Homework 3 Solutions

Due: Thursday 2/20/20 by 8:30am

Rubric:

- Maximum of 2 points each for 1. and 3., determined as follows:
  - 0 points for no solutions whatsoever or incomplete solutions;
  - 1 point for solutions provided for each part, but at least one incorrect solution;
  - 2 points for correct solutions to each part;
- Maximum of 3 points for 2. and 4., determined as follows:
  - 0 points for no solutions whatsoever or `R` output only;
  - 1 point for an honest effort but very few correct answers or `R` output only plus a figure;
  - 2 points for mostly correct answers but at least one substantial issue;
  - 3 points for nearly/exactly correct.

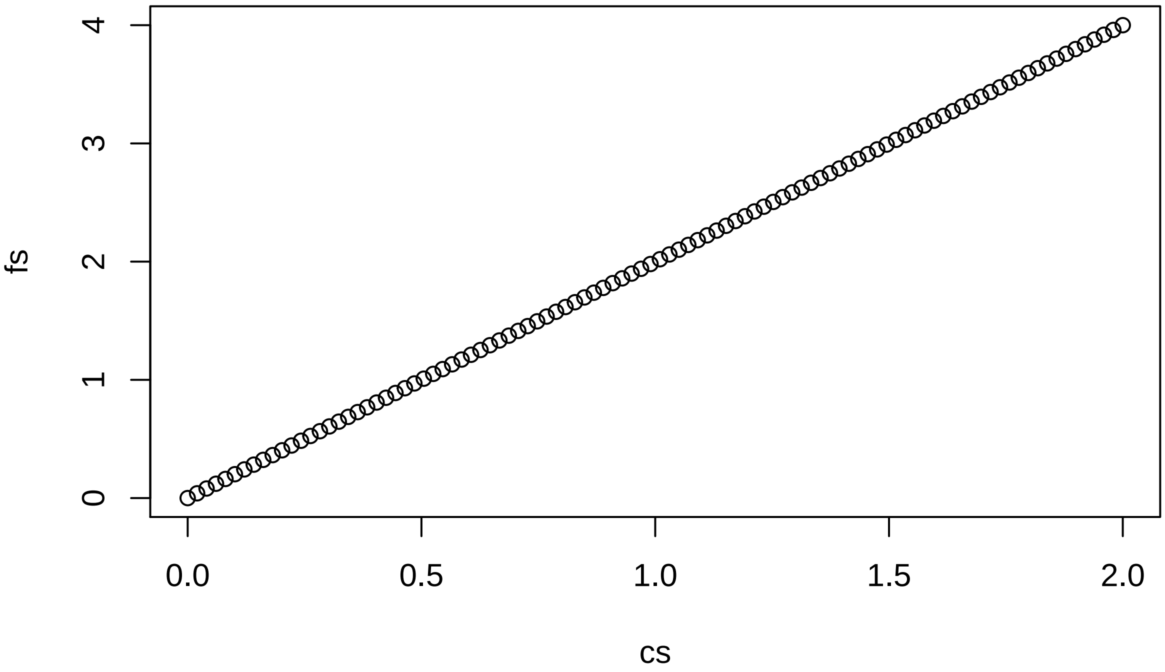
This homework assignment focuses on material covered in Chapter 1 of the textbook. You may find the following three `R` hints useful:

- All text on a line that follows the `#` sign is treated as a comment, i.e. ignored by `R` when you run the line of code.
- Suppose you are asked to plot a function of some variable, let's call it $f(c) = 2c$. Then you will need to choose the values of c to consider, and then evaluate the function at each of those values.

