

20240408_synbot_puncta_processor

Justin Savage

2024-04-08

```
#author Justin Savage
#js664@duke.edu
#Version 1.0
#8/16/21

#This script is for combining the data from the coloc_results file output of the
# Syn_Bot ImageJ Macro. You should be able to simply select the entire script (Ctrl/Cmnd + A)
# and click the Run button in the top right corner. You will then select the same folder
# that was selected for the Syn_Bot run. This will read in the data and write a
# summary.csv file similar to the one that is made by Syn_Bot itself.

#Further analysis can be performed on the data from individual images using similar
# techniques to those shown here. Those unfamiliar with R are encouraged to learn
# more about it at https://r4ds.had.co.nz/

#loaded RStudioAPI to use for selecting working directory
if("rstudioapi" %in% rownames(installed.packages()) == FALSE)
{install.packages("rstudioapi")}
library(rstudioapi)
if("xlsx" %in% rownames(installed.packages()) == FALSE)
{install.packages("xlsx")}
library(xlsx)
if("reshape" %in% rownames(installed.packages()) == FALSE)
{install.packages("reshape")}
library(reshape)
if("data.table" %in% rownames(installed.packages()) == FALSE)
{install.packages("data.table")}
library(data.table)

##
## Attaching package: 'data.table'

## The following object is masked from 'package:reshape':
##
##      melt

if("stringi" %in% rownames(installed.packages()) == FALSE)
