Bios 6301: Assignment 9

Max Rohde

```
library(tidyverse)
library(cowplot)
library(glue)
library(extrafont)
library(directlabels)
library(resample)
```

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
}
```

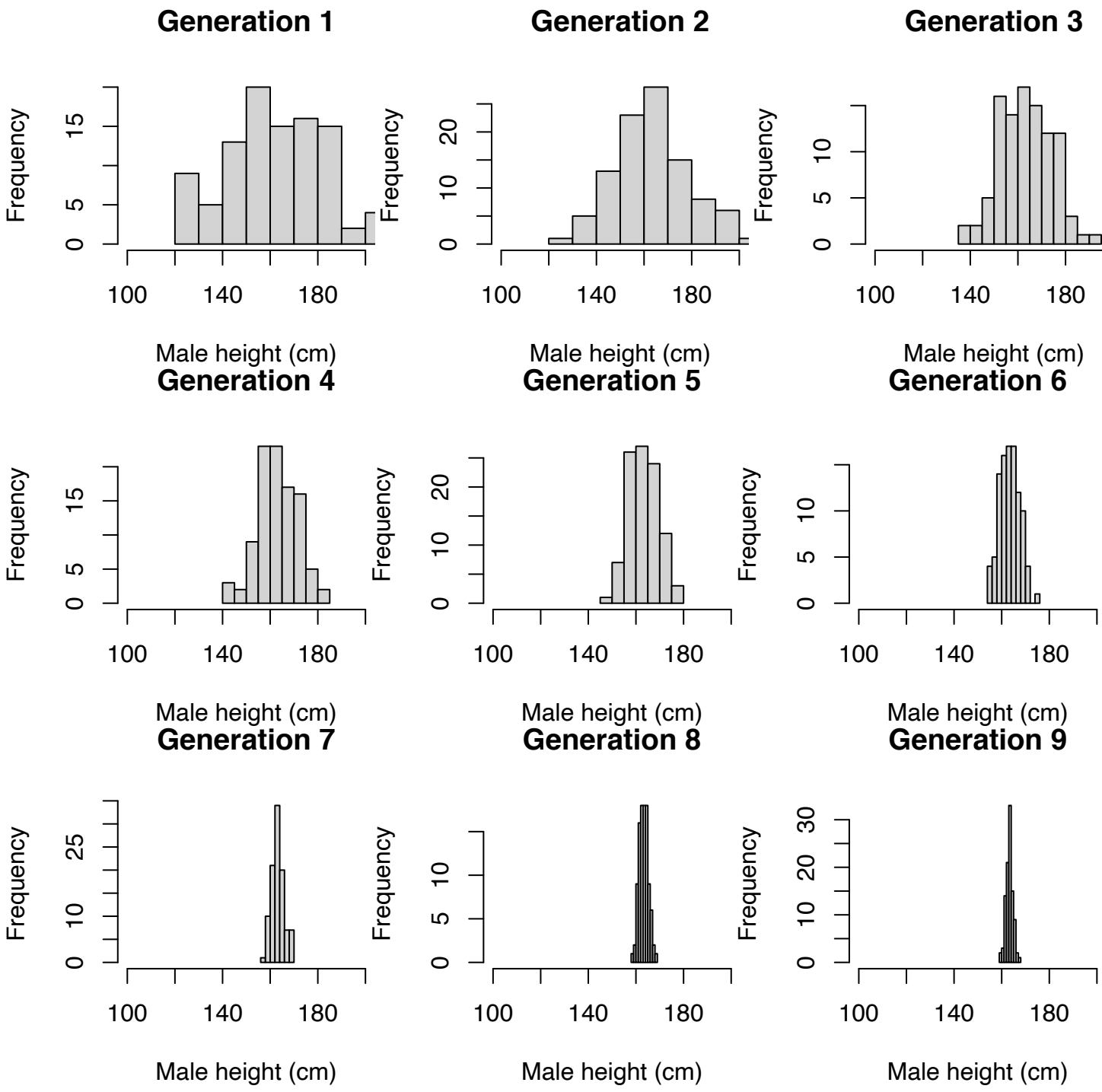
Use the function `next_gen` to generate nine generations (you already have the first), then use the function `hist` to plot the distribution of male heights in each generation (this will require multiple calls to `hist`). The phenomenon you see is called regression to the mean. Provide (at least) minimal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

```
x <- list(pop)

