STAT 8020 R Lab 3: Simple Linear Regression III

Whitney

August 26, 2020

Contents

Understanding Sampling Distributions and Confident Intervals via simulation

Simulate the "data" $\{x_i, y_i\}_{i=1}^n$ where $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$, $\varepsilon \sim \mathrm{N}(0, \sigma^2)$. Repeat this process N times.

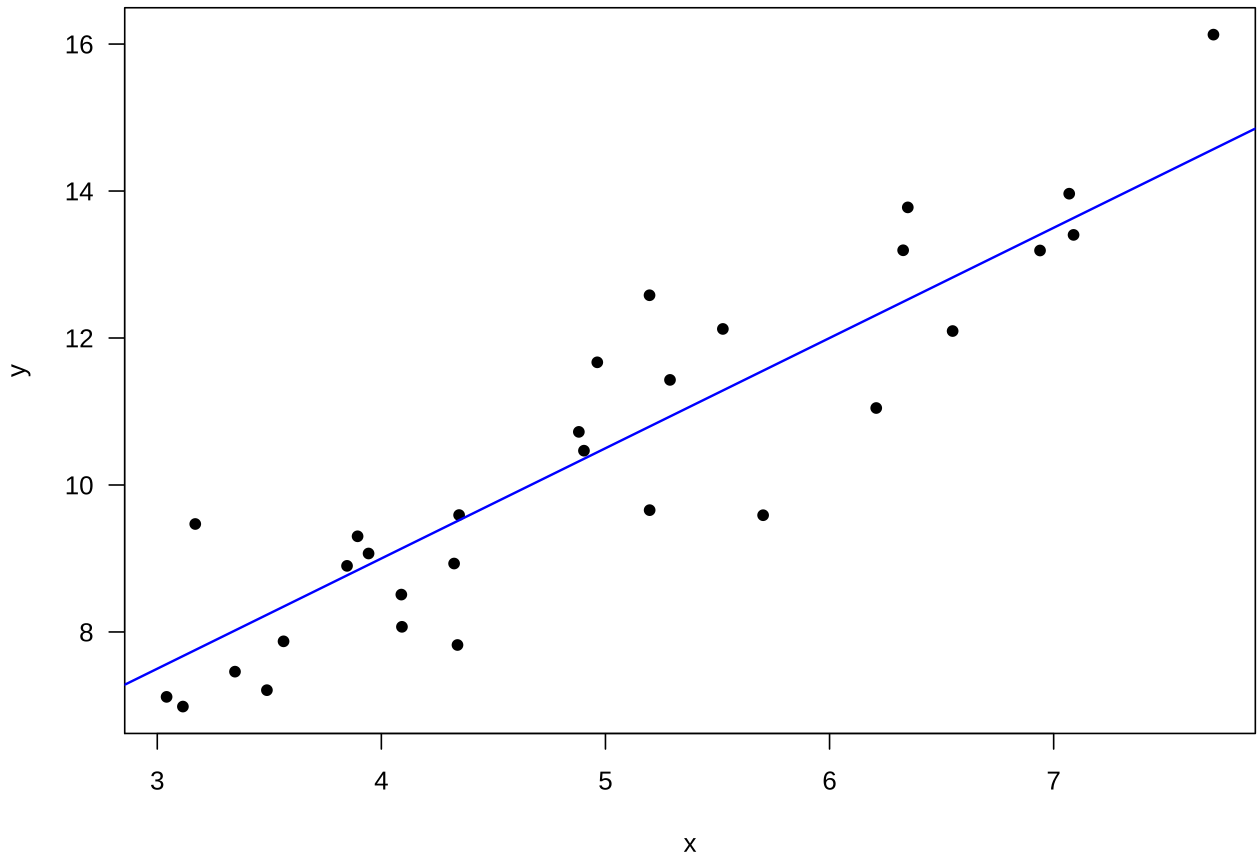
Generate data in R

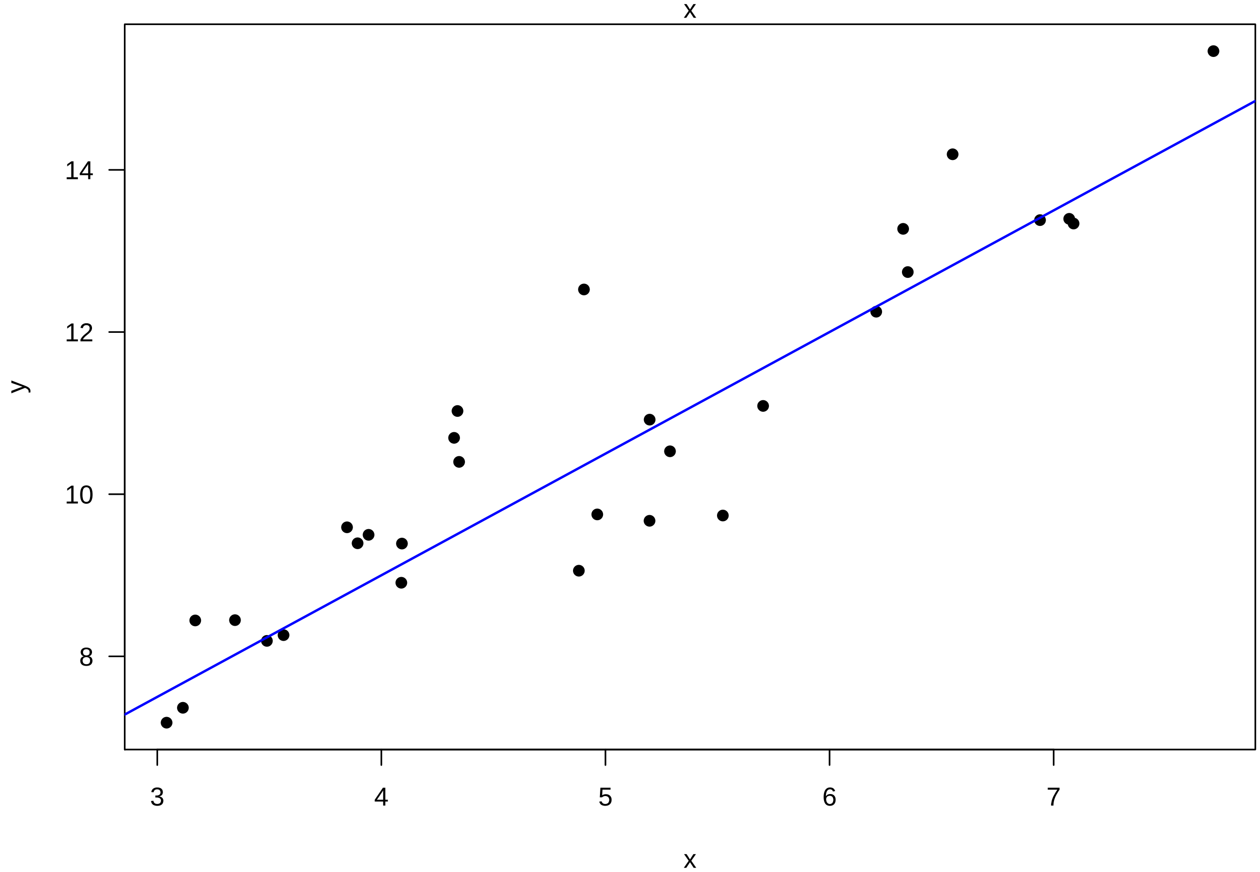
```
set.seed(12)
n = 30; beta0 = 3; beta1 = 1.5; N = 100; sigma2 = 1
x <- 3 + 5 * runif(n)
set.seed(123)
y <- replicate(N, beta0 + beta1 * x + rnorm(n, mean = 0, sd = sqrt(sigma2)))
dim(y)
```

