Problem Set #12

Econ 103

Part I – Problems from the Textbook

Since I have assigned two even-numbered problems from the book, I will post solutions to these along with those for the "Additional Problems."

12-2, 12-3, 13-5, 13-12, 14-1, 14-3, 14-5

For 13-5, part (e) you can download the data from my website as follows:

```
data.url <- 'http://www.ditraglia.com/econ103/ex_13_5.csv'
election <- read.csv(data.url)
head(election)
```

```
##   year    y x1  x2
## 1 1946  7.3 32 -40
## 2 1950  2.0 43 100
## 3 1954  2.3 65 -10
## 4 1958  5.9 56 -10
## 5 1962 -0.8 67  60
## 6 1966  1.7 48 100
```

Similarly, the data for 14-5 can be downloaded as follows:

```
data.url <- 'http://www.ditraglia.com/econ103/ex_14_5.csv'
bpdata <- read.csv(data.url)
head(bpdata)
```

```
##   D WEIGHT BP
## 1 0    180 81
## 2 0    150 75
## 3 0    210 83
## 4 0    140 74
```

```
## 5 0    160 72
## 6 0    160 80
```

Solution: 12-2 All of these are based on the approximation:

$$SE(\widehat{\beta}_1) \approx \frac{\sigma}{\sqrt{n}} \cdot \frac{1}{s_X}$$

Solution: 12-2(a) Since n is multiplied by four, the new SE is half as big as before.

Solution: 12-2(b) This multiplies s_x by four, so the new SE is one fourth as large as before.

Solution: 12-2(c) Here, n is divided by two, which increases the SE by a factor of $\sqrt{2}$. At the same time, however, s_x increases by a factor of 2. The net effect is a decrease in SE by a factor of $\sqrt{2}$.

Solution: 12-2(d) Since σ^2 is reduced by a factor two, σ is reduced by a factor of $\sqrt{2}$. At the same time, s_x increases by a factor of five. The net effect is a decrease in SE by a factor $5\sqrt{2}$.

Solution: 13-12 Each of these answers refers to the following fitted regression:

$$\begin{array}{rccccccccccc} \widehat{S} = & 230B & + & 18A & + & 100E & + & 490D & + & 190Y & + & 50T & + & \ldots \\ & (86) & & (8) & & (28) & & (60) & & (17) & & (370) & & \end{array}$$

where the standard errors appear in parentheses below each coefficient estimate.

Solution: 13-12(a)

	B	A	E	D	Y	T
SE	86	8	28	60	17	370
95% CI	230 ± 172	18 ± 16	100 ± 56	490 ± 120	190 ± 34	50 ± 740
t ratio	2.67	2.25	3.57	8.17	11.18	0.14
p-value	0.007	0.024	< 0.001	< 0.001	< 0.001	0.893

Solution: 13-12(b) We have strong evidence that each of the predictors except T, teaching score, is associated with higher professor salaries. The predictors that seem to be associated with the largest differences in salary are books written, PhDs supervised, and years of experience.

Solution: 13-12(c)

(i) FALSE. The last sentence should be changed to: "The distribution of these estimates would be centered around the true population value." The point is that we do not know what the true population value is equal to: the value 230 is merely our *estimate* from a particular study.

(ii) FALSE. Each instance of "one or more" should be changed to "one more." Consider two professors, A and B, who have written the sample number of ordinary articles and the same number of excellent articles, who supervise the same number of PhDs, who have the same number of years experience, and who have the same teaching score. Professor B, however, has written *one more book* than Professor A. Then, we would predict that Professor B earns an average of 230 dollars more per year than Professor A.

(iii) TRUE.

Solution: 13-12(d) See my answer to part (ii) of (c) above. Just change the predictor that is allowed to vary by one unit and hold the others constant across professors A and B.