# Generate next 8 generations
for (i in 1:8) {
  x[i + 1] <- list(next_gen(x[[i]]))
}

for (i in 1:9) {
  df <- x[[i]]
```

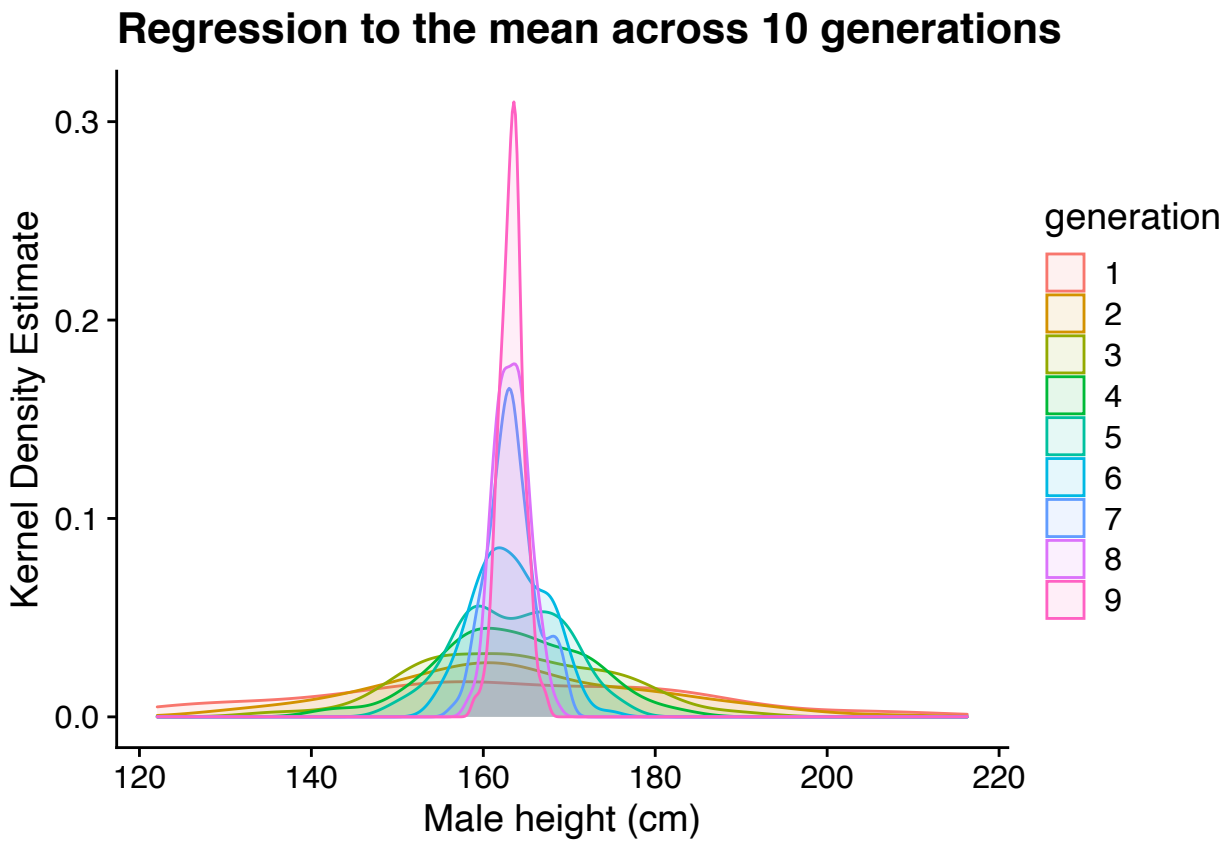
```
  hist(df$m,
    xlim = c(100, 200),
    main = glue("Generation {i}"),
    xlab = "Male height (cm)"
  )
}
```




```
# Here's a version for fun using ggplot
```

```r
# Merge into a single data frame
df <- map_dfr(x, ~.x, .id = "generation")
df$generation <- as.numeric(df$generation) %>% as.factor()