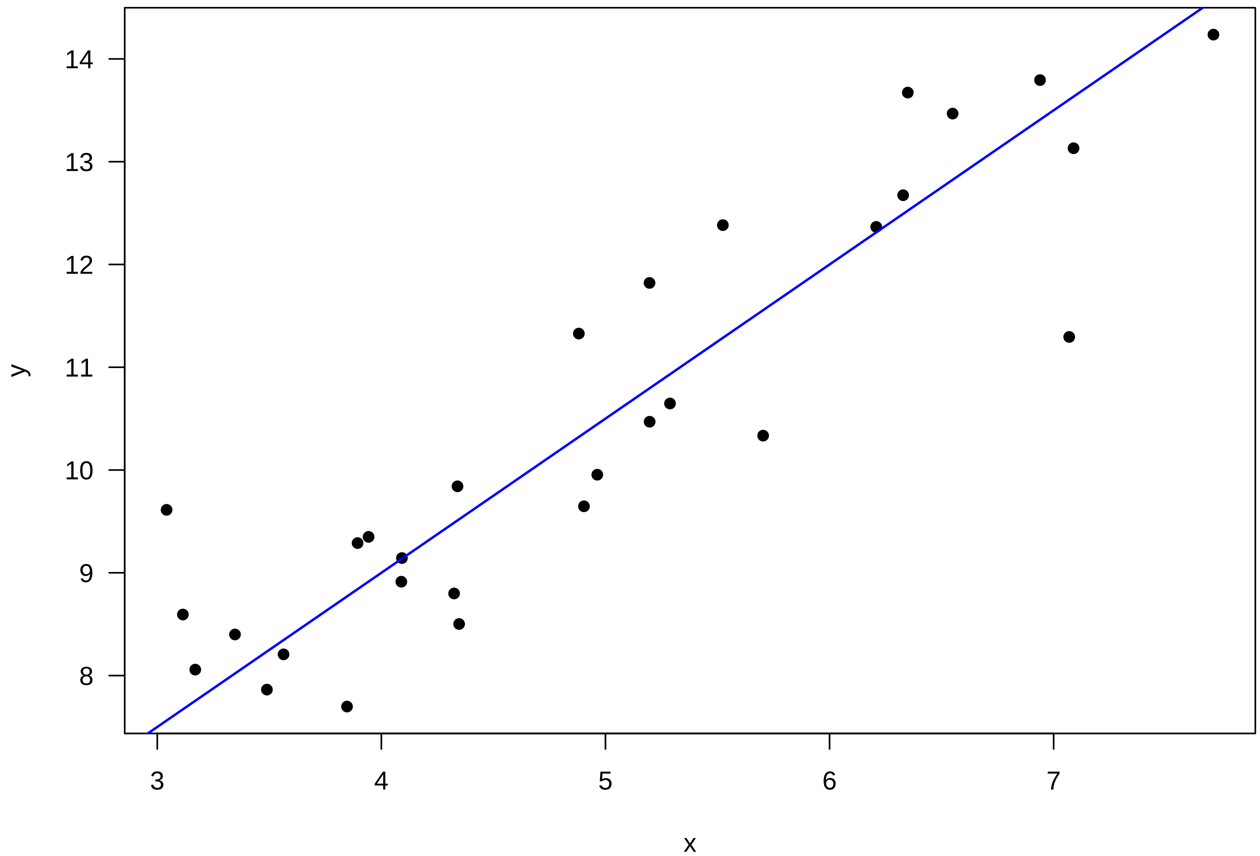
```
## [1]  30 100
```

Plot the first few simulated datasets

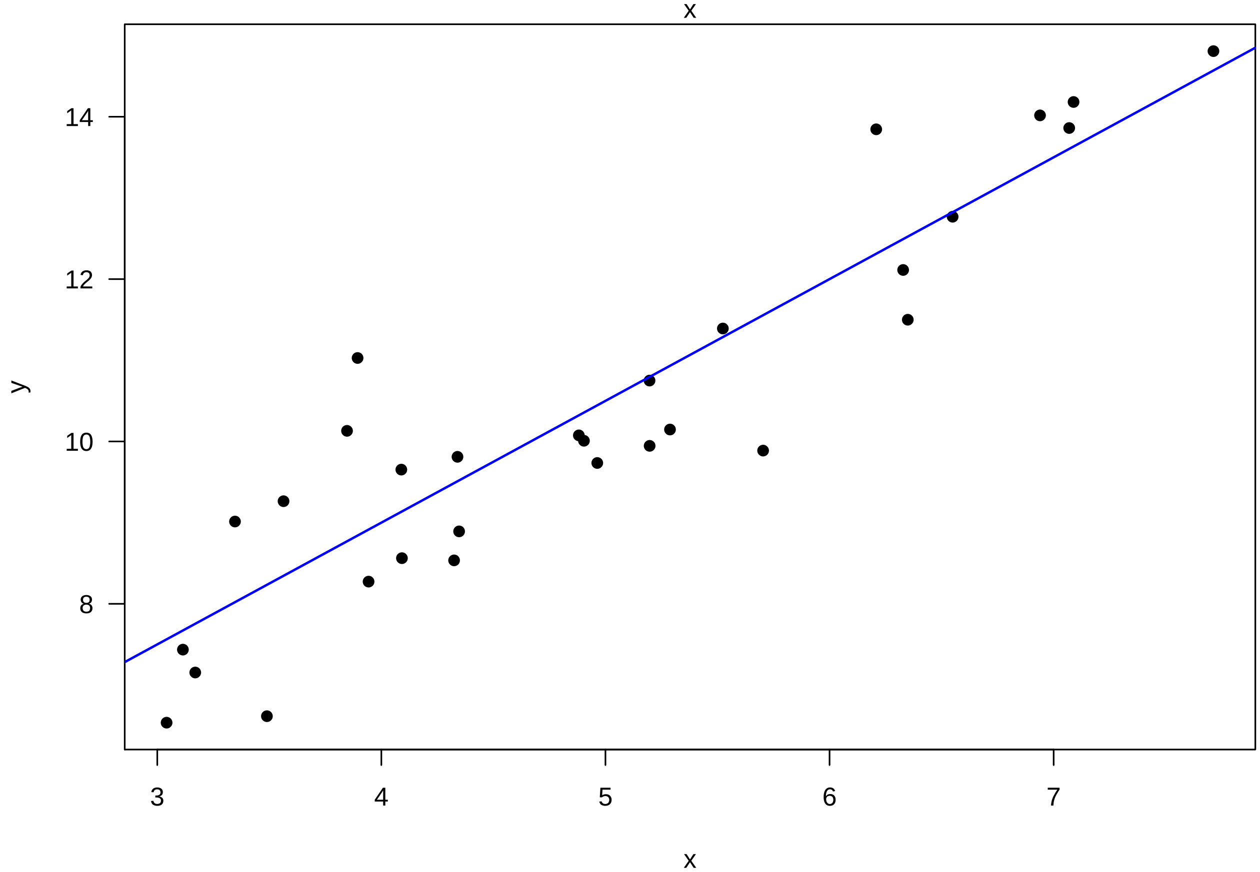
```
for (i in 1:5){
  plot(x, y[, i], pch = 16, las = 1, ylab = "y")
  abline(3, 1.5, col = "blue", lwd = 1.5)
}
```


