 0.018698205  
## 50 4109.925 0.019740823  
## 51 4110.025 0.020860230  
## 52 4110.126 0.022063586  
## 53 4110.226 0.023358864  
## 54 4110.327 0.024754969  
## 55 4110.427 0.026261857  
## 56 4110.528 0.027890689  
## 57 4110.628 0.029654006  
## 58 4110.729 0.031565925  
## 59 4110.829 0.033642382  
## 60 4110.930 0.035901406  
## 61 4111.030 0.038363450  
## 62 4111.131 0.041051776  
## 63 4111.231 0.043992918  
## 64 4111.332 0.047217224  
## 65 4111.432 0.050759518  
## 66 4111.533 0.054659876  
## 67 4111.633 0.058964580  
## 68 4111.734 0.063727252  
## 69 4111.834 0.069010248  
## 70 4111.935 0.074886350  
## 71 4112.035 0.081440837  
## 72 4112.136 0.088774046  
## 73 4112.236 0.097004528  
## 74 4112.337 0.106272988  
## 75 4112.437 0.116747198  
## 76 4112.538 0.128628183  
## 77 4112.638 0.142158047  
## 78 4112.739 0.157629915  
## 79 4112.839 0.175400664  
## 80 4112.940 0.195907302  
## 81 4113.040 0.219688176  
## 82 4113.141 0.247410591  
## 83 4113.241 0.279906981  
## 84 4113.342 0.318222533  
## 85 4113.442 0.363678166  
## 86 4113.543 0.417954087  
## 87 4113.643 0.483200833  
## 88 4113.744 0.562186625  
## 89 4113.844 0.658491761  
## 90 4113.945 0.776761627  
## 91 4114.045 0.923027411  
## 92 4114.146 1.105092288  
## 93 4114.246 1.332949270  
## 94 4114.347 1.619122047  
## 95 4114.447 1.978660321  
## 96 4114.548 2.428213627  
## 97 4114.648 2.983099692  
## 98 4114.749 3.650677215  
## 99 4114.849 4.418285445  
## 100 4114.950 5.236344107  
## 101 4115.050 6.004459385  
## 102 4115.151 6.578876320  
## 103 4115.251 6.818593986  
## 104 4115.352 6.655971392  
## 105 4115.452 6.137468185  
## 106 4115.553 5.395226077  
## 107 4115.653 4.577373715  
## 108 4115.754 3.794474754  
## 109 4115.854 3.105346544  
## 110 4115.955 2.528485943  
## 111 4116.055 2.059325912  
## 112 4116.156 1.683438478  
## 113 4116.256 1.384115443  
## 114 4116.357 1.145875892  
## 115 4116.457 0.955681346  
## 116 4116.558 0.803062961  
## 117 4116.658 0.679820220  
## 118 4116.759 0.579606145  
## 119 4116.859 0.497530678  
## 120 4116.960 0.429826329  
## 121 4117.060 0.373582397  
## 122 4117.161 0.326539968  
## 123 4117.261 0.286936225  
## 124 4117.362 0.253387031  
## 125 4117.462 0.224798575  
## 126 4117.563 0.200300814  
## 127 4117.663 0.179197178  
## 128 4117.764 0.160926405  
## 129 4117.864 0.145033433  
## 130 4117.965 0.131147072  
## 131 4118.065 0.118962781  
## 132 4118.166 0.108229297  
## 133 4118.266 0.098738201  
## 134 4118.367 0.090315725  
## 135 4118.467 0.082816272  
## 136 4118.568 0.076117276  
## 137 4118.668 0.070115095  
## 138 4118.769 0.064721709  
## 139 4118.869 0.059862061  
## 140 4118.970 0.055471892  
## 141 4119.070 0.051495980  
## 142 4119.171 0.047886702  
## 143 4119.271 0.044602840  
## 144 4119.372 0.041608607  
## 145 4119.472 0.038872830  
## 146 4119.573 0.036368273  
## 147 4119.673 0.034071070  
## 148 4119.774 0.031960248  
## 149 4119.874 0.030017331  
## 150 4119.975 0.028225995  
## 151 4120.075 0.026571783  
## 152 4120.176 0.025041866  
## 153 4120.276 0.023624823  
## 154 4120.377 0.022310475  
## 155 4120.477 0.021089722  
## 156 4120.578 0.019954415  
## 157 4120.678 0.018897244  
## 158 4120.779 0.017911635  
## 159 4120.879 0.016991668  
## 160 4120.980 0.016132001  
## 161 4121.080 0.015327804  
## 162 4121.181 0.014574706  
## 163 4121.281 0.013868741  
## 164 4121.382 0.013206307  
## 165 4121.482 0.012584128  
## 166 4121.583 0.011999218  
## 167 4121.683 0.011448854  
## 168 4121.784 0.010930547  
## 169 4121.884 0.010442021  
## 170 4121.985 0.009981190  
## 171 4122.085 0.009546141  
## 172 4122.186 0.009135116  
## 173 4122.286 0.008746501  
## 174 4122.387 0.008378807  
## 175 4122.487 0.008030665  
## 176 4122.588 0.007700810  
## 177 4122.688 0.007388073  
## 178 4122.789 0.007091375  
## 179 4122.889 0.006809717  
## 180 4122.990 0.006542171  
## 181 4123.090 0.006287881  
## 182 4123.191 0.006046048  
## 183 4123.291 0.005815932  
## 184 4123.392 0.005596845  
## 185 4123.492 0.005388145  
## 186 4123.593 0.005189236  
## 187 4123.693 0.004999561  
## 188 4123.794 0.004818602  
## 189 4123.894 0.004645872  
## 190 4123.995 0.004480920  
## 191 4124.095 0.004323321  
## 192 4124.196 0.004172680  
## 193 4124.296 0.004028624  
## 194 4124.397 0.003890805  
## 195 4124.497 0.003758898  
## 196 4124.598 0.003632596  
## 197 4124.698 0.003511612  
## 198 4124.799 0.003395675  
## 199 4124.899 0.003284531  
## 200 4125.000 0.003177941

