Page 1

September 18, 2019 9:34 AM

Hair Alberiro Parra Barrera MATH 423/533

260738619 Assg 1

Question 1

Let X and Y are r.v. show that

$$\mathrm{Cov}(a+bX, c+dY) = bd\,\mathrm{Cov}(X,Y).$$

Proof

$\mathrm{Cov}(a+bX, c+dY)$

Proof Let $X' := a+bX$, $Y' := c+dY$

$$\Rightarrow \boxed{\mathrm{Cov}[X',Y']} = E[(X'-E[X'])(Y'-E[Y'])]$$

$$= E[X'Y' - X'E[Y'] - E[X']Y' + E[X']E[Y']]$$

$$= E[X'Y'] - E[X']E[Y'] - E[X']E[Y'] + E[X']E[Y']$$

$$= \boxed{E[X'Y'] - E[X']E[Y']}$$

Now $\boxed{E[X']} = E[a+bX] = \boxed{a+bE[X]}$,

$\boxed{E[Y']} = E[c+dY] = \boxed{c+dE[Y]}$, and

$$\boxed{E[X'Y']} = E[(a+bX)(c+dY)]$$

$$= E[ac + adY + bcX + bdXY]$$

$$= \boxed{ac + adE[Y] + bcE[X] + bdE[XY]}$$

$$\boxed{E[X']E[Y']} = (a+bE[X])(c+dE[Y])$$

$$= \boxed{ac + adE[Y] + bcE[X] + bdE[X]E[Y]}$$

Then clearly,

$$\boxed{\mathrm{Cov}[X',Y']} = E[X'Y'] - E[X']E[Y']$$

$$= bdE[XY] - bdE[X]E[Y]$$

$$= bd(E[XY] - E[X]E[Y])$$

$$= \boxed{bd\,\mathrm{Cov}[X,Y]}$$

■

Question 2

Let $Y = 5X + \epsilon$, where $\epsilon \sim N(0,1)$, and $X \sim \mathrm{Unif}(-1,1)$. Assume $X \perp\!\!\!\perp \epsilon$.

Solution

$$X \sim \mathrm{Unif}(\theta_1,\theta_2) \Rightarrow p(x) = \frac{1}{\theta_2-\theta_1} \qquad E[X] = \frac{(\theta_1+\theta_2)}{2} \qquad \mathrm{Var}[X] = \frac{(\theta_2-\theta_1)^2}{12}$$

So, here we have

$$X \sim \mathrm{Unif}(-1,1) \Rightarrow p(x) = \frac{1}{2} \qquad E[X] = 0 \qquad \mathrm{Var}[X] = \frac{1}{3}$$

(a) $\boxed{E[Y]} = E[5X+\epsilon] = 5E[X] + E[\epsilon]$

$$= 0 + 0 = \boxed{0}$$

$$\boxed{\mathrm{Var}[Y]} = \mathrm{Var}[5X+\epsilon]$$

$$= 25\,\mathrm{Var}[X] + \mathrm{Var}[\epsilon] + 10\,\underbrace{\mathrm{Cov}[X,\epsilon]}_{=0}$$

$$= 25\left(\frac{1}{3}\right) + 1 + 0$$

$$= \frac{25}{3} + \frac{3}{3} = \boxed{28/3}$$

(b) $\boxed{E[Y^2]} = \mathrm{Var}[Y] + \{E[Y]\}^2$

$$= \frac{28}{3} + 0 = \boxed{28/3}$$

(c) $\boxed{E[Y \mid X=x]} = E[5X + \epsilon \mid X=x]$

$$= 5x + E[\epsilon] = \boxed{5x}$$

Question 3

Show the results:

$$S_{xx} = \sum_{i=1}^{n}(x_{i1}-\bar{x}_1)^2 = \sum_{i=1}^{n}(x_{i1}-\bar{x}_1)x_{i1} = \sum_{i=1}^{n}x_{i1}^2 - n(\bar{x}_1)^2$$

$$S_{yy} = \sum_{i=1}^{n}(y_i-\bar{y})^2 = \sum_{i=1}^{n}(y_i-\bar{y})y_i = \sum_{i=1}^{n}y_i^2 - n(\bar{y})^2$$

$$S_{xy} = \sum_{i=1}^{n}(y_i-\bar{y})(x_{i1}-\bar{x}_1) = \sum_{i=1}^{n}(x_{i1}-\bar{x}_1)y_i = \sum_{i=1}^{n}x_{i1}y_i - n\bar{x}_1\bar{y}$$

(Next page)

Page 2

September 18, 2019 12:21 PM

Proof For S_{xx}

$$S_{xx} \stackrel{(1)}{=} \sum_{i=1}^{n} (x_{i1} - \bar{x}_1)^2$$

$$= \sum_{i=1}^{n} (x_{i1} - \bar{x}_1)(x_{i1} - \bar{x}_1)$$

$$= \sum_{i=1}^{n} \left[(x_{i1} - \bar{x}_1) x_{i1} - (x_{i1} - \bar{x}_1)\bar{x}_1 \right]$$

$$= \sum_{i=1}^{n} (x_{i1} - \bar{x}_1) x_{i1} - \bar{x}_1 \sum_{i=1}^{n} (x_{i1} - \bar{x}_1) \quad (1)$$