16
14
12
10
8
y
3
4
5
6
7
x

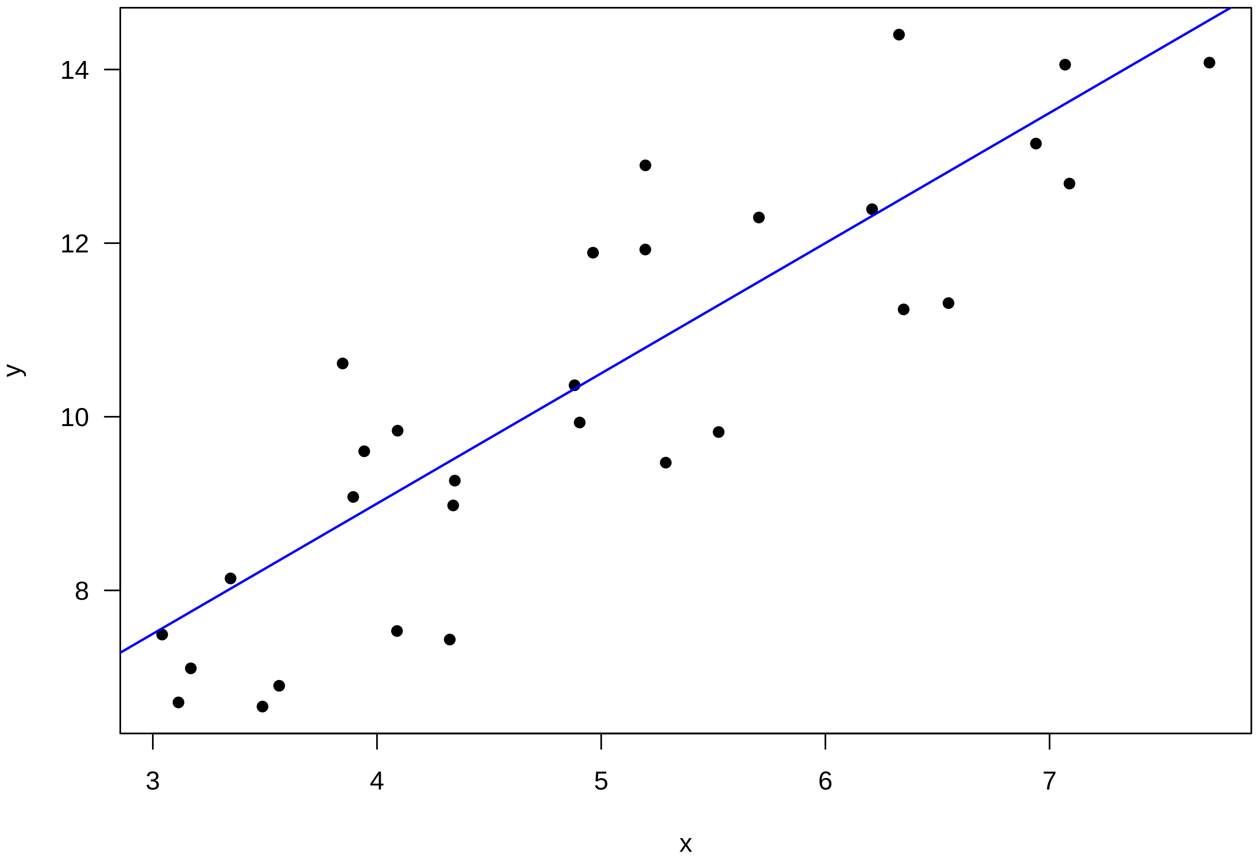

14
12
10
8
y
3
4
5
6
7
x



14
13
12
11
10
9
8
y
3
4
5
6
7
x



14
12
10
8
y
3
4
5
6
7
x

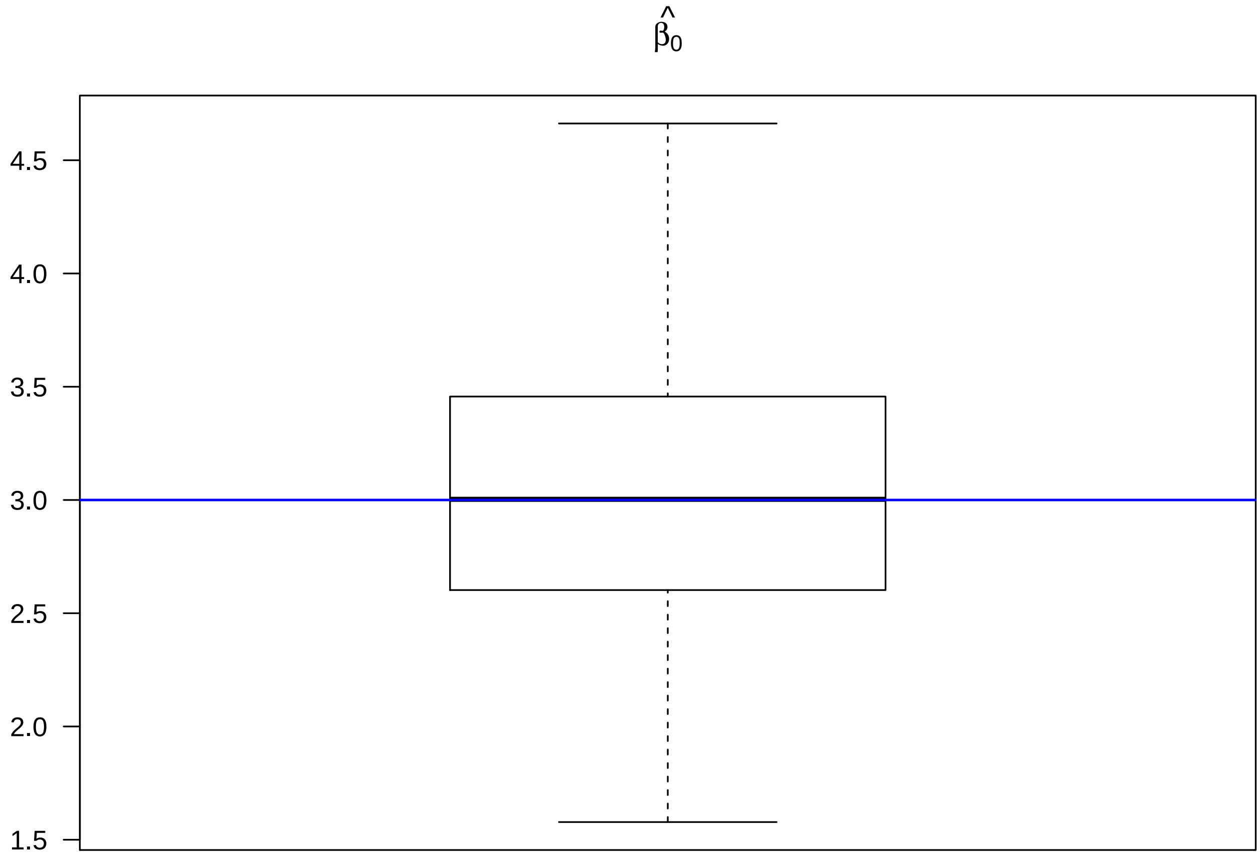


Estimate the β_0, β_1, and σ^2 for each simulated dataset

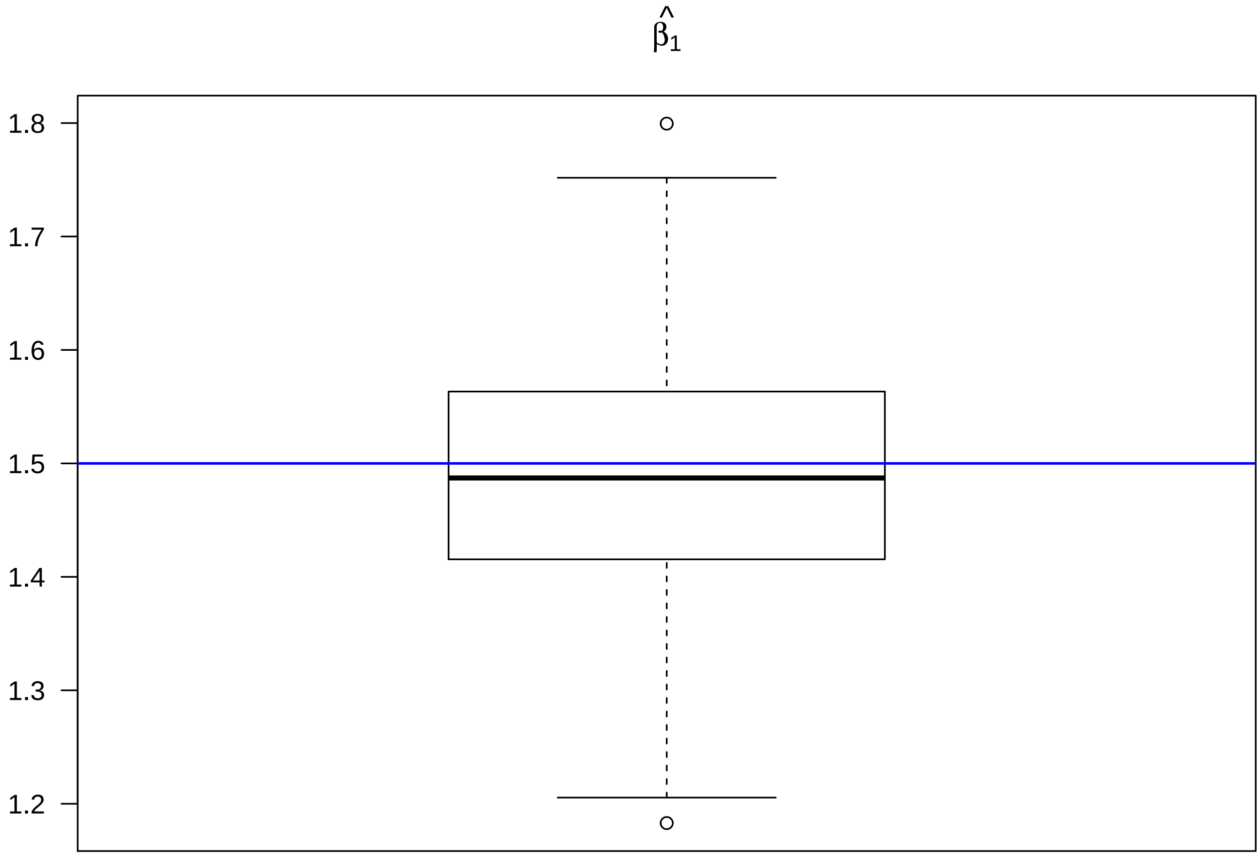
```
beta0_hat <- beta1_hat <- sigma2_hat <- se_beta1 <- numeric(N)
for (i in 1:100){
  fit <- lm(lm(y[, i] ~ x))
  beta0_hat[i] <- summary(fit)[["coefficients"]][, 1][1]
  beta1_hat[i] <- summary(fit)[["coefficients"]][, 1][2]
  se_beta1[i] <- summary(fit)[["coefficients"]][, 2][2]
  sigma2_hat[i] <- summary(fit)[["sigma"]]^2
}
```