Part II – Additional Problems

1. This question is based on the dataset on child test scores and mother characteristics we studied during our final lecture of the semester. Before working on this question, make

sure you've installed the package `arm` in RStudio. You can download the data from:

`www.ditraglia.com/econ103/child_test_data.csv`

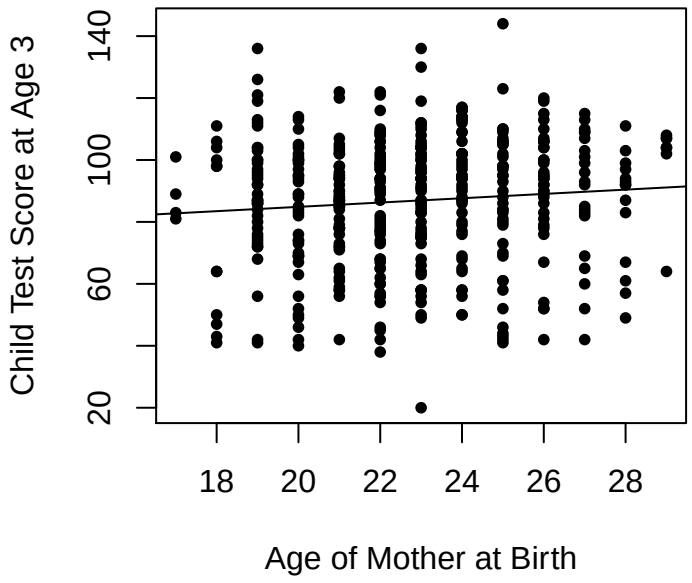
The columns contained in this dataset are as follows:

Variable Name	Description
`kid.score`	Child's Test Score at Age 3
`mom.age`	Age of Mother at Birth of Child
`mom.hs`	Mother Completed High School? (1 = Yes)
`mom.iq`	Mother's IQ Score

(a) Run a regression of `kid.score` on `mom.age`. Plot both the data and the fitted regression line, making sure to label the axes. Interpret the results. At what age do you recommend mothers give birth? What assumptions must you make to justify your recommendation?

Solution:

```
library(arm)
data.url <- "http://www.ditraglia.com/econ103/child_test_data.csv"
data <- read.csv(data.url)
attach(data)
## The following objects are masked from data (position 8):
##
##     kid.score, mom.age, mom.hs, mom.iq
## The following objects are masked from data (position 9):
##
##     kid.score, mom.age, mom.hs, mom.iq
reg1 <- lm(kid.score ~ mom.age)
display(reg1)
## lm(formula = kid.score ~ mom.age)
##             coef.est coef.se
## (Intercept) 70.96     8.31
## mom.age      0.70     0.36
## ---
## n = 434, k = 2
## residual sd = 20.35, R-Squared = 0.01
plot(mom.age, kid.score, pch = 20, xlab = 'Age of Mother at Birth',
     ylab = 'Child Test Score at Age 3')
coefficients(reg1)
## (Intercept)     mom.age
##     70.9569      0.6952
intercept <- coef(reg1)[1]
slope <- coef(reg1)[2]
abline(a = intercept, b = slope)
```

