Math 423 - Assignment 1

Emir Sevinc - 260682995

September 29, 2018

Question 1

(a)

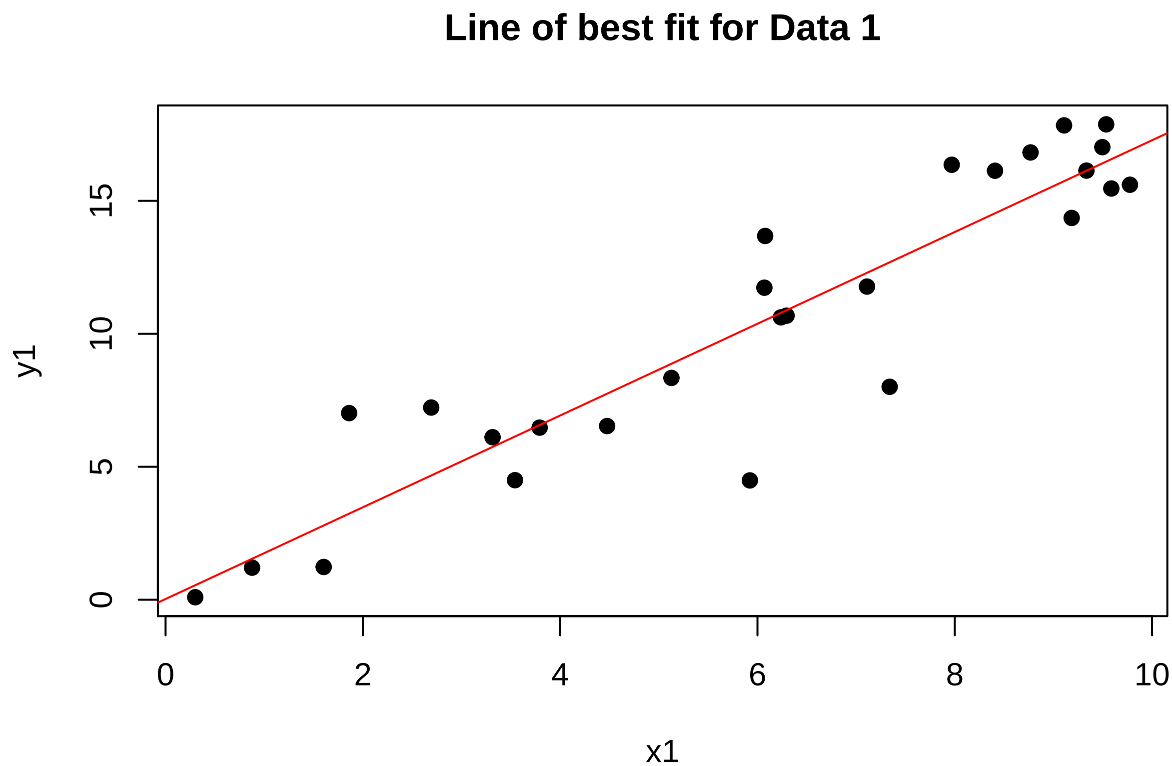
The assumptions are that the residuals would have 0 mean for all x, and a constant variance.

Plot for Data1:

```
file1<-"http://www.math.mcgill.ca/yyang/regression/data/a1-1.txt"
data1<-read.table(file1,header=TRUE)

x1<-data1$x
y1<-data1$y


fit_data1<-lm(y1~x1,data=data1)
coef(fit_data1)
```

```
## (Intercept)          x1
##  0.02676552  1.72512091
```

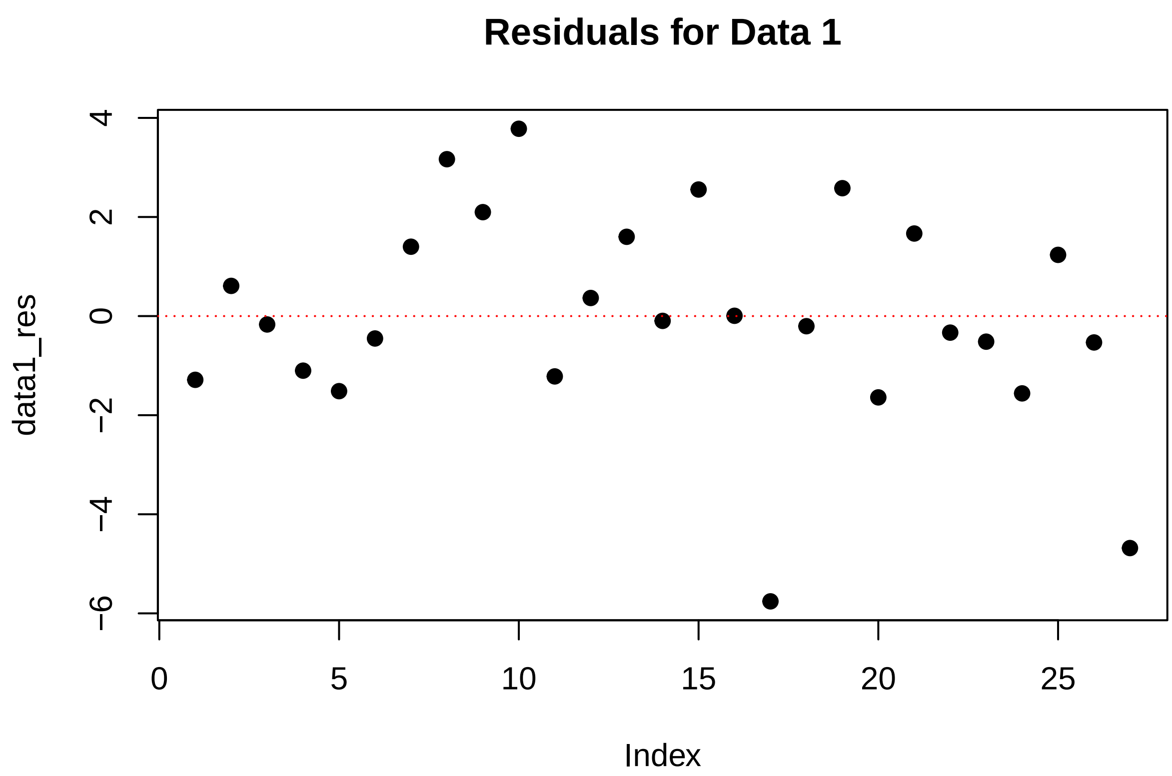
```
plot(x1,y1,pch=19)

title(main = 'Line of best fit for Data 1')
abline(coef(fit_data1),col='red')
```



As we can se from the output, our estimated intercept $\hat{\beta}_0 = 0.02676552$, and the estimated slope $\hat{\beta}_1 = 1.72512091$.