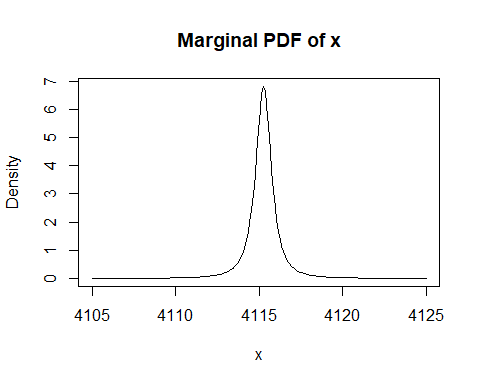
#checks - the sum of each marginal distribution should equal the sum of the whole area.  
sum(marginalX)

## [1] 98.31611

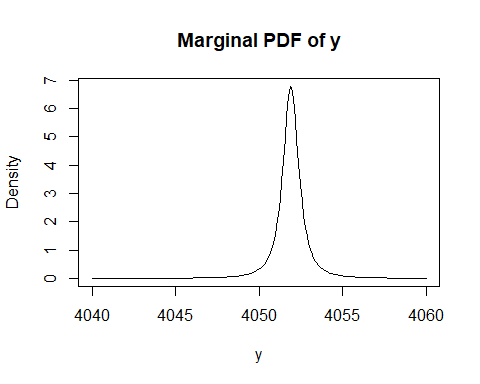
sum(marginalY)

## [1] 98.31611

plot(xRange, marginalX, type = "l", xlab = "x", ylab = "Density", main = "Marginal PDF of x")



plot(yRange, marginalY, type = "l", xlab = "y", ylab = "Density", main = "Marginal PDF of y")

 Obtaining Marginal PDFs - Technique 2

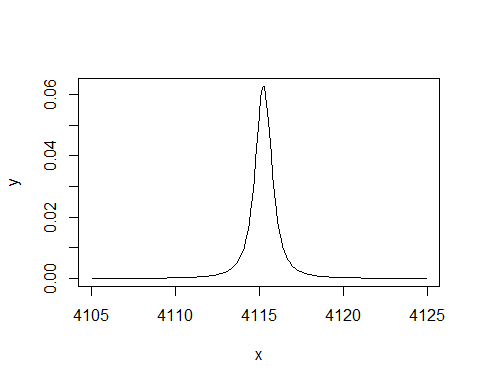
xRange <- seq(4105, 4125, length = 200)  
yRange <- seq(4040, 4060, length = 200)  
  
Rx <- length(xRange) - 1  
Ry <- length(yRange) - 1  
  
marginalX <- numeric(length = Rx)  
for(i in 1:Rx){  
 marginalX[i] <- integral2(Moffat2DPDF, xmin = xRange[i], xmax = xRange[i+1], ymin = 4040, ymax = 4060, parameters = optimPOI2Par)[[1]]  
}  
  
sum(marginalX) #checking it sums to 1 - expect it to be a little less as it isn't summed over infinite (x,y) plain

## [1] 0.9487426

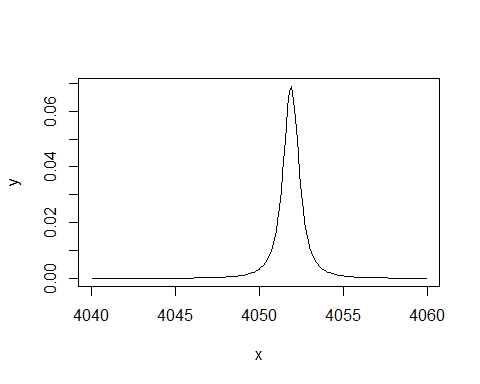
marginalY <- numeric(length = Ry)  
for(i in 1:Ry){  
 marginalY[i] <- integral2(Moffat2DPDF, xmin = 4105, xmax = 4125, ymin = yRange[i], ymax = yRange[i+1], parameters = optimPOI2Par)[[1]]  
}  
  
sum(marginalY) #checking it sums to 1

## [1] 0.9947681

xMids <- numeric(length = Rx)  
for(i in 1:Rx){  
 xMids[i] <- (xRange[i]+xRange[i+1])/2  
}  
yMids <- numeric(length = Ry)  
for(i in 1:Ry){  
 yMids[i] <- (yRange[i]+yRange[i+1])/2  
}  
  
dataframeMarginalX <- data.frame(x = xMids, y = marginalX)  
dataframeMarginalY <- data.frame(x = yMids, y = marginalY)  
  
plot(dataframeMarginalX, type = "l")

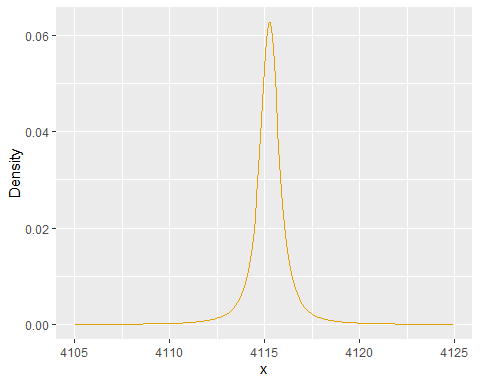


plot(dataframeMarginalY, type = "l")