Assess the estimation perfromance

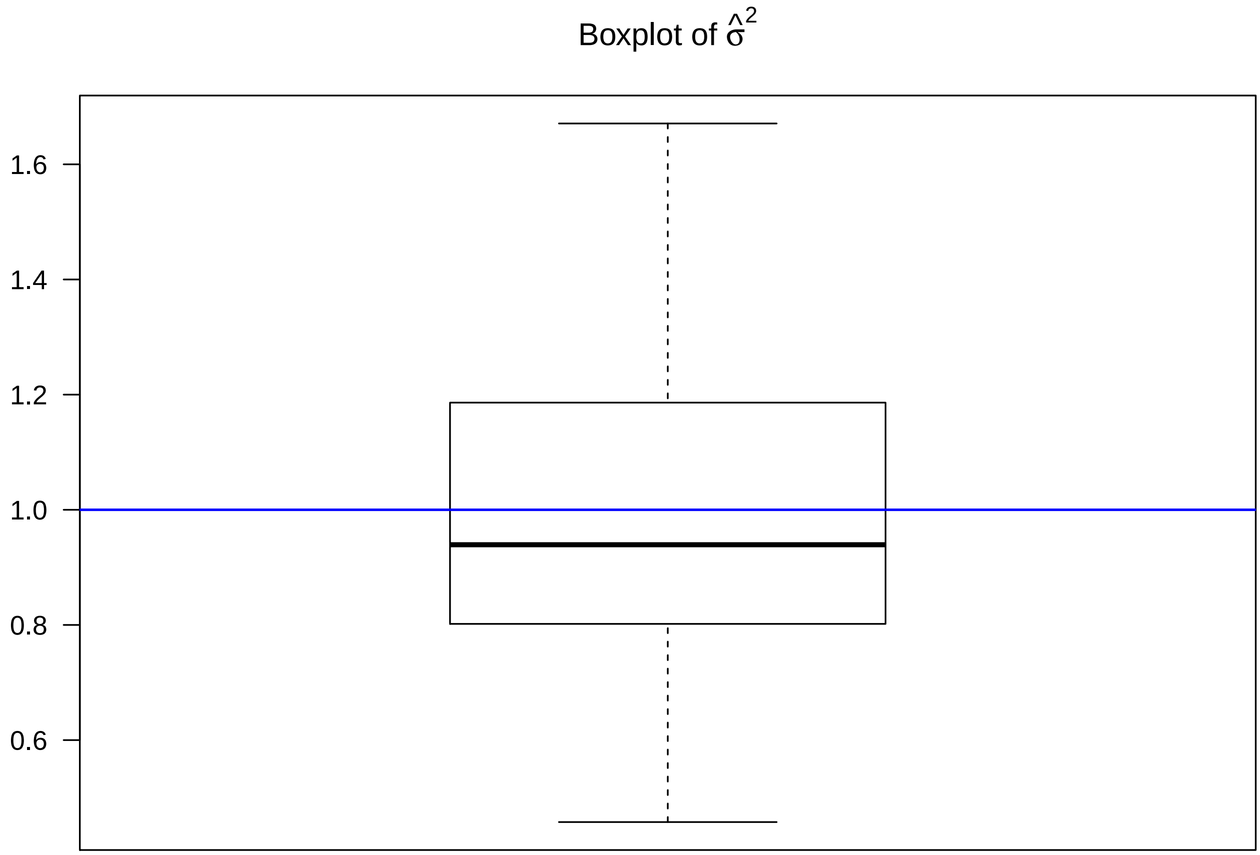
```
boxplot(beta0_hat, las = 1, main = expression(hat(beta[0])))
abline(h = beta0, col = "blue", lwd = 1.5)
```




```r
boxplot(beta1_hat, las = 1, main = expression(hat(beta[1])))
abline(h = beta1, col = "blue", lwd = 1.5)
```