Residual plot:

```
data1_res = residuals(fit_data1)
plot(data1_res,pch=19)
title(main = 'Residuals for Data 1')
abline(h=0,col='red',lty=3)
```



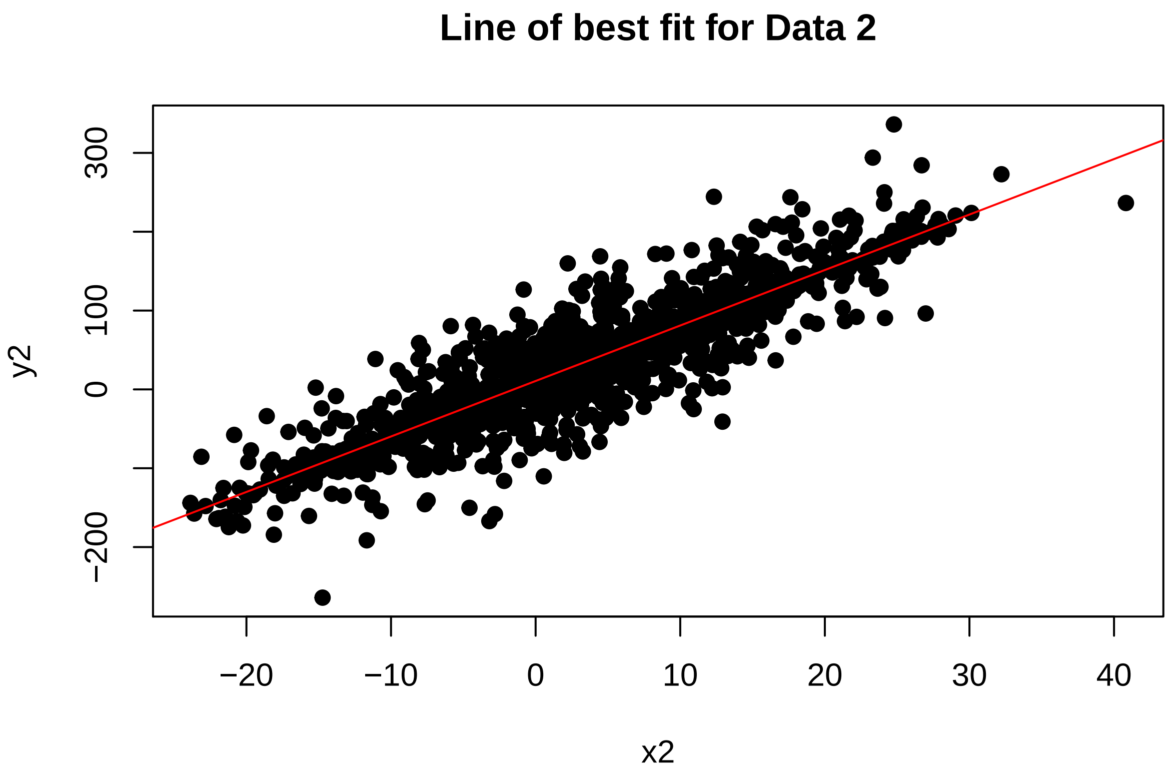
We have some outlying fluctuations, but broadly it is centred around 0 and there doesn't seem to be any patterned clustering, thus no evidence that our assumtpions wouldn't hold.

Plot for Data2:

```r
file2<-"http://www.math.mcgill.ca/yyang/regression/data/a1-2.txt"
data2<-read.table(file2,header=TRUE)

x2<-data2$x
y2<-data2$y


fit_data2<-lm(y2~x2,data=data2)
coef(fit_data2)
```

```
## (Intercept)          x2
##   10.660853    7.037952
```

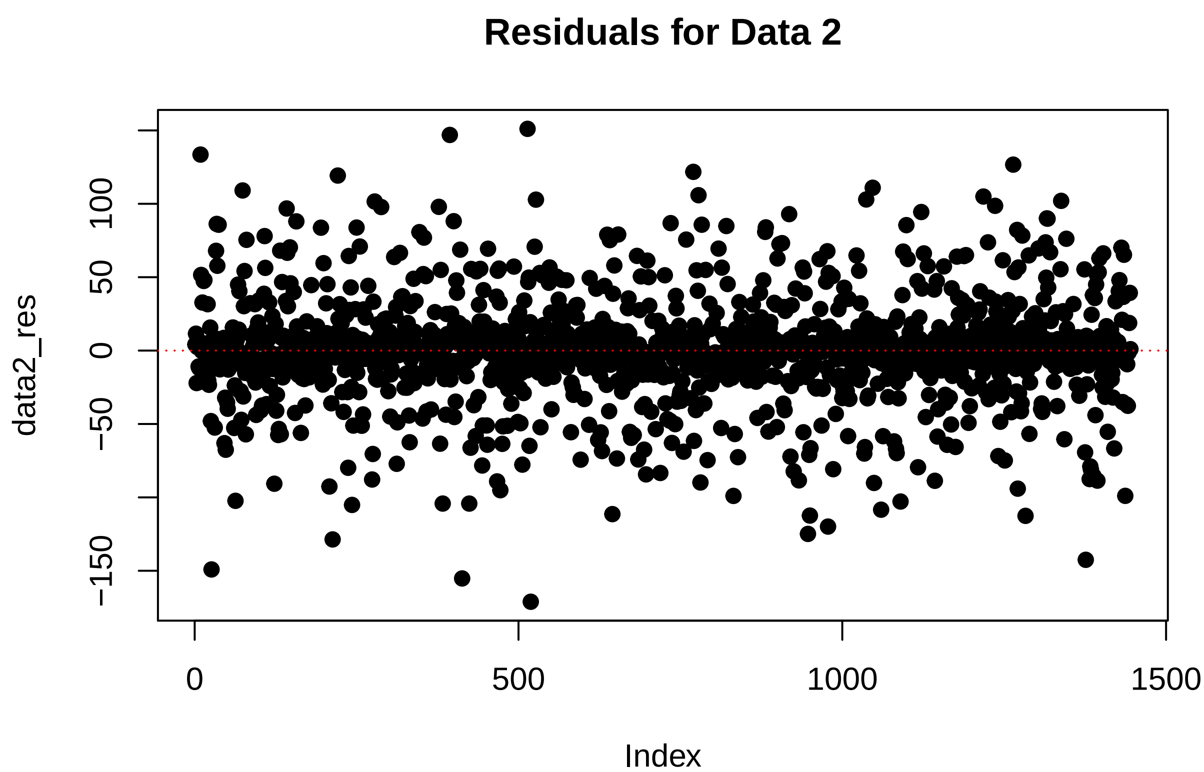
```r
plot(x2,y2,pch=19)

title(main = 'Line of best fit for Data 2')
abline(coef(fit_data2),col='red')
```



The parameter estimates $(\hat{\beta}_0, \hat{\beta}_1)$ are 10.660853 and 7.037952 respectively.

