STAT 206 Lab 3

Due Monday, October 23, 5:00 PM

General instructions for labs: You are encouraged to work in pairs to complete the lab. Labs must be completed as an R Markdown file. Be sure to include your lab partner (if you have one) and your own name in the file. Give the commands to answer each question in its own code block, which will also produce plots that will be automatically embedded in the output file. Each answer must be supported by written statements as well as any code used.

Agenda: Writing functions to automate repetitive tasks; fitting statistical models.

The ***gamma*** distributions are a family of probability distributions defined by the density functions,

$$f(x) = \frac{x^{a-1}e^{-x/s}}{s^a \Gamma(a)}$$

where the ***gamma function*** $\Gamma(a) = \int_0^\infty u^{a-1}e^{-u}du$ is chosen so that the total probability of all non-negative x is 1. The parameter a is called the ***shape***, and s is the ***scale***. When $a = 1$, this becomes the exponential distributions we saw in the first lab. The gamma probability density function is called `dgamma()` in R. You can prove (as a calculus exercise) that the expectation value of this distribution is as, and the variance as^2. If the mean and variance are known, μ and σ^2, then we can solve for the parameters,

$$a = \frac{a^2s^2}{as^2} = \frac{\mu^2}{\sigma^2}$$
$$s = \frac{as^2}{as} = \frac{\sigma^2}{\mu}$$

In this lab, you will fit a gamma distribution to data, and estimate the uncertainty in the fit.

Our data today are measurements of the weight of the hearts of 144 cats.

Part I

1. The data is contained in a data frame called `cats`, in the R package `MASS`. (This package is part of the standard R installation.) This records the sex of each cat, its weight in kilograms, and the weight of its heart in grams. Load the data as follows:

```
library(MASS)
data(cats)
```

```
library(MASS)
data(cats)
#cats

#after taking a look at the dataset we can see the raw data only has decimals out to the tens place
```

Run `summary(cats)` and explain the results.

```
summary(cats)
```

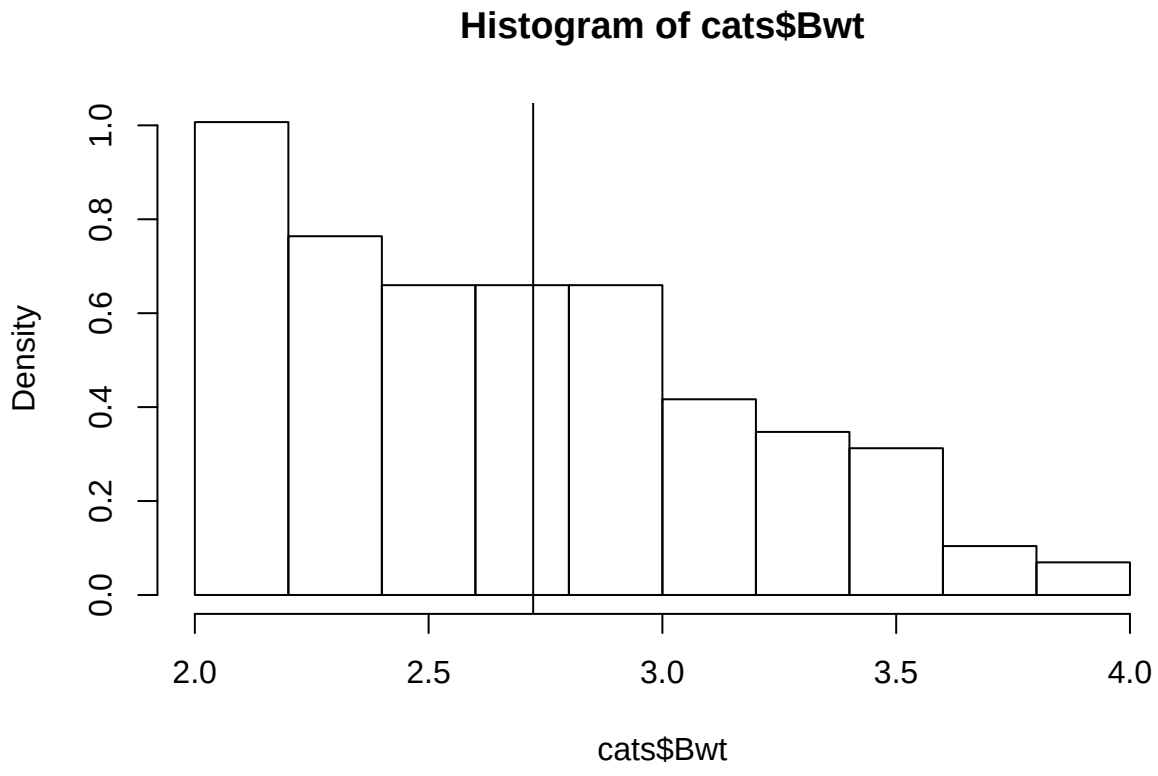
```
##  Sex         Bwt             Hwt
##  F:47   Min.   :2.000   Min.   : 6.30
##  M:97   1st Qu.:2.300   1st Qu.: 8.95
```

```
##          Median :2.700   Median :10.10
##          Mean   :2.724   Mean   :10.63
##          3rd Qu.:3.025   3rd Qu.:12.12
##          Max.   :3.900   Max.   :20.50
```