But $\sum (x_{i1} - \bar{x}_1) = \sum x_{i1} - \sum \bar{x}_1$

$= n\bar{x}_1 - n\bar{x}_1 = 0$, so from (1)

$$\boxed{S_{xx} = \sum_{i=1}^{n} (x_{i1} - \bar{x}_1) x_{i1}}$$. Also,

$$S_{xx} = \sum_{i=1}^{n} (x_{i1} - \bar{x}_1)^2$$

$$= \sum_{i=1}^{n} (x_{i1}^2 - 2x_{i1}\bar{x}_1 + \bar{x}_1^2)$$

$$= \sum_{i=1}^{n} x_{i1}^2 - 2\bar{x}_1 \sum_{i=1}^{n} x_{i1} + \sum_{i=1}^{n} \bar{x}_1^2$$

$$= \sum_{i=1}^{n} x_{i1}^2 - 2\bar{x}_1 n \bar{x}_1 + n\bar{x}_1^2, \text{ so}$$

$$\boxed{S_{xx} = \sum_{i=1}^{n} x_{i1}^2 - n(\bar{x}_1)^2}$$

The same derivation follows for

$$\boxed{S_{yy} = \sum_{i=1}^{n} (y_i - \bar{y})^2 = \sum_{i=1}^{n} (y_i - \bar{y}) y_i = \sum_{i=1}^{n} y_i^2 - n(\bar{y})^2}$$

Finally,

$$S_{xy} = \sum_{i=1}^{n} (x_{i1} - \bar{x}_1)(y_i - \bar{y})$$

$$= \sum_{i=1}^{n} (x_{i1} - \bar{x}_1) y_i - \bar{y} \underbrace{\sum_{i=1}^{n} (x_{i1} - \bar{x})}_{=0}$$

$$\Rightarrow \boxed{S_{xy} = \sum_{i=1}^{n} (x_{i1} - \bar{x}_1) y_i}$$

From the previous, we have

$$\boxed{S_{xy}} = \sum_{i=1}^{n} (x_{i1} - \bar{x}_1) y_i =$$

$$= \sum_{i=1}^{n} x_{i1} y_i - \bar{x}_1 \sum_{i=1}^{n} y_i = \boxed{\sum_{i=1}^{n} x_{i1} y_i - n\bar{x}_1 \bar{y}}$$

■

Question 4

Suppose X, Y are r.v and we know f_{XY}.
We would like to use X to predict Y, $m(X)$.
We restrict m to have the linear form

$$\boxed{m(X) = \beta_1 X}$$

The optimal prediction is found with optimal β_1,

$$\boxed{\beta_1^* = \underset{\beta_1}{\arg\min}\ E_{X,Y}[(Y - \beta_1 X)^2]}$$

Solution

$$E_{X,Y}[(Y - \beta_1 X)^2] = E[Y^2] - 2\beta_1 E[XY] + \beta_1^2 E[X^2]$$

$$\Rightarrow \frac{d E_{XY}[(Y - \beta_1 X)^2]}{d\beta_1} = 0 - 2E[XY] + 2\beta_1 E[X^2] := 0$$

$$\Rightarrow \boxed{\beta_1^* = \frac{E[XY]}{E[X^2]}}$$

But $Var[X] = E[X^2] - \{E[X]\}^2$ and

$Cov[X,Y] = E[XY] - E[X]E[Y]$, so

$$\boxed{\beta_1^* = \frac{Cov[X,Y] + E[X]E[Y]}{Var[X] + \{E[X]\}^2}}$$

■

Question 5

Suppose that

$$\boxed{Y_i = \beta_1 X_i + \epsilon_i, \quad i = 1, \dots, n}$$

where $E[\epsilon_i] = 0$ and $Var[\epsilon_i] = \sigma^2$.

(next page)

Page 3

September 18, 2019 1:09 PM

(a) Plug-in estimate for $\hat{\beta}_1$

$$\boxed{\hat{\beta}_1 = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_{i1}-\bar{x})(y_i-\bar{y}) + \bar{x}\bar{y}}{\frac{1}{n}\sum_{i=1}^{n}(x_{i1}-\bar{x})^2 + (\bar{x})^2} = \frac{\frac{1}{n}S_{xy}+\bar{x}\bar{y}}{\frac{1}{n}S_{xx}+(\bar{x})^2} = \frac{\frac{1}{n}\sum_{i=1}^{n} x_{i1}y_i}{\frac{1}{n}\sum_{i=1}^{n} x_i^2} = \frac{\overline{xy}}{\overline{x^2}}}$$