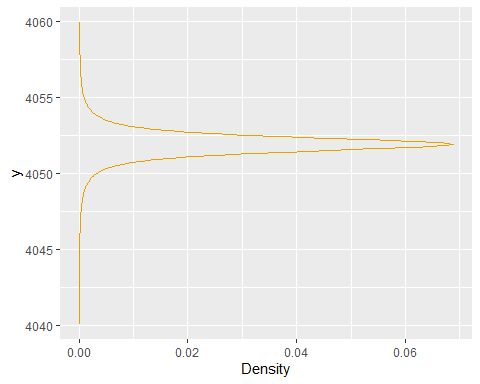


Plotting Marginal PDFs and histograms from the model

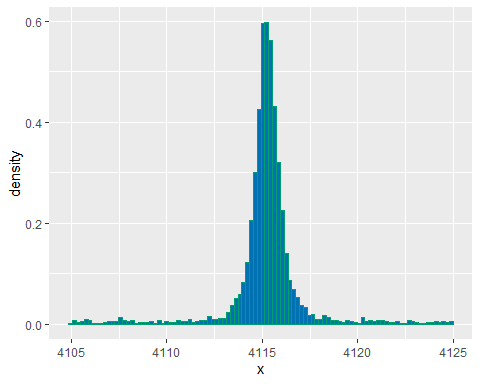
margX <- ggplot(dataframeMarginalX) +  
 geom\_line(aes(x = xMids, y = marginalX), col = cbPalette[2]) +  
 labs(x = "x", y = "Density", main = "Marginal PDF of x")   
  
margY <- ggplot(dataframeMarginalY) +  
 geom\_line(aes(x = yMids, y = marginalY), col = cbPalette[2]) +  
 labs(x = "y", y = "Density", main = "Marginal PDF of y") +  
 rotate()  
  
histX <- ggplot(POI2, aes(x = x)) +  
 geom\_histogram(aes(y = after\_stat(density)), bins = 100, col = cbPalette[3], fill = cbPalette[5])  
  
histY <- ggplot(POI2, aes(x = y)) +  
 geom\_histogram(aes(y = after\_stat(density)), bins = 100, col = cbPalette[3], fill = cbPalette[5]) +   
 rotate()  
  
margX



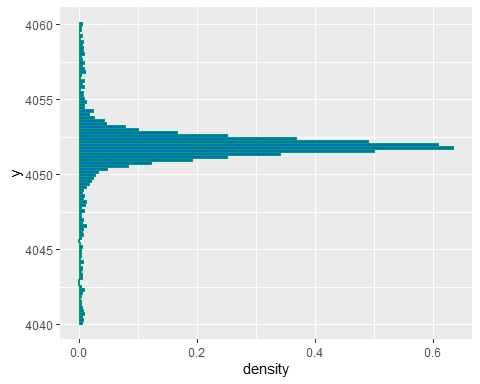
margY



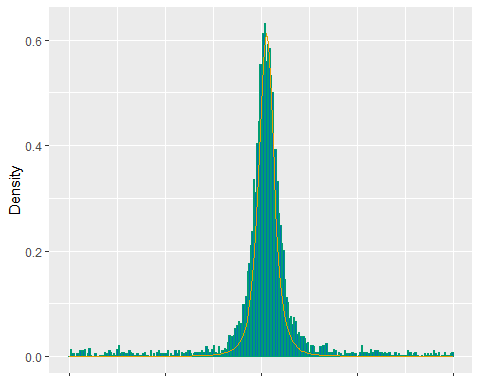
histX



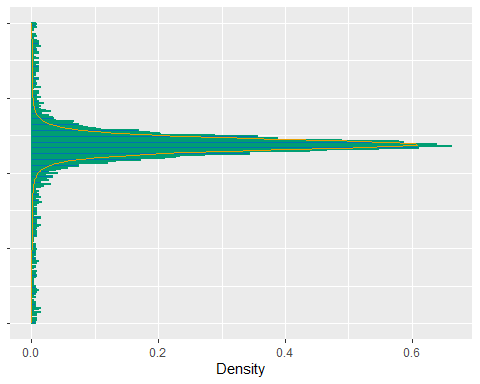
histY



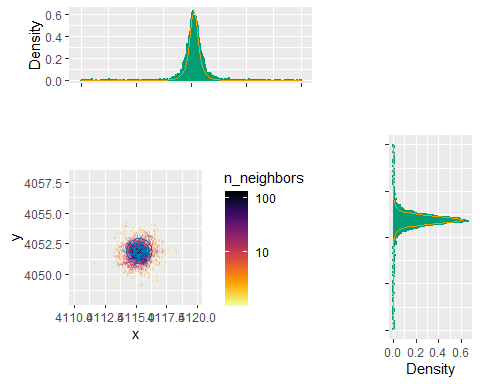
xRange <- seq(4105, 4125, length = 200)  
yRange <- seq(4040, 4060, length = 200)  
xMarg <- numeric(length = 200)  
for(i in 1:200){  
 xMarg[i] <- Moffat2DPDF(optimPOI2Par, xRange[i], optimPOI2Par[2])  
}  
  
xMargdf <- data.frame(xRange, xMarg)  
  
histogramWithMarginalX <- ggplot(POI2, aes(x = x)) +  
 geom\_histogram(aes(y = after\_stat(density)), bins = 250, col = cbPalette[3], fill = cbPalette[5]) +  
 geom\_line(data = xMargdf, aes(x = xRange, y = xMarg), col = cbPalette[2]) +  
 labs(x = NULL, y = "Density") +   
 theme(axis.text.x = element\_blank(), axis.title.x = element\_blank())  
  
histogramWithMarginalX



yMarg <- numeric(length = 200)  
for(i in 1:200){  
 yMarg[i] <- Moffat2DPDF(optimPOI2Par, optimPOI2Par[1], yRange[i])  
}  
  
yMargdf <- data.frame(yRange, yMarg)  
  
histogramWithMarginalY <- ggplot(POI2, aes(x = y)) +  
 geom\_histogram(aes(y = after\_stat(density)), bins = 250, col = cbPalette[3], fill = cbPalette[5]) +  
 geom\_line(data = yMargdf, aes(x = yRange, y = yMarg), col = cbPalette[2]) +  
 labs(x = NULL , y = "Density") +   
 theme(axis.text.y = element\_blank(), axis.title.y = element\_blank()) +  
 rotate()  
   
