LM HW 4

Joshua Ingram

3/2/2020

Problem 1

Generating x values

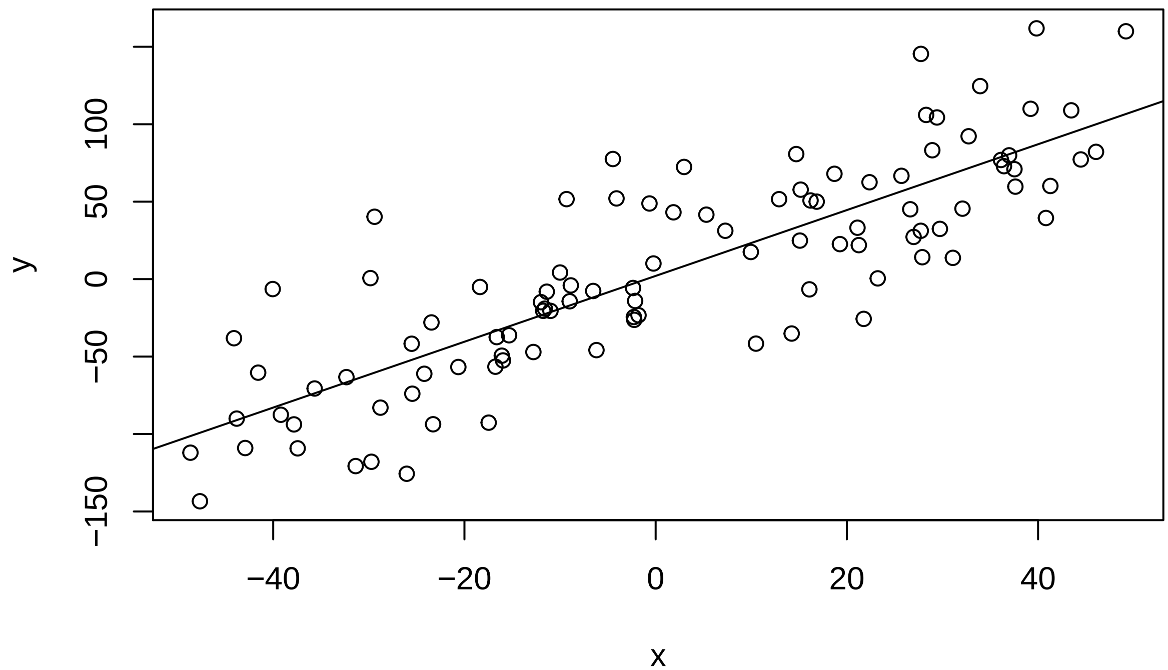
```
set.seed(1)
x <- runif(100, min = -50, max = 50)
```

Model fitting and visualizations

```
eps <- rnorm(100, 0, 40)
y <- 3 + (2 * x) + eps

lm_fit <- lm(y ~ x)
summary(lm_fit)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -73.991 -22.489  -3.483  20.971 100.664
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.0738     3.7727    0.55    0.584
## x             2.1249     0.1414   15.03   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37.64 on 98 degrees of freedom
## Multiple R-squared:  0.6974, Adjusted R-squared:  0.6943
## F-statistic: 225.9 on 1 and 98 DF,  p-value: < 2.2e-16
```