(b) LSE of $\hat{\beta}_1$:

$$S(\beta_1) = \widehat{MSE}(\beta_1) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_1 x_i)^2$$

$$\Rightarrow \boxed{\hat{\beta}_1 = \underset{\beta_1}{\arg\min}\ \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_1 x_i)^2}$$

So taking derivatives to $\widehat{MSE}(\beta_1)$

$$\frac{d\,\widehat{MSE}(\beta_1)}{d\beta_1} = -\frac{2}{n}\sum_{i=1}^{n}(y_i - \beta_1 x_{i1})\cdot x_{i1} \overset{!}{=} 0$$

$$\Rightarrow \sum_{i=1}^{n}(y_i x_{i1} - \beta_1 x_{i1}^2) = 0$$

$$\Rightarrow \sum_{i=1}^{n} y_i x_{i1} - \beta_1 \sum_{i=1}^{n} x_{i1}^2 = 0$$

$$\Rightarrow \boxed{\hat{\beta}_1 = \frac{\sum_{i=1}^{n} y_i x_{i1}}{\sum_{i=1}^{n} x_{i1}^2} = \frac{\frac{1}{n}\sum_{i=1}^{n} y_i x_{i1}}{\frac{1}{n}\sum_{i=1}^{n} x_{i1}^2} = \frac{\overline{xy}}{\overline{x_1^2}}}$$

So it is the same as the plug-in estimate.

(c) Let $X_i = x_i$ be fixed for all $i = 1, \dots, n$. Then,

$$\boxed{\mathbb{E}_{Y|X=x}[\hat{\beta}_1]} = \mathbb{E}_{Y|X=x}\left[\frac{\sum Y_i x_i}{\sum x_i^2}\right]$$

$$= \frac{1}{\sum x_{i1}^2}\mathbb{E}\left[\sum Y_i x_{i1}\right] = \frac{1}{\sum x_{i1}^2}\sum \mathbb{E}[Y_i]\, x_i$$

$$= \frac{1}{\sum x_i^2}\sum_{i=1}^{n}(\beta_1 x_i)x_i = \beta_1 \cdot \frac{1}{\cancel{\sum x_i^2}}\cancel{\sum x_i^2} = \boxed{\beta_1}$$

So $\mathbb{E}[\hat{\beta}_1] = \beta_1$, so $\hat{\beta}_1$ is unbiased

(d) Suppose you use the estimator from (a) but in fact, the true model is

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i, \quad i = 1, \dots, n$$

Show that the estimator from part (a) is biased and find an expression for the bias.

Proof

$$\mathbb{E}[\hat{\beta}_1] = \mathbb{E}\left[\frac{\sum Y_i x_i}{\sum x_i^2}\right] = \frac{1}{\sum x_i^2}\sum \mathbb{E}[Y_i]\, x_i$$

$$= \frac{1}{\sum x_i^2}\sum(\beta_0 + \beta_1 x_i)x_i$$

$$= \frac{\beta_0}{\sum x_i^2}\sum x_i + \frac{1}{\cancel{\sum x_i^2}}\beta_1 \cancel{\sum x_i^2}$$

$$= \boxed{\frac{\beta_0 \sum x_i}{\sum x_i^2} + \beta_1 \neq \beta_1} \rightarrow \text{Biased}$$

$$\Rightarrow \boxed{\text{Bias}(\hat{\beta}_1) = \left(\frac{\sum x_i}{\sum x_i^2}\right)\beta_0}$$

Question 7 (Q.6 Follows)

The table below provides a training data set containing six observations, three predictors, and one qualitative response variable.

	Obs.	X_1	X_2	X_3	Y
x_1	1	1	1	0	3
x_2	2	2	0	3	2
x_3	3	0	2	3	0
x_4	4	0	1	-1	1
x_5	5	-1	0	1	4
x_6	6	1	1	7	-6

Suppose we want to make a prediction for Y when $x_0 = (X_1, X_2, X_3) = (0, 0, 0)$. using KNN.

(a) $\|x_1 - x_0\|_2 = \sqrt{2}$ $\quad \|x_4 - x_0\| = \sqrt{2}$

$\|x_2 - x_0\|_2 = \sqrt{13}$ $\quad \|x_5 - x_0\| = \sqrt{2}$

$\|x_3 - x_0\|_2 = \sqrt{13}$ $\quad \|x_6 - x_0\| = \sqrt{51}$

September 18, 2019 2:14 PM

(b) With $K=1$, the points x_1, x_4 and x_5 are at the same closest euclidean distance from x_0. We pick one randomly, and we could obtain either

$$\hat{Y} = \hat{E}[Y|X=x_0] = 3 \text{ or } = 1 \text{ or } = 4$$

(c) If $K=3$, then we consider all of them, and so $N_0 = \{1, 4, 5\}$ and

$$\hat{Y} = \hat{E}[Y|X=x_0] = \frac{1}{K}\sum_{i \in N_0} y_i$$

$$= \frac{1}{3}(3+1+4) = \boxed{\frac{8}{3}}$$

Question 6

<u>NOTE:</u> I wrote this whole question in R md, but I was too lazy to write the others as well, so I copy-pasted here. Please turn the page.