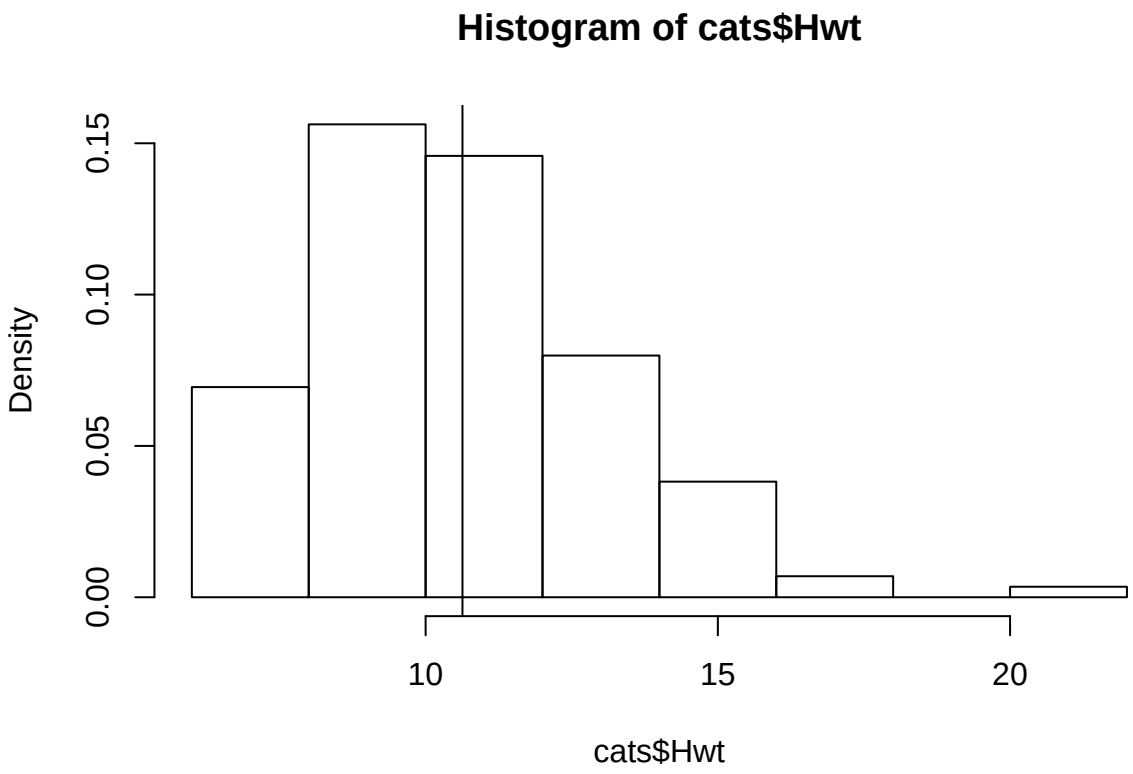
```
#Running summary(cats) gives us some descriptive statistics of the data set. We can see that there are
#3 catagories, 1. Sex 2. Bwt and 3 Hwt. We can also see that we have one categorical attribute and 2
#quantitative attributes. We need to look into help(cats) to get a better description of what Bwt and H
#mean, in this case I know already that it is for body weight and heart weight
#from the help("cats") it also specifies that body weight is in kg and heart weight is in g
#NOTE: i used help("cats") in the r console as to not clutter up the lab assignment
```