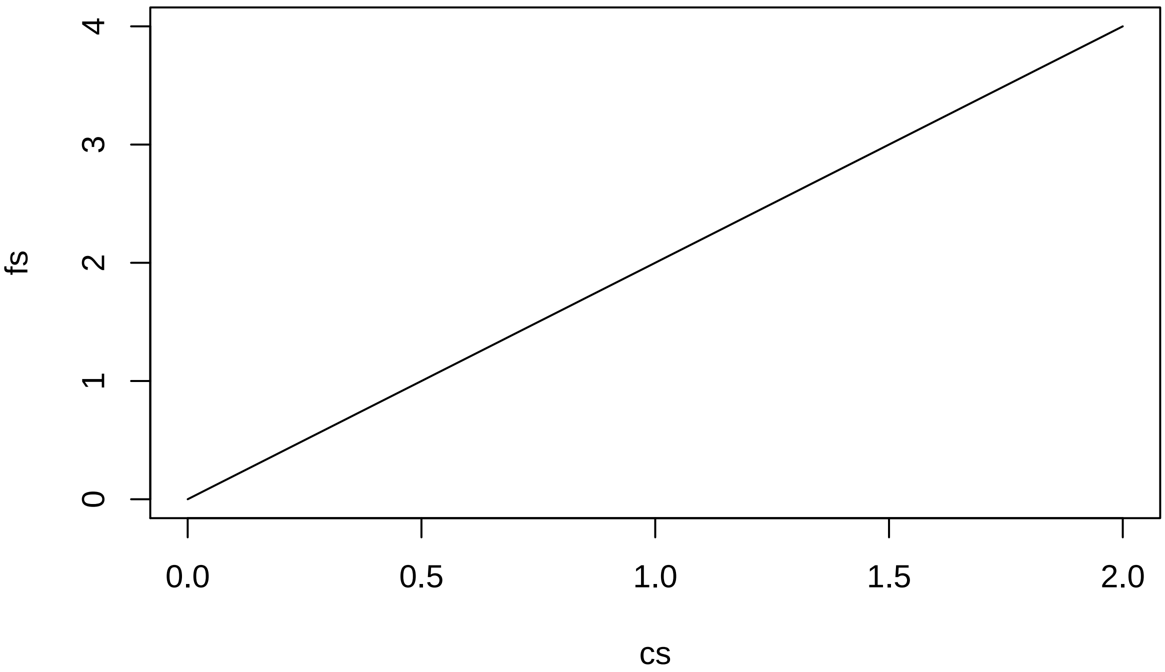
```r
# First, choose values of c to consider. Let's look at values of c between 0 and 2
cs <- seq(0, 2, length.out = 100) # This returns 100 values from 0 to 2
# Alternatively: cs <- seq(0, 2, by = 0.01) # This returns values from 0 to 2,
# each 0.01 apart

# Now, we need to create an empty vector to store the function values in
# Note: We want to have the same number of function values as values of c
fs <- rep(NA, length(cs)) # This makes a vector with the same length as cs,
# Each element fs[i] for i = 1,...,100 is defined as NA, which is R's way
# of indicating a missing value

# Now we're going to go one-by-one through the values in cs and record the
# function value in fs
for (i in 1:length(cs)) {
  fs[i] <- cs[i]*2
}
# Ok! Now we can make our plot by just plotting cs and fs against each other
plot(cs, fs) # The default is to make a plot of points
```



```
plot(cs, fs, type = "l") # Sometimes we might prefer to make a line plot by connecting the points
```



- Suppose you are asked to simulate n independent normal random variables $x_1, \ldots, x_n$ with different means m_i, but the same variance $\sigma^2 = 1$. Suppose that you already have a variable `n` that gives the number of random variables you want to simulate n, and an $n \times 1$ vector `m` that gives the means for each of the n random variables. You can do this as follows:

```
x <- rnorm(n, m, 1) # Simulates the n random variables
x[1] # Extracts and prints the first value
x[2] # Extracts and prints the second value
```