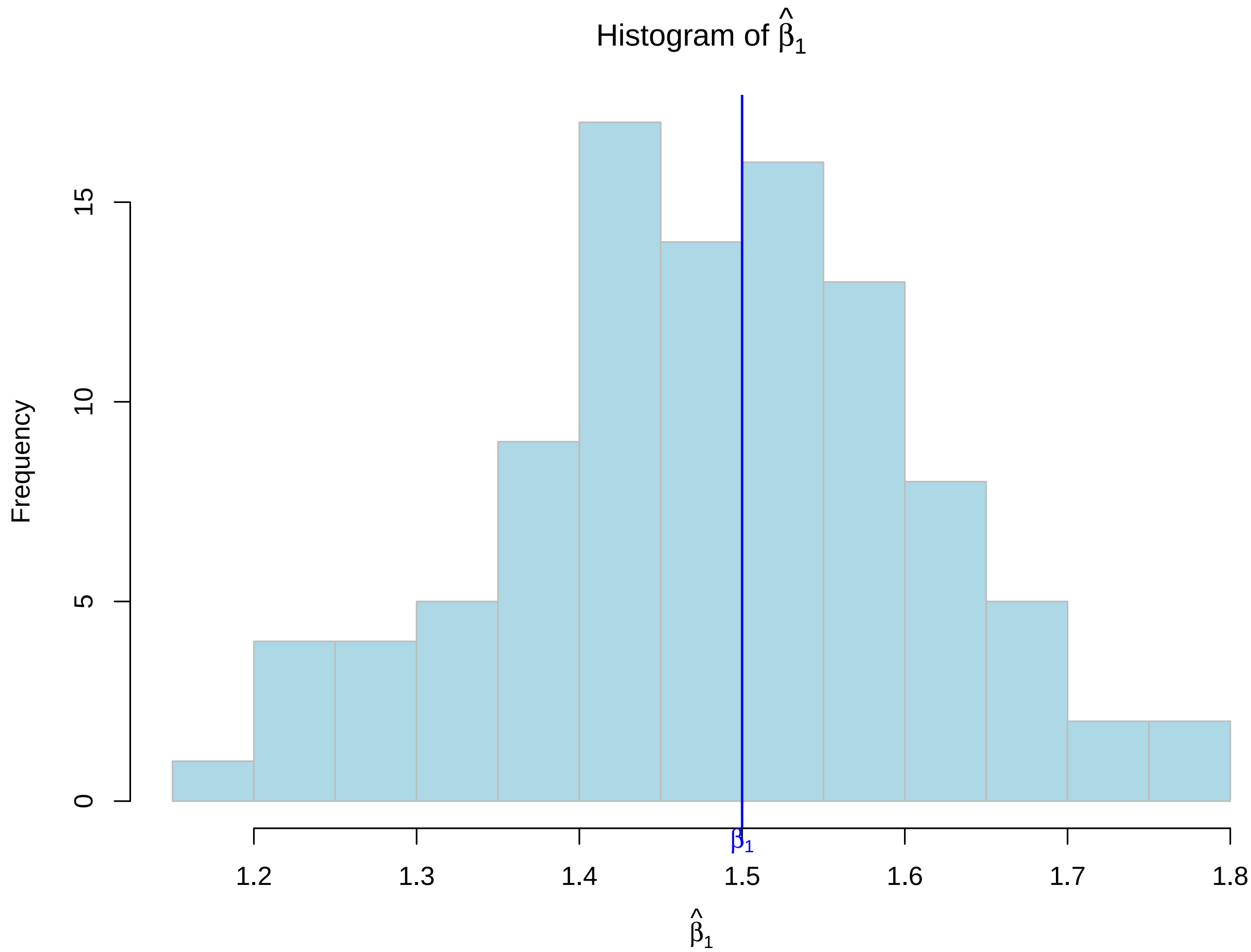



```
boxplot(sigma2_hat, las = 1, main = expression(paste("Boxplot of ", hat(sigma)^2)))
abline(h = sigma2, col = "blue", lwd = 1.5)
```



Sampling distribution

```r
hist(beta1_hat, 16, col = "lightblue", border = "gray",
     main = expression(paste("Histogram of ", hat(beta)[1])),
     xlab = expression(hat(beta)[1]))
abline(v = beta1, col = "blue", lwd = 1.5)
mtext(expression(beta[1]), 1, at = beta1, col = "blue")
```

