STAT 206 Lab 3_Lihua_Xu

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)
```

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

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

```r
summary(cats)
```

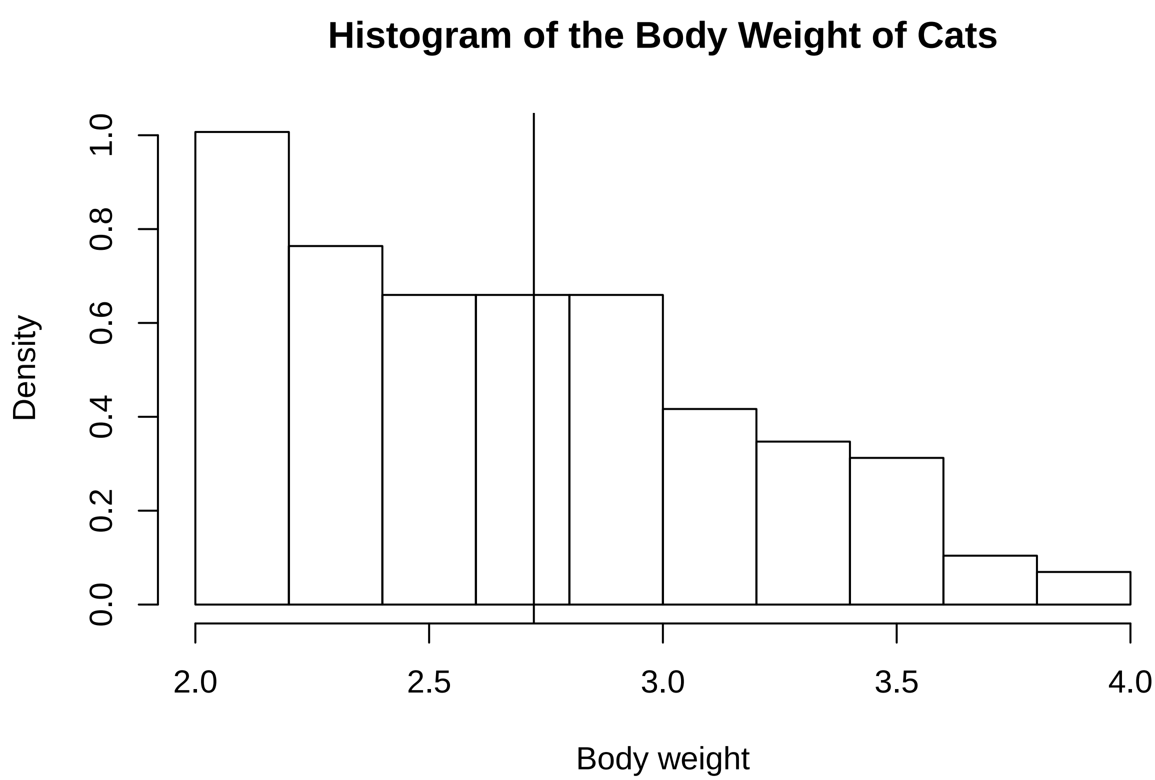
```
##  Sex         Bwt             Hwt       
##  F:47   Min.   :2.000   Min.   : 6.30  
##  M:97   1st Qu.:2.300   1st Qu.: 8.95  
##         Median :2.700   Median :10.10  
##         Mean   :2.724   Mean   :10.63  
##         3rd Qu.:3.025   3rd Qu.:12.12  
```