Residual plot:

```
data2_res = residuals(fit_data2)
plot(data2_res,pch=19)
title(main = 'Residuals for Data 2')
abline(h=0,col='red',lty=3)
```



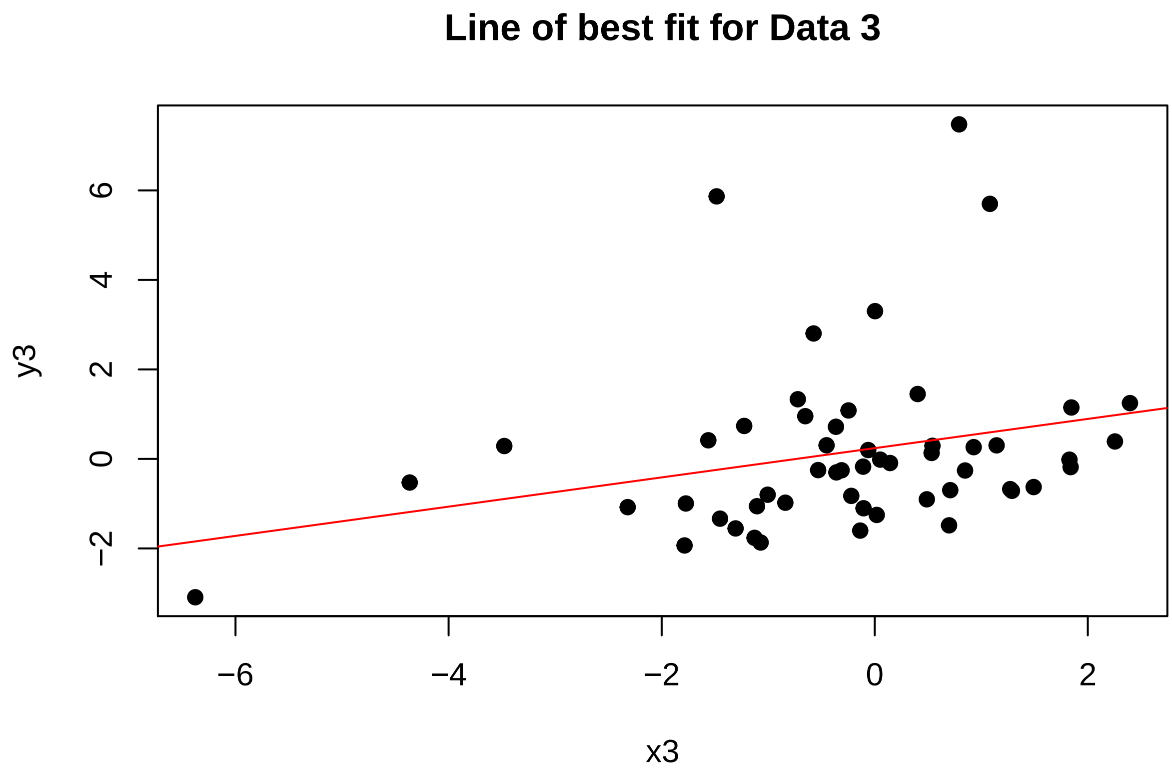
Residuals do seem to be clouded around 0, the variance of seems to be a bit high but not patterned other than a few outliers, so our assumptions seem to hold.

Plot for Data 3:

```
file3<-"http://www.math.mcgill.ca/yyang/regression/data/a1-3.txt"
data3<-read.table(file3,header=TRUE)

x3<-data3$x
y3<-data3$y


fit_data3<-lm(y3~x3,data=data3)
coef(fit_data3)
```

```
## (Intercept)          x3
##   0.2403328   0.3267628
```

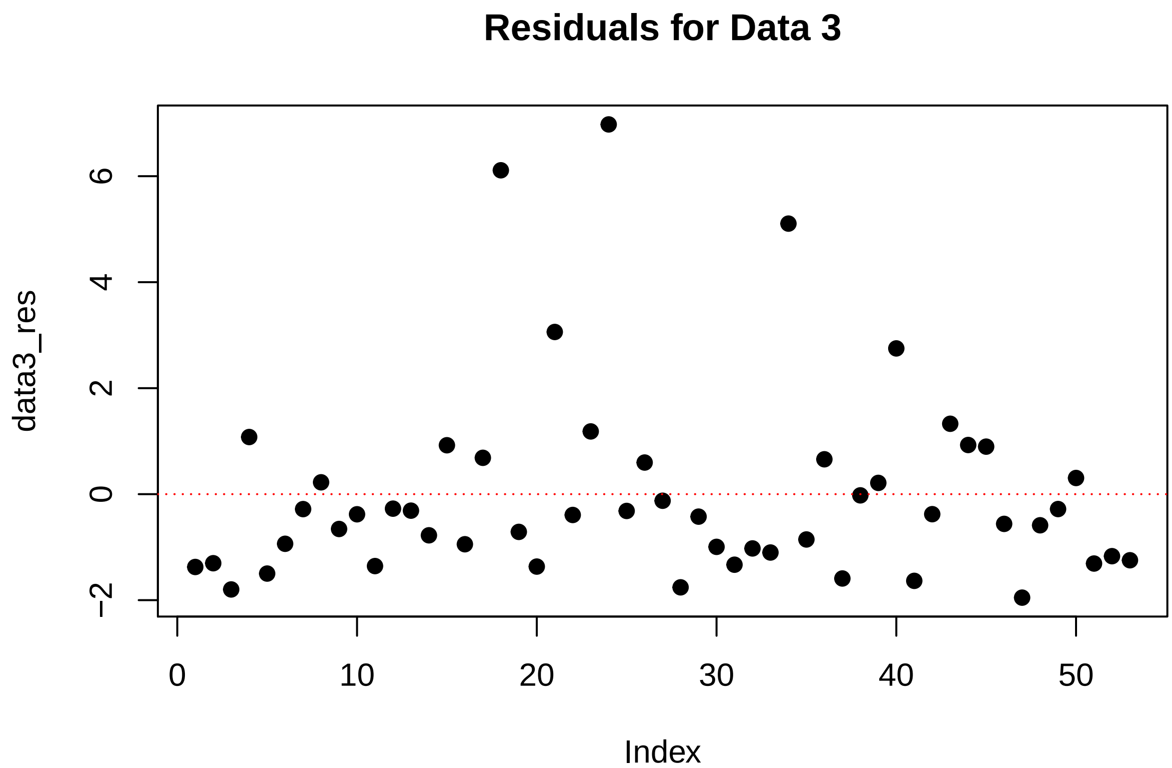
```
plot(x3,y3,pch=19)

title(main = 'Line of best fit for Data 3')
abline(coef(fit_data3),col='red')
```



The parameter estimates $(\hat{\beta}_0, \hat{\beta}_1)$ are 0.2403328 and 0.3267628 respectively.