1. Suppose Instagram magically knew that every time the number of times user i purchases a product,

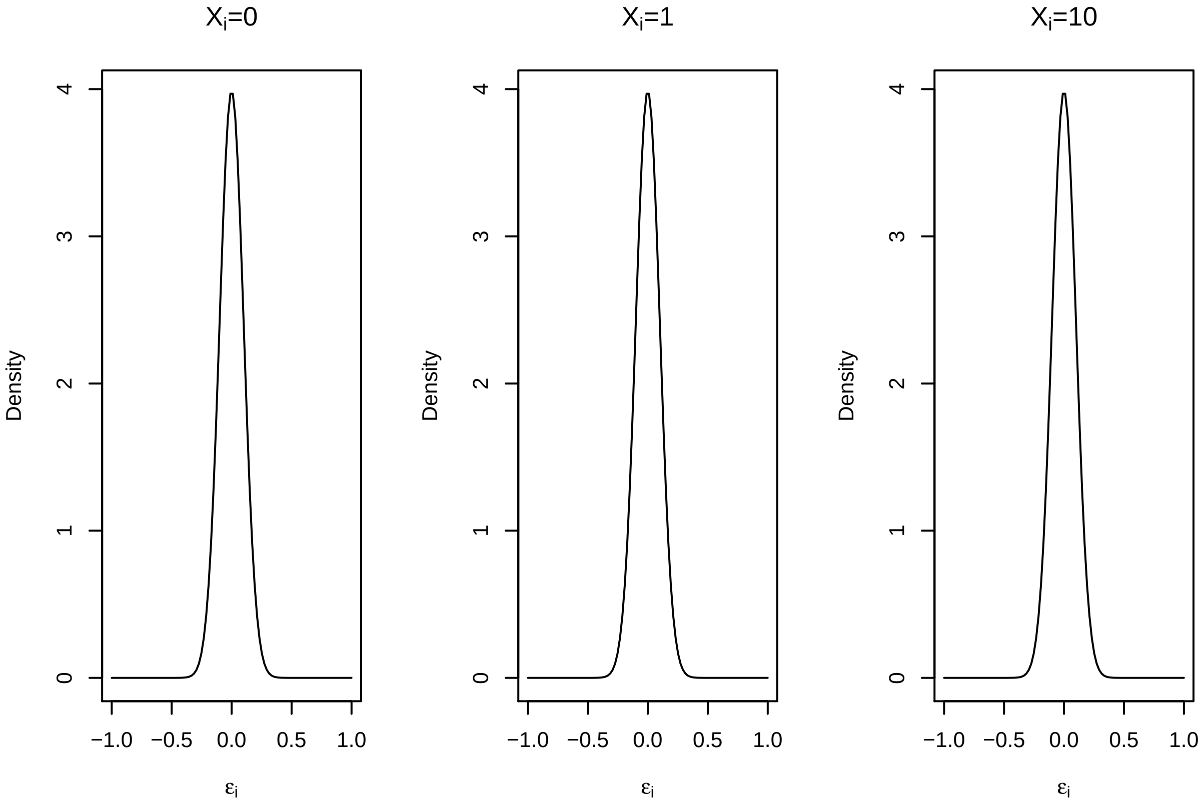
denoted by Y_i, is related to the number of times the product has been advertised to user i, denoted by X_i, as follows:

$$Y_i = 1 + 2X_i + \epsilon_i$$

where ϵ_i is a *normal* random error term with mean $E\{\epsilon_i\} = 0$ and variance $\sigma^2\{\epsilon_i\} = 0.1$; ϵ_i and ϵ_j are uncorrelated so that their covariance is zero (i.e., $\sigma\{\epsilon_i, \epsilon_j\} = 0$ for all $i \neq j$) for $i = 1, \dots n$.

(a) Using R, make a plot with three panels. You can make a single plot with three panels by typing `par(mfrow = c(1, 3))` before running any lines of code that create plots. Plot the density of the errors ϵ_i for $X_i = 0$, $X_i = 1$, and $X_i = 10$, using a separate panel for each value of X_i. Ensure that the axes are the same across all three plots.

```
par(mfrow = c(1, 3))
eps.vals <- seq(-1, 1, length.out = 100)
dens.vals <- rep(NA, length(eps.vals))
for (i in 1:length(dens.vals)) {
  dens.vals[i] <- dnorm(eps.vals[i], mean = 0, sd = sqrt(0.01))
}
plot(eps.vals, dens.vals, type = "l", xlab = expression(epsilon[i]), ylab = "Density",
     main = expression(paste(X[i], "=0", sep = "")))
plot(eps.vals, dens.vals, type = "l", xlab = expression(epsilon[i]), ylab = "Density",
     main = expression(paste(X[i], "=1", sep = "")))
plot(eps.vals, dens.vals, type = "l", xlab = expression(epsilon[i]), ylab = "Density",
     main = expression(paste(X[i], "=10", sep = "")))
```



The density of the errors ϵ_i does not depend on X_i at all, so all three plots are identical.

(b) Based on the assumed model and the information provided, can we conclude that the number of times user i purchases a product Y_i is independent of the number of times user j purchases a product Y_j?