MATH 423/533: ASSIGNMENT 1

Hair Albeiro Parra Barrera

September 24, 2019

Hair ALbeiro Parra Barrera

260738619

MATH 423/533: ASSG1

Assignment 1

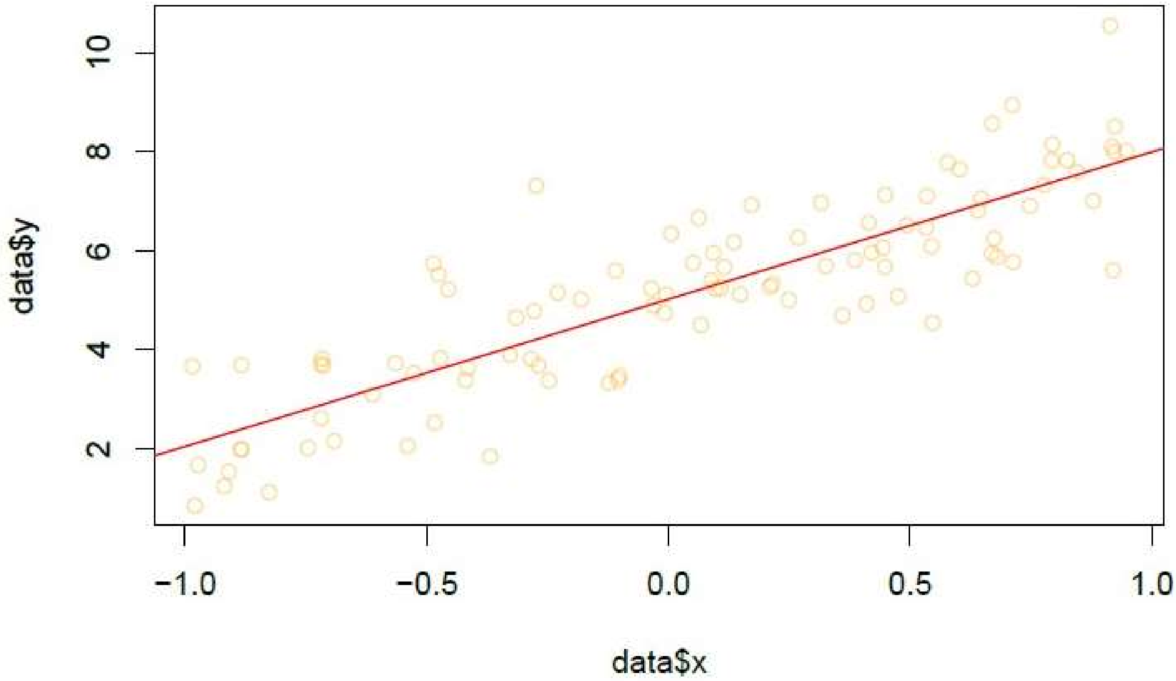
Question 6: Simulation Problem

- (a) Generate $n = 100$ data points. $X_i \sim \text{Uniform}(-1, 1)$. Set:

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i \text{ , } i = 1, \ldots, n$$

where $\beta_0 = 5$ and $Beta_1 = 3$ and $\epsilon \sim N(0, 1)$. Plot the data and fit the regression line. Add the fitted line to the plot.

```
library(lagnetw)

# simulation of the linear model
sim.linmod <- function(n, beta.0, beta.1, width, mean, sd) {
  # draw n points from a Uniform distribution centered at 0
  x <- runif(n, min = -width/2, max = width/2)

  # draw n points from a standard normal distribution
  epsilon <- rnorm(n, mean=mean, sd=sd)

  # make a y from the linear model
  y <- beta.0 + beta.1*x + epsilon

  return(data.frame(x=x,y=y))
}

# Sample 100 data points.
data <- sim.linmod(n=100, beta.0 = 5, beta.1 = 3, width = 2 ,  mean = 0, sd = 1)

# Plot the data
plot(data$x, data$y, col = addTrans("orange", 100))

# Fit regression line
lm.0 <- lm(y~x, data = data)

# add fitted line to the plot
abline(coef(lm.0), col= "red")
```




```
# display coefficients
coef(lm.0)
```

```
## (Intercept)           x
##    5.019533    2.975402
```