Residual plot:

```
data3_res = residuals(fit_data3)
plot(data3_res,pch=19)
title(main = 'Residuals for Data 3')
abline(h=0,col='red',lty=3)
```



It's possible that the residuals have 0 mean; while the variance seems to vary a bit more but it's hard to definitely conclude.

(b)

(i)

(1)

$x_i = x_i - m$ means that our estimators will change accordingly. Suppose $\tilde{\beta_0}$ and $\tilde{\beta_1}$ are the new parameters replacing β_0 and β_1 respectively. So let's take a look at how they relate. So our new model is
$Y_i = \tilde{\beta_0} + \tilde{\beta_1}(x_i - m) + \epsilon_i$
$= \tilde{\beta_0} + \tilde{\beta_1}x_i - \tilde{\beta_1}m + \epsilon_i$
$= (\tilde{\beta_0} - \tilde{\beta_1}m) + \tilde{\beta_1}x_i + \epsilon_i$. So this gives us $\beta_0 = \tilde{\beta_0} - \tilde{\beta_1}m$ and $\beta_1 = \tilde{\beta_1}$.

$\implies \tilde{\beta_1} = \beta_1$, and $\tilde{\beta_0} = \beta_0 + \tilde{\beta_1}m \implies \tilde{\beta_0} = \beta_0 + \beta_1 m$. So the new estimates become $\hat{\beta_1}new = \hat{\beta_1}$ (unchanged) and $\hat{\beta_0}new = \hat{\beta_0} + \hat{\beta_1}m$.

(2)

similarly, if $x_i = lx_i$ means that our estimators will also change accordingly. Suppose $\tilde{\beta_0}$ and $\tilde{\beta_1}$ are the new parameters replacing β_0 and β_1 respectively. Then our new model is $Y_i = \tilde{\beta_0} + \tilde{\beta_1}lx \implies \beta_0 = \tilde{\beta_0}$ and $\beta_1 = l\tilde{\beta_1} \implies \tilde{\beta_1} = \frac{1}{l}\beta_1$. So our new estimates are $\hat{\beta_0}new = \hat{\beta_0}$ and $\hat{\beta_1}new = \frac{1}{l}\hat{\beta_1}$.

Now for the shifting by m case, let's take a look at the new parameters.

```
coef(fit_data1)
```

```
## (Intercept)          x1
```

```
## 0.02676552 1.72512091
```

```r
fit_shiftedm<-lm(y1~I(x1-2),data=data1) #subtracted 2 from all x's in the data
coef(fit_shiftedm)
```

```
## (Intercept)   I(x1 - 2)
##    3.477007    1.725121
```

As we can see, the slope β_1 did not change, but the new intercept is $0.02676552 + (1.72512091) * 2 = \beta_0 + \beta_1 m = 3.477007$. So this indeed agrees with our result.

(ii)

For the shift by m case, we had that $\hat{\beta}_1 new = \hat{\beta}_1$ and $\hat{\beta}_0 new = \hat{\beta}_0 + \hat{\beta}_1 m$. Then,
$E[\hat{\beta}_1 new] = E[\hat{\beta}_1] = \beta_1 = \tilde{\beta}_1$
$E[\hat{\beta}_0 new] = E[\hat{\beta}_0 + \hat{\beta}_1 m] = \beta_0 + m\beta_1 = \tilde{\beta}_0$
$Var(\hat{\beta}_1 new) = Var(\hat{\beta}_1) = \frac{\sigma^2}{nS_x^2}$
$Var(\hat{\beta}_0 new) = Var(\hat{\beta}_0) + m^2 Var(\hat{\beta}_1) + 2mCov(\hat{\beta}_0, \hat{\beta}_1) = \frac{\sigma^2}{n^2} \frac{\sum_{i=1}^n x_i^2}{S_x^2} + \frac{m^2\sigma^2}{nS_x^2} + 2m(\frac{-\sigma^2(1/n)\sum_{i=1}^n x_i}{nS_x^2})$.

And for the scale by l case, we had that $\hat{\beta}_0 new = \hat{\beta}_0$ and $\hat{\beta}_1 new = \frac{1}{l}\hat{\beta}_1$. Then,
$E[\hat{\beta}_0 new] = E(\hat{\beta}_0) = \beta_0 = \tilde{\beta}_0$
$E[\hat{\beta}_1 new] = \frac{1}{l}E(\hat{\beta}_1) = \frac{1}{l}\beta_1 = \tilde{\beta}_1$
$Var[\hat{\beta}_0 new] = Var(\hat{\beta}_0) = \frac{\sigma^2}{n^2} \frac{\sum_{i=1}^n x_i^2}{S_x^2}$
$Var[\hat{\beta}_1 new] = \frac{1}{l^2} Var(\hat{\beta}_1) = \frac{1}{l^2} \frac{\sigma^2}{nS_x^2}$.

As the expected values in both cases ended up being the same as the new parameters ($\tilde{\beta}_0$ and $\tilde{\beta}_1$), our new estimators remain unbiased, however the variances changed as shown above.

Question 2

First note that $Cov(X,Y) = E(X.Y) - E(X).E(Y)$ (definition of Covariance), and
$E[(k_1X_1) + (k_2X_2)] = k_1E(X_1) + k_2E(X_2)$ (Linearity of Expectation) for any 2 random variables X_1, X_2 and constants k_1, k_2. Now: $Cov(a + bX, c + dy) = E[(a + bX) * (c + dY)] - E(a + bX).E(c + dY)$
$= E(ac + adY + bcX + bdXY) - [(a + bE(X)).(c + dE(Y))]$
$= ac + adE(Y) + bcE(X) + bdE(X.Y) - [ac + adE(Y) + bcE(X) + bdE(X).E(Y)]$
$= ac + adE(Y) + bcE(X) + bdE(X.Y) - ac - adE(Y) - bcE(X) - bdE(X).E(Y)$
$= bdE(X.Y) - bdE(X).E(Y) = bd[E(X.Y) - E(X).E(Y)]$
$= bdCov(X,Y)$ by above.