{install.packages("stringi")}
library(stringi)
if("tidyverse" %in% rownames(installed.packages()) == FALSE)
{install.packages("tidyverse")}
library(tidyverse)

## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
```

```
## v dplyr      1.1.4      v readr      2.1.5
## v forcats   1.0.0      v stringr   1.5.1
## v ggplot2    3.5.0      v tibble    3.2.1
## v lubridate  1.9.3      v tidyr     1.3.1
## v purrr      1.0.2

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::between()      masks data.table::between()
## x tidyr::expand()       masks reshape::expand()
## x dplyr::filter()       masks stats::filter()
## x dplyr::first()        masks data.table::first()
## x lubridate::hour()     masks data.table::hour()
## x lubridate::isoweek()  masks data.table::isoweek()
## x dplyr::lag()          masks stats::lag()
## x dplyr::last()         masks data.table::last()
## x lubridate::mday()     masks data.table::mday()
## x lubridate::minute()   masks data.table::minute()
## x lubridate::month()    masks data.table::month()
## x lubridate::quarter()  masks data.table::quarter()
## x dplyr::rename()       masks reshape::rename()
## x lubridate::second()   masks data.table::second()
## x lubridate::stamp()    masks reshape::stamp()
## x purrr::transpose()    masks data.table::transpose()
## x lubridate::wday()     masks data.table::wday()
## x lubridate::week()     masks data.table::week()
## x lubridate::yday()     masks data.table::yday()
## x lubridate::year()     masks data.table::year()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
```

