STT465\_HW3

Sam Isken

October 16, 2019

STT 465 Fall 2019 Homework 3 Due 10/23/2019 (In Class) Instruction:

When using R in any problem, copy the code and results onto your word document under that question number and add any required comments. You will lose points if I do not see your codes. You should present a stapled document when multiple pages are used. The grader will not be held responsible for any loss of pages.

The crab data set (in D2L) contains information on the Crab width in cm per female crab. After downloading onto your computer, import the data into R using:

crab <- read.table("crab.txt",header=TRUE)  
head(crab)

## color spine width nSatellites weight  
## 1 3 3 28.3 8 3050  
## 2 4 3 22.5 0 1550  
## 3 2 1 26.0 9 2300  
## 4 4 3 24.8 0 2100  
## 5 4 3 26.0 4 2600  
## 6 3 3 23.8 0 2100

1. Maximum Likelihood Inference of the mean in a Gaussian Model

Use the carapace width variable to answer the following questions:

1. Provide summary statistics (mean, variance, range, and histogram).

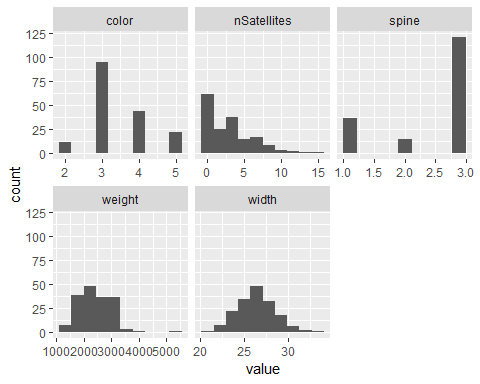
mean\_list <- lapply(crab, mean)  
variance\_list <- lapply(crab,var)  
range\_list <- lapply(crab,range)  
list\_all <- list(mean\_list,variance\_list,range\_list)  
result <- setNames(do.call(rbind.data.frame, list\_all), c("color", "spine","width","nSatellites","weight"))  
result$Category <- c("Means","Variances","Range\_lower","Range\_upper")  
result

## color spine width nSatellites weight Category  
## 2 3.4393064 2.485549 26.298844 2.919075 2437.191 Means  
## 21 0.6430972 0.681476 4.448138 9.912018 332958.097 Variances  
## 3 2.0000000 1.000000 21.000000 0.000000 1200.000 Range\_lower  
## 4 5.0000000 3.000000 33.500000 15.000000 5200.000 Range\_upper

library(tidyr)  
library(ggplot2)  
  
crab %>% gather() %>% head()

## key value  
## 1 color 3  
## 2 color 4  
## 3 color 2  
## 4 color 4  
## 5 color 4  
## 6 color 3

ggplot(gather(crab), aes(value)) +   
 geom\_histogram(bins = 10) +   
 facet\_wrap(~key, scales = 'free\_x')

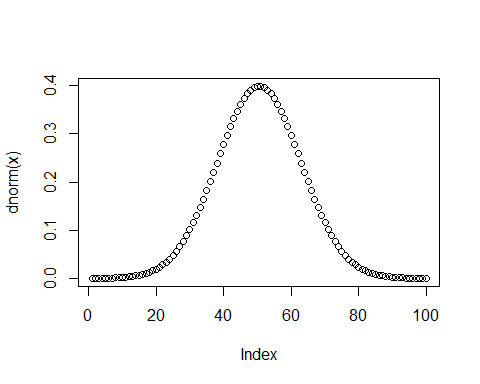


1. Write the likelihood function under Gaussian assumptions.

The Gaussian Assumptions attempt to estimate such that our data is distributed as close to normally as possible. (Thus having a pdf of:

and a plotted distribution of:

x <- seq(-4, 4, length=100)  
plot(dnorm(x))



).

To do this we make 2 assumptions:

(mu and sigma must be known)

1. Data must be independently distributed.
2. Data must be identically distributed.

(This can be summarized as iid)

1. Derive the Maximum Likelihood estimator of the mean parameter (recall the steps:
2. write the likelihood,

Let us first identify our likelihood function to estimate assuming a Gaussian distribution

1. simplify as much as possible,

We can simplify this to be:

1. take the log, take derivative with respect to the mean,

Now let us take the natural log of

Now let us take the derivitive with respect to (because of this many of the early terms drop out since they are constants)

1. set the derivative equal to zero,
2. solve for the mean ).

This is expected as the MLE for a Gaussian distribution for is the sample mean.

1. Provide the Max. Likelihood estimate of the mean for this data set together with an approximate 95% CI (assume Central Limit Theorem).