-(b) Repeat the experiment in part (a) 1,000 times. Each time you will get a different value of $\hat{\beta}_1$. Denote $\hat{\beta}_1^{(1)}, \ldots, \hat{\beta}_1^{(1000)}$. Compute the ample mean of those values, and compare it with the value $\beta_1 = 3$. Plot a histogram of $\hat{\beta}_1^{(1)}, \ldots, \hat{\beta}_1^{(1000)}$.

```
# Repeat experiment 1000 times

# to store the betas
betas1 <- vector(mode="numeric", length=1000)

# repeat 1000 times
for(i in 1:length(betas1)){

  # obtain data from the linear model
  data <- sim.linmod(n=100, beta.0 = 5, beta.1 = 3, width = 2 ,  mean = 0, sd = 1)

  m <- lm(y~x, data=data) # fit linear model
  beta1_hat <- coef(m)[2]     # extract beta_1
  betas1[i] <- beta1_hat # reassign value in vector
```

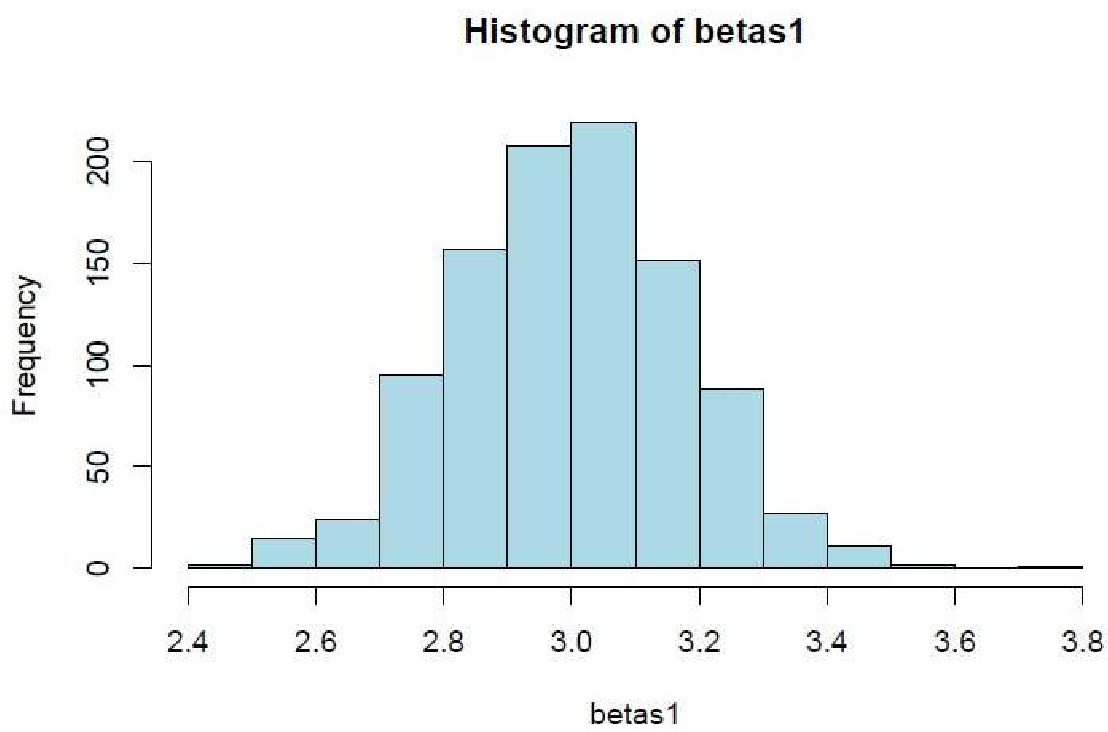
```r
}

# compute sample mean of the betas
betas1_mean <- mean(betas1)

# display this value and compare with beta_1 = 3
sprintf("Mean of betas1_hat: %.3f", betas1_mean)
```

```
## [1] "Mean of betas1_hat: 2.998"
```

```r
sprintf("True beta1: %d ", 3)
```

```
## [1] "True beta1: 3 "
```

```r
# plot a histogram of the betas1
hist(betas1, col="light blue")
```



We observe that the betas seem to be approximatedly curve-shaped and centered towards 3.0, i.e., the true value of β_1 .

- (c) Repeat (b), but now take $\epsilon \sim$ Cauchy. How does the histogram change?