Question 3

(a)

Note that $X \sim Unif(-1,1)$ has mean $\mu_x = (1-1)/2 = 0$, and variance $\sigma_x^2 = 1/12 * (-1-1)^2 = 4/12 = 1/3$. Also, $Var(k_1X_1 + k_2X_2) = k_1^2 Var(X_1) + k_2^2 Var(X_2) + 2k_1k_2Cov(X_1, X_2)$ for any random variables X_1, X_2 and constants k_1, k_2.

$E(Y) = E(5X + \epsilon) = 5E(X) + E(\epsilon) = 0 + 0 = 0$.
$Var(Y) = Var(5X + \epsilon) = 5^2 Var(X) + Var(\epsilon) + 10Cov(X, \epsilon)$. But if X and ϵ are independent, this implies $Cov(X, \epsilon) = 0$.
So we'll have $Var(Y) = 25Var(X) + Var(\epsilon) = 25.(1/3) + 1 = 25/3 + 3/3 = 28/3 \approx 9.33$.

(b)

Note that $Var(Y) = E(Y^2) - [E(Y)]^2 \implies E(Y^2) = Var(Y) + [E(Y)]^2$.
So $E(Y^2) = Var(Y) + [E(Y)]^2 = 28/3 + 0^2 = 28/3$
.

(c)

Given that X takes some value x, we'll have that $E[Y/X = x] = E(5x) + E(\epsilon) = 5x$.

Question 4

(a)

The least square term we need to minimize will be the following:

$\sum_{i=1}^{n}(y_i - \hat{y_i})^2$ where $\hat{y_i}$ is our "prediction" from the model, and y_i is the actual relation. $\sum_{i=1}^{n}(y_i - \hat{y_i})^2 =$ $\sum_{n=1}^{n}(y_i - (\hat{\beta_1}x_i))^2$. Differentiating this with respect to $\hat{\beta_1}$ gives $\sum_{i=1}^{n} 2[y_i - \hat{\beta_1}x_i] * -x_i$ by chain rule, so we have $-2\sum_{i=1}^{n}[y_i - \hat{\beta_1}x_i] * x_i = -2\sum_{i=1}^{n}[y_ix_i - \hat{\beta_1}x_i^2] = -2[\sum_{i=1}^{n} y_ix_i - \hat{\beta_1}\sum_{i=1}^{n} x_i^2]$, and equating this to 0 will optimize it. So let $-2[\sum_{i=1}^{n} y_ix_i - \hat{\beta_1}\sum_{i=1}^{n} x_i^2] = 0 \implies [\sum_{i=1}^{n} y_ix_i - \hat{\beta_1}\sum_{i=1}^{n} x_i^2] = 0 \implies$ $\hat{\beta_1}\sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} y_ix_i \implies \hat{\beta_1} = \frac{\sum_{i=1}^{n} y_ix_i}{\sum_{i=1}^{n} x_i^2}$.

(b)

we need $E\frac{\sum_{i=1}^{n} y_ix_i}{\sum_{i=1}^{n} x_i^2}$.Since $y_i = \beta_1x_i + \epsilon_i$ we can rewrite this as $\frac{\sum_{i=1}^{n}(\beta_1x_i+\epsilon_i)x_i}{\sum_{i=1}^{n} x_i^2}$
$= \frac{\sum_{i=1}^{n}\beta_1x_i^2+\sum_{i=1}^{n}x_i\epsilon_i}{\sum_{i=1}^{n} x_i^2}$
$= \frac{\beta_1\sum_{i=1}^{n}x_i^2+\sum_{i=1}^{n}x_i\epsilon_i}{\sum_{i=1}^{n} x_i^2}$
$= \frac{\beta_1\sum_{i=1}^{n}x_i^2}{\sum_{i=1}^{n} x_i^2} + \frac{\sum_{i=1}^{n}x_i\epsilon_i}{\sum_{i=1}^{n} x_i^2}$
$= \beta_1 + \frac{\sum_{i=1}^{n}x_i\epsilon_i}{\sum_{i=1}^{n} x_i^2}$.

So we got our "constant+noise". Now we need $E[\beta_1 + \frac{\sum_{i=1}^{n}x_i\epsilon_i}{\sum_{i=1}^{n} x_i^2}]$.The right side term is equal to $\frac{x_1\epsilon_1+x_2\epsilon_2+...+x_n\epsilon_n}{\sum_{i=1}^{n} x_i^2}$. Now since $E(\epsilon_i = 0)$ for all i, the right term will shrink to zero when we take the expected value, and so extected value is simply β_1 since it's a constant. This shows that our estimator is unbiased for β_1.

(c)

Bias $= E(\hat{\theta}) - \theta$ where $\hat{\theta}$ is our estimator and θ is what we're trying to estimate. So $\hat{\theta} = \frac{\sum_{i=1}^{n} y_ix_i}{\sum_{i=1}^{n} x_i^2}$ and $\theta = \beta_1$.

Under the condition that $y_i = \beta_0 + \beta_1x_i + \epsilon_i$, our estimator will now be $\frac{\sum_{i=1}^{n}(\beta_0+\beta_1x_i+\epsilon_i)x_i}{\sum_{i=1}^{n} x_i^2}$
$= \frac{\sum_{i=1}^{n}\beta_0x_i+\sum_{i=1}^{n}\beta_1x_i^2+\sum_{i=1}^{n}x_i\epsilon_i}{\sum_{i=1}^{n} x_i^2}$
$= \frac{\beta_0\sum_{i=1}^{n}x_i}{\sum_{i=1}^{n} x_i^2} + \frac{\beta_1\sum_{i=1}^{n}x_i^2}{\sum_{i=1}^{n} x_i^2} + \frac{\sum_{i=1}^{n}x_i\epsilon_i}{\sum_{i=1}^{n} x_i^2}$

$= \beta_1 + \frac{\beta_0 \sum_{i=1}^n x_i}{\sum_{i=1}^n x_i^2} + \frac{\sum_{i=1}^n x_i \epsilon_i}{\sum_{i=1}^n x_i^2}$. Using the fact that $\sum_{i=1}^n x_i = n\bar{x}$, This equals
$\beta_1 + \frac{\beta_0 n\bar{x}}{nx^2} + \frac{\sum_{i=1}^n x_i \epsilon_i}{\sum_{i=1}^n x_i^2}$
$= \beta_1 + \frac{\beta_0 \bar{x}}{x^2} + \frac{\sum_{i=1}^n x_i \epsilon_i}{\sum_{i=1}^n x_i^2}$. Taking the expected value of this, once again the term with ϵ_i's will shrink to 0, so we'll be left with $\beta_1 + \frac{\beta_0 \bar{x}}{x_i^2}$. Thus the bias is $\beta_1 + \frac{\beta_0 \bar{x}}{x_i^2} - \beta_1 = \frac{\beta_0 \bar{x}}{x_i^2}$. Since this isn't only β_1, we say that this estimator is biased.

