bios6301hw9

Haoyang Yi

11/9/2022

Question 1

15 points

Consider the following very simple genetic model (*very* simple – don't worry if you're not a geneticist!). A population consists of equal numbers of two sexes: male and female. At each generation men and women are paired at random, and each pair produces exactly two offspring, one male and one female. We are interested in the distribution of height from one generation to the next. Suppose that the height of both children is just the average of the height of their parents, how will the distribution of height change across generations?

Represent the heights of the current generation as a dataframe with two variables, m and f, for the two sexes. We can use `rnorm` to randomly generate the population at generation 1:

```
pop <- data.frame(m = rnorm(100, 160, 20), f = rnorm(100, 160, 20))
```

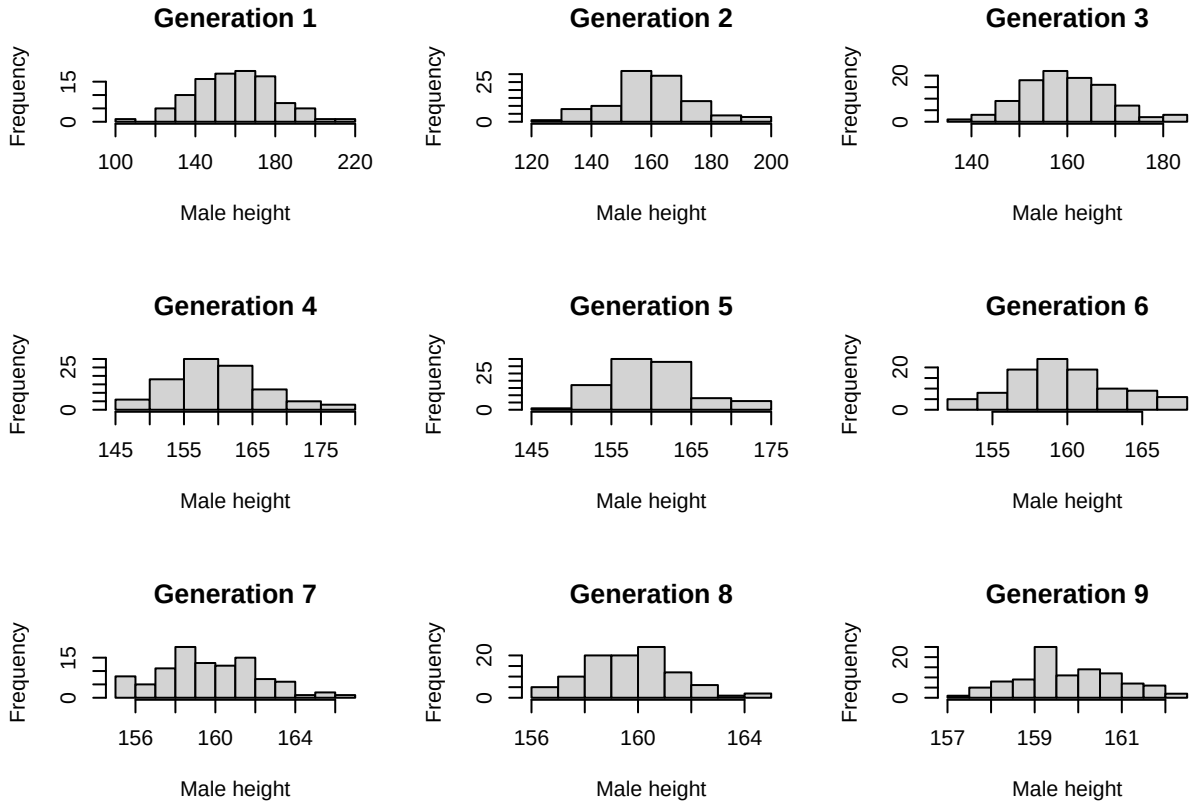
The following function takes the data frame `pop` and randomly permutes the ordering of the men. Men and women are then paired according to rows, and heights for the next generation are calculated by taking the mean of each row. The function returns a data frame with the same structure, giving the heights of the next generation.

```
next_gen <- function(pop) {
    pop$m <- sample(pop$m)
    pop$m <- rowMeans(pop)
    pop$f <- pop$m
    pop
}
```