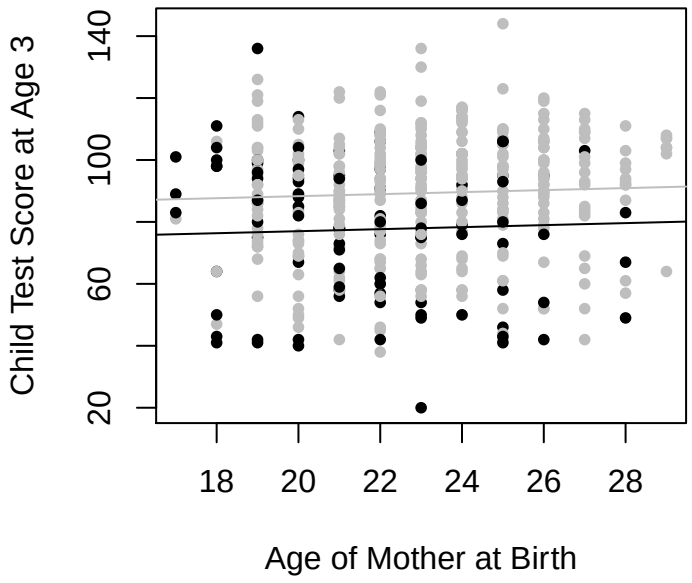


Our model suggests that the children of mothers who were older when they gave birth tend to score higher. In particular, comparing two children whose mothers' age at birth differed by one year, we would predict that the child of the older mother will score, on average, 0.7 points higher. The standard error associated with the estimate, however, is fairly large. An approximate 95% CI would just barely include zero. Nevertheless, this result is suggestive that the children of older mothers do better on the test. This would seem to suggest that women should wait to have children until they are as old as possible. However, for this advice to truly be valid, it would have to be the case that being older when you give birth *caused* your child to have higher test scores. This seems unlikely. For one, we know that the incidence of birth defects (including those that affect mental ability), increases with mother's age during pregnancy. Further, teenage pregnancy is correlated with economic disadvantage and lower levels of education. There are many possible confounders here.

(b) Augment your model from part (a) by allowing a different intercept for children whose mother completed high school. Plot the data along with the regression lines for each group (those whose mother completed high school and those whose mother did not). Interpret your results and compare them to those you got in part (a).

Solution:

```
reg2 <- lm(kid.score ~ mom.hs + mom.age)
display(reg2)
## lm(formula = kid.score ~ mom.hs + mom.age)
##             coef.est coef.se
## (Intercept) 70.48     8.11
## mom.hs      11.31     2.38
## mom.age      0.33     0.36
## ---
## n = 434, k = 3
## residual sd = 19.86, R-Squared = 0.06
coef(reg2)
## (Intercept)      mom.hs     mom.age
##     70.4787     11.3112      0.3261
slope <- coef(reg2)[3]
intercept.hs <- coef(reg2)[1] + coef(reg2)[2]
intercept.no.hs <- coef(reg2)[1]
colors <- ifelse(mom.hs == 1, 'gray', 'black')
plot(mom.age, kid.score, , ylab = 'Child Test Score at Age 3', xlab = 'Age of M
abline(a = intercept.hs, b = slope, col = 'gray')
abline(a = intercept.no.hs, b = slope, col = 'black')
```

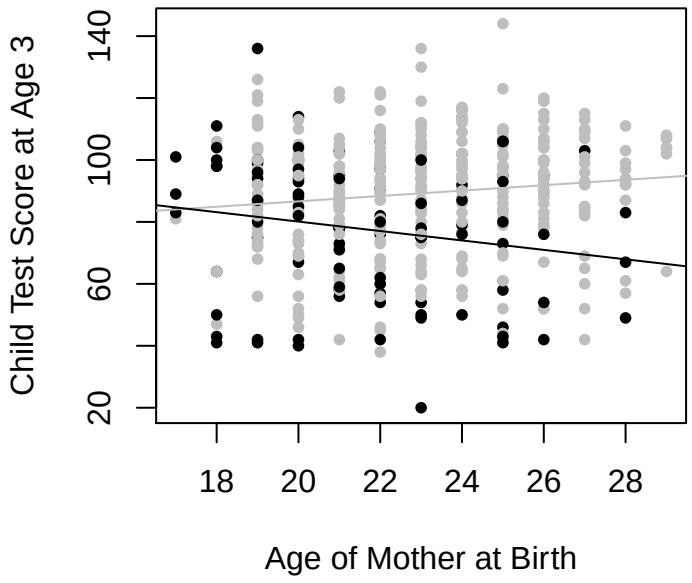


By adding a dummy variable that equals one if a child's mother completed high school, we have controlled for one of the possible confounders from above: mother's level of education. We have done this by allowing the regression line to have a different intercept depending on mother's education. Comparing two children whose mothers are of the same age but only one whom attended high school, we predict that the child of the better educated mother will score, on average, 11 points higher. The standard error associated with this estimate is quite small, yielding a 95% CI that is is nowhere near zero. We have strong evidence of a large effect from mother's education level. In contrast, once we've controlled from mother's education, the estimated effect of `mom.age` falls substantially while the associated standard error stays the same. This results in an approximate 95% CI that includes many negative values. After controlling for mother's education, there is much less evidence to suggest that older mothers have higher-scoring children. In terms of predictive accuracy, the second model is slightly better but neither is particularly effective: we are only predicting test scores to an accuracy of about 20 points.