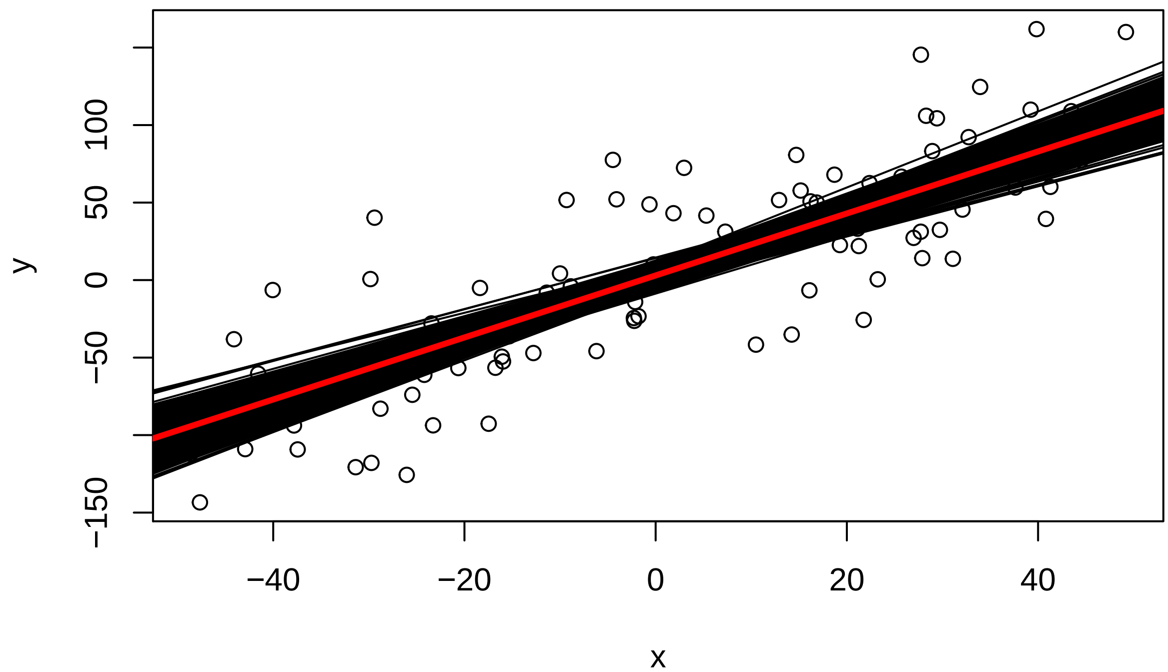
```
plot(y ~ x)
abline(lm_fit)
```



1000 simulations

```
set.seed(1)
beta_hats <- numeric(1000)
beta_ci <- matrix(0, nrow = 1000, ncol = 2)
alpha_hats <- numeric(1000)
alpha_ci <- matrix(0, nrow = 1000, ncol = 2)

plot(y ~ x)

for (i in 1:1000){

  eps <- rnorm(100, 0, 40)

  y <- 3 + (2 * x) + eps

  ## Fit least squares regression y ~ x, plot the resulting fit.
  lm_fit <- lm(y ~ x)

  conf.ints <- confint(lm_fit)
  #plot(y ~ x)
  abline(lm_fit)

  beta_hats[i] <- lm_fit$coefficients[2]
```

```r
  beta_ci[i,] <- conf.ints[2,]

  alpha_hats[i] <- lm_fit$coefficients[1]

  alpha_ci[i,] <- conf.ints[1,]
}

# overlaying the population line
abline(3, 2, lw = 3, col = "red")
```



Lab with beta estimations

Practical and Theoretical Comparisons

Expected Values Practical:

```r
mean(beta_hats)
```

```
## [1] 1.998186
```

Theoretical:

$E[\hat{\beta}] = \beta = 2$

These two are very close (the practical value is .002 off).

Variances Practical:

```r
var(beta_hats)
```

```
## [1] 0.02198641
```

Theoretical:

$V[\hat{\beta}] = \frac{\sigma^2}{\Sigma_i (x_i - \bar{x})^2}$

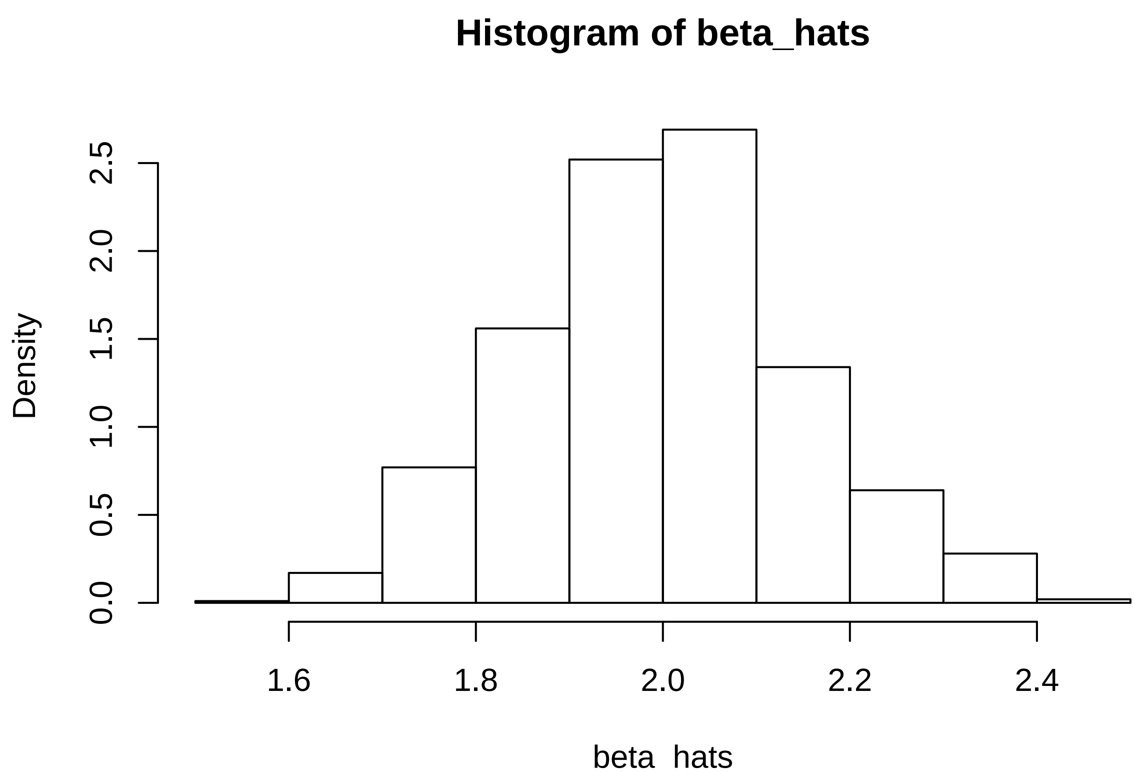
```r
theor_var_beta <- 1600/sum((x - mean(x))^2)
theor_var_beta
```

```
## [1] 0.02257158
```

The practical and theoretical values are virtually the same.

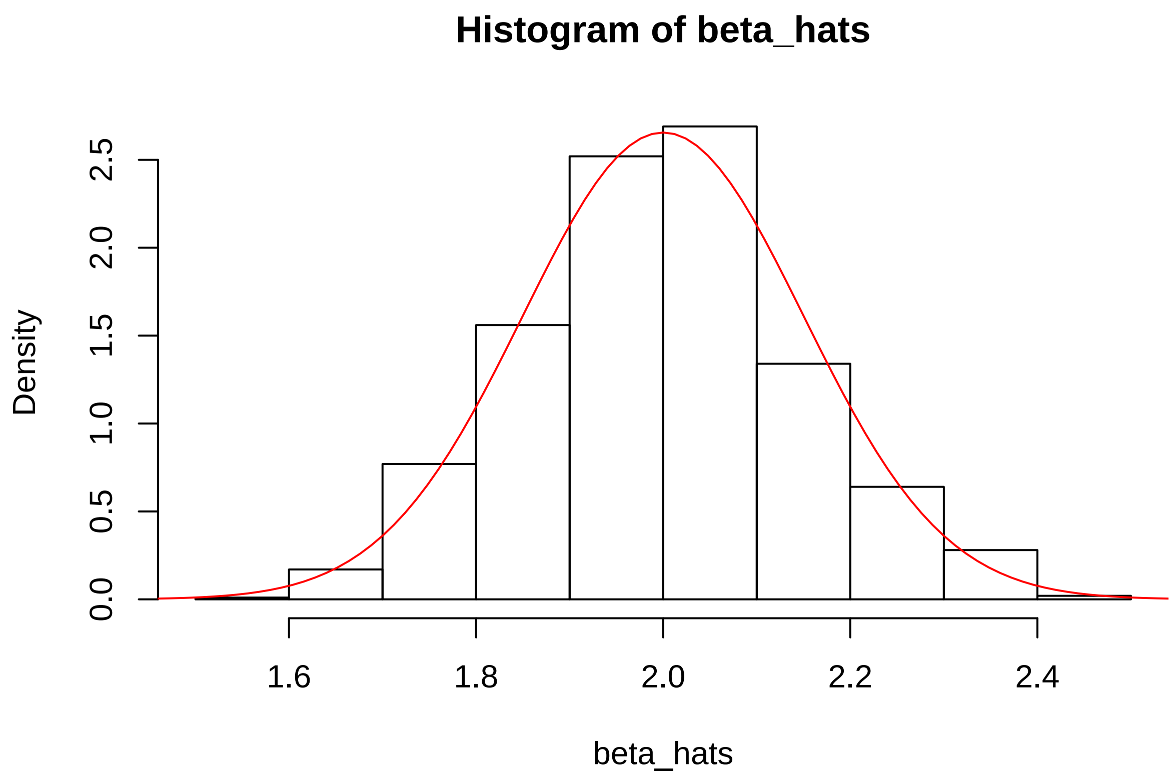
Distribution Practical:

```r
hist(beta_hats, freq = F)
```