Use the function `next_gen` to generate nine generations (you already have the first), then use the function `hist` to plot the distribution of male heights in each generation (this will require multiple calls to `hist`). The phenomenon you see is called regression to the mean. Provide (at least) minimal decorations such as title and x-axis labels.

```
gen1 = pop
gen2 = next_gen(gen1)
gen3 = next_gen(gen2)
gen4 = next_gen(gen3)
gen5 = next_gen(gen4)
gen6 = next_gen(gen5)
gen7 = next_gen(gen6)
gen8 = next_gen(gen7)
gen9 = next_gen(gen8)
par(mfrow = c(3,3))
hist(gen1$m,main = paste("Generation", 1),xlab = "Male height")
hist(gen2$m,main = paste("Generation", 2),xlab = "Male height")
hist(gen3$m,main = paste("Generation", 3),xlab = "Male height")
hist(gen4$m,main = paste("Generation", 4),xlab = "Male height")
```

```
hist(gen5$m,main = paste("Generation", 5),xlab = "Male height")
hist(gen6$m,main = paste("Generation", 6),xlab = "Male height")
hist(gen7$m,main = paste("Generation", 7),xlab = "Male height")
hist(gen8$m,main = paste("Generation", 8),xlab = "Male height")
hist(gen9$m,main = paste("Generation", 9),xlab = "Male height")
```