```
dataFile <- selectDirectory(
  caption = "Select Directory",
  label = "Select",
  path = getActiveProject()
)
setwd(dataFile)

allFiles <- dir()
imageIndices <- 0

for (i in 1:length(allFiles)){
  currentFile <- allFiles[i]
  imageIndices <- append(imageIndices, i)
}

for(i in 2:length(imageIndices)){
  if(i == 2){
    imageFiles <- allFiles[imageIndices[i]]
  }
  if(i != 2){
    imageFiles <- append(imageFiles, allFiles[imageIndices[i]])
  }
}
```

```

}

#imageFiles is all folders that aren't the key
for(i in 1:length(imageFiles)) {
  currentFile <- imageFiles[i]
  pathOutput <- paste(currentFile, 'Output', sep = "/")
  if (i == 1) {
    fileList <- dir(pathOutput)
    for(j in 1:length(fileList)){
      fileList[j] <- paste(pathOutput, fileList[j], sep = "/")
    }
  }
  if (i != 1) {
    newFiles <- dir(pathOutput)
    for(j in 1:length(newFiles)){
      newFiles[j] <- paste(pathOutput, newFiles[j], sep = "/")
    }
    fileList <- append(fileList, newFiles)
  }
}

getwd()

## [1] "C:/Users/Justin/Desktop/inside_only_cleared_for_synbot"

#redFiles and greenFiles stores the names of images
redFiles <- str_subset(fileList, pattern = 'redResults')
greenFiles <- str_subset(fileList, pattern = 'greenResults')