2. Plot a histogram of these weights using the `probability=TRUE` option. Add a vertical line with your calculated mean using `abline(v=yourmeanvaluehere)`. Does this calculated mean look correct?

```
#First we will make a histogram for Bwt. The attribute can be specified after the data with the $
#also the histogram does not have a argument that can handle adding a vertical line of the mean to the
#so we can just use the abline function on the line after the call to the function hist()
#according to the summary above the value does look correct and I specifically typed the values I wante
#from there to verify (commented out now)

hist(cats$Bwt, probability =TRUE)
#abline(v=2.724)
abline(v=mean(cats$Bwt))
```




```
hist(cats$Hwt, probability = TRUE)
#abline(v=10.63)
abline(v=mean(cats$Hwt))
```



3. Define two variables, `fake.mean <- 10` and `fake.var <- 8`. Write an expression for a using these placeholder values. Does it equal what you expected given the solutions above? Once it does, write another such expression for s and confirm.

```
fake.mean = 10
fake.var = 8
input = c(fake.mean, fake.var)

shape = function (input) {
  answer = ((input[1]^2) / input[2])
  return(answer)
}

shape(input)
```

```
## [1] 12.5
```

```
#The first step was to create the two test variables to make sure that the function works.
#we know that the answer will be 100/8 which is equal to 12.5
#testing the function with fake.mean and fake.var confirmsv this

scale = function (input) {
  answer = (input[2]/input[1])
  return(answer)
}

scale(input)
```

```
## [1] 0.8
```

```
#this time when when the function is written we must make sure to keep the arguments in the same order
#is not confusing. This time we know the answer should be  8/10 or .8
#running the fuction with our test variables confirms this
```

4. Calculate the mean, standard deviation, and variance of the heart weights using R's existing functions for these tasks. Plug the mean and variance of the cats' hearts into your formulas from the previous question and get estimates of a and s. What are they? Do not report them to more significant digits than is reasonable.

```
#to begin we make object variables to contain the mean, sd and var of the cats heart weight data

avg = round(mean(cats$Hwt),1)
avg
```

```
## [1] 10.6
```

```
std = round(sd(cats$Hwt),1)
std
```

```
## [1] 2.4
```

```
vari = round(var(cats$Hwt),1)
vari
```

```
## [1] 5.9
```

```
input = c(avg, vari)

#The code below is to check the values to make sure I get what I expect

#avg
#std
#vari

round(shape(input), 1)
```

```
## [1] 19
```

```
round(scale(input), 1)
```

```
## [1] 0.6
```

```
#now that out functions are tested and proven to work we use the actual cats data for heart weight
#the shape is about 19.1 and the scale is about 0.6
#from earlier we saw in the raw data that the values were only given to the tens decimal place
#we use a round functions so we do not report higher percision numbers than the raw data
```

5. Write a function, `cat.stats()`, which takes as input a vector of numbers and returns the mean and variances of these cat hearts. (You can use the existing mean and variance functions within this function.) Confirm that you are returning the values from above.

```
cat.stats = function(data) {
  avg = round(mean(data),1)
  vari = round(var(data),1)
  answer = c(avg, vari)
  return(answer)
}
```

```
cat.stats(cats$Hwt)
```

```
## [1] 10.6  5.9
```

```
#The function above takes an attribute from a data frame and makes it into a vector first
#then it extracts the mean and variance from that vector
#next the function calculates the shape and scale from our previously defined functions
#it also rounds the answer to match significant figures
#finally it returns a list to us with our answers
#the answers match our test case as above
```