histogramWithMarginalY

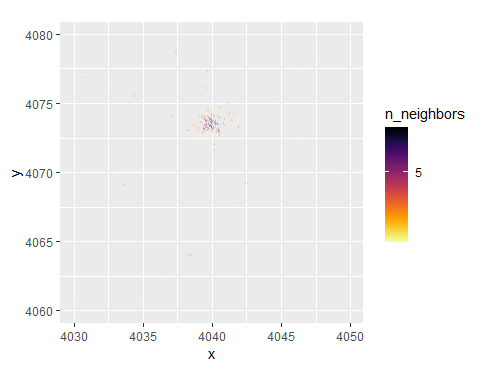


plot\_grid(histogramWithMarginalX, NULL, combined, histogramWithMarginalY, ncol = 2, align = "hv", rel\_widths = c(2, 1), rel\_heights = c(1, 2), axis = c("l", "b"))

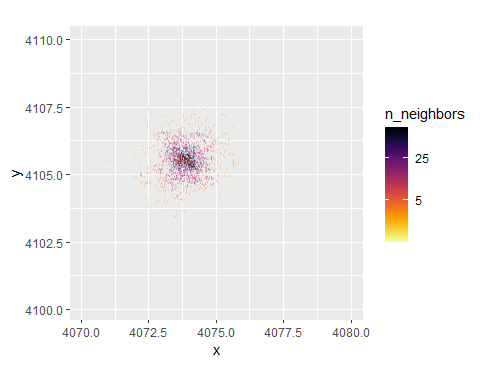


celestialSource1 <- T5 %>%  
 filter(x>4030 & x<4050 & y>4060 & y<4080)  
celestialSource2 <- T5 %>%  
 filter(x>4070 & x<4080 & y>4100 & y<4110)  
celestialSource3 <- T5 %>%  
 filter(x>4070 & x<4080 & y>4150 & y<4160)

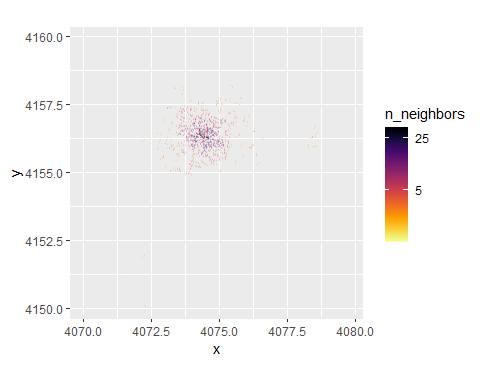
ggplot(celestialSource1, aes(x,y)) +  
 geom\_pointdensity(adjust = 0.05, size = 0.1, shape = "1") +   
 scale\_color\_viridis(direction = -1, option = "B", trans = "log", breaks = c(5, 25, 125)) +   
 coord\_fixed()



ggplot(celestialSource2, aes(x,y)) +  
 geom\_pointdensity(adjust = 0.05, size = 0.1, shape = "1") +   
 scale\_color\_viridis(direction = -1, option = "B", trans = "log", breaks = c(5, 25, 125)) +   
 coord\_fixed()



ggplot(celestialSource3, aes(x,y)) +  
 geom\_pointdensity(adjust = 0.05, size = 0.1, shape = "1") +   
 scale\_color\_viridis(direction = -1, option = "B", trans = "log", breaks = c(5, 25, 125)) +   
 coord\_fixed()



Looking at other sources Celestial Source 1

mean(celestialSource1$x)

## [1] 4039.865

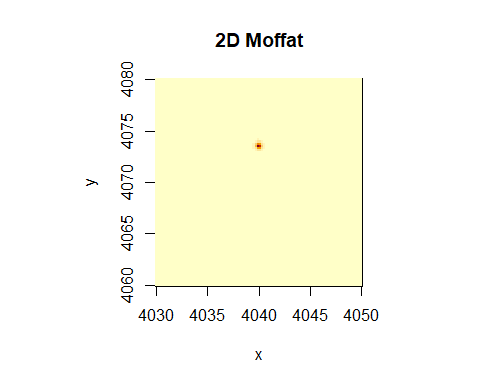
mean(celestialSource1$y)

## [1] 4071.066

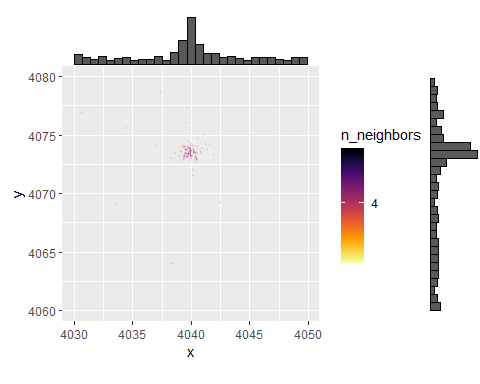
initGuess <- c(x0 = 4039.865, y0 = 4071.066, alpha = 1, beta = 2)  
lowerBounds <- c(0, 0, 1e-6, 1+1e-6)  
  
result2D <- optim(par = initGuess, fn = NLL2D, gr = deriv2DNLL, x = celestialSource1$x, y = celestialSource1$y, method = "L-BFGS-B", lower = lowerBounds)  
optimCS1Par <- result2D$par  
optimCS1Par