colocFiles <- str_subset(fileList, pattern = 'colocResults')
redDataIn <- lapply(redFiles, read.csv)
greenDataIn <- lapply(greenFiles, read.csv)
colocDataIn <- lapply(colocFiles, read.csv)

for (i in 1:length(redDataIn)){
  currentData <- redDataIn[[i]]
  #Removes all columns but puncta number, Area, X, Y
  redDataIn[[i]] <- currentData[,c(2,3,6,7)]
  redDataIn[[i]]$Image <- i
  setorder(redDataIn[[i]],Image,X)
}

for (i in 1:length(greenDataIn)){
  currentData <- greenDataIn[[i]]
  #Removes all columns but puncta number, Area, X, Y
  greenDataIn[[i]] <- currentData[,c(2,3,6,7)]
  greenDataIn[[i]]$Image <- i
  setorder(greenDataIn[[i]],Image,X)
}

for (i in 1:length(colocDataIn)){
  currentData <- colocDataIn[i]
  colocDataIn[[i]]$Image <- i
  setorder(colocDataIn[[i]],Image,X)
}

```

```

for (i in 1:length(colocDataIn)) {
  currentColocData <- colocDataIn[[i]]
  colocTot <- length(currentColocData$X)
  if (i == 1){
    colocVector <- c(colocTot)
  }
  if (i != 1){
    colocVector <- append(colocVector, colocTot)
  }
}

# #not working, probably because not every image has colocs

# colocData <- tibble(
#   Image = redFiles,
#   ColocCount = colocVector
# )
#
# #also count red puncta
#
# for (i in 1:length(redDataIn)) {
#   currentRedData <- redDataIn[[i]]
#   redTot <- length(currentRedData$X)
#   if (i == 1){
#     redVector <- c(redTot)
#   }
#   if (i != 1){