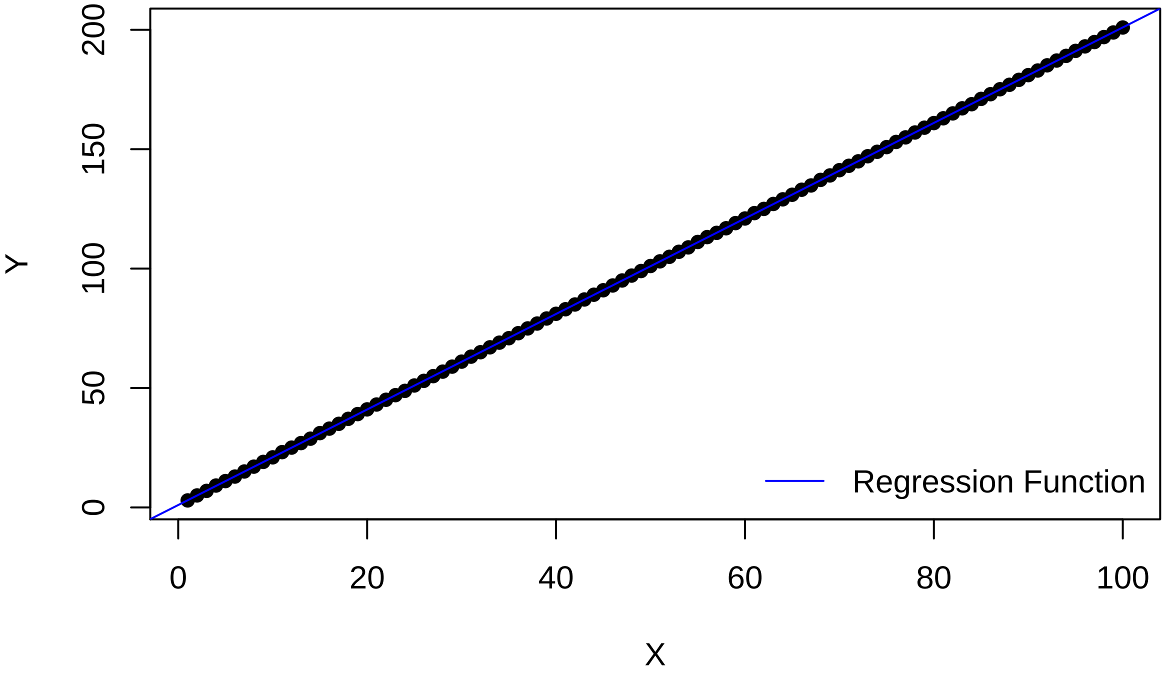
Yes.

(c) Based on the assumed model and the information provided, can we state the exact probability that a single value Y_i will be greater than 4 given that $X_i = 1$?

Yes.

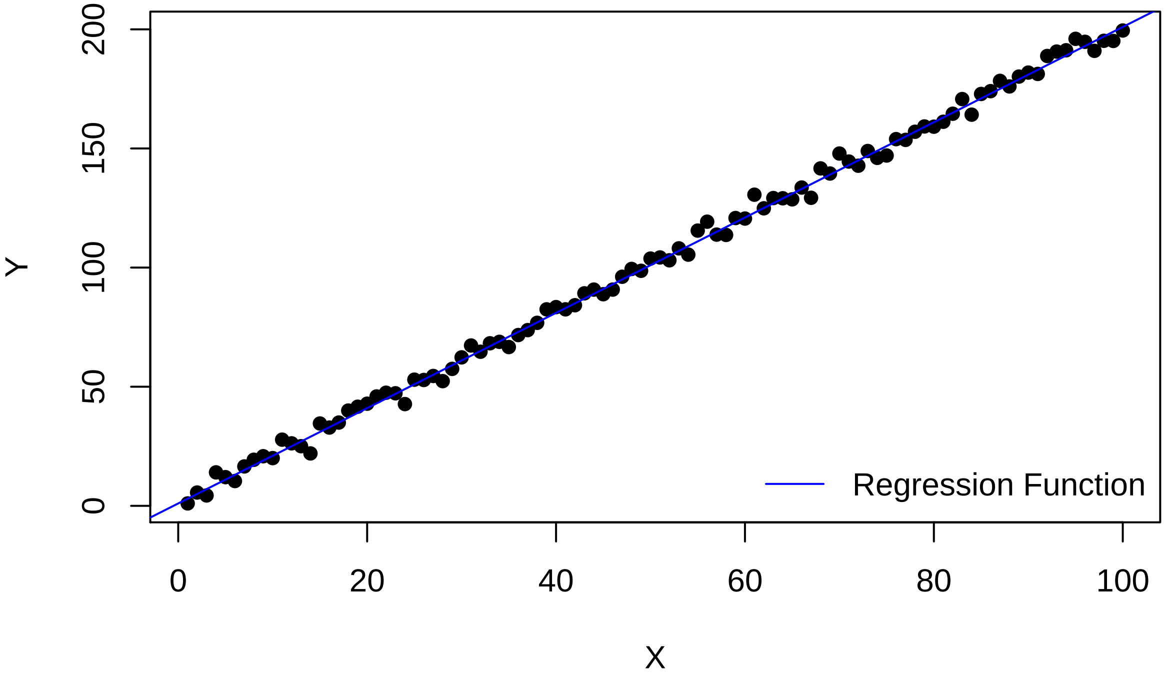
(d) Simulate $n = 100$ observations from the model, with $X_i = i$. Using `R`, make a scatter plot of the data and overlay the regression function on the scatterplot.

```r
set.seed(1)
n <- 100
X <- seq(1, n, by = 1)
means <- 1 + 2*X
Y <- rnorm(n, means, sd = sqrt(0.01))
plot(X, Y, xlab = "X", ylab = "Y", pch = 16)
abline(a = 1, b = 2, col = "blue")
legend("bottomright", lty = 1, col = "blue", legend = "Regression Function",
       bty = "n")
```