## x0 y0 alpha beta   
## 4039.9739489 4073.5419540 0.2196664 1.2197412

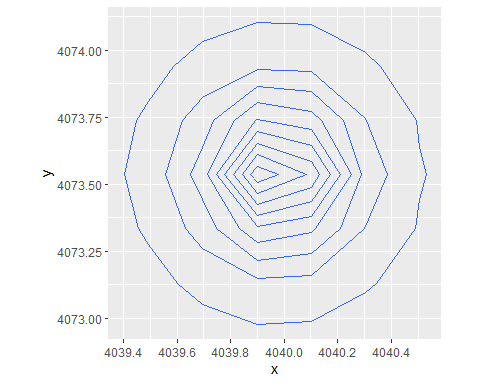
xRange <- seq(4030, 4050, length = 100)  
yRange <- seq(4060, 4080, length = 100)  
grid <- expand.grid(x = xRange, y = yRange)  
  
amplitude <- (optimCS1Par[[4]]-1)/(pi\*optimCS1Par[[3]]^2)  
parameters <- c(amplitude, optimCS1Par)  
  
zCS1<- Moffat2D(grid, parameters)  
zCS1Matrix <- matrix(zCS1, nrow = length(xRange), ncol = length(yRange))  
par(pty = "s")  
image(xRange, yRange, zCS1Matrix, xlab = "x", ylab = "y", main = "2D Moffat")



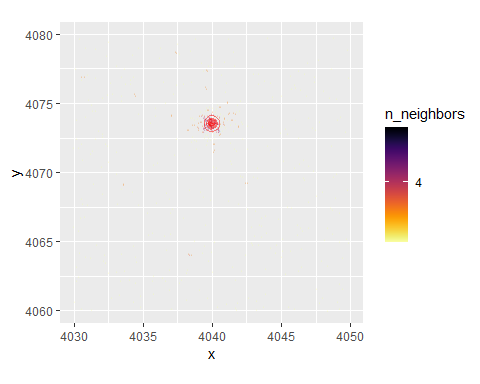
xRange <- seq(4030, 4050, length = 100)  
yRange <- seq(4060, 4080, length = 100)  
datac1 <- expand.grid(x = xRange, y = yRange)  
  
model <- ggplot(celestialSource1, aes(x,y)) +  
 geom\_pointdensity(adjust = 0.05, size = 0.1, shape = "1") +   
 scale\_color\_viridis(direction = -1, option = "B", trans = "log", breaks = c(4, 16, 64)) +   
 coord\_fixed()  
  
contours <- ggplot(datac1, aes(x = x, y = y)) +   
 stat\_contour(aes(z = zCS1), bins = 10) +  
 coord\_fixed()  
  
ggMarginal(model, type = "histogram")



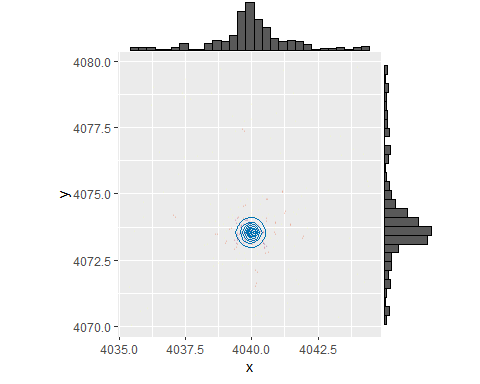
contours



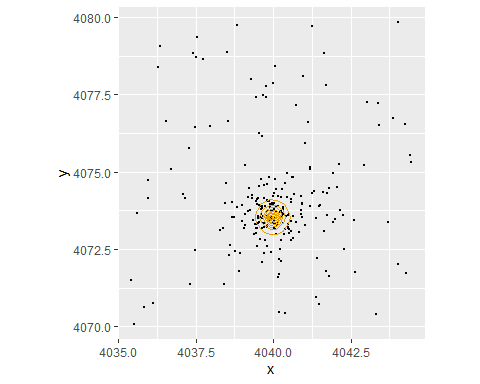
contourData <- ggplot\_build(contours)$data[[1]]  
combined <- model + geom\_path(data = contourData, aes(x = x, y = y, group = group), color = "red", alpha = 0.5)  
combined



#refiltering Celestial Source 1  
CS1ZOOM <- celestialSource1 %>%  
 filter(x>4035 & x<4045 & y>4070 & y<4080)  
  
#creating a point density plot  
modelZoomedInCS1 <- ggplot(CS1ZOOM, aes(x,y)) +  
 geom\_pointdensity(adjust = 0.05, size = 0.1, shape = "1") +   
 scale\_color\_viridis(direction = -1, option = "B", trans = "log", breaks = c(10, 100, 1000)) +   
 coord\_fixed()  
  
contourData <- ggplot\_build(contours)$data[[1]]  
  
#combining the contour plot and the point density plot  
combined <- modelZoomedInCS1 + geom\_path(data = contourData, aes(x = x, y = y, group = group), color = cbPalette[5], alpha = 1, linewidth= 0.3)  
  
#viewing  
ggMarginal(combined, type = "histogram")



#plotting contours on the raw data  
ggplot(CS1ZOOM, aes(x,y)) +  
 geom\_point(size = 0.2) +   
 geom\_path(data = contourData, aes(x = x, y = y, group = group), color = cbPalette[2], alpha = 1) +  
 coord\_fixed()



Celestial Source 2

mean(celestialSource2$x)

## [1] 4073.925

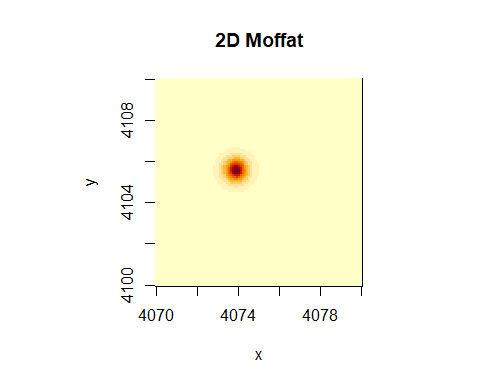
mean(celestialSource2$y)