Part II

6. Now, use your existing function as a template for a new function, `gamma.cat()`, that calculates the mean and variances and returns the estimate of a and s. What estimates does it give on the cats' hearts weight? Should it agree with your previous calculation?

```
gamma.cat = function(data) {
  mean_vari = cat.stats(data)
  shape = round(shape(mean_vari),1)
  scale = round(scale(mean_vari),1)
  return(c(shape, scale))


}

gamma.cat(cats$Hwt)
```

```
## [1] 19.0  0.6
```

```
#this new function first calls our previous function to get the mean and variance of the data
#next it uses our first function to get the shape and scale
#finally it returns the shape and scale of the data. It should agree with what we had found in
#our test cases
```

7. Estimate the a and s separately for all the male cats and all the female cats, using `gamma.cat()`. Give the commands you used and the results.

```
males = subset(cats, cats$Sex == "M", select = c(Hwt))
boy = gamma.cat(males[,'Hwt'])
boy
```

```
## [1] 19.6  0.6
```

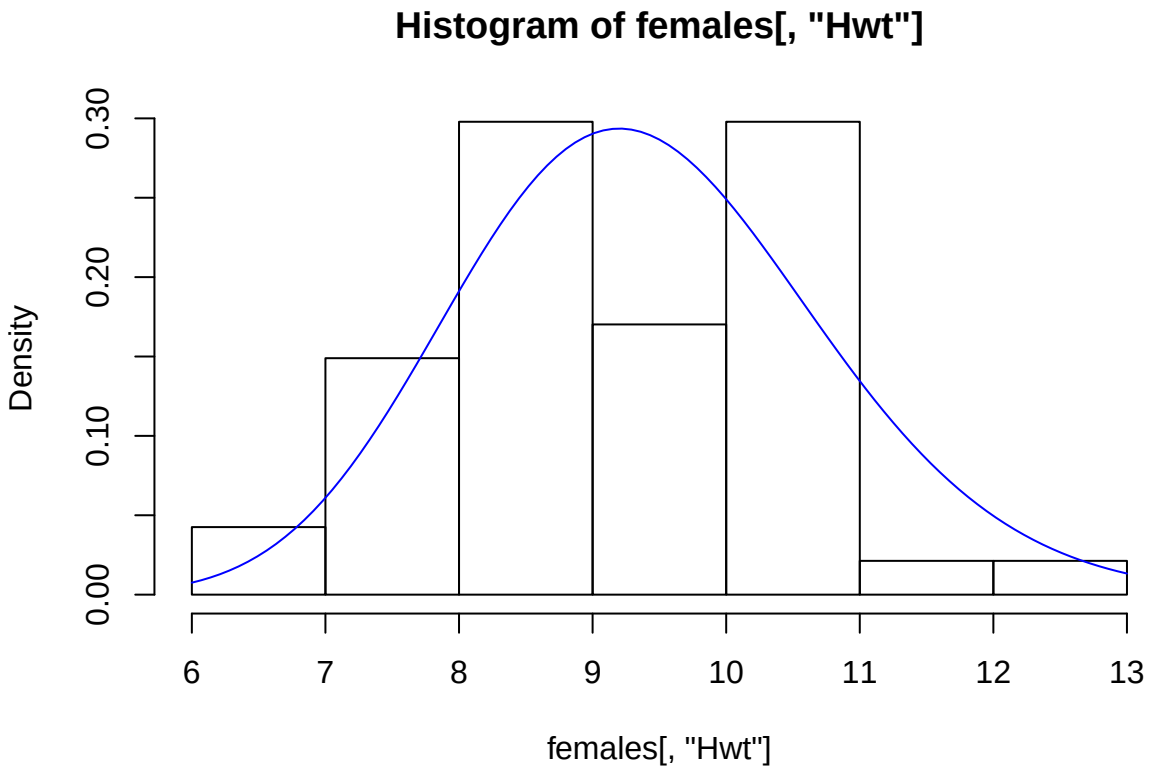
```
females = subset(cats, cats$Sex == "F", select = c(Hwt))

girl = gamma.cat(females[,'Hwt'])
girl
```

```
## [1] 47.0  0.2
```

```
#The results for male and female cats can be found by calling for a subset of the data. Then we can
#use the subsets as input for our gamma.cat function
```