(c) Now allow different slopes as well as intercepts for each group (those whose mother completed high school and those whose mother did not). Plot the data and the regression lines for each group and interpret your results.

Solution:

```
reg3 <- lm(kid.score ~ mom.hs + mom.age + mom.hs:mom.age)
display(reg3)
## lm(formula = kid.score ~ mom.hs + mom.age + mom.hs:mom.age)
##                coef.est coef.se
## (Intercept)    110.54    16.45
## mom.hs         -41.29    18.99
## mom.age         -1.52     0.75
## mom.hs:mom.age   2.39     0.86
## ---
## n = 434, k = 4
## residual sd = 19.70, R-Squared = 0.07
coef(reg3)
##    (Intercept)         mom.hs        mom.age mom.hs:mom.age
##        110.542        -41.287         -1.522          2.391
intercept.no.hs <- coef(reg3)[1]
intercept.hs <- coef(reg3)[1] + coef(reg3)[2]
slope.no.hs <- coef(reg3)[3]
slope.hs <- coef(reg3)[3] + coef(reg3)[4]
plot(mom.age, kid.score, xlab = 'Age of Mother at Birth', pch = 20, col = color
abline(a = intercept.hs, b = slope.hs, col = 'gray')
abline(a = intercept.no.hs, b = slope.no.hs, col = 'black')
```



This is very interesting! When we allow for different slopes as well as intercepts, by adding an *interaction* between `mom.hs` and `mom.hs`, namely `mom.hs:mom.age`, we find very different results depending on mother's education. (There is strong evidence that we should allow for different slopes, since the approximate 95% CI for the interaction does not include zero.) For children whose mothers attended high school, there is a *positive* relationship between mother's age at birth and child's test score. For children whose mothers did not attend high school, the relationship is *negative*. For children whose mothers were 18 then they gave birth, there is essentially *no* impact from mother's education level. As age of mother at birth increases, the impact of mother's education widens.

2. This example is based on 12-1 from WW4, but has been adapted somewhat for you to carry out in R. Suppose that the following expression gives the true relationship, i.e. the long-run average, between corn yield in tons per acre (Y) and the amount of fertilizer used in hundreds of pounds per acre (X)

$$Y = 2.40 + 0.30X$$

This means that the population regression parameters are $\beta_0 = 2.40$ and $\beta_1 = 0.30$. Normally we don't know these parameters but rather use data to estimate them. In this question, however, we will pretend that we know these parameters and carry out a

Monte Carlo simulation to understand how sampling variability works in the context of regression.

(a) Write an R function called `y.plus.noise` that takes as its input a vector `x` of X-values and returns the corresponding Y values from the above equation *plus a standard normal error term*. The error term should be a *different* random number for each element.

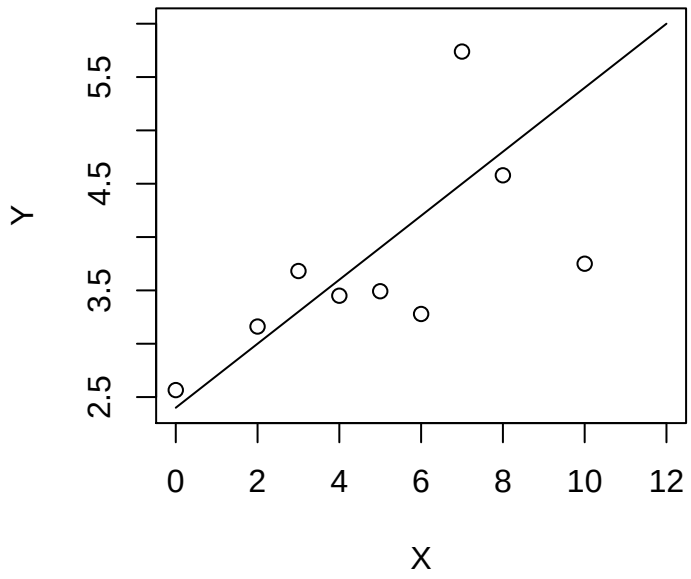
Solution:

```
y.plus.noise <- function(x){
  2.4 + 0.3 * x + rnorm(length(x))
}
```