Theoretical: $\hat{\beta} \quad N(\beta, V[\hat{\beta}])$

```r
hist(beta_hats, freq = F)
my.dnorm <- function(z) dnorm(z, 2, sqrt(theor_var_beta))
curve(my.dnorm,
      from = 1.4, to = 2.6,
      add = T, col = "red")
```



Theoretical distribution is in red… histograms are the practical distribution… which very closely follows a normal distribution.

Confidence Interval Coverage Practical:

```r
mean(beta_ci[,1] < 2 & beta_ci[,2] > 2)
```

```
## [1] 0.943
```

Theoretical: 0.95

The practical estimate is .007 off, but extremely close to 95% coverage.

Lab with alpha estimations

Practical and Theoretical Comparisons

Expected Values Practical:

```r
mean(alpha_hats)
```

```
## [1] 2.913474
```

Theoretical: $E[\hat{\alpha}] = \alpha = 3$

Very close in values

Variances Practical:

```r
var(alpha_hats)
```

```
## [1] 15.09815
```

Theoretical: $V[\hat{\beta}] = \frac{\sigma^2 \Sigma_i x_i^2}{n \Sigma_i (x_i - \bar{x})^2}$

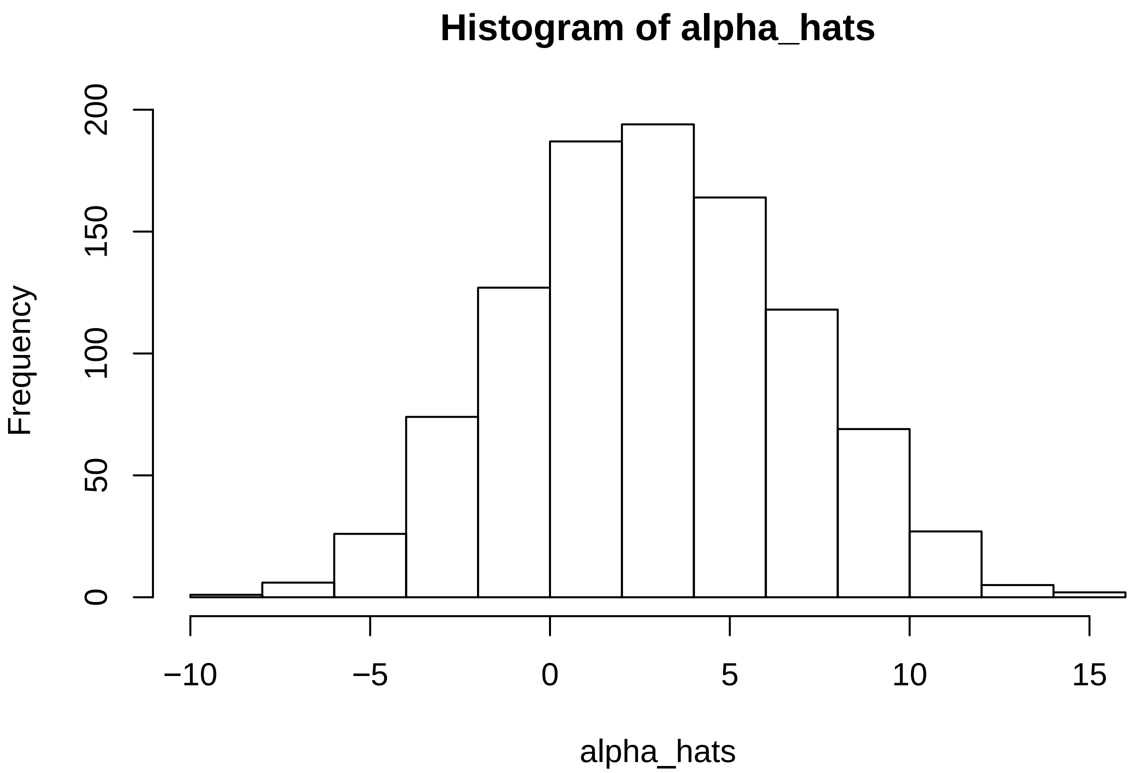
```r
theor_var_alpha <- (1600 * sum(x^2))/(length(x)*sum((x - mean(x))^2))
theor_var_alpha
```

```
## [1] 16.07189
```