# Generate the plot
df %>%
  ggplot(aes(x = m, color = generation, fill = generation)) +
  geom_density(alpha = 0.1) +
  theme_cowplot() +
  labs(
    title = "Regression to the mean across 10 generations",
    x = "Male height (cm)",
    y = "Kernel Density Estimate"
  )
```



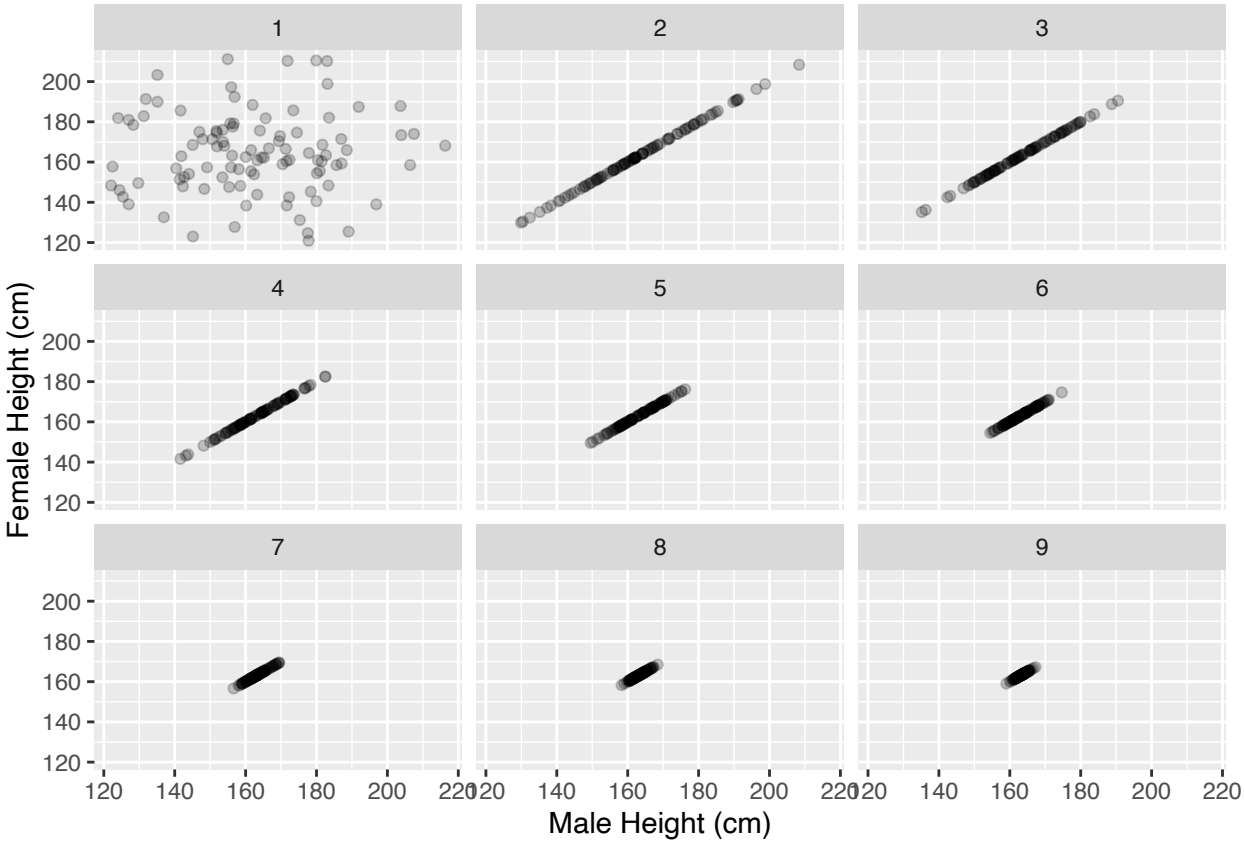
Question 2

10 points

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

```r
df %>%
  ggplot(aes(x = m, y = f)) +
  geom_point(alpha = 0.2) +
  facet_wrap(~generation) +
  labs(
    x = "Male Height (cm)",
    y = "Female Height (cm)"
```

)



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)

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)

```
set.seed(7)

# A function to generate 95% Bootstrap intervals for the mean
# Uses the `resample` package
get_boot_ci <- function(x, boot_size) {
  boot <- bootstrap(data = x, R = boot_size, statistic = mean)
  ci <- CI.percentile(boot, probs = c(0.025, 0.975))
  out <- tibble(ll = ci[1], ul = ci[2])
  return(out)
}
```

```r
# Generates the mean and CIs for a given sample size
generate <- function(n) {
  boot_size <- 1000
  group <- sample(c("Control", "Treatment"), n, replace = TRUE)
  outcome <- rnorm(n, 60, 20)
  outcome[group == "Treatment"] <- outcome[group == "Treatment"] + 5

  df <- tibble(group, outcome)

  df %>%
    group_by(group) %>%
    summarize(
      mean = mean(outcome),
      get_boot_ci(outcome, boot_size)
    ) %>%
    mutate(
      sample_size = n,
      boot_size = boot_size
    ) -> out
}

# Generate the data for each sample size
df <- map_df(seq(250, 2500, by = 250), ~ generate(.x))

head(df)
```

group	mean	ll	ul	sample_size	boot_size
Control	62.43368	58.22468	66.59220	250	1000
Treatment	63.39512	60.25821	66.58103	250	1000
Control	59.48290	57.18607	61.62245	500	1000
Treatment	64.51204	62.03150	66.92945	500	1000
Control	61.07223	58.90373	63.10327	750	1000
Treatment	65.60852	63.60847	67.70256	750	1000

```r
# Plot the data
df %>%
  group_by(group, sample_size) %>%
  ggplot(aes(x = sample_size, y = mean, color = group)) +
  geom_point() +
  geom_line() +
  geom_ribbon(aes(ymin = ll, ymax = ul, fill = group), alpha = 0.3, size = 0) +
  scale_color_brewer(palette = "Dark2") +
  scale_fill_brewer(palette = "Dark2") +
  theme_cowplot() +
  annotate(
    geom = "text", x = 2350, y = 67, label = "Treatment",
    color = "#d85e01", family = "Source Sans Pro"
  ) +
  annotate(
    geom = "text", x = 2350, y = 61.5, label = "Control",
    color = "#1c9e77", family = "Source Sans Pro"
  ) +
  theme(
```

```
  text = element_text(size = 12, family = "Source Sans Pro"),
  legend.position = "none"
) +
labs(
  title = "95% Bootstrap intervals for increasing sample sizes",
  x = "Sample Size",
  y = "Mean"
)
```