#     redVector <- append(redVector, redTot)
#   }
# }
#
# colocData <- colocData %>% add_column(RedCount = redVector)
# colocData$RedCount = redVector
#
# #also count green puncta
#
# for (i in 1:length(greenDataIn)) {
#   currentGreenData <- greenDataIn[[i]]
#   greenTot <- length(currentGreenData$X)
#   if (i == 1){
#     greenVector <- c(greenTot)
#   }
#   if (i != 1){
#     greenVector <- append(greenVector, greenTot)
#   }
# }
#
# colocData <- colocData %>% add_column(GreenCount = greenVector)
# colocData$GreenCount = greenVector
# write.csv(colocData, file="summary.csv")

```

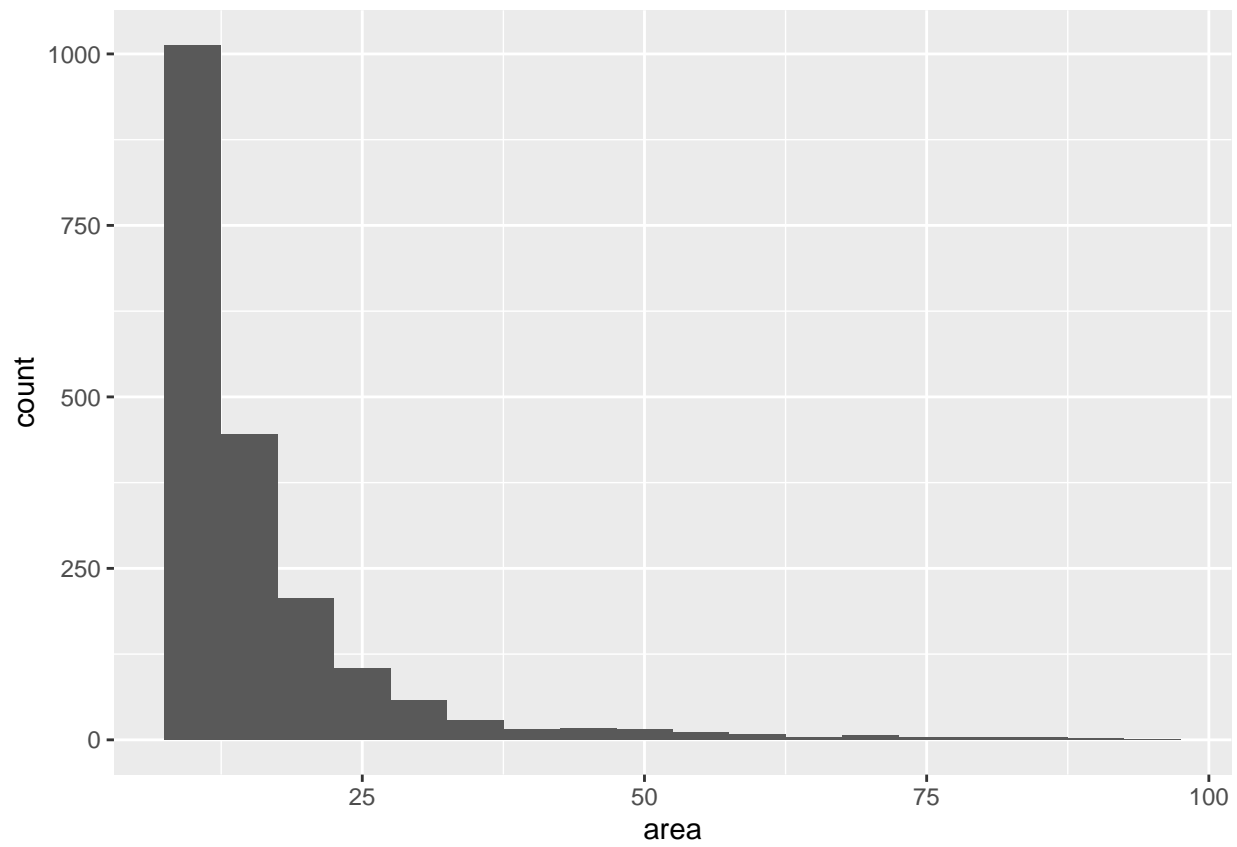
```

GScount <- 0
WTcount <- 0
#get red puncta area by condition
for (i in 1:length(redFiles)){
  currentImage <- redFiles[i]
  currentAreas <- redDataIn[[i]]$Area
  if (grepl("GS", currentImage)){
    currentCondition <- "GS"
    if (GScount == 0){
      redAreaVector_GS <- currentAreas
    }
    if (GScount > 0){
      redAreaVector_GS <- append(redAreaVector_GS, currentAreas)
    }
    GScount <- GScount + 1
  }
  else{
    currentCondition <- "WT"
    if (WTcount == 0){
      redAreaVector_WT <- currentAreas
    }
    if (WTcount > 0){
      redAreaVector_WT <- append(redAreaVector_WT, currentAreas)
    }
    WTcount <- WTcount + 1
  }
}

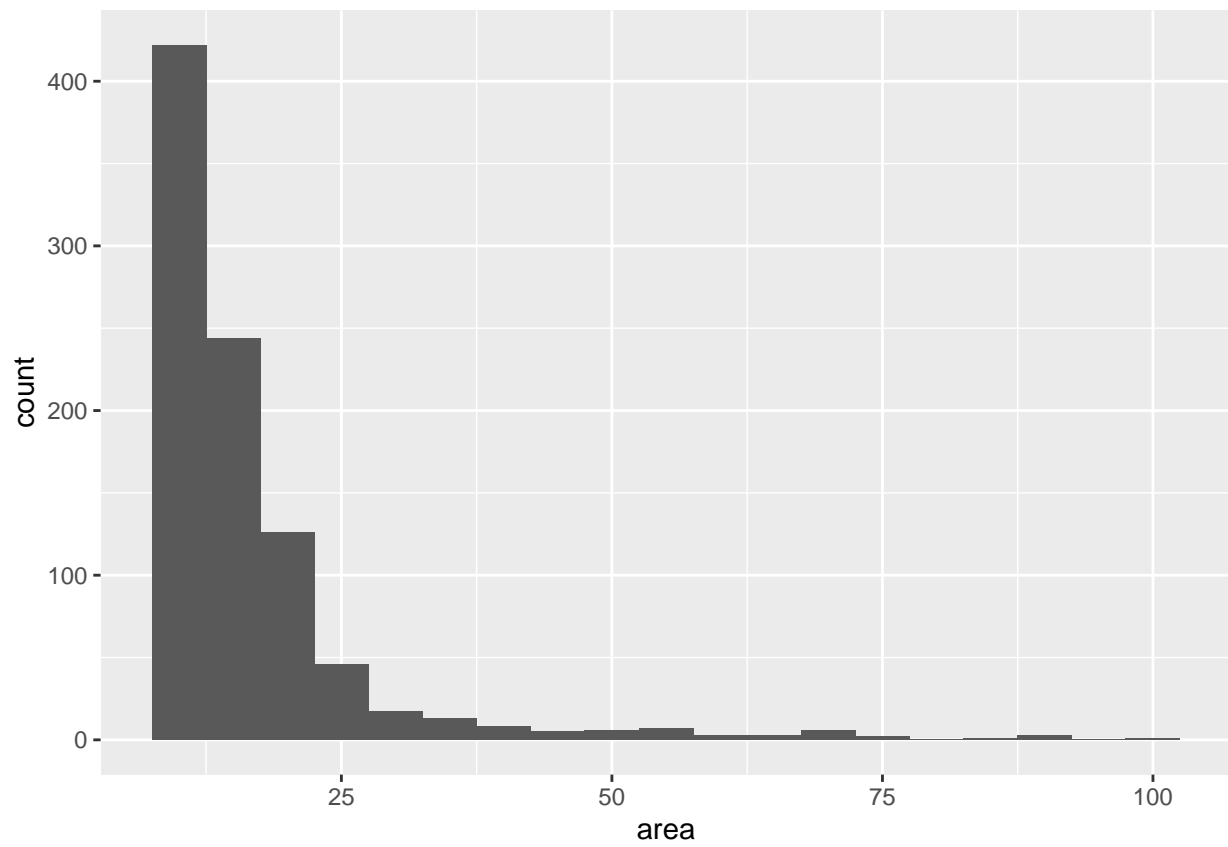
redArea_GS.df <- data.frame(redAreaVector_GS)
colnames(redArea_GS.df) <- c("area")