(b) Define `x.test <- 0:12`, a vector containing all the integers from 0 to 12. Test our your function from part (a) by inputting `x.test` and assigning the result to `y.sim`. Make a plot of the function $Y = 2.40 + 0.30X$ along with the points `x.test` and `y.sim`. Try repeating this: you should get a different result because the error terms are *random*.

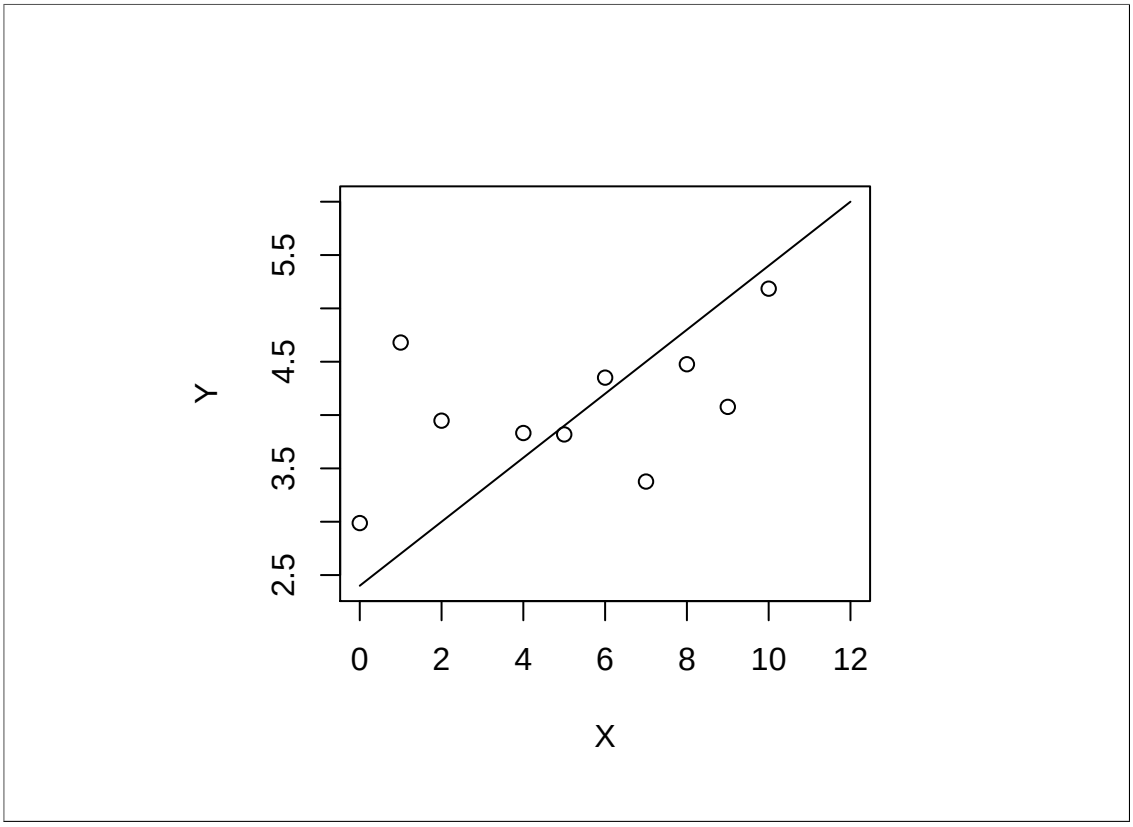
Solution:

```
x.test <- 0:12
y.sim <- y.plus.noise(x.test)
plot(x.test, 2.4 + 0.3 * x.test, type = 'l', xlab = 'X', ylab = 'Y')
points(x.test, y.sim)
```