```
##         Max.   :3.900   Max.   :20.50
```

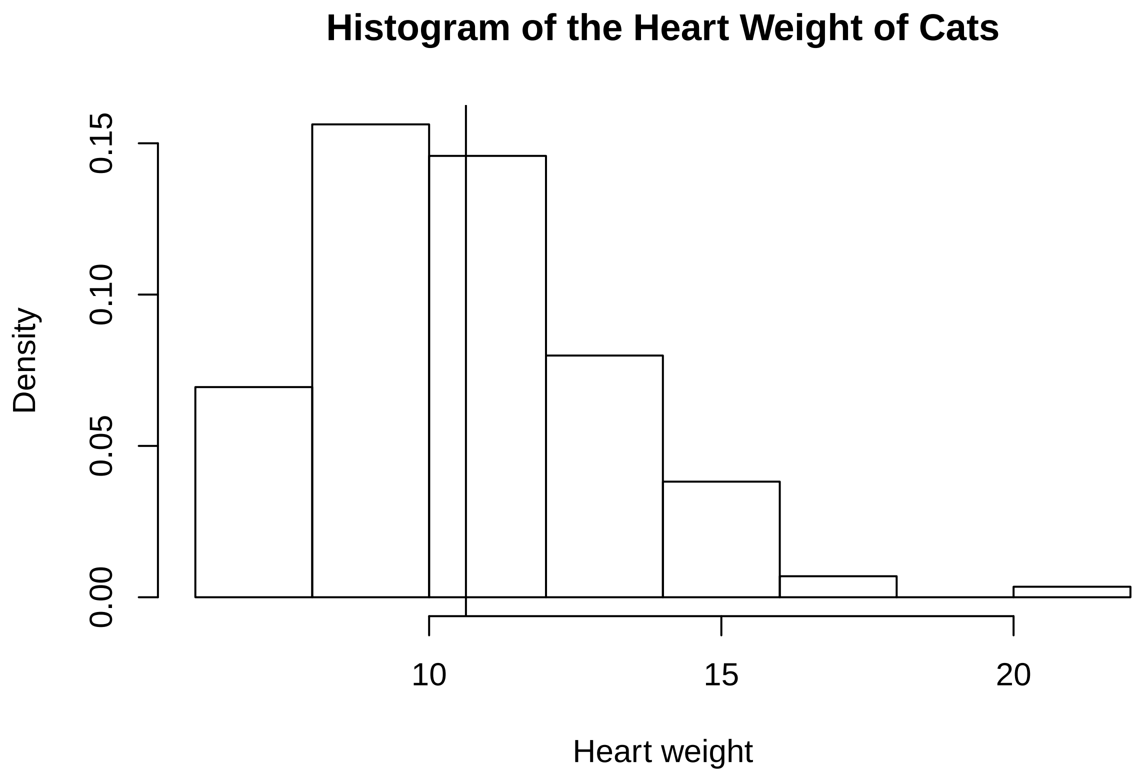
```
#The amount of the male cat is 97 and the amount of the female cat is 47.
#"Bwt" stands for the body weight:
#The minimum weight among these data is 2.
#The median number among these data is 2.7.
#The mean value aong these data is 2.724.
#The maximun weight among all these data is 3.9.
#The first quartile weight among these data is 2.3.
#The third quartile weight among these data is 3.025.

#"Hwt" stand for the heart weight:
#The minimum weight among these data is 6.3.
#The median number among these data is 10.10.
#The mean value aong these data is 10.63.
#The maximun weight among all these data is 20.5.
#The first quartile weight among these data is 8.95.
#The third quartile weight among these data is 12.12.
```

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?

```
hist(cats$Bwt,probability=TRUE,xlab="Body weight",main="Histogram of the Body Weight of Cats")
abline(v=2.724)
```




```
hist(cats$Hwt,probability=TRUE,xlab="Heart weight",main="Histogram of the Heart Weight of Cats")
abline(v=10.63)
```



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
#Writing a function for obtainning a
a <- function(mu,sigma)
{a_value=mu^2/sigma^2
return(a_value)}
a(fake.mean,fake.var)
```

```
## [1] 1.5625
```

```
#It does give me the solution I want.

#Writing a function for obtaining s
s <- function(sigma,mu)
{s_value=sigma^2/mu
return(s_value)}
s(fake.var,fake.mean)
```

```
## [1] 6.4
```

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.

```
mean_hw <- mean(cats$Hwt)
mean_hw
```

```
## [1] 10.63056
```

```
#The mean of the heart weights is 10.63056.

sd_hw <- sd(cats$Hwt)
sd_hw
```

```
## [1] 2.434636
```

```
#The standard deviation of the heart weights is 2.434636.

var_hw <- var(cats$Hwt)
var_hw
```

```
## [1] 5.927451
```

```
#The variance of the heart weights is 5.927451.

cat_hw_a <- a(mean_hw,var_hw)
cat_hw_a
```

```
## [1] 3.216443
```

```
#The a value should be 3.216443.

cat_hw_s <- s(var_hw,mean_hw)
cat_hw_s
```

```
## [1] 3.305065
```

```
#The s value should be 3.305065.
```