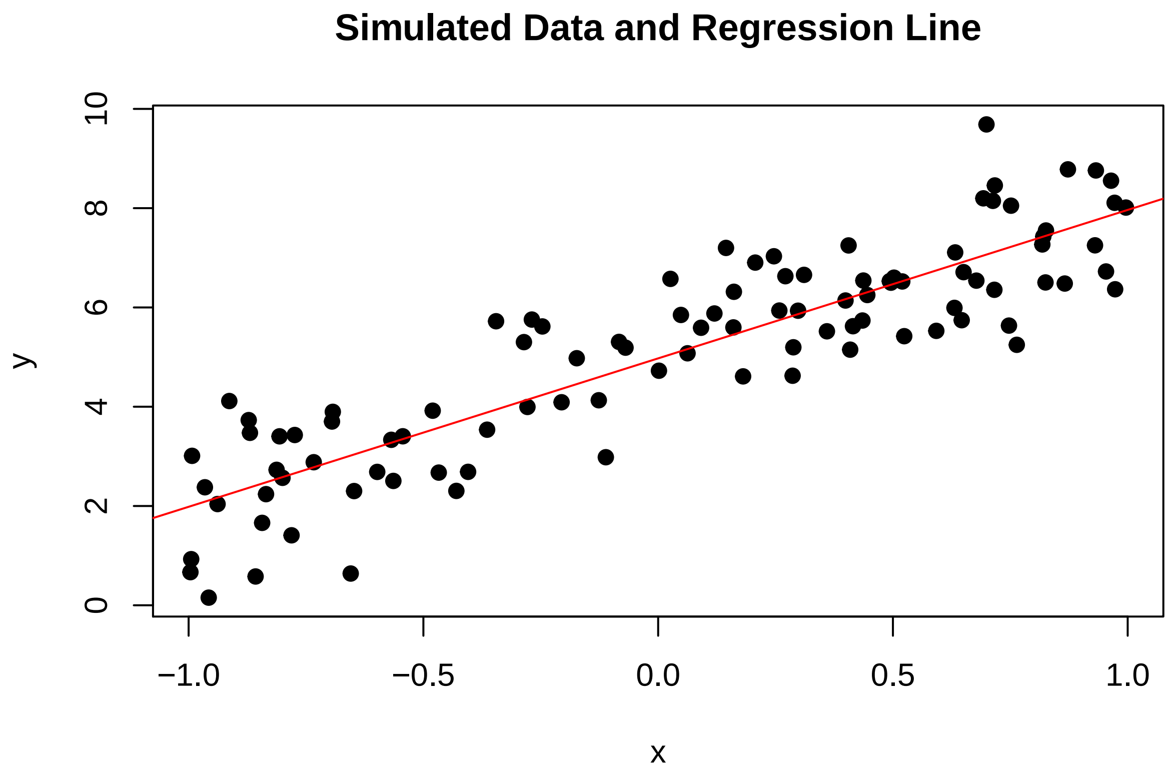
Question 5

The left term is equal to $\frac{1}{n}\sum_{i=1}^n \hat{\beta}_0 + \hat{\beta}_1 x_i$
$= \frac{1}{n}\sum_{i=1}^n \hat{\beta}_0 + \frac{1}{n}\sum_{i=1}^n \hat{\beta}_1 x_i$
$= \frac{1}{n} n\hat{\beta}_0 + \hat{\beta}_1 \frac{1}{n}\sum_{i=1}^n x_i$
$= \hat{\beta}_0 + \hat{\beta}_1 \bar{x}$. By the definition of our least square estimators $\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \implies LHS = \bar{y} - \hat{\beta}_1 \bar{x} + \hat{\beta}_1 \bar{x} = \bar{y}$. But since $\frac{1}{n}\sum_{i=1}^n y_i = \bar{y}$, this is equal to RHS.

Question 6

a)

```
smlt <- function(n, beta_0, beta_1, width) {
#defining our function to simulate data
x_sim <- runif(n, min = -width/2, max = width/2)
#randomly picking n from a uniform distribution
epsilon <- rnorm(n,mean=0,sd=1)
#picking n points from a standard normal distribution
y_sim <- beta_0 + beta_1 * x_sim + epsilon
#establishing our linear model

return(data.frame(x_sim = x_sim, y_sim = y_sim))

}


data_a <- smlt(n = 100,beta_0 = 5, beta_1 = 3, width = 2)
plot(data_a, pch=19,xlab="x",ylab="y")
title(main = 'Simulated Data and Regression Line')
abline(lm(y_sim ~ x_sim, data = data_a),col='red')
```



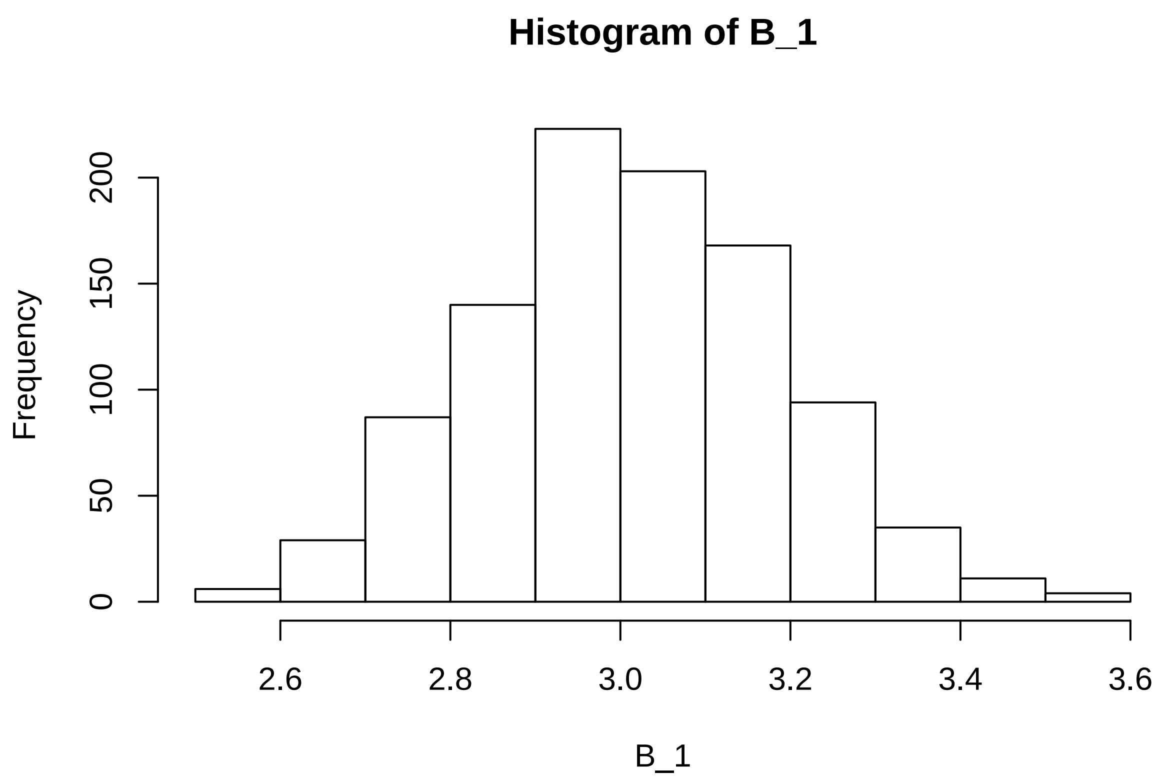

```
#The usual simple linear regression
```