Running it a second time:

```
y.sim <- y.plus.noise(x.test)
plot(x.test, 2.4 + 0.3 * x.test, type = 'l', xlab = 'X', ylab = 'Y')
points(x.test, y.sim)
```



(c) Run a regression of `y.sim` on `x.test` using the R command `lm(y.sim  x.test)`. Compare the estimated coefficients to the population regression parameters from above by adding the *fitted* regression line to your plot from part (b) using code similar to the following:

```
reg <- lm(y.sim ~ x.test)
estimates <- coefficients(reg)
a.estimate <- estimates[1]
b.estimate <- estimates[2]
abline(a = a.estimate, b = b.estimate, lty = 2)
```

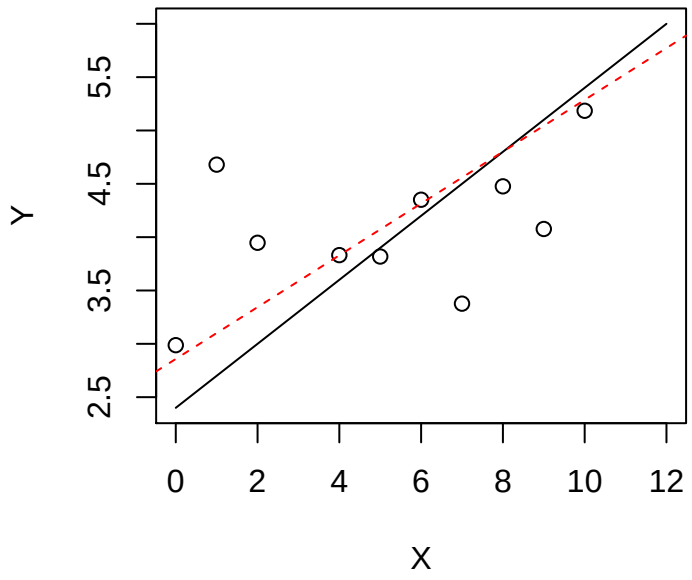
The command `coefficients` extracts the estimated regression coefficients as a numeric vector, while `abline` plots a line based on its intercept (a) and slope (b). Setting the parameter `lty = 2` gives a dashed line. If you repeat part (b) followed by part (c) what changes in the picture and what stays the same?

Solution:

```r
lm(y.sim ~ x.test)
```

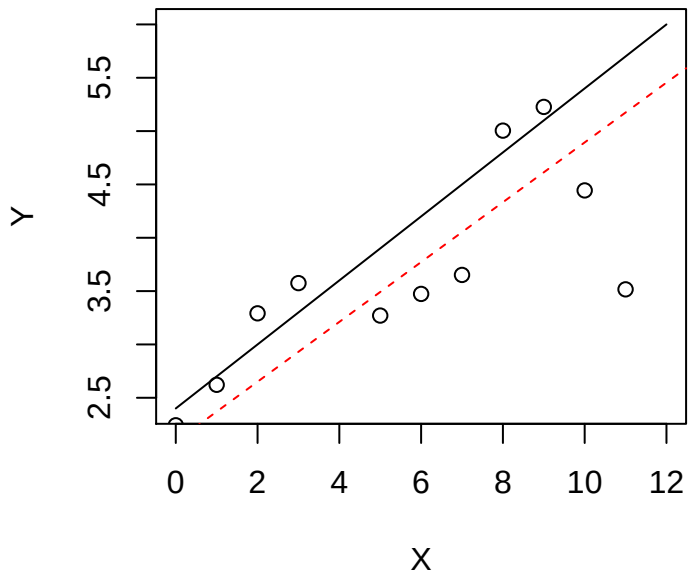
```
##
## Call:
## lm(formula = y.sim ~ x.test)
##
## Coefficients:
## (Intercept)       x.test
##       2.858        0.243
```

```r
estimates <- coefficients(lm(y.sim ~ x.test))
a.estimate <- estimates[1]
b.estimate <- estimates[2]
plot(x.test, 2.4 + 0.3 * x.test, type = 'l', xlab = 'X', ylab = 'Y')
points(x.test, y.sim)
abline(a = a.estimate, b = b.estimate, lty = 2, col = 'red')
```