redArea_WT.df <- data.frame(redAreaVector_WT)
colnames(redArea_WT.df) <- c("area")

ggplot(data = redArea_GS.df) + geom_histogram(aes(x = area), binwidth = 5)

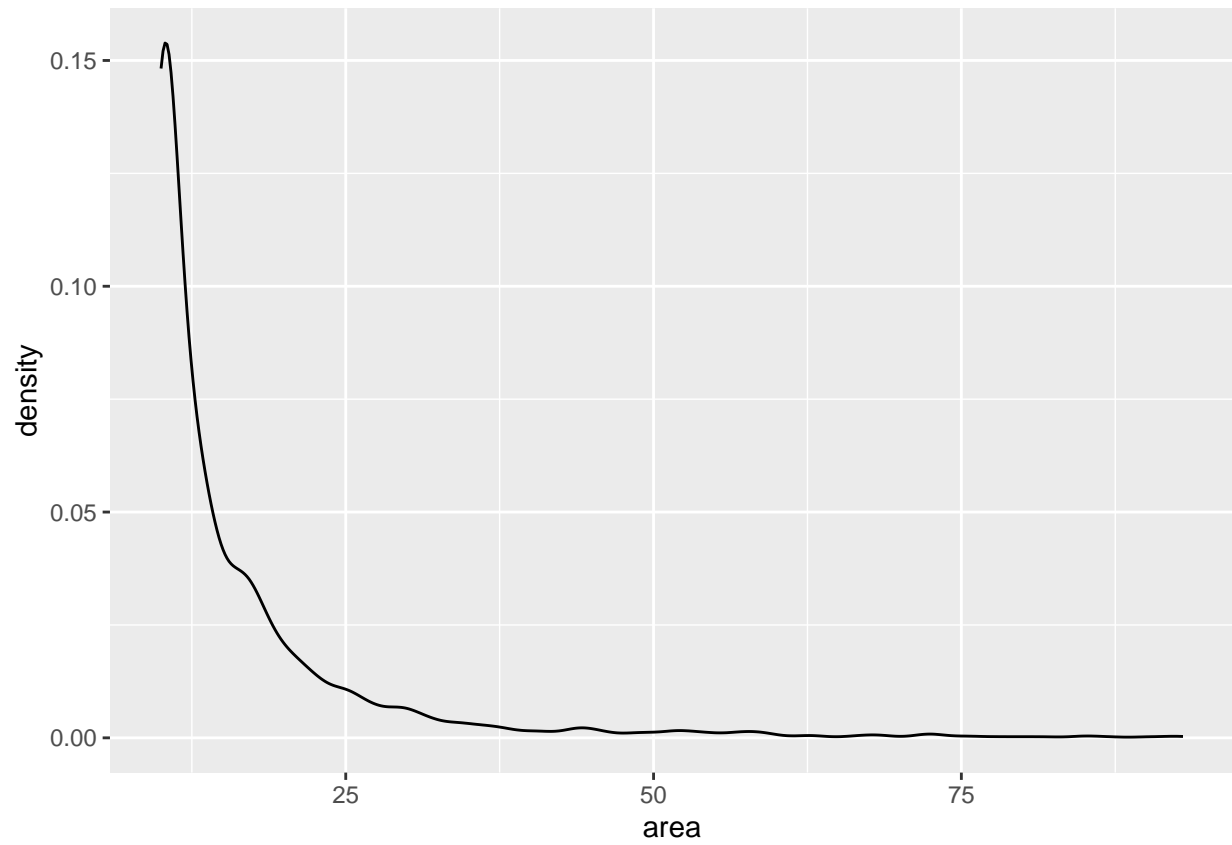
```



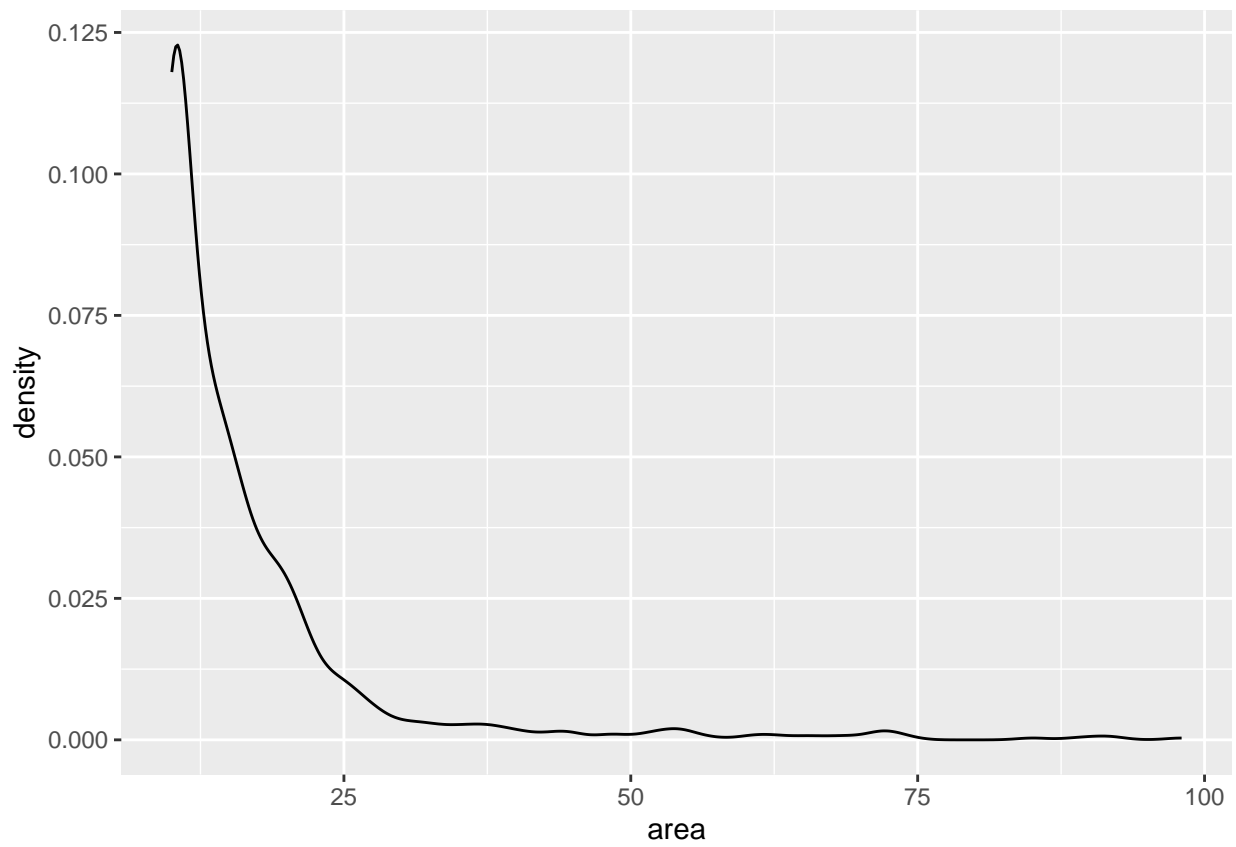
```
ggplot(data = redArea_WT.df) + geom_histogram(aes(x = area), binwidth = 5)
```



```
ggplot(data = redArea_GS.df) + geom_density(aes(x = area))
```



```
ggplot(data = redArea_WT.df) + geom_density(aes(x = area))
```

```
ggplot(data = redArea_GS.df) + geom_density(aes(x = area), fill = "magenta") + geom_density(data = redA
```



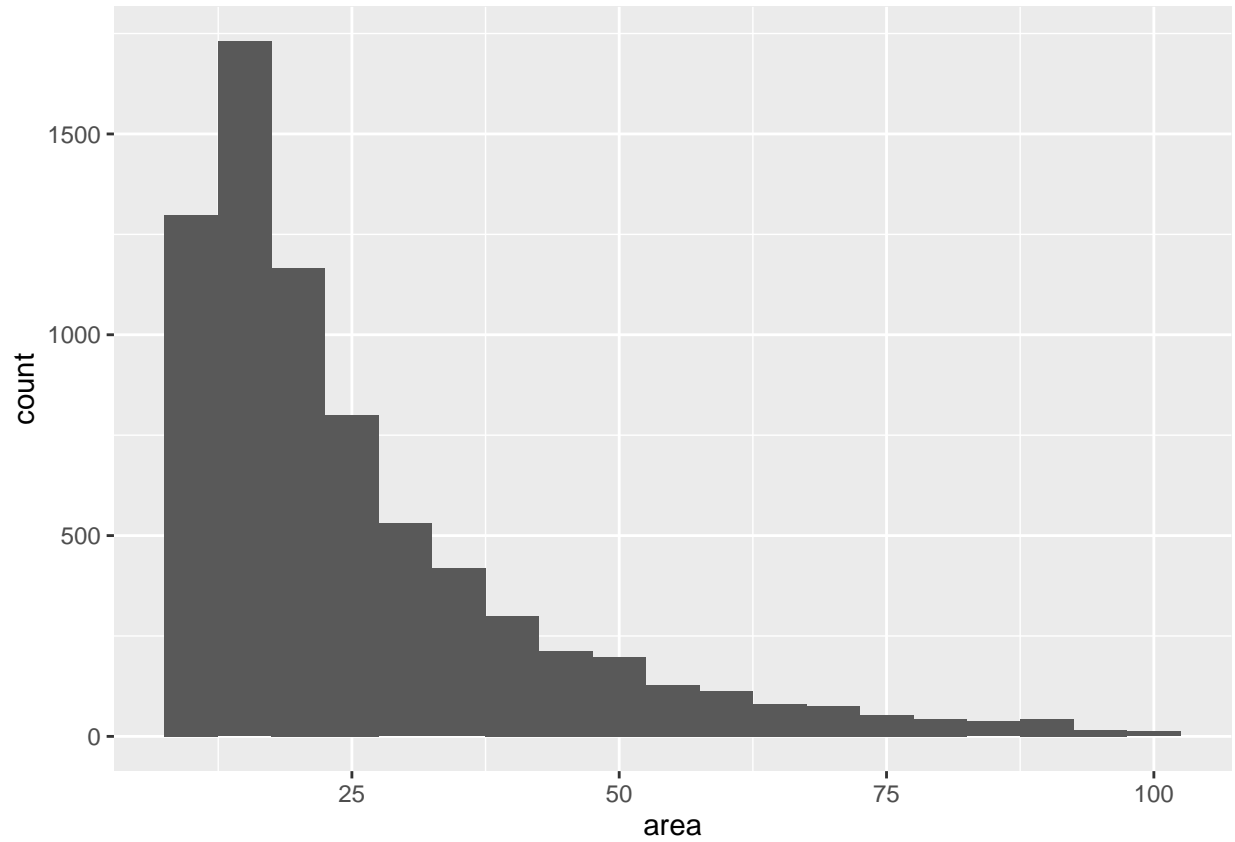
```

GScount <- 0
WTcount <- 0
#get green puncta area by condition
for (i in 1:length(greenFiles)){
  currentImage <- greenFiles[i]
  currentAreas <- greenDataIn[[i]]$Area
  if (grepl("GS", currentImage)){
    currentCondition <- "GS"
    if (GScount == 0){
      greenAreaVector_GS <- currentAreas
    }
    if (GScount > 0){
      greenAreaVector_GS <- append(greenAreaVector_GS, currentAreas)
    }
    GScount <- GScount + 1
  }
  else{
    currentCondition <- "WT"
    if (WTcount == 0){
      greenAreaVector_WT <- currentAreas
    }
    if (WTcount > 0){
      greenAreaVector_WT <- append(greenAreaVector_WT, currentAreas)
    }
    WTcount <- WTcount + 1
  }
}

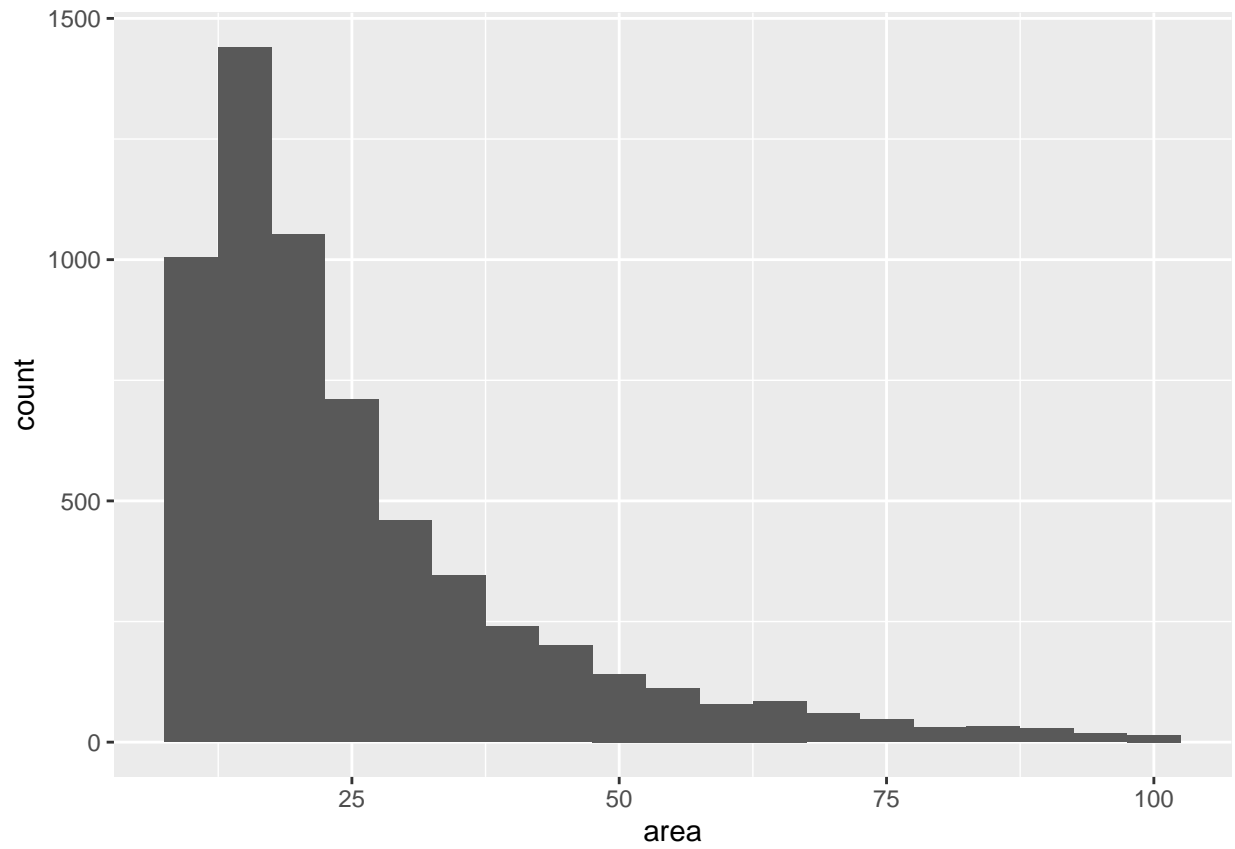
```

```
greenArea_GS.df <- data.frame(greenAreaVector_GS)