b)

```
B_0 <- rep(0, 1000) #here we create out betas
B_1 <- rep(0, 1000)
for (i in 1:1000) {
  model_b <- lm(y_sim ~ x_sim, data = smlt(n = 100, beta_0 = 5, beta_1 = 3, width = 2))
  B_0[i] <- model_b$coef[1]
  B_1[i] <- model_b$coef[2] #we add the results to out beta vectors in this loop
}
mean_B1 = mean(B_1)
print(mean_B1)
```

```
## [1] 3.009951
```

```
hist(B_1)
```



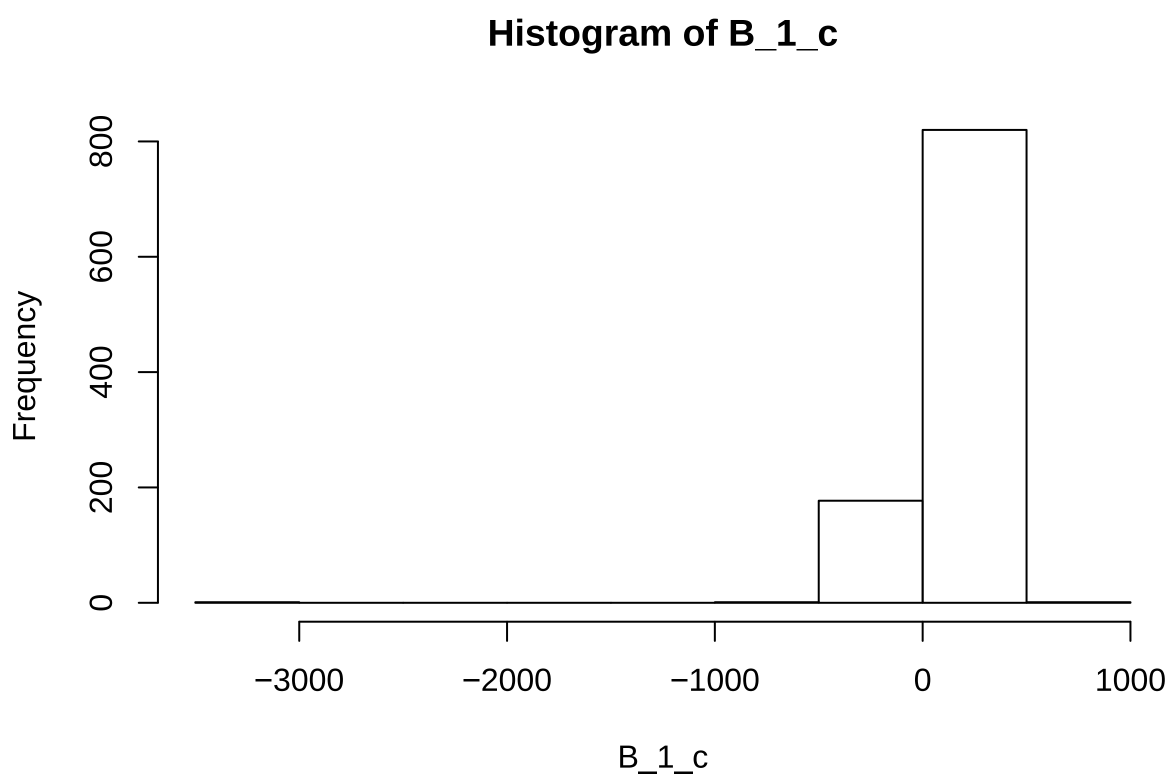
As we can see from the output, the mean we got is very close to the actual value of β_1=3.

c)

```
smlt_c <- function(n, beta_0, beta_1, width) {

x_simc <- runif(n, min = -width/2, max = width/2)
# draw n points from a Cauchy distribution this time
epsilon <- rcauchy(n, location = 0, scale = 1)

y_simc <- beta_0 + beta_1 * x_simc + epsilon

return(data.frame(x_simc = x_simc, y_simc = y_simc))
}

B_0_c <- rep(0, 1000)
B_1_c <- rep(0, 1000)
for (i in 1:1000) {
 model_c <- lm(y_simc ~ x_simc, data = smlt_c(n = 100, beta_0 = 5, beta_1 = 3, width = 2))
  B_0_c[i] <- model_c$coef[1]
  B_1_c[i] <- model_c$coef[2]
}

mean_B1_c = mean(B_1_c)
print(mean_B1_c)
```