(e) Repeat (d), but instead of assuming that $\sigma^2\{\epsilon_i\} = 0.1$, assume that $\sigma^2\{\epsilon_i\} = 10$. In at most one sentence, describe how increasing $\sigma^2\{\epsilon_i\}$ changes how the regression function relates to the scatter plot.

```r
set.seed(1)
n <- 100
X <- seq(1, n, by = 1)
means <- 1 + 2*X
Y <- rnorm(n, means, sd = sqrt(10))
plot(X, Y, xlab = "X", ylab = "Y", pch = 16)
abline(a = 1, b = 2, col = "blue")
legend("bottomright", lty = 1, col = "blue", legend = "Regression Function",
       bty = "n")
```



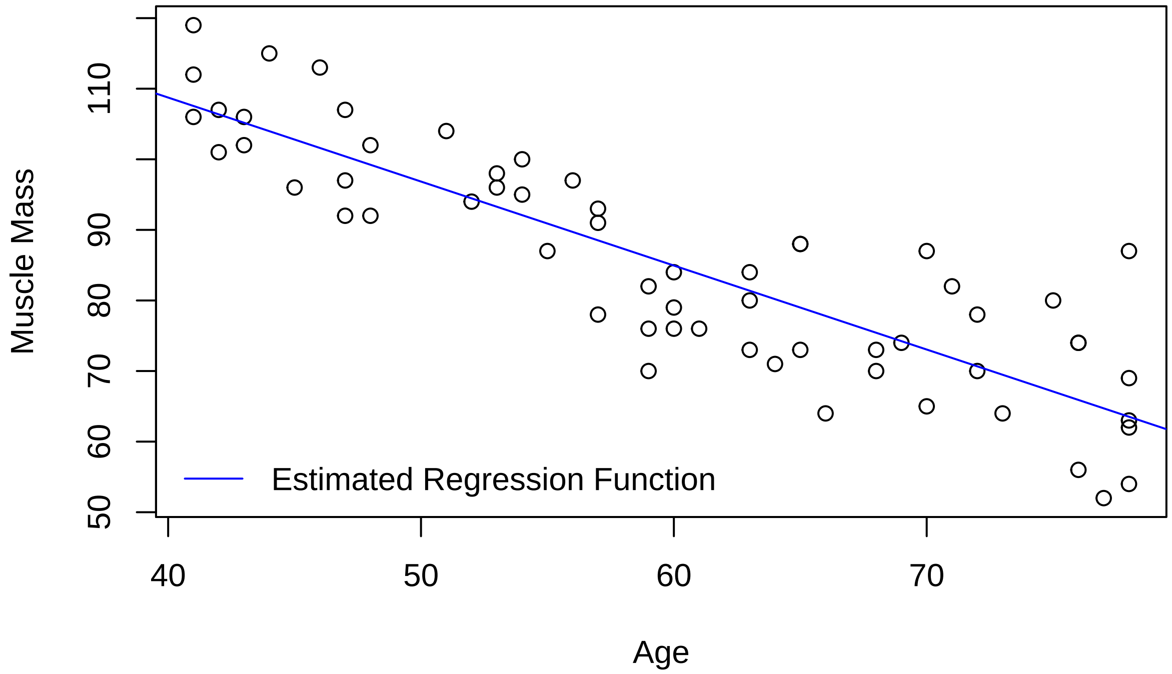
When we increase σ^2, the deviations of the response `Y` from the regression line increase in magniture, i.e. the errors become bigger.