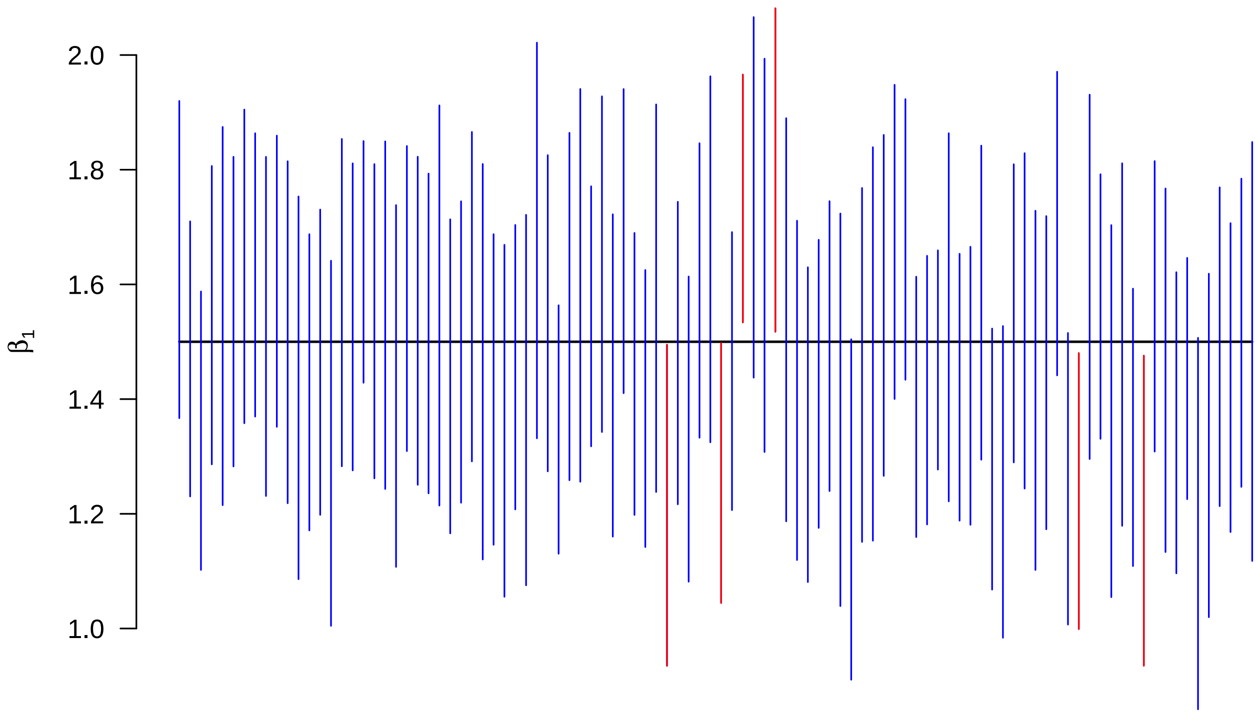


CI's for all the simulated datasets

```
t <- qt(1 - 0.05 / 2, n - 2)
LL <- beta1_hat - t * se_beta1
UL <- beta1_hat + t * se_beta1
miss <- which((LL - beta1) * (UL - beta1) > 0)


par(las = 1)
plot(1:100, rep(beta1, N), type = "l", bty = "n", xaxt = "n", xlab = "",
     lwd = 1.5, ylab = expression(hat(beta)[1]))
for (i in 1:100){
  segments(i, LL[i], i, UL[i], col = "blue")
}

for (i in miss){
  segments(i, LL[i], i, UL[i], col = "red")
}
```



Maximum Heart Rate vs. Age Example

First Step: Load the data

```r
dat <- read.csv('http://whitneyhuang83.github.io/STAT8010/Data/maxHeartRate.csv', header = T)
head(dat)
```

```
##   Age MaxHeartRate
## 1  18          202
## 2  23          186
## 3  25          187
## 4  35          180
## 5  65          156
## 6  54          169
```

```r
attach(dat)
```

Fitting a simple linear regression

```r
fit <- lm(MaxHeartRate ~ Age)
summary(fit)
```

```
##
## Call:
## lm(formula = MaxHeartRate ~ Age)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.9258 -2.5383  0.3879  3.1867  6.6242
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 210.04846    2.86694   73.27  < 2e-16 ***
## Age          -0.79773    0.06996  -11.40 3.85e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.578 on 13 degrees of freedom
## Multiple R-squared:  0.9091, Adjusted R-squared:  0.9021
## F-statistic:   130 on 1 and 13 DF,  p-value: 3.848e-08
```

Confidence Interval

β_1

```
alpha = 0.05
beta1_hat <- summary(fit)[["coefficients"]][, 1][2]
se_beta1 <- summary(fit)[["coefficients"]][, 2][2]
CI_beta1 <- c(beta1_hat - qt(1 - alpha / 2, 13) * se_beta1,
              beta1_hat + qt(1 - alpha / 2, 13) * se_beta1)
CI_beta1
```

```
##        Age        Age
## -0.9488720 -0.6465811
```

$Y_h | X_h = 40$

```
Age_new = data.frame(Age = 40)
hat_Y <- fit$coefficients[1] + fit$coefficients[2] * 40
hat_Y
```

```
## (Intercept)
##    178.1394
```

```
predict(fit, Age_new, interval = "confidence")
```

```
##        fit      lwr      upr
## 1 178.1394 175.5543 180.7245
```

```
predict(fit, Age_new, interval = "predict")
```

```
##        fit      lwr      upr
## 1 178.1394 167.9174 188.3614
```

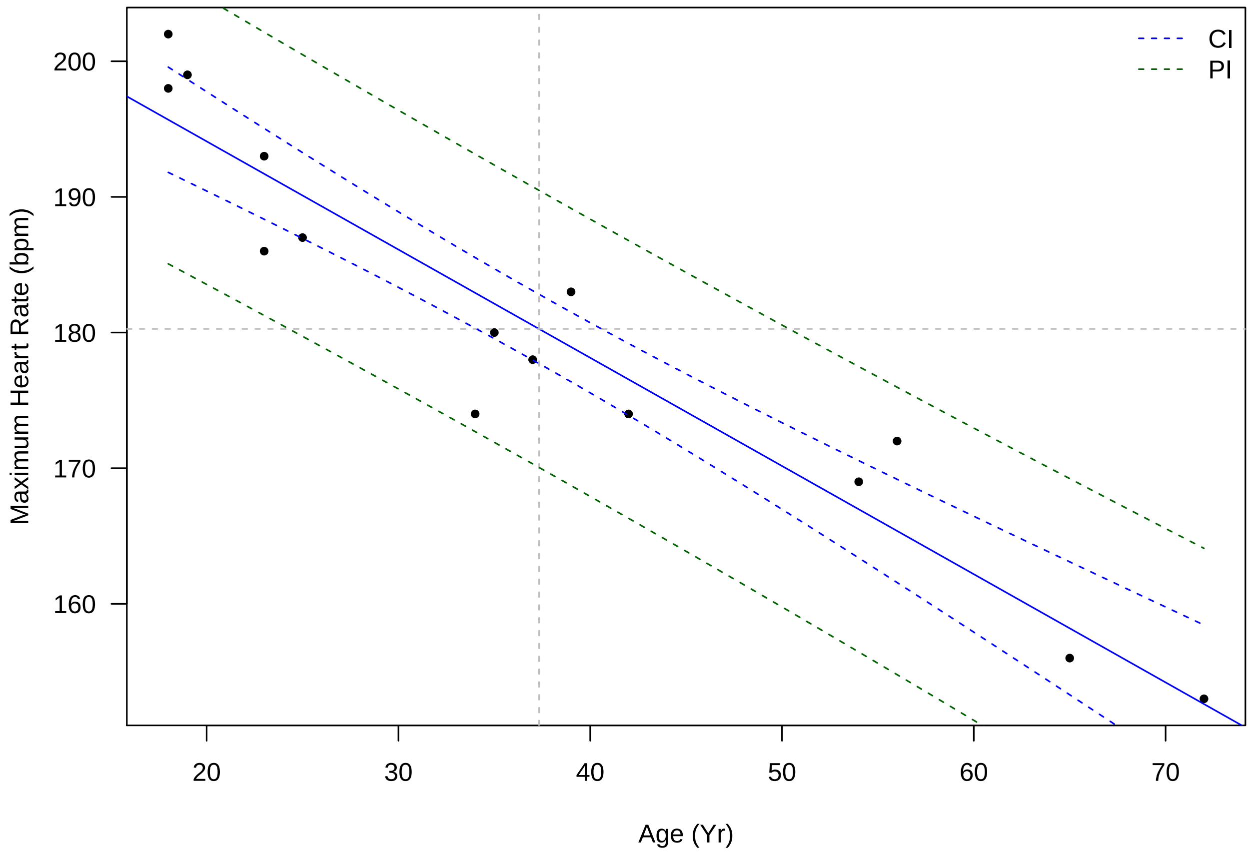
Check

```
sd <- sqrt((sum(fit$residuals^2) / 13))
ME <- qt(1 - alpha / 2, 13) * sd * sqrt(1 + 1 / 15 + (40 - mean(Age))^(2) / sum((Age - mean(Age))^2))
c(hat_Y - ME, hat_Y + ME)
```

```
## (Intercept) (Intercept)
##    167.9174    188.3614
```

```
Age_grid = data.frame(Age = 18:72)
CI_band <- predict(fit, Age_grid, interval = "confidence")
PI_band <- predict(fit, Age_grid, interval = "predict")
```

```
plot(dat$Age, dat$MaxHeartRate, pch = 16, cex = 0.75,
     xlab = "Age (Yr)", ylab = "Maximum Heart Rate (bpm)", las = 1)
abline(fit, col = "blue")
abline(v = mean(dat$Age), lty = 2, col = "gray")
abline(h = mean(dat$MaxHeartRate), lty = 2, col = "gray")
lines(18:72, CI_band[, 2], lty = 2, col = "blue")
lines(18:72, CI_band[, 3], lty = 2, col = "blue")
lines(18:72, PI_band[, 2], lty = 2, col = "darkgreen")
lines(18:72, PI_band[, 3], lty = 2, col = "darkgreen")
legend("topright", legend = c("CI", "PI"), col = c("blue", "darkgreen"),
       lty = 2, bty = "n")
```