## [1] 4105.563

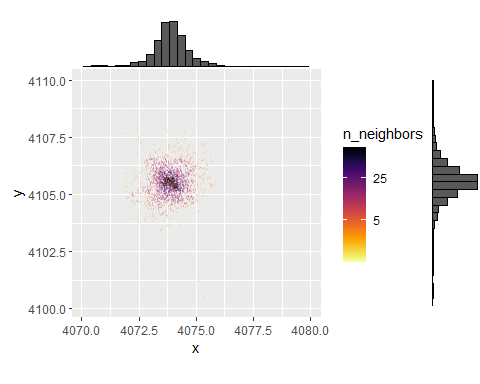
initGuess <- c(x0 = 4073.925, y0 = 4105.563, alpha = 1, beta = 2)  
lowerBounds <- c(0, 0, 1e-6, 1+1e-6)  
  
result2D <- optim(par = initGuess, fn = NLL2D, gr = deriv2DNLL, x = celestialSource2$x, y = celestialSource2$y, method = "L-BFGS-B", lower = lowerBounds)  
optimCS2Par <- result2D$par  
optimCS2Par

## x0 y0 alpha beta   
## 4073.8955778 4105.5827057 0.6865981 1.9625294

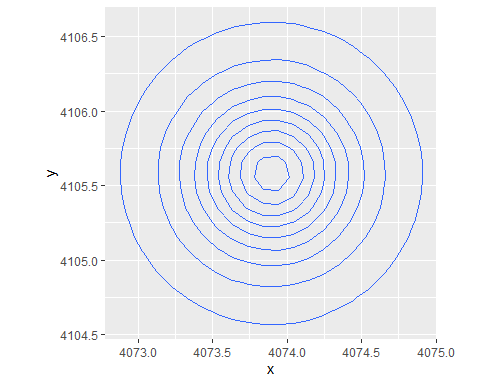
xRange <- seq(4070, 4080, length = 100)  
yRange <- seq(4100, 4110, length = 100)  
grid <- expand.grid(x = xRange, y = yRange)  
  
amplitude <- (optimCS2Par[[4]]-1)/(pi\*optimCS1Par[[3]]^2)  
parameters <- c(amplitude, optimCS2Par)  
  
zCS2<- Moffat2D(grid, parameters)  
zCS2Matrix <- matrix(zCS2, nrow = length(xRange), ncol = length(yRange))  
par(pty = "s")  
image(xRange, yRange, zCS2Matrix, xlab = "x", ylab = "y", main = "2D Moffat")



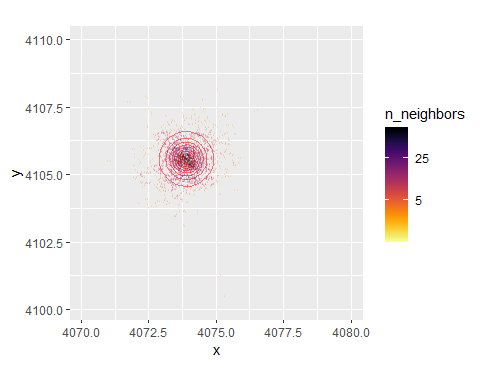
xRange <- seq(4070, 4080, length = 100)  
yRange <- seq(4100, 4110, length = 100)  
datacs2 <- expand.grid(x = xRange, y = yRange)  
  
model <- ggplot(celestialSource2, aes(x,y)) +  
 geom\_pointdensity(adjust = 0.05, size = 0.1, shape = "1") +   
 scale\_color\_viridis(direction = -1, option = "B", trans = "log", breaks = c(5, 25, 125)) +   
 coord\_fixed()  
  
contours <- ggplot(datacs2, aes(x = x, y = y)) +   
 stat\_contour(aes(z = zCS2), bins = 10) +  
 coord\_fixed()  
  
ggMarginal(model, type = "histogram")



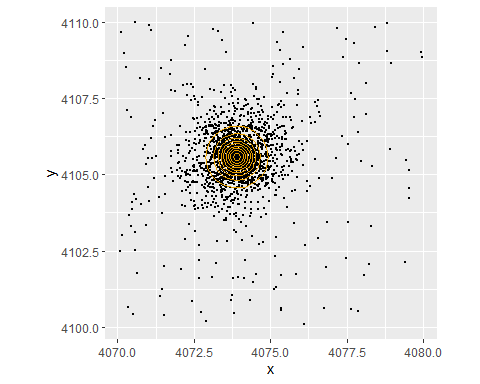
contours



contourData <- ggplot\_build(contours)$data[[1]]  
combined <- model + geom\_path(data = contourData, aes(x = x, y = y, group = group), color = "red", alpha = 0.5)  
combined



#plotting contours on the raw data  
ggplot(celestialSource2, aes(x,y)) +  
 geom\_point(size = 0.2) +   
 geom\_path(data = contourData, aes(x = x, y = y, group = group), color = cbPalette[2], alpha = 1) +  
 coord\_fixed()



Celestial Source 3

mean(celestialSource3$x)

## [1] 4074.631

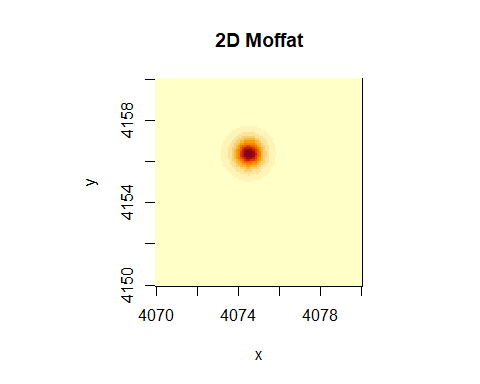
mean(celestialSource3$y)

## [1] 4156.233

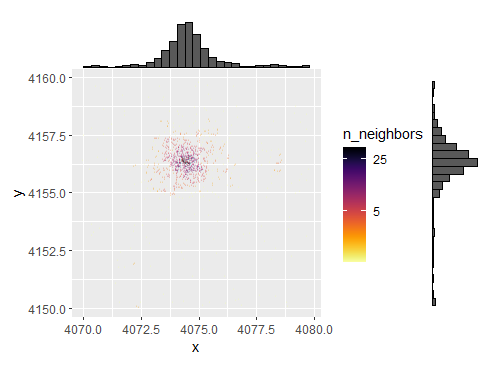
initGuess <- c(x0 = 4074.631, y0 = 4156.233, alpha = 1, beta = 2)  
lowerBounds <- c(0, 0, 1e-6, 1+1e-6)  
  
result2D <- optim(par = initGuess, fn = NLL2D, gr = deriv2DNLL, x = celestialSource3$x, y = celestialSource3$y, method = "L-BFGS-B", lower = lowerBounds)  
optimCS3Par <- result2D$par  
optimCS3Par