colnames(greenArea_GS.df) <- c("area")
greenArea_WT.df <- data.frame(greenAreaVector_WT)
colnames(greenArea_WT.df) <- c("area")

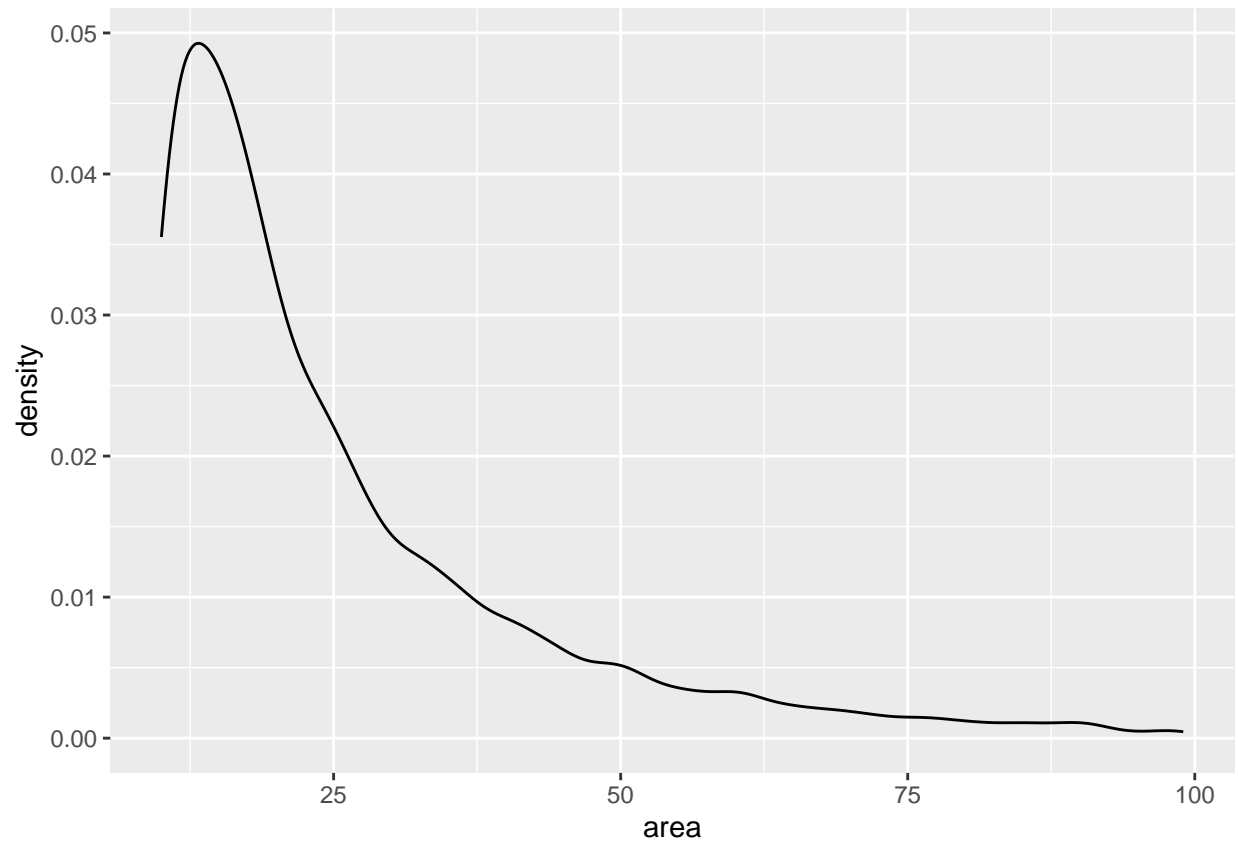
ggplot(data = greenArea_GS.df) + geom_histogram(aes(x = area), binwidth = 5)
```



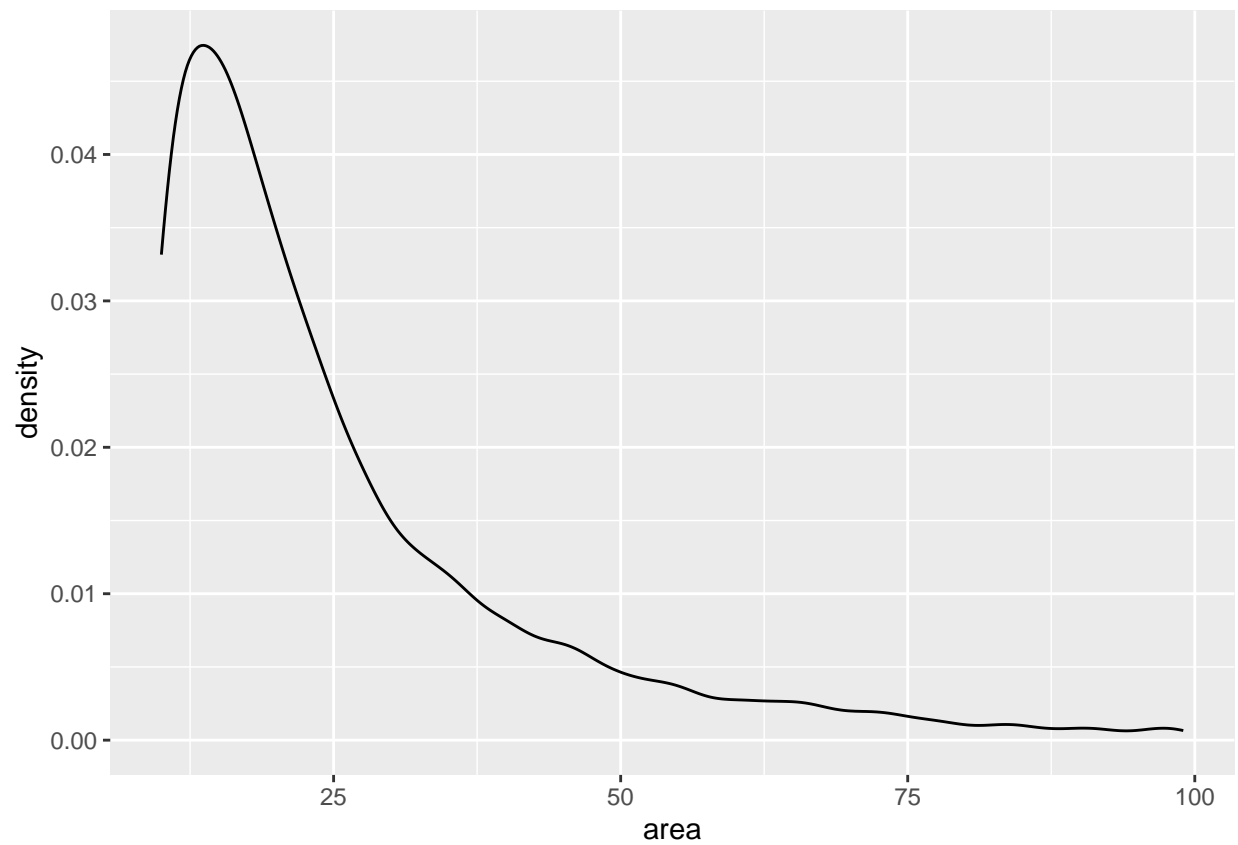
```
ggplot(data = greenArea_WT.df) + geom_histogram(aes(x = area), binwidth = 5)
```



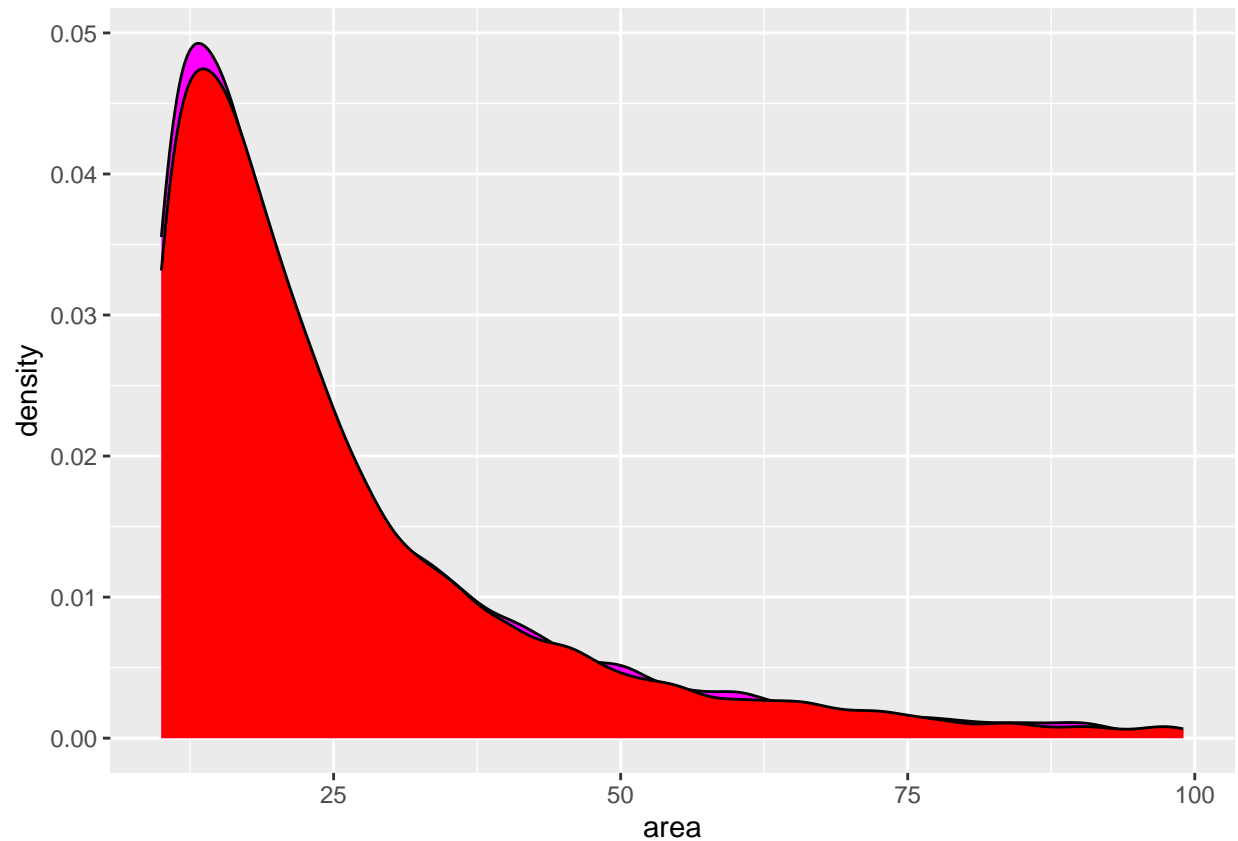
```
ggplot(data = greenArea_GS.df) + geom_density(aes(x = area))
```



```
ggplot(data = greenArea_WT.df) + geom_density(aes(x = area))
```



```
ggplot(data = greenArea_GS.df) + geom_density(aes(x = area), fill = "magenta") + geom_density(data = gr
```



```

GScount <- 0
WTcount <- 0
#get coloc puncta area by condition
for (i in 1:length(colocFiles)){
  currentImage <- colocFiles[i]
  currentAreas <- colocDataIn[[i]]$Area
  if (grepl("GS", currentImage)){
    currentCondition <- "GS"
    if (GScount == 0){
      colocAreaVector_GS <- currentAreas
    }
    if (GScount > 0){
      colocAreaVector_GS <- append(colocAreaVector_GS, currentAreas)
    }
    GScount <- GScount + 1
  }
  else{
    currentCondition <- "WT"
    if (WTcount == 0){
      colocAreaVector_WT <- currentAreas
    }
    if (WTcount > 0){
      colocAreaVector_WT <- append(colocAreaVector_WT, currentAreas)
    }
    WTcount <- WTcount + 1
  }
}

```

```

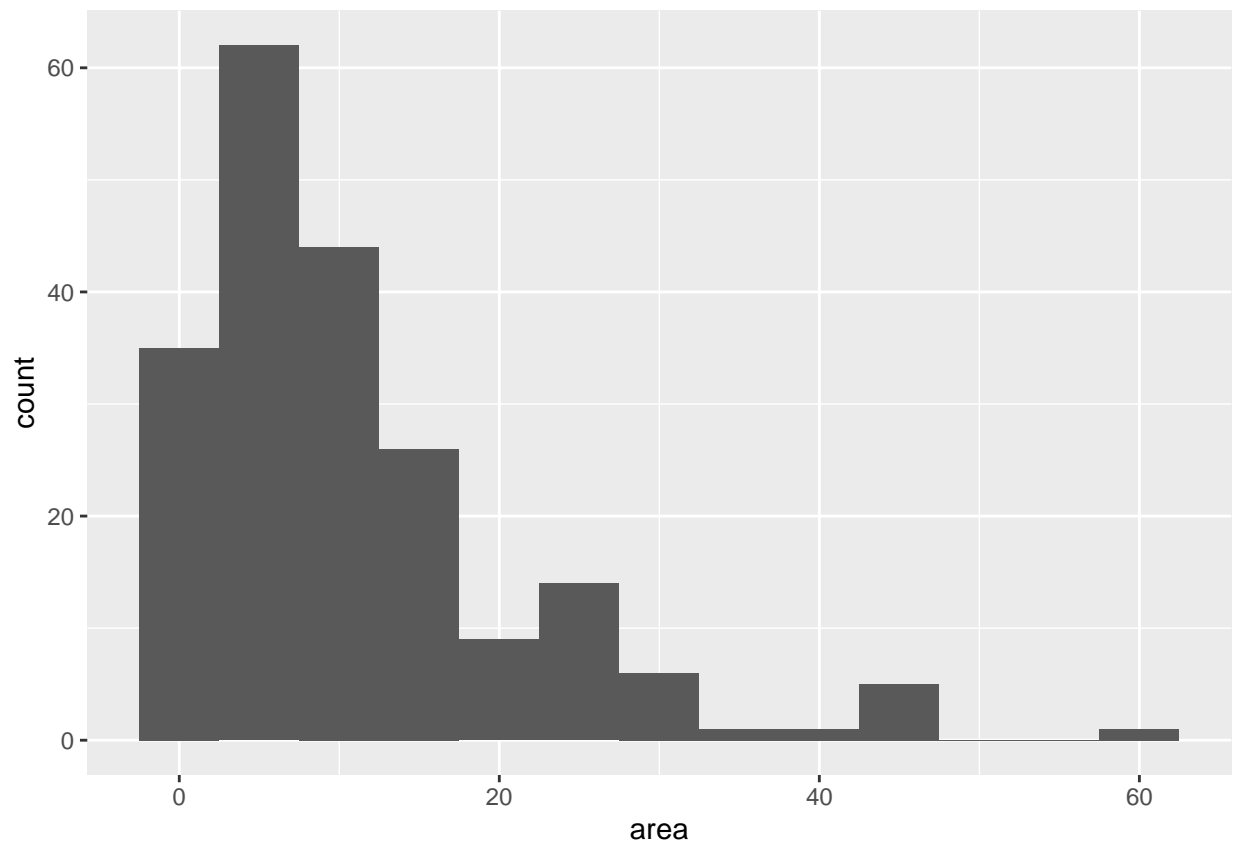
}