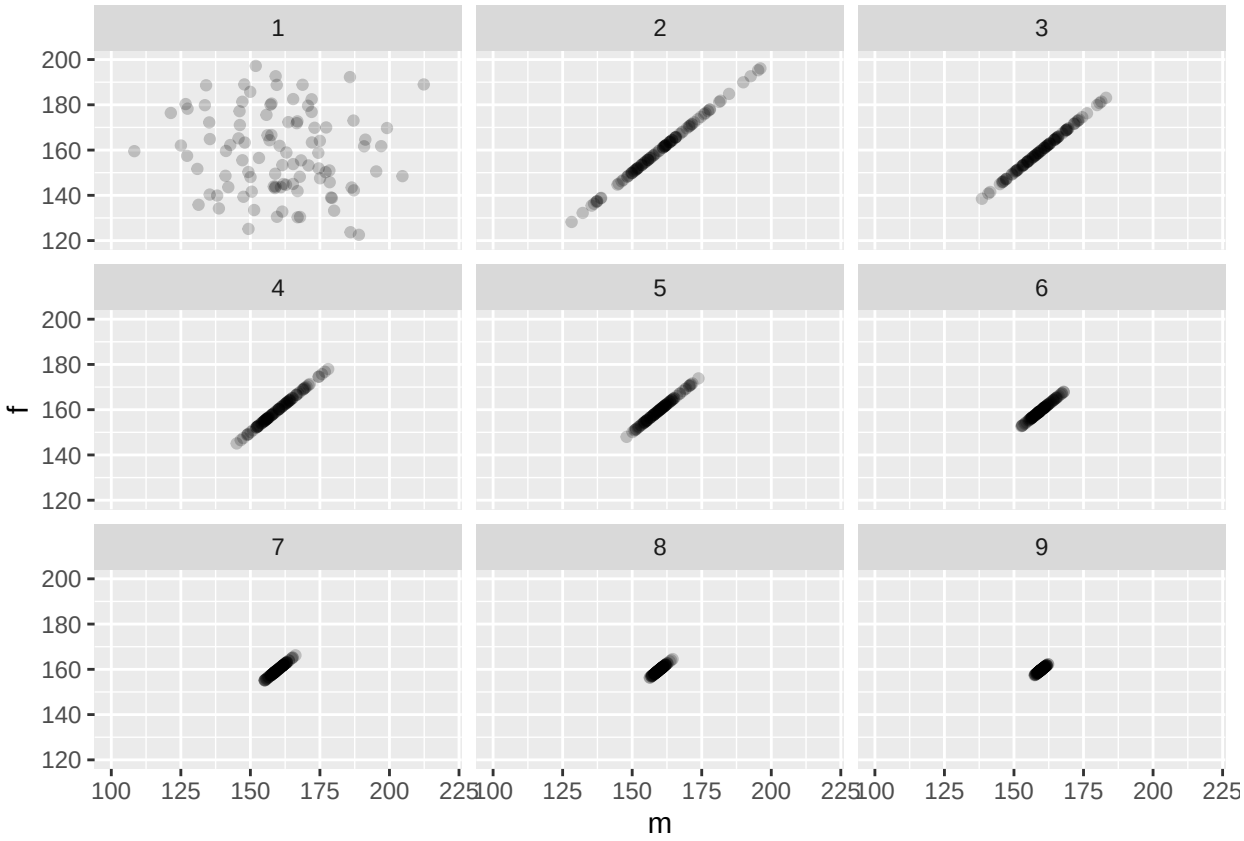
Question 2

10 points

Use the simulated results from question 1 to reproduce (as closely as possible) the following plot in ggplot2.

```
df_gen = rbind(gen1,gen2,gen3,gen4,gen5,gen6,gen7,gen8,gen9)
df_gen$generation = c(rep(1,100),rep(2,100),rep(3,100),rep(4,100),
                      rep(5,100),rep(6,100),rep(7,100),rep(8,100),rep(9,100))
df_gen %>%
  ggplot(aes(x=m,y=f))+geom_point(alpha = 0.2)+facet_wrap(~generation)+xlim(c(100,220))+ylim(c(120,200))
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```



Question 3

15 points

You calculated the power of a study design in question #1 of assignment 3. The study has two variables, treatment group and outcome. There are two treatment groups (0, 1) and they should be assigned randomly with equal probability. The outcome should be a random normal variable with a mean of 60 and standard deviation of 20. If a patient is in the treatment group, add 5 to the outcome.

Starting with a sample size of 250, create a 95% bootstrap percentile interval for the mean of each group. Then create a new bootstrap interval by increasing the sample size by 250 until the sample is 2500. Thus you will create a total of 10 bootstrap intervals. Each bootstrap should create 1000 bootstrap samples. (9 points)

```
set.seed(2022)
intervals = data.frame(matrix(NA,10,7))
colnames(intervals)=c("size","2.5%_treatment0","97.5%_treament0",
                      "2.5%_treatment1","97.5%_treatment1","mean_0","mean_1")
for (i in 1:10){
n = 250*i
treat = rbinom(n,1,0.5)
outcome = rnorm(n,60,20)+5*treat
res = data.frame(matrix(NA,1000,2))
colnames(res) = c("treatment_0","treatment_1")
for (j in 1:1000){
  newrow = sample(n,n,replace = TRUE)
  res[j,] = tapply(outcome[newrow],treat[newrow],mean)
}
```

```
intervals[i,1]=n
intervals[i,2:3]=quantile(res[,1],c(0.025,0.975))
intervals[i,4:5]=quantile(res[,2],c(0.025,0.975))
intervals[i,6] = mean(res[,1])
intervals[i,7] = mean(res[,2])
}