## x0 y0 alpha beta   
## 4074.5159021 4156.3826608 0.8414085 1.9560738

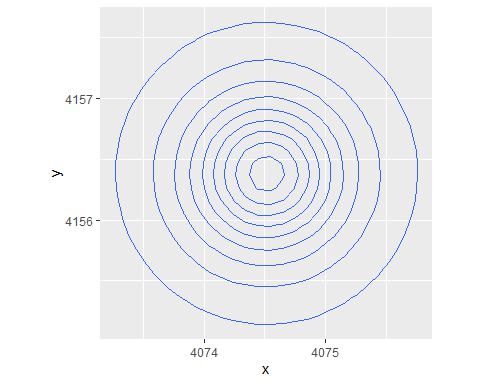
xRange <- seq(4070, 4080, length = 100)  
yRange <- seq(4150, 4160, length = 100)  
grid <- expand.grid(x = xRange, y = yRange)  
  
amplitude <- (optimCS3Par[[4]]-1)/(pi\*optimCS1Par[[3]]^2)  
parameters <- c(amplitude, optimCS3Par)  
  
zCS3<- Moffat2D(grid, parameters)  
zCS3Matrix <- matrix(zCS3, nrow = length(xRange), ncol = length(yRange))  
par(pty = "s")  
image(xRange, yRange, zCS3Matrix, xlab = "x", ylab = "y", main = "2D Moffat")



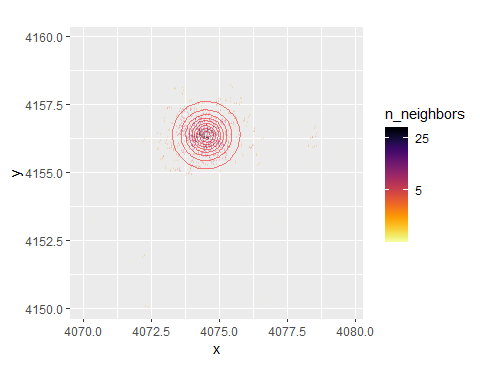
xRange <- seq(4070, 4080, length = 100)  
yRange <- seq(4150, 4160, length = 100)  
datac1 <- expand.grid(x = xRange, y = yRange)  
  
model <- ggplot(celestialSource3, aes(x,y)) +  
 geom\_pointdensity(adjust = 0.05, size = 0.1, shape = "1") +   
 scale\_color\_viridis(direction = -1, option = "B", trans = "log", breaks = c(5, 25, 125)) +   
 coord\_fixed()  
  
contours <- ggplot(datac1, aes(x = x, y = y)) +   
 stat\_contour(aes(z = zCS3), bins = 10) +  
 coord\_fixed()  
  
ggMarginal(model, type = "histogram")



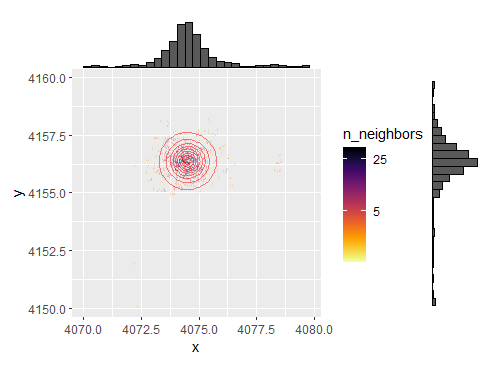
contours



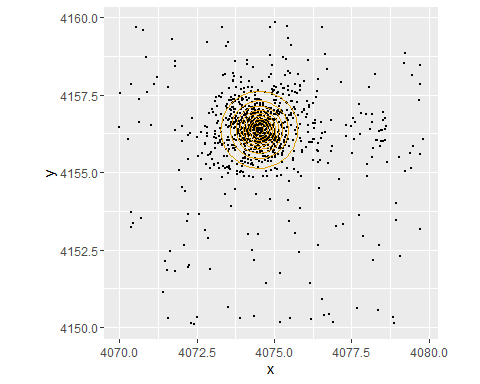
contourData <- ggplot\_build(contours)$data[[1]]  
combined <- model + geom\_path(data = contourData, aes(x = x, y = y, group = group), color = "red", alpha = 0.5)  
combined



#viewing  
ggMarginal(combined, type = "histogram")



#plotting contours on the raw data  
ggplot(celestialSource3, aes(x,y)) +  
 geom\_point(size = 0.2) +   
 geom\_path(data = contourData, aes(x = x, y = y, group = group), color = cbPalette[2], alpha = 1) +  
 coord\_fixed()