2. Problem 1.27 from the `.pdf` version of the textbook. Requires use of the `muscle` data that has been posted on the Homework page.

(a)

```r
load("~/Dropbox/Teaching/STAT525-2020/stat525/content/homework/muscle.RData")
X <- data$X
Y <- data$Y
linmod <- lm(Y~X)

plot(X, Y, xlab = "Age", ylab = "Muscle Mass")
abline(a = linmod$coef[1], b = linmod$coef[2], col = "blue")
legend("bottomleft", lty = 1, col = "blue", legend = "Estimated Regression Function",
       bty = "n")
```



A linear regression does appear to give a good fit here, because the errors appear to be evenly scattered about the regression line across all of the ages. The plot does support my anticipation that muscle mass decreases with age, because the estimated regression line is decreasing with age.

(b)

```
b1 <- linmod$coef[2]
b0 <- linmod$coef[1]
y.hat <- b0 + b1*60
R <- Y - b0 - b1*X
n <- length(Y)
s.sq <- sum(R^2)/(n - 2)
```

We obtain (a) $b_1 = -1.19$ for a point estimate of the difference in th emean muscle mass for women differing in age by one year, (2) $\hat{Y} = 84.95$ for a point estimate of the mean muscle mass for women aged $X = 60$ years, (3) $r_8 = 4.44$ for the value of the residual for the eight case, and $s^2 = 66.8$ for a point estimate of σ^2.

3. Problem 1.34 from the `.pdf` version of the textbook.

The least squares estimator of b_0 of β_0 is unbiased if $E\left\{b_0\right\} = \beta_0$. The solution below does not assume $\beta_1 = 0$, but you received full credit if you assumed $\beta_1 = 0$.

$$\begin{aligned}
E\left\{b_0\right\} &= E\left\{\bar{Y} - b_1\bar{X}\right\} \\
&= \frac{1}{n}\sum_{i=1}^{n} E\left\{Y_i\right\} - E\left\{b_1\right\}\bar{X} \\
&= \frac{1}{n}\sum_{i=1}^{n} E\left\{\beta_0 + X_i\beta_1 + \epsilon_i\right\} - E\left\{b_1\right\}\bar{X} \\
&= \beta_0 + \left(\beta_1 - E\left\{b_1\right\}\right)\bar{X}
\end{aligned}$$

We've almost shown that $E\left\{b_0\right\} = \beta_0$, we just need to show that $E\left\{b_1\right\} = \beta_1$.

$$
\begin{aligned}
E\left\{b_1\right\} &= E\left\{\frac{\sum_{i=1}^{n}\left(X_i-\bar{X}\right)\left(Y_i-\bar{Y}\right)}{\sum_{i=1}^{n}\left(X_i-\bar{X}\right)^2}\right\} \\
&= \sum_{i=1}^{n}\left(\frac{\left(X_i-\bar{X}\right)}{\sum_{i=1}^{n}\left(X_i-\bar{X}\right)^2}\right)E\left\{\left(Y_i-\bar{Y}\right)\right\} \\
&= \sum_{i=1}^{n}\left(\frac{\left(X_i-\bar{X}\right)}{\sum_{i=1}^{n}\left(X_i-\bar{X}\right)^2}\right)\left(\beta_0+\beta_1X_i-\beta_0-\beta_1\bar{X}\right) \\
&= \sum_{i=1}^{n}\left(\frac{\left(X_i-\bar{X}\right)}{\sum_{i=1}^{n}\left(X_i-\bar{X}\right)^2}\right)\beta_1\left(X_i-\bar{X}\right) \\
&= \beta_1
\end{aligned}
$$

Plugging this in to our simplified expression for $E\left\{b_0\right\}$, we get

$$
\begin{aligned}
E\left\{b_0\right\} &= \beta_0+\left(\beta_1-\beta_1\right)\bar{X} \\
&= \beta_0,
\end{aligned}
$$

i.e. b_0 is an unbiased estimator of β_0.