Theoretical variance and practical variance are very close in value as expected.

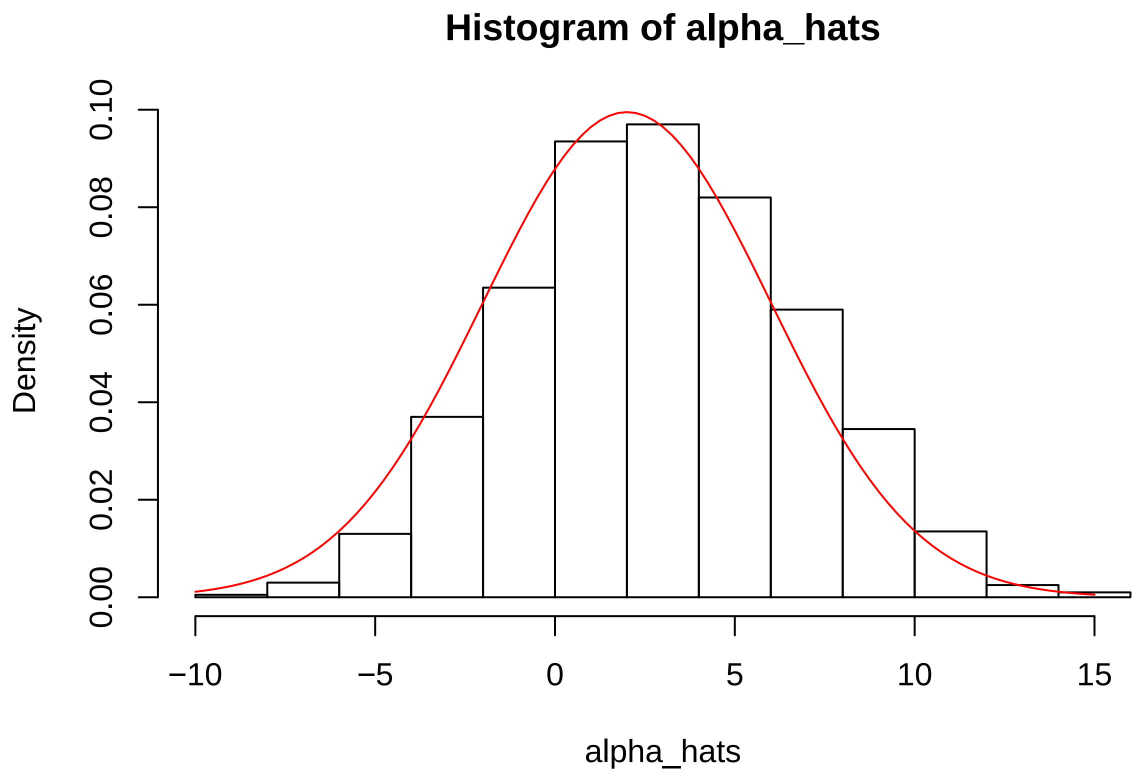
Distribution Practical:

```r
hist(alpha_hats)
```



Theoretical: $\hat{\alpha}$ $N(\alpha, V[\hat{\alpha}])$

```
hist(alpha_hats, freq = F)
my.dnorm <- function(z) dnorm(z, 2, sqrt(theor_var_alpha))
curve(my.dnorm,
      from = -10, to = 15,
      add = T, col = "red")
```



Both follow the expected distributions.

Confidence Interval Practical:

```
mean(alpha_ci[,1] < 2 & alpha_ci[,2] > 2)
```

```
## [1] 0.952
```

Theoretical: 0.95

Practical is only .002 off, so very close.