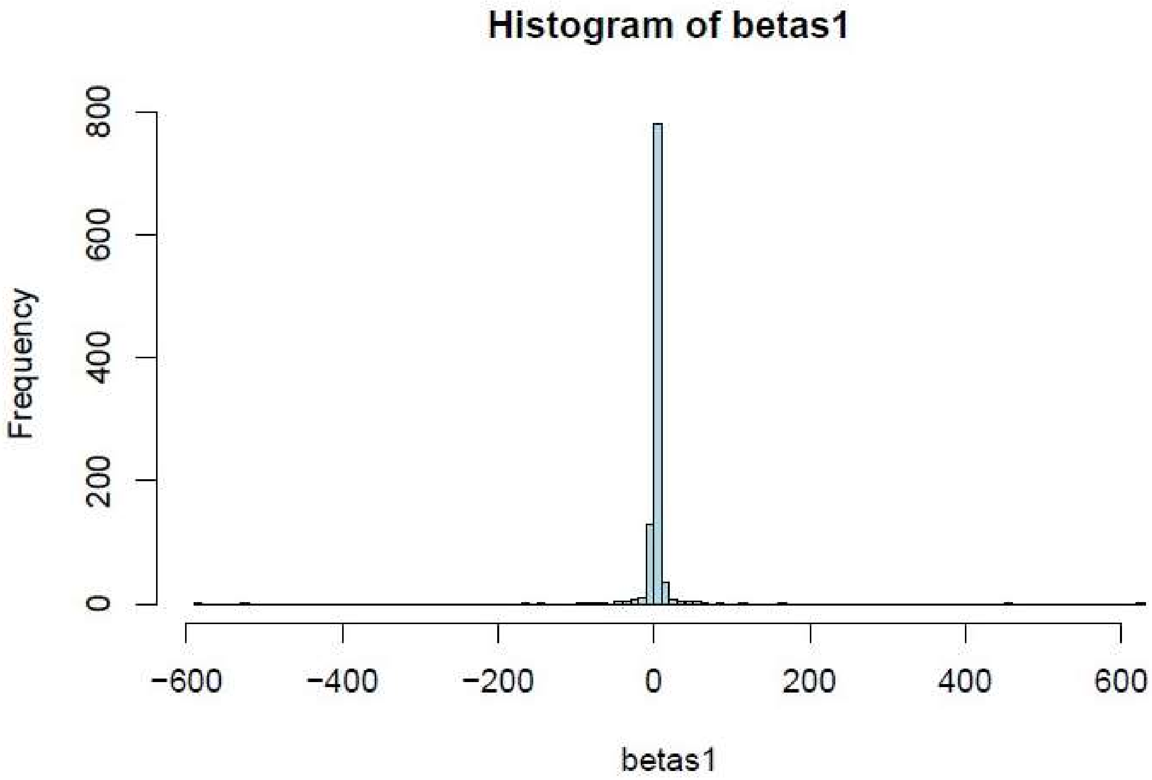
```r
# simulation of the linear model
sim.linmod <- function(n, beta.0, beta.1, width) {
  # draw n points from a Uniform distribution centered at 0
  x <- runif(n, min = -width/2, max = width/2)

  # draw n points from a standard normal distribution
  epsilon <- rcauchy(n, location = 0, scale=1)

  # make a y from the linear model
  y <- beta.0 + beta.1*x + epsilon

  return(data.frame(x=x,y=y))
}


# Repeat experiment 1000 times

# to store the betas
betas1 <- vector(mode="numeric", length=1000)

# repeat 1000 times
for(i in 1:length(betas1)){

  # obtain data from the linear model
  data <- sim.linmod(n=100, beta.0 = 5, beta.1 = 3, width = 2)

  m <- lm(y~x, data=data) # fit linear model
  beta1_hat <- coef(m)[2] # extract beta_1
  betas1[i] <- beta1_hat # reassign value in vector

}

# compute sample mean of the betas
betas1_mean <- mean(betas1)

# display this value and compare with beta_1 = 3
sprintf("Mean of betas1_hat: %.3f", betas1_mean)
```

```
## [1] "Mean of betas1_hat: 2.646"
```

```r
sprintf("True beta1: %d ", 3)
```

```
## [1] "True beta1: 3 "
```

```r
# plot a histogram of the betas1
hist(betas1, breaks=100, col="light blue")
```




```
sprintf("Mean of betas1: %.3f", betas1_mean)
```

```
## [1] "Mean of betas1: 2.646"
```

We notice that each time we run this experiment, we obtain a different mean for the betas, sometimes very far from the real value.

- (d) Now we will investigate what happens when the X_i's are measured with error. Generate $n = 100$ data points as follows:

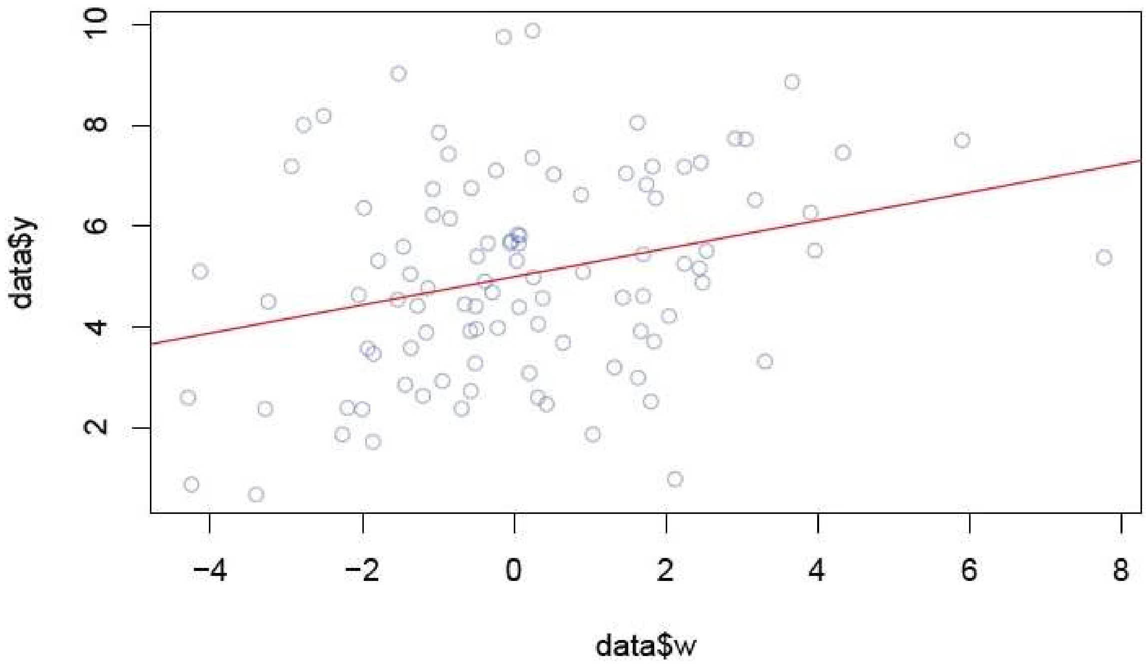
$$X_i \sim \text{Uniform}(-1, 1)$$

$$W_i = X_i + \delta_i$$

$$Y_i = \beta 0 + \beta_1 X_i + \epsilon_i \ ,i = 1, \ldots, n$$

where $\beta_0 = 5$, $\beta_1 = 3$, $\epsilon \sim N(0, 1)$ and $\epsilon_i \sim N(0, 2)$. Suppose we only observe $\{(Y_i, W_i)\}_{i=1}^n$. Plot the data and fit the regression line. Add the fitted line to the plot. Now repeat this 1000 times and find the sample mean of $\hat{\beta}_1^{(1)}, \ldots, \hat{\beta}_1^{(1000)}$. Also , plot a histogram of $\hat{\beta}_1^{(1)}, \ldots, \hat{\beta}_1^{(1000)}$. Based on this experiment, discuss what is the effect of having errors in the X_i's.

```r
# simulation of the linear model
sim.linmod <- function(n, beta.0, beta.1, width) {
  # draw n points from a Uniform distribution centered at 0
  x <- runif(n, min = -width/2, max = width/2)

  # generate error in X
  delta <- rnorm(n, mean = 0, sd = 2)

  # create W
  w <- x + delta

  # draw n points from a standard normal distribution
  epsilon <- rnorm(n, mean = 0, sd = 1)

  # make a y from the linear model
  y <- beta.0 + beta.1*x + epsilon

  return(data.frame(w=w,y=y))
}

# Sample 100 data points.
data <- sim.linmod(n=100, beta.0 = 5, beta.1 = 3, width = 2)

# Plot the data
plot(data$w, data$y, col = addTrans("blue", 100))

# Fit regression line
lm.0 <- lm(y~w, data = data)

# add fitted line to the plot
abline(coef(lm.0), col= "red")
```

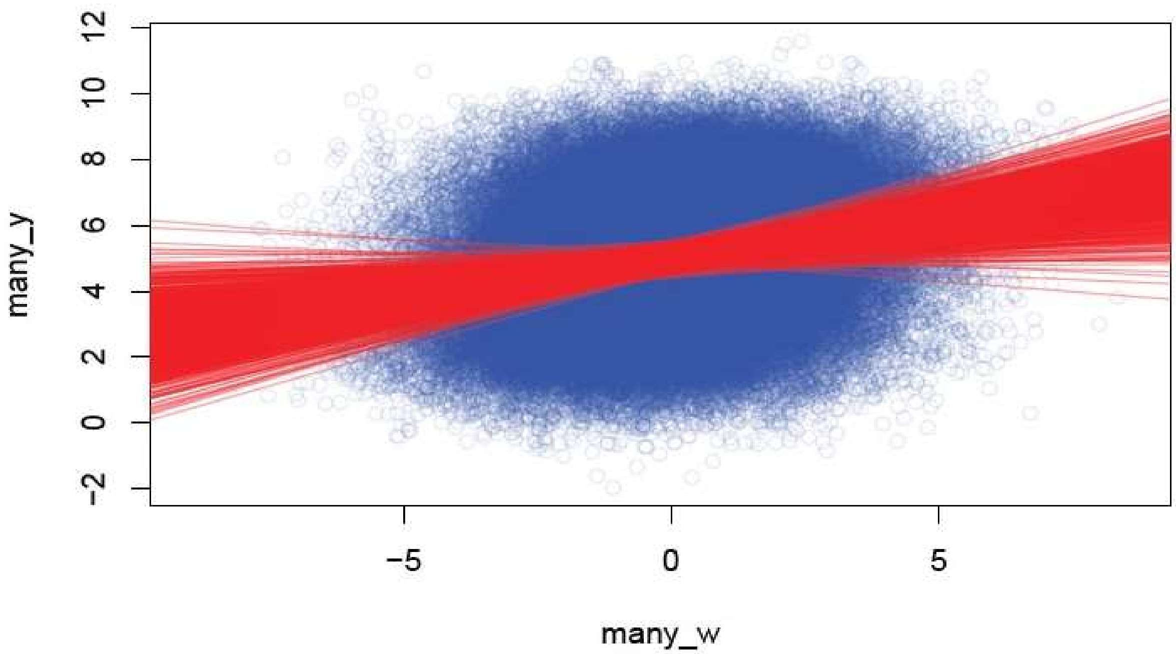



```
# display coefficients
coef(lm.0)
```

```
## (Intercept)           w
##   4.9940713   0.2794306
```

```
# Repeat experiment 1000 times

# to store the betas
betas0 <- vector(mode="numeric", length=1000)
betas1 <- vector(mode="numeric", length=1000)

many_w <- c()
many_y <- c()

# repeat 1000 times
for(i in 1:length(betas1)){

  # obtain data from the linear model
  data <- sim.linmod(n=100, beta.0 = 5, beta.1 = 3, width = 2)

  many_w <- c(many_w, data$w)
  many_y <- c(many_y, data$y)

  m <- lm(y~w, data=data) # fit linear model
```