8. Now, produce a histogram for the female cats. On top of this, add the shape of the gamma PDF using `curve()` with its first argument as `dgamma()`, the known PDF for the Gamma distribution. Is this distribution consistent with the empirical probability density of the histogram?

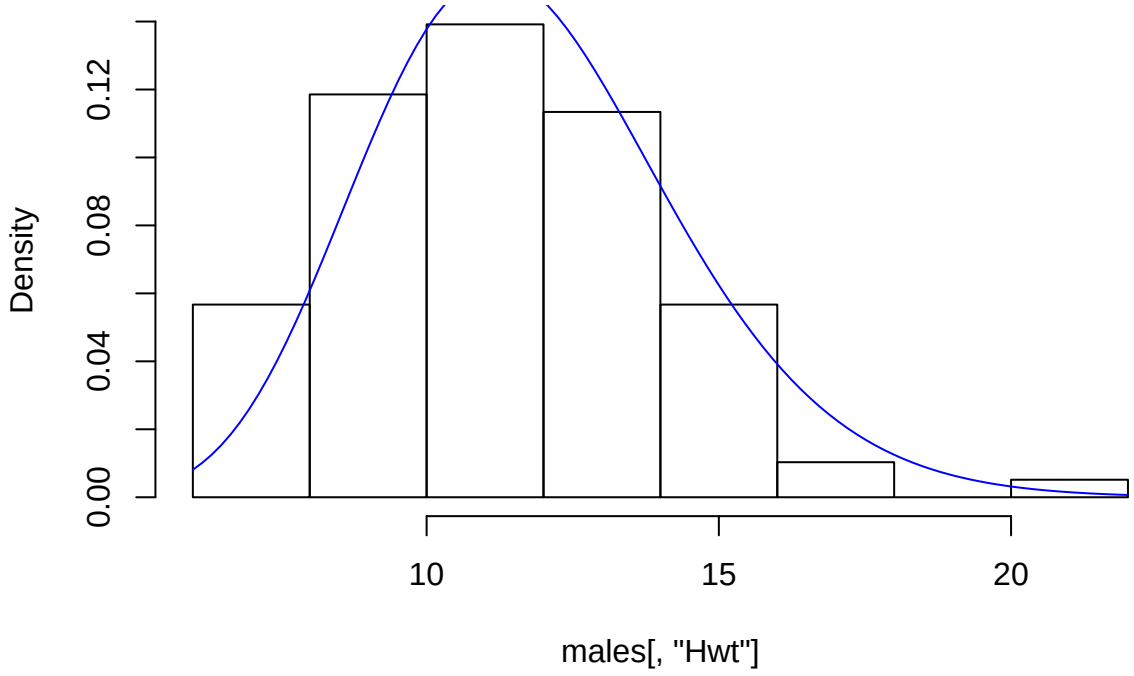
```
hist(females[,'Hwt'], probability = TRUE)
curve(dgamma(x, shape=girl[1], scale=girl[2]),add=TRUE,col="blue")
```




```
#From the histogram we can see that females have a bimodal distribution. However, with the dgamma line
#does not quite reflect
```

9. Repeat the previous step for male cats. How do the distributions compare?

```
hist(males[,'Hwt'], probability = TRUE)
curve(dgamma(x, shape=boy[1], scale=boy[2]),add=TRUE,col="blue")
```




```
#The histrogram shows a slightly negatively skewed distribution. The dgamma line follows the distributi
#However, it doesn't follow the the last bin at the end
```