intervals
```

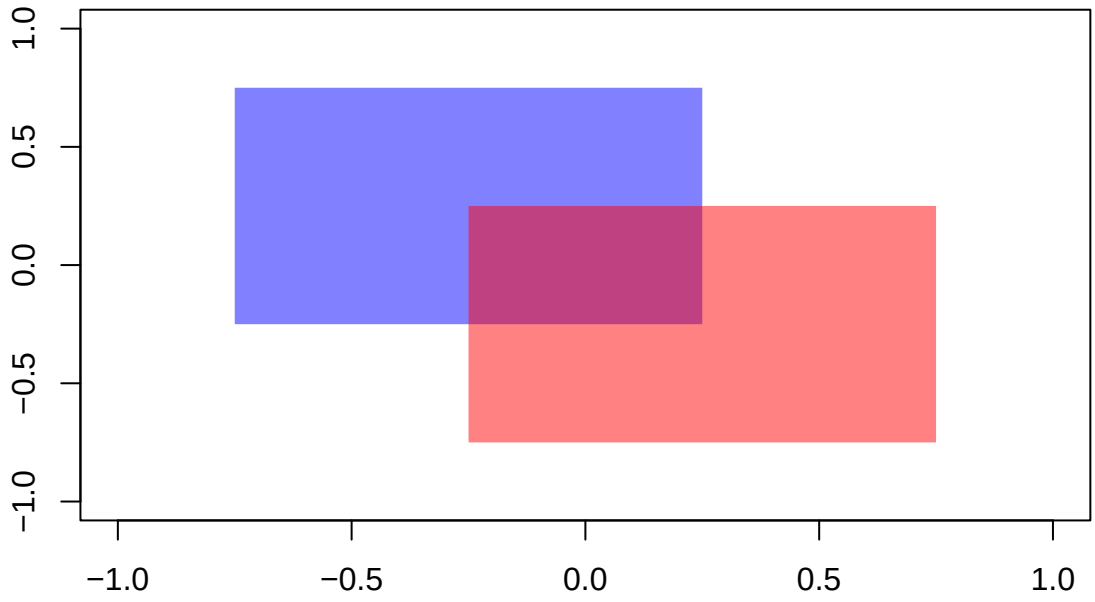
```
##    size 2.5%_treatment0 97.5%_treament0 2.5%_treatment1 97.5%_treatment1
## 1   250        53.99656        60.98433        59.12599         65.06593
## 2   500        59.27101        64.18093        64.47157         69.05103
## 3   750        58.02260        62.12314        63.61589         67.82770
## 4  1000        57.73422        61.19670        63.23148         66.94054
## 5  1250        56.70952        59.69581        63.81686         66.68974
## 6  1500        59.73296        62.74158        62.71206         65.48744
## 7  1750        58.69097        61.35978        62.46932         65.08222
## 8  2000        59.81898        62.19992        64.12481         66.55308
## 9  2250        60.15856        62.54635        63.00229         65.36832
## 10 2500        57.68994        59.81357        63.61398         65.68806
##      mean_0   mean_1
## 1  57.66908 62.08083
## 2  61.75156 66.79496
## 3  60.07992 65.72524
## 4  59.55441 65.14387
## 5  58.17597 65.28930
## 6  61.30692 64.10961
## 7  59.98083 63.71939
## 8  60.98619 65.30477
## 9  61.34470 64.11646
## 10 58.72048 64.67747
```

Produce a line chart that includes the bootstrapped mean and lower and upper percentile intervals for each group. Add appropriate labels and a legend. (6 points)

Here's an example of how you could create transparent shaded areas.

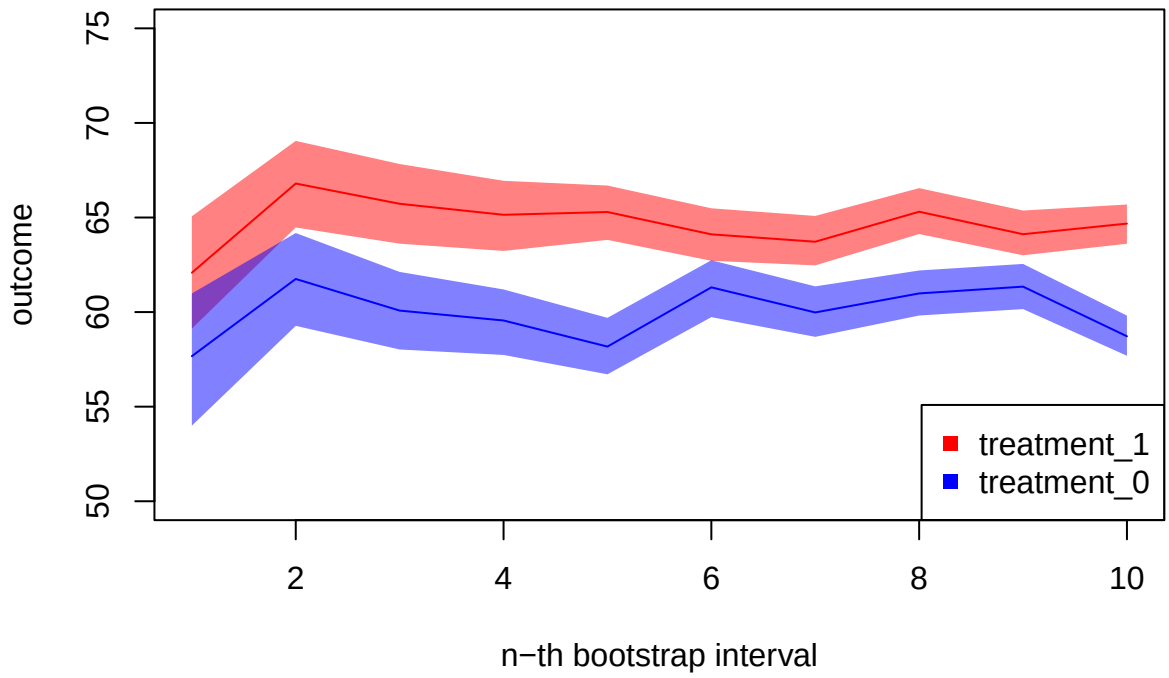
```
makeTransparent = function(..., alpha=0.5) {
  if(alpha<0 | alpha>1) stop("alpha must be between 0 and 1")
  alpha = floor(255*alpha)
  newColor = col2rgb(col=unlist(list(...)), alpha=FALSE)
  .makeTransparent = function(col, alpha) {
    rgb(red=col[1], green=col[2], blue=col[3], alpha=alpha, maxColorValue=255)
  }
  newColor = apply(newColor, 2, .makeTransparent, alpha=alpha)
  return(newColor)
}
par(new=FALSE)
plot(NULL,
  xlim=c(-1, 1),
  ylim=c(-1, 1),
  xlab="",
  ylab=""
)
polygon(x=c(seq(-0.75, 0.25, length.out=100), seq(0.25, -0.75, length.out=100)),
```