Modelled Areas

ggplot(T5,aes(x,y)) +   
 geom\_pointdensity(adjust = 0.05, size = 0.1, shape = "1") +   
 scale\_color\_viridis(direction = -1, option = "B", trans = "log", breaks = c(10, 100, 1000, 10000)) +   
 coord\_fixed() +   
 #POI1  
 geom\_segment(aes(x = 4160, y = 4180, xend = 4180, yend = 4180)) +  
 geom\_segment(aes(x = 4160, y = 4200, xend = 4180, yend = 4200)) +  
 geom\_segment(aes(x = 4160, y = 4180, xend = 4160, yend = 4200)) +  
 geom\_segment(aes(x = 4180, y = 4180, xend = 4180, yend = 4200)) +   
 #POI2  
 geom\_segment(aes(x = 4105, y = 4040, xend = 4125, yend = 4040)) +  
 geom\_segment(aes(x = 4105, y = 4060, xend = 4125, yend = 4060)) +  
 geom\_segment(aes(x = 4105, y = 4040, xend = 4105, yend = 4060)) +  
 geom\_segment(aes(x = 4125, y = 4040, xend = 4125, yend = 4060)) +   
 #CS1  
 geom\_segment(aes(x = 4030, y = 4060, xend = 4050, yend = 4060)) +  
 geom\_segment(aes(x = 4030, y = 4080, xend = 4050, yend = 4080)) +  
 geom\_segment(aes(x = 4030, y = 4060, xend = 4030, yend = 4080)) +  
 geom\_segment(aes(x = 4050, y = 4060, xend = 4050, yend = 4080)) +   
 #CS2  
 geom\_segment(aes(x = 4070, y = 4100, xend = 4080, yend = 4100)) +  
 geom\_segment(aes(x = 4070, y = 4110, xend = 4080, yend = 4110)) +  
 geom\_segment(aes(x = 4070, y = 4100, xend = 4070, yend = 4110)) +  
 geom\_segment(aes(x = 4080, y = 4100, xend = 4080, yend = 4110)) +   
 #CS3  
 geom\_segment(aes(x = 4070, y = 4150, xend = 4080, yend = 4150)) +  
 geom\_segment(aes(x = 4070, y = 4160, xend = 4080, yend = 4160)) +  
 geom\_segment(aes(x = 4070, y = 4150, xend = 4070, yend = 4160)) +  
 geom\_segment(aes(x = 4080, y = 4150, xend = 4080, yend = 4160))

## Warning in geom\_segment(aes(x = 4160, y = 4180, xend = 4180, yend = 4180)): All aesthetics have length 1, but the data has 114699 rows.  
## ℹ Did you mean to use `annotate()`?

## Warning in geom\_segment(aes(x = 4160, y = 4200, xend = 4180, yend = 4200)): All aesthetics have length 1, but the data has 114699 rows.  
## ℹ Did you mean to use `annotate()`?

## Warning in geom\_segment(aes(x = 4160, y = 4180, xend = 4160, yend = 4200)): All aesthetics have length 1, but the data has 114699 rows.  
## ℹ Did you mean to use `annotate()`?

## Warning in geom\_segment(aes(x = 4180, y = 4180, xend = 4180, yend = 4200)): All aesthetics have length 1, but the data has 114699 rows.  
## ℹ Did you mean to use `annotate()`?

## Warning in geom\_segment(aes(x = 4105, y = 4040, xend = 4125, yend = 4040)): All aesthetics have length 1, but the data has 114699 rows.  
## ℹ Did you mean to use `annotate()`?

## Warning in geom\_segment(aes(x = 4105, y = 4060, xend = 4125, yend = 4060)): All aesthetics have length 1, but the data has 114699 rows.  
## ℹ Did you mean to use `annotate()`?

## Warning in geom\_segment(aes(x = 4105, y = 4040, xend = 4105, yend = 4060)): All aesthetics have length 1, but the data has 114699 rows.  
## ℹ Did you mean to use `annotate()`?

## Warning in geom\_segment(aes(x = 4125, y = 4040, xend = 4125, yend = 4060)): All aesthetics have length 1, but the data has 114699 rows.  
## ℹ Did you mean to use `annotate()`?

## Warning in geom\_segment(aes(x = 4030, y = 4060, xend = 4050, yend = 4060)): All aesthetics have length 1, but the data has 114699 rows.  
## ℹ Did you mean to use `annotate()`?

## Warning in geom\_segment(aes(x = 4030, y = 4080, xend = 4050, yend = 4080)): All aesthetics have length 1, but the data has 114699 rows.  
## ℹ Did you mean to use `annotate()`?

## Warning in geom\_segment(aes(x = 4030, y = 4060, xend = 4030, yend = 4080)): All aesthetics have length 1, but the data has 114699 rows.  
## ℹ Did you mean to use `annotate()`?

## Warning in geom\_segment(aes(x = 4050, y = 4060, xend = 4050, yend = 4080)): All aesthetics have length 1, but the data has 114699 rows.  
## ℹ Did you mean to use `annotate()`?

## Warning in geom\_segment(aes(x = 4070, y = 4100, xend = 4080, yend = 4100)): All aesthetics have length 1, but the data has 114699 rows.  
## ℹ Did you mean to use `annotate()`?

## Warning in geom\_segment(aes(x = 4070, y = 4110, xend = 4080, yend = 4110)): All aesthetics have length 1, but the data has 114699 rows.  
## ℹ Did you mean to use `annotate()`?

## Warning in geom\_segment(aes(x = 4070, y = 4100, xend = 4070, yend = 4110)): All aesthetics have length 1, but the data has 114699 rows.  
## ℹ Did you mean to use `annotate()`?

## Warning in geom\_segment(aes(x = 4080, y = 4100, xend = 4080, yend = 4110)): All aesthetics have length 1, but the data has 114699 rows.  
## ℹ Did you mean to use `annotate()`?

## Warning in geom\_segment(aes(x = 4070, y = 4150, xend = 4080, yend = 4150)): All aesthetics have length 1, but the data has 114699 rows.  
## ℹ Did you mean to use `annotate()`?

## Warning in geom\_segment(aes(x = 4070, y = 4160, xend = 4080, yend = 4160)): All aesthetics have length 1, but the data has 114699 rows.  
## ℹ Did you mean to use `annotate()`?

## Warning in geom\_segment(aes(x = 4070, y = 4150, xend = 4070, yend = 4160)): All aesthetics have length 1, but the data has 114699 rows.  
## ℹ Did you mean to use `annotate()`?

## Warning in geom\_segment(aes(x = 4080, y = 4150, xend = 4080, yend = 4160)): All aesthetics have length 1, but the data has 114699 rows.  
## ℹ Did you mean to use `annotate()`?