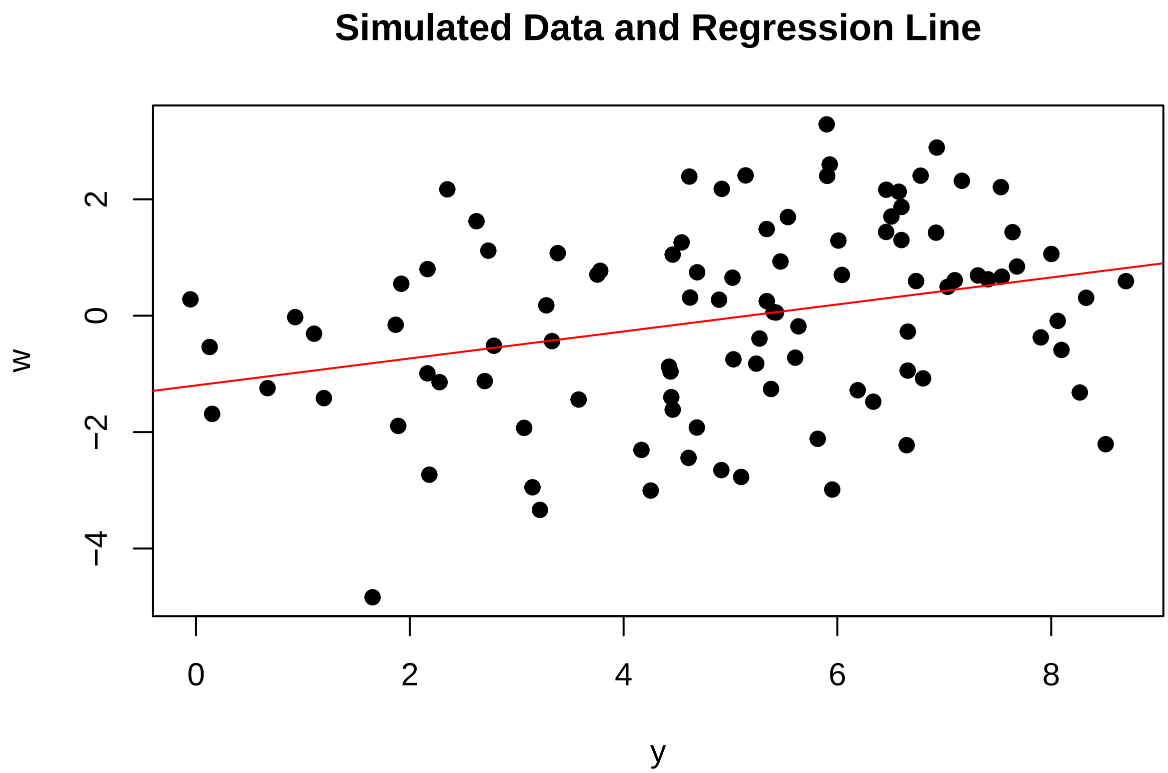
```
## [1] -1.881902
```

```r
hist(B_1_c)
```



The histogram got clustered and it looks a whole lot less like the histogram of a normal distribution (bell curve).

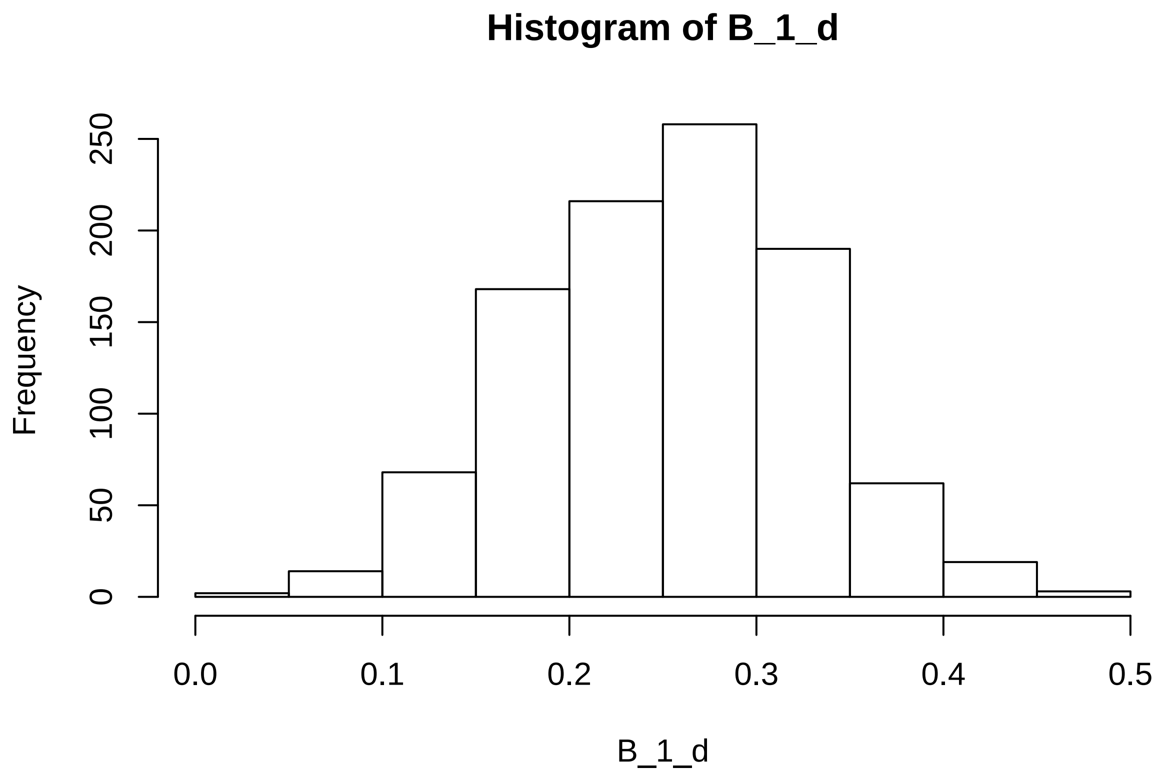
```r
smlt_d <- function(n, beta_0, beta_1, width) {

x_simd <- runif(n, min = -width/2, max = width/2)

epsilon <- rnorm(n,mean=0,sd=1)
delta <- rnorm(n,mean=0,sd=(sqrt(2))) #establishing the errors as specified
#creating our linear model
y_simd <- beta_0 + beta_1 * x_simd + epsilon
w_i <- x_simd + delta #the required Wi

return(data.frame(y_simd = y_simd, w_i=w_i))
}

data_d <- smlt_d(n = 100,beta_0 = 5, beta_1 = 3, width = 2)
plot(data_d, pch=19,xlab="y",ylab="w")
title(main = 'Simulated Data and Regression Line')
abline(lm(w_i ~ y_simd, data = data_d),col='red')
```




```
B_0_d <- rep(0, 1000)
B_1_d <- rep(0, 1000)
for (i in 1:1000) {
 model_d <- lm(w_i ~ y_simd, data = smlt_d(n = 100, beta_0 = 5, beta_1 = 3, width = 2))
  B_0_d[i] <- model_d$coef[1]
  B_1_d[i] <- model_d$coef[2]
}

mean_B1_d = mean(B_1_d)
print(mean_B1_d)
```

```
## [1] 0.2531238
```

```
hist(B_1_d)
```



The histogram we got this time is still centred, somewhere between 0.2 and 0.3, and looks decently"normal" in the sense that we get this bell curve pattern. As we can see from the output, the mean is indeed very off although it wouldn't be as far off if we multiplied it by 10; somewhat less than the originally approximated mean. It seems as the error shifted things to the left and scaled them down by a factor of 10. It's a linear effect, predictably so.