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(vector_of_values)
{mean_hw_fun <- mean(vector_of_values)
 var_hw_fun <- var(vector_of_values)
 mean_var_cat <- c(mean_of_cat_hearts=mean_hw_fun,variance_of_cat_hearts=var_hw_fun)
 return(mean_var_cat)}

cat.stats(cats$Hwt)
```

```
##     mean_of_cat_hearts variance_of_cat_hearts
##              10.630556               5.927451
```

```
#Values for the mean and varience of cat heart weight are the same as the above solution.
```

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?

```r
gamma.cat <- function(vector_of_values)
{mean_hw_fun <- mean(vector_of_values)
 var_hw_fun <- var(vector_of_values)
 a <- mean_hw_fun^2/var_hw_fun^2
 s <- var_hw_fun^2/mean_hw_fun
 estimate_vlues <- c(a=a,s=s)
 return(estimate_vlues)}

gamma.cat(cats$Hwt)
```

```
##        a        s
## 3.216443 3.305065
```

```r
#The estimated values for a and s of the caat hearts weight are 3.216443 and 3.305065 seperately.
#The results are the same as what we calculated before.
#So it should agree with our previous calculations.
```

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.

```r
gamma.cat(cats[cats$Sex=="F",]$Hwt)
```

```
##          a          s
## 24.9232733  0.3692183
```

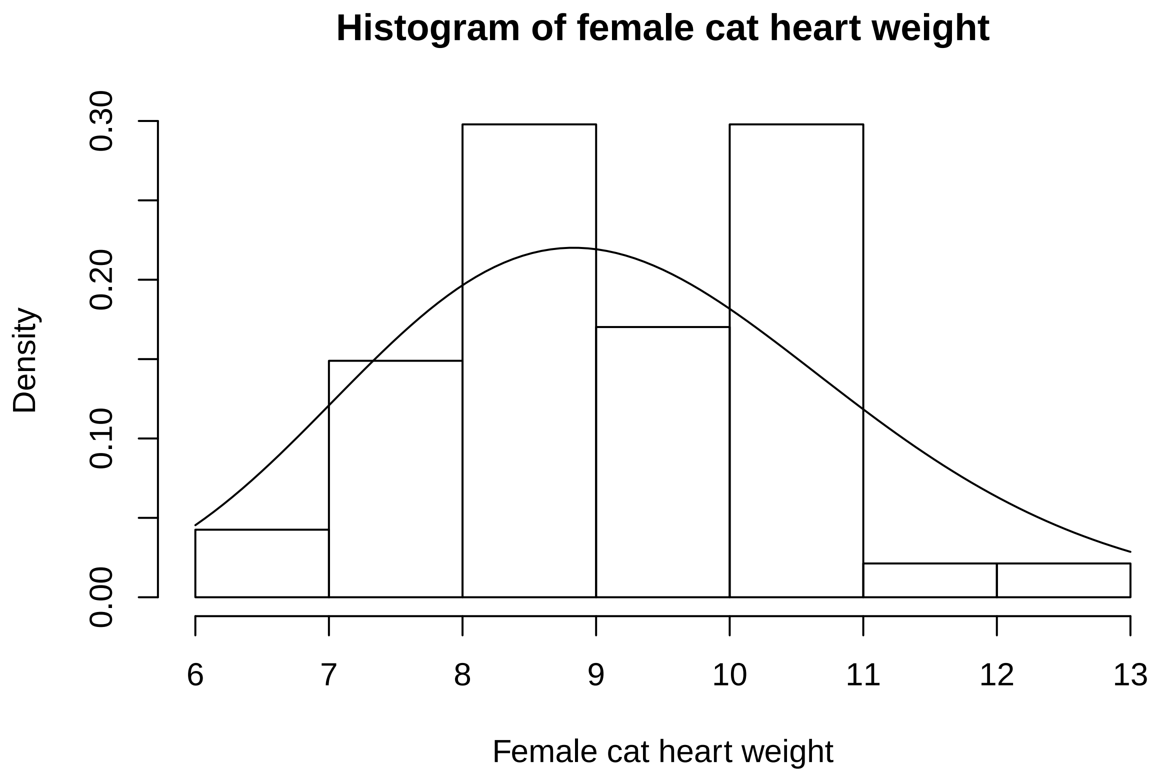
```r
#The estimation value for a and s for all female catas are 24.9232733 and 0.3692183 seperately.
gamma.cat(cats[cats$Sex=="M",]$Hwt)
```