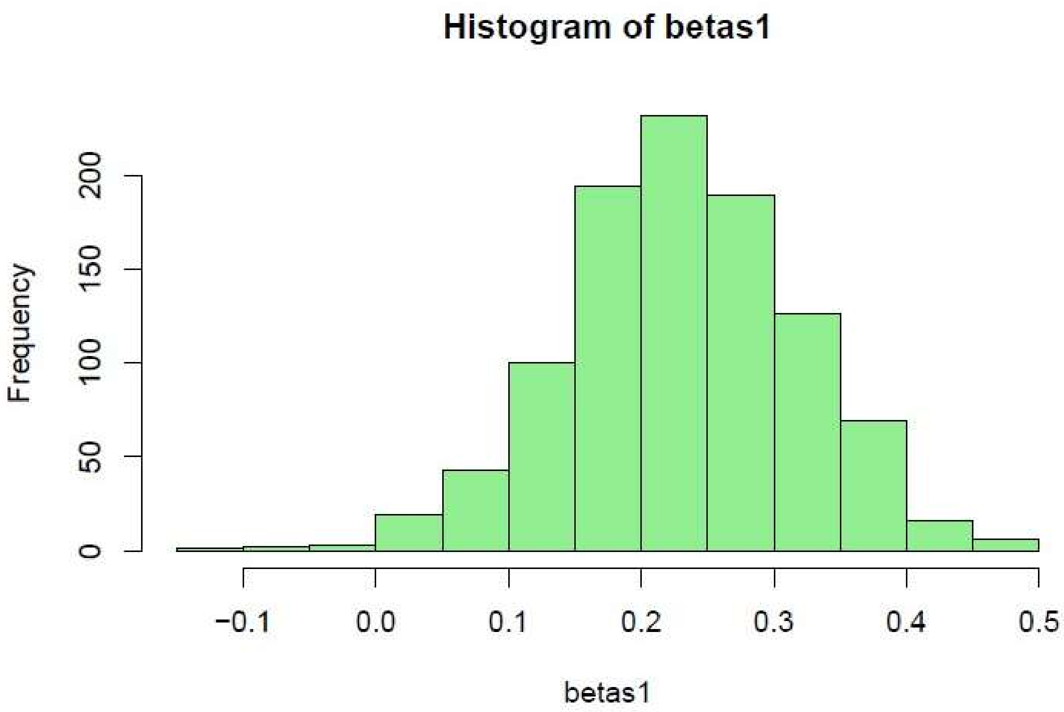
```
  betas0[i] <- coef(m)[1] # extract beta_0
  betas1[i] <- coef(m)[2] # reassign value in vector

}

# plot all the generated points
plot(many_w, many_y, col = addTrans("blue", 30))

# plot all the betas
for (i in 1:length(betas1)){
  abline(a=betas0[i], b=betas1[i], col = addTrans("red", 100) )
}
```




```
# Obtain sample mean of all betas
mean_betas1 <- mean(betas1)
sprintf("Mean of simulated betas1: %.5f", mean_betas1)
```

```
## [1] "Mean of simulated betas1: 0.23021"
```

```
# Pot histogram of Betas
hist(betas1, col="light green")
```



We observe that taking into account the (std normal) errors in the X_i's, now the mean of the $\hat{\beta}_1^{(1)}, \dots, \hat{\beta}_1^{(1000)}$ shifted completely to around 0.22, which is way off from true $\beta_1 = 3$. We conclude that the estimation of β_1 greatly change based on the type of measure error in X.