#Did for all variables   
mle\_list <- lapply(crab, mean)  
print(mle\_list)

## $color  
## [1] 3.439306  
##   
## $spine  
## [1] 2.485549  
##   
## $width  
## [1] 26.29884  
##   
## $nSatellites  
## [1] 2.919075  
##   
## $weight  
## [1] 2437.191

n <- nrow(crab)  
  
mle\_color <- 3.439306 # xbar  
mle\_spine <- 2.485549 # xbar  
mle\_width <- 26.29884 # xbar  
mle\_nSatellites <- 2.919075 # xbar  
mle\_weight <- 2437.191 # xbar  
  
sigma\_color <- sqrt(var(crab$color)) # sigma   
sigma\_spine <- sqrt(var(crab$spine)) # sigma  
sigma\_width <- sqrt(var(crab$width)) # sigma  
sigma\_nSatellites <- sqrt(var(crab$nSatellites)) # sigma  
sigma\_weight <- sqrt(var(crab$weight)) # sigma  
  
# Confidence Intervals   
color\_CI <- c(mle\_color - (1.96\*(sigma\_color/sqrt(n))),mle\_color + (1.96\*(sigma\_color/sqrt(n))))  
spine\_CI <- c(mle\_spine - (1.96\*(sigma\_spine/sqrt(n))),mle\_spine + (1.96\*(sigma\_spine/sqrt(n))))  
width\_CI <- c(mle\_width - (1.96\*(sigma\_width/sqrt(n))),mle\_width + (1.96\*(sigma\_width/sqrt(n))))  
nSatellites\_CI <- c(mle\_nSatellites - (1.96\*(sigma\_nSatellites/sqrt(n))),mle\_nSatellites + (1.96\*(sigma\_nSatellites/sqrt(n))))  
weight\_CI <- c(mle\_weight - (1.96\*(sigma\_weight/sqrt(n))),mle\_weight + (1.96\*(sigma\_weight/sqrt(n))))

1. Bayesian Inference: Normal Model
2. Assume the normal model with inference for where is known. Analyze the carapace width of crabs with the normal model with conjugate prior distribution for in which and . Find the posterior distribution for .

We know that:

Let us find our n:

n <- length(crab$width)  
n

## [1] 173

thus, n = 173.

Let us now calculate our credibility factor .

Let us plug that into our formula for to solve for .

sigma\_sq <- var(crab$width)

(Scratch work shown on attached sheet)

Thus we discover:

So

Let us now calculate the posterior mean, median, mode and variance.

where ${Y} $ is

ybar <- mean(crab$width)  
ybar

## [1] 26.29884

thus

mustar <- ((1-.99425)\*20) + (.99425\*26.29884)  
mustar

## [1] 26.26262

Since the posterior distribution is normal, the posterior mean = posterior median = posterior mode.

Let us now calculate the posterior variance:

sigma\_squared\_star <- (.99425 \* 1.2)/173  
sigma\_squared\_star

## [1] 0.006896532

Thus the posterior distribution of is:

1. Construct a 90% credibility interval for ð and interpret your result.

abs(qnorm(.05,lower.tail=TRUE))

## [1] 1.644854

Thus:

#Credibility Interval Calculated  
lower\_bound <- 26.26262 - (1.644854\*(sqrt(0.006896532)))  
upper\_bound <- 26.26262 + (1.644854\*(sqrt(0.006896532)))  
lower\_bound

## [1] 26.12602

upper\_bound

## [1] 26.39922

1. Plot the prior density, likelihood and posterior density on a single plot clearly indicating each graph.

y = crab$width # data  
n = length(y)  
sigma2= 1.2 # known mean  
  
#### Parameters for prior density (usually called hyperparameters)  
a=3  
b=2  
  
a0=3 # prior shape parameter  
b0=2 # prior scale(rate) parameter  
options(digits=4)  
a=a0+n/2;   
c(a,sigma2,b)

## [1] 89.5 1.2 2.0

# Inference on sigma2 = 1/lambda   
  
sig2priormean=b0/(a0-1) # Prior mean for sigma^2  
  
sig2postmean=b/(a-1)  
  
### 95% Cred. Int  
  
sig2cr=1/qgamma(c(0.975,0.025),a,b)  
  
c(ybar, mustar, sig2cr)

## [1] 26.29884 26.26262 0.01835 0.02781

sig2v=seq(0.01,10,0.01) # HERE   
prior=dgamma(1/sigma\_squared\_star,a0,b0)/sigma\_squared\_star^2  
post=dgamma(1/sigma\_squared\_star,a,b)/sigma\_squared\_star^2;  
like=dgamma(1/sigma\_squared\_star,a-a0-1,b-b0+0)/sigma\_squared\_star^2  
plot(c(0,10),c(0,1.2),type="n",  
 main="Inference on the model variance parameter",  
 xlab="sigma^2 = 1/lambda",ylab="density/likelihood")  
lines(sigma\_squared\_star,prior,lty=3,lwd=2); lines(sigma\_squared\_star,like,lty=3,lwd=2)  
lines(sigma\_squared\_star,post,lty=3,lwd=2)  
  
points(sig2cr,c(0,0),pch=rep(16,2),cex=rep(1.5,2))  
legend(1.8,1.2,  
 c("Prior density","Likelihood function (normalised)","Posterior density"),  
 lty=c(1,2,3),lwd=c(2,2,2))