4. Problem 1.42 from the `.pdf` version of the textbook.

(a)

The likelihood function for the six Y observations, for $\sigma^2 = 16$, is

$$
\prod_{i=1}^{6}\frac{1}{\sqrt{2\pi 16}}\exp\left\{-\frac{1}{2\times 16}\left(Y_i-\beta_1X_i\right)^2\right\} = \frac{1}{\sqrt{2\pi 16}^6}\exp\left\{-\frac{1}{2\times 16}\sum_{i=1}^{6}\left(Y_i-\beta_1X_i\right)^2\right\}
$$

(b)

```
Y <- c(128, 213, 75, 250, 446, 540)
X <- c(7, 12, 4, 14, 25, 30)

beta1s <- c(17, 18, 19)
lls <- rep(NA, 3)
for (i in 1:length(lls)) {
  lls[i] <- (1/sqrt(2*pi*16))^6*exp(-(1/(2*16))*sum((Y - beta1s[i]*X)^2))
}
```

The likelihood function for $\beta_1 = 17$, 18, and 19 is given by $9.4513295 \times 10^{-30}$, 2.6490426×10^{-7}, and $3.0472851 \times 10^{-37}$, respectively. The likelihood function is largest for $\beta_1 = 18$.

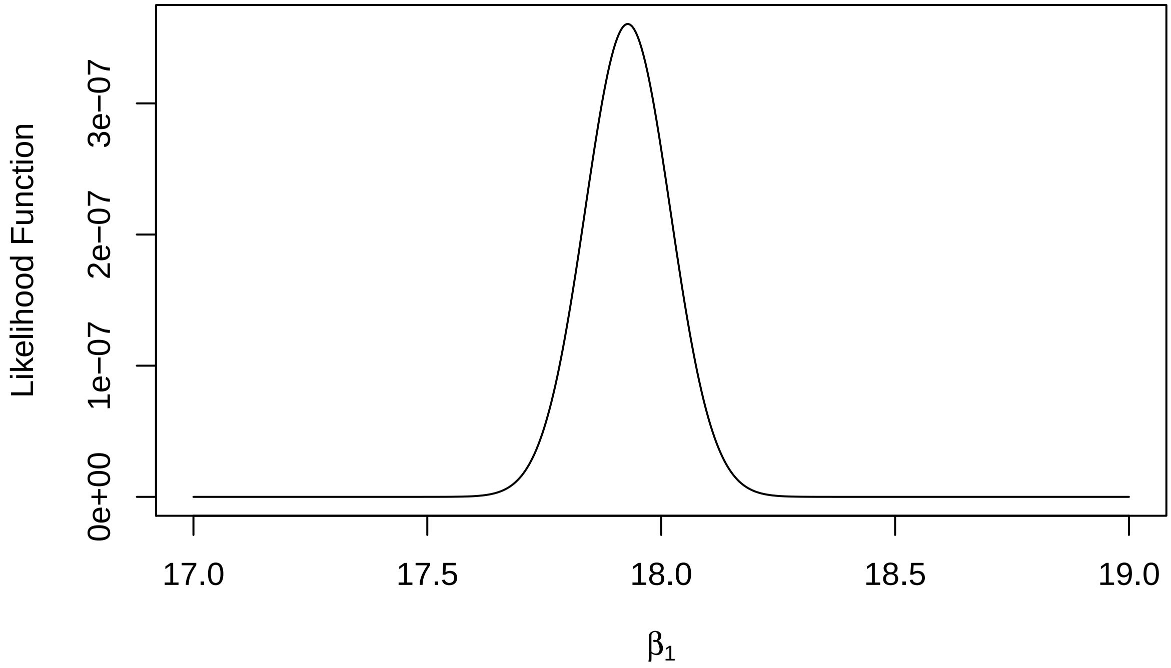
(c)

```
b1 <- sum(X*Y)/(sum(X^2))
```

The maximum likelihood estimator is $b_1 = 17.93$, which is very close to 18, which is the value of β_1 for which the likelihood was largest in (b).

(d)

```
beta1s <- seq(17, 19, length.out = 500)
lls <- rep(NA, length(beta1s))
for (i in 1:length(lls)) {
  lls[i] <- (1/sqrt(2*pi*16))^6*exp(-(1/(2*16))*sum((Y - beta1s[i]*X)^2))
}
plot(beta1s, lls, type = "l", xlab = expression(beta[1]), ylab = "Likelihood Function")
```



The likelihood is maximized at $\beta_1 = 17.93$, which is identical to the maximum likelihood estimate found in (c).

5. Integrative Experience Step 2, as described in `ieproject.pdf` on the Project page.