```
        y=c(rep(-0.25, 100), rep(0.75, 100)), border=NA, col=makeTransparent('blue',alpha=0.5))
polygon(x=c(seq(-0.25, 0.75, length.out=100), seq(0.75, -0.25, length.out=100)),
        y=c(rep(-0.75, 100), rep(0.25, 100)), border=NA, col=makeTransparent('red',alpha=0.5))
```




```
plot(NULL,
xlim=c(1,10),
ylim=c(50,75),
xlab="n-th bootstrap interval",
ylab="outcome"
)
lines(x=seq(1:10), y = intervals$mean_1, col = "red")
polygon(x=c(seq(1,10), seq(10,1)), y = c(intervals$`2.5%_treatment1`, rev(intervals$`97.5%_treatment1`))
border=NA, col=makeTransparent('red',alpha=0.5))
lines(x=seq(1:10), y = intervals$mean_0, col = "blue")
polygon(x=c(seq(1,10), seq(10,1)), y = c(intervals$`2.5%_treatment0`, rev(intervals$`97.5%_treament0`))
border=NA, col=makeTransparent('blue',alpha=0.5))
lab <- c('treatment_1', 'treatment_0')

legend("bottomright", lab, pch=c(15,15), col=c('red','blue'))
```


outcome
75
70
65
60
55
50
2
4
6
8
10
n−th bootstrap interval
treatment_1
treatment_0