colocArea_GS.df <- data.frame(colocAreaVector_GS)
colnames(colocArea_GS.df) <- c("area")
colocArea_WT.df <- data.frame(colocAreaVector_WT)
colnames(colocArea_WT.df) <- c("area")

colocArea_GS.df$genotype <- "GS"
colocArea_WT.df$genotype <- "WT"

ggplot(data = colocArea_GS.df) + geom_histogram(aes(x = area), binwidth = 5)

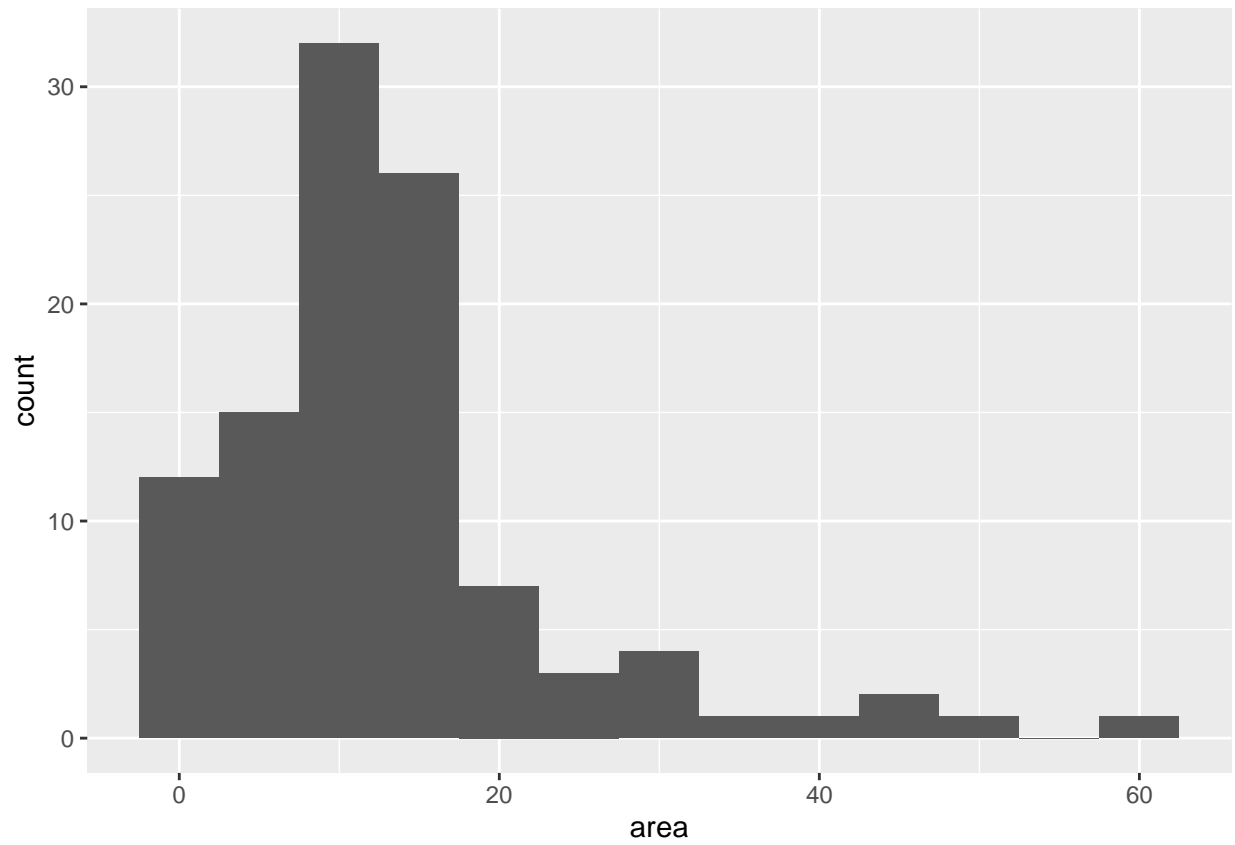
```



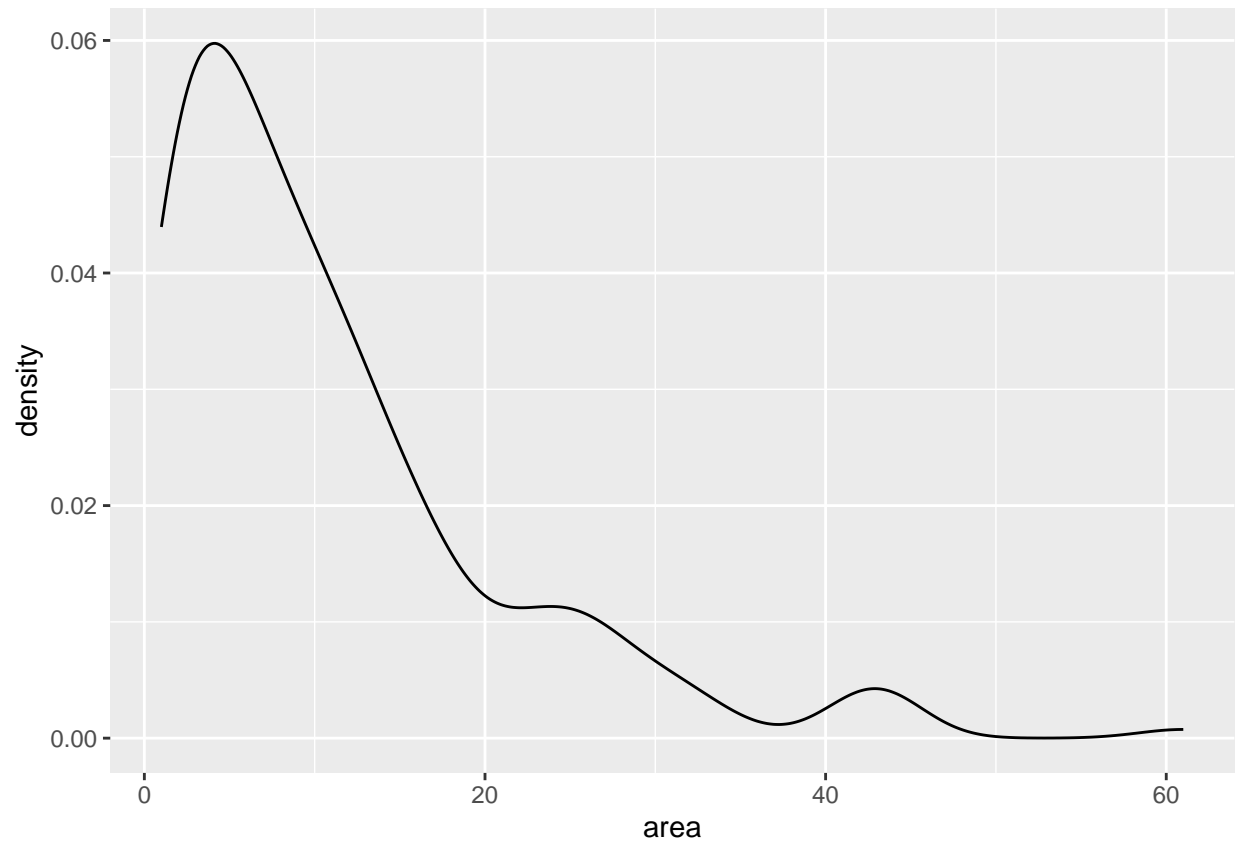
```

ggplot(data = colocArea_WT.df) + geom_histogram(aes(x = area), binwidth = 5)

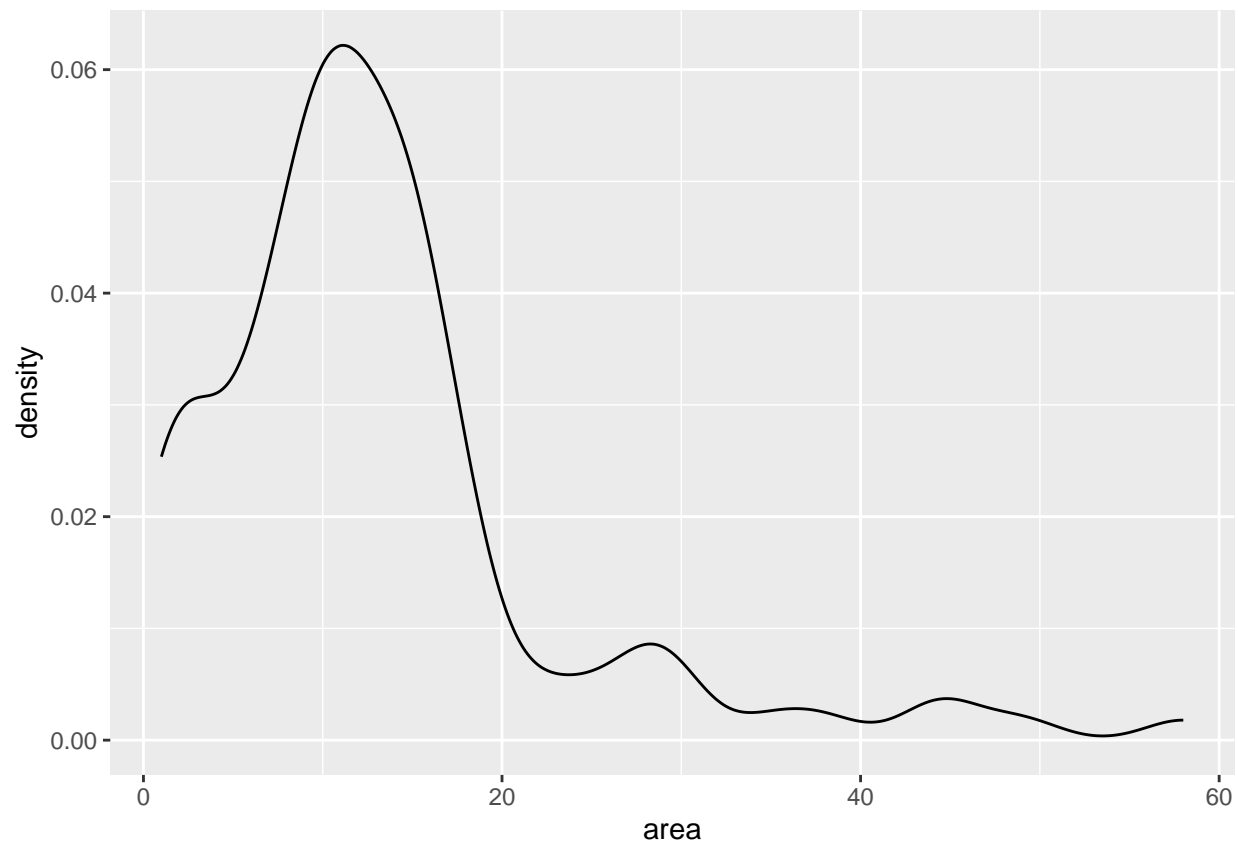
```

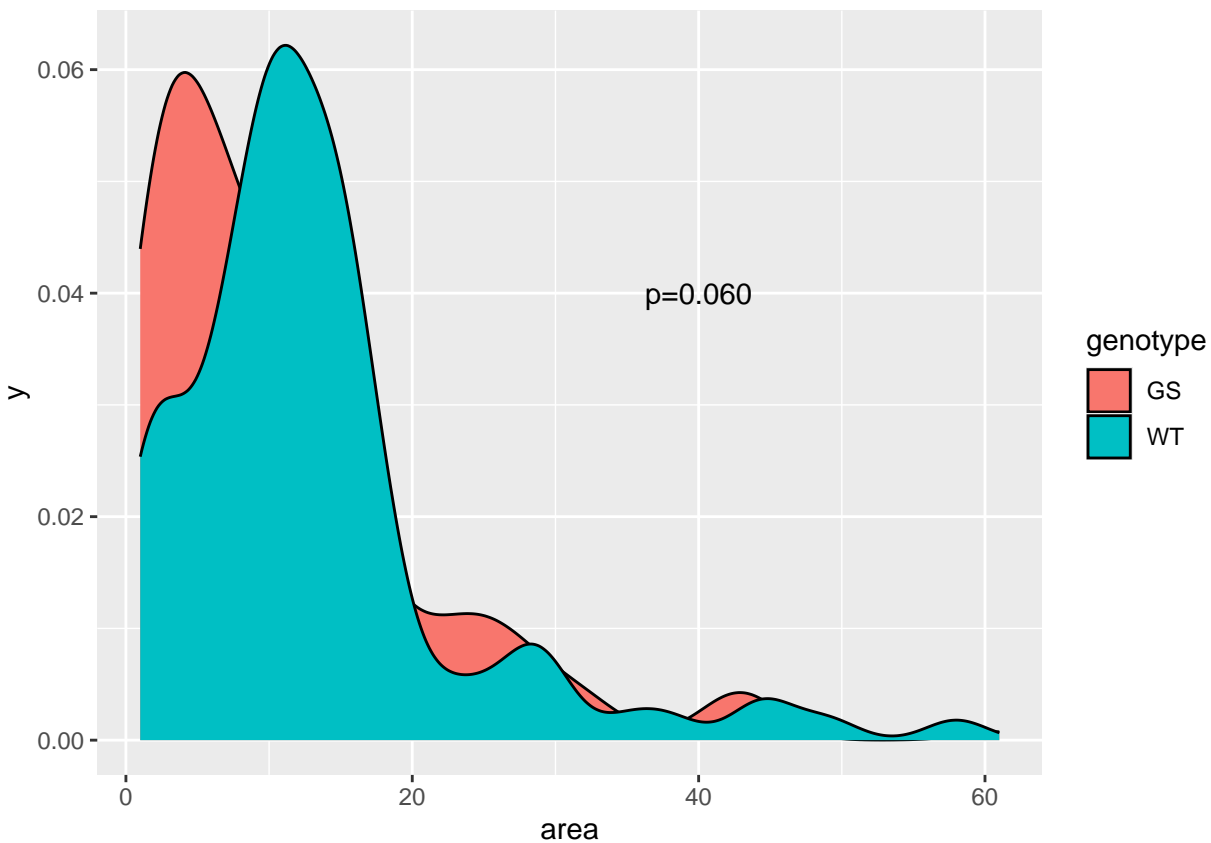
```
ggplot(data = colocArea_GS.df) + geom_density(aes(x = area))
```



```
ggplot(data = colocArea_WT.df) + geom_density(aes(x = area))
```



```
ggplot(data = colocArea_GS.df) + geom_density(aes(x = area, fill = genotype)) + geom_density(data = colocArea_GS.df)
```



```
colocArea_GS.df$genotype <- "GS"
colocArea_WT.df$genotype <- "WT"

colocArea_combined.df <- rbind(colocArea_GS.df, colocArea_WT.df)

t.test(area ~ genotype, colocArea_combined.df)

##
##  Welch Two Sample t-test
##
## data:  area by genotype
## t = -1.8903, df = 209.13, p-value = 0.0601
## alternative hypothesis: true difference in means between group GS and group WT is not equal to 0
## 95 percent confidence interval:
##   -4.75901509  0.09991145
## sample estimates:
## mean in group GS mean in group WT
##      10.96569      13.29524
```