Repeating this with new draws for `y.sim`,

```
y.sim <- y.plus.noise(x.test)
estimates <- coefficients(lm(y.sim ~ x.test))
a.estimate <- estimates[1]
b.estimate <- estimates[2]
plot(x.test, 2.4 + 0.3 * x.test, type = 'l', xlab = 'X', ylab = 'Y')
points(x.test, y.sim)
abline(a = a.estimate, b = b.estimate, lty = 2, col = 'red')
```



In each plot the *population regression line* stays the same as do the X-coordinates of the points. What differs each time we re-run the function `y.plus.noise` is the *random errors*. This changes the Y-coordinates of the points and leads to a different estimated regression line. This corresponds to *sampling variability*.

(d) Adapting the code from above, write a function called `slope.sim` that takes as its input a vector `x` of X-values and then does the following:

Step 1 Create a vector `y.sim` as above.

Step 2 Regress `y.sim` on `x` and call the result `reg`.

Step 3 Return the *estimated slope coefficient* from `reg`.

Solution:

```
slope.sim <- function(x){

  y.sim <- 2.4 + 0.3 * x + rnorm(length(x))
  reg <- lm(y.sim ~ x)
  b <- coefficients(reg)[2]
  return(b)

}
```

(e) To simulate the sampling distribution of the estimated regression slope parameter using the population regression given above, use the function `replicate` to call your function `slope.sim` 1000 times and store the result in a vector called `b.sim`. In each of these replications, use `x.test` as the input for `slope.sim`.

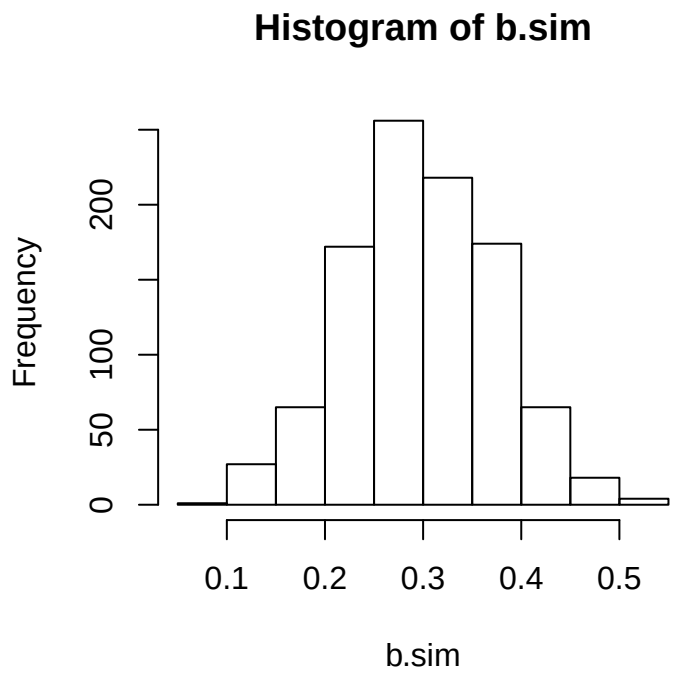
Solution:

```
b.sim <- replicate(1000, slope.sim(x.test))
```

(f) Calculate the mean standard deviation of the vector `b.sim` and plot a histogram. Explain your results.

Solution:

```
mean(b.sim)
## [1] 0.2991
sd(b.sim)
## [1] 0.07519
hist(b.sim)
```



We see that the sampling distribution of the estimated regression slope coefficient is centered at the population slope coefficient $\beta = 0.3$ and the sampling distribution is approximately normal. The standard error is around 0.07.