```
##        a        s
## 3.069017 3.689351
```

```r
#The estimation value for a and s for all male catas are 3.069017 and 3.689351 seperately.
```

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?

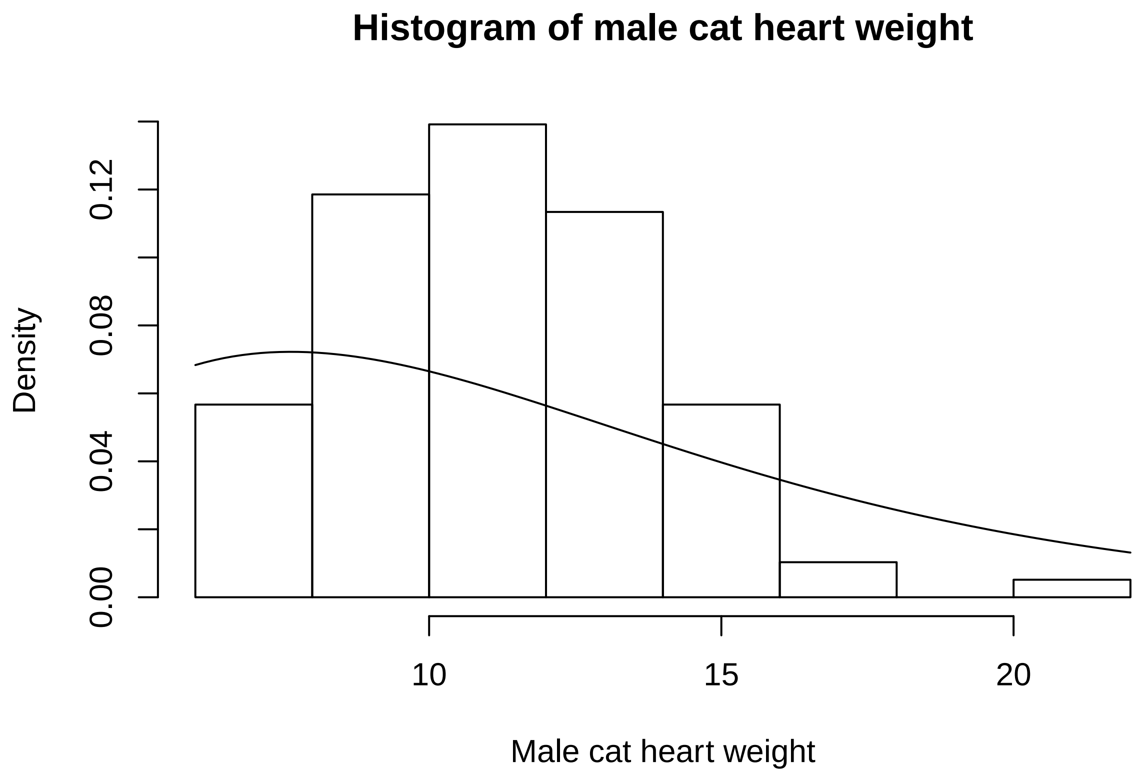
```r
hist(cats[cats$Sex=="F",]$Hwt,freq=FALSE,main="Histogram of female cat heart weight",
     xlab="Female cat heart weight")
curve(dgamma(x, scale=0.3692183, shape=24.9232733), add=TRUE)
```




```
#The is the plot adding the gamma PDF using 'curve()'
#The gamma distribution plot is consistent with the empirical probability density of the histogram.
```

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

```
hist(cats[cats$Sex=="M",]$Hwt,freq=FALSE,main="Histogram of male cat heart weight",
     xlab="Male cat heart weight")
curve(dgamma(x, scale=3.689351, shape=3.069017), add=TRUE)
```




```
#The is the plot adding the gamma PDF using 'curve()'
#The gamma distribution plot is consistent with the empirical probability density of the histogram.
```