Hypothesis Tests for β_1

$H_0 : \beta_1 = -1$ vs. $H_a : \beta_1 \neq -1$ with $\alpha = 0.05$

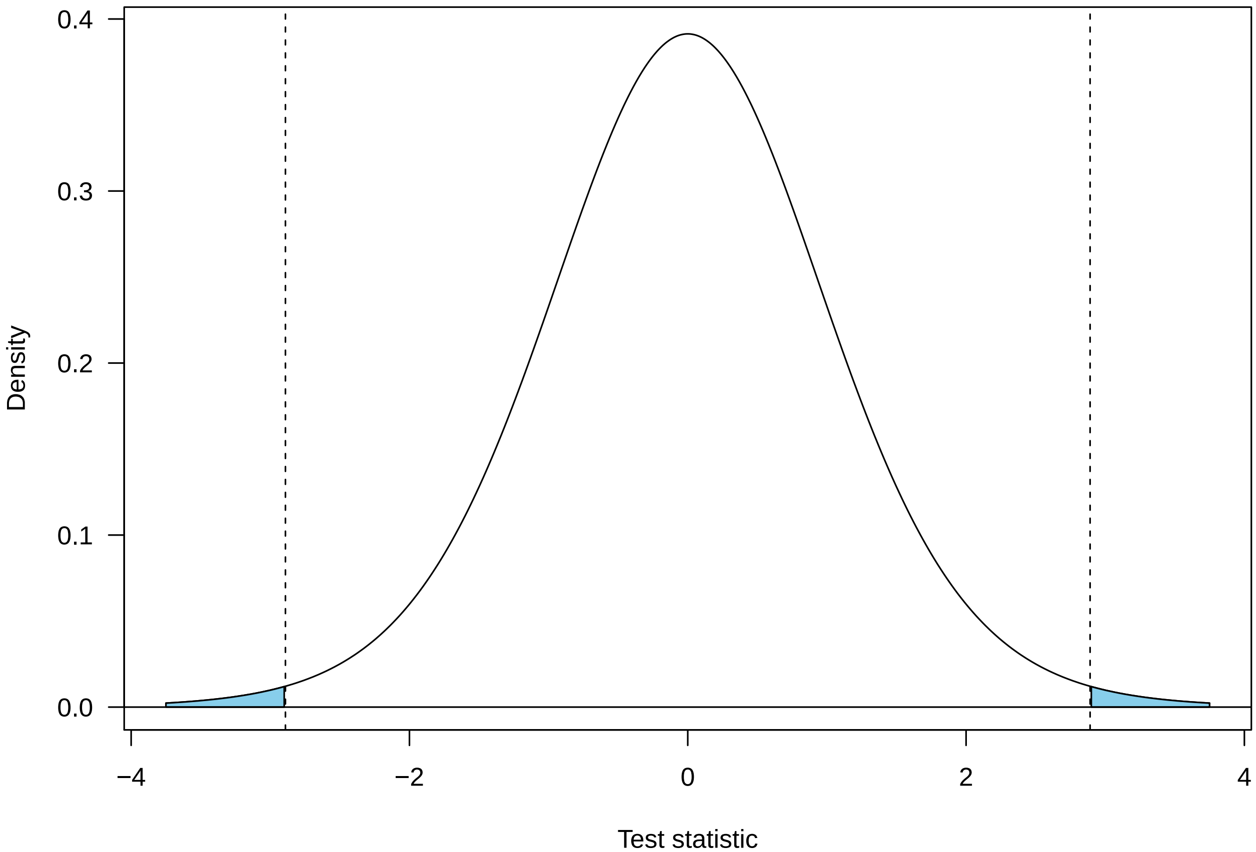
```
beta1_null <- -1
t_star <- (beta1_hat - beta1_null) / se_beta1
p_value <- 2 * pt(t_star, 13, lower.tail = F)
p_value
```

```
##        Age
## 0.01262031
```

```
par(las = 1)
x_grid <- seq(-3.75, 3.75, 0.01)
y_grid <- dt(x_grid, 13)
```

```r
plot(x_grid, y_grid, type = "l", xlab = "Test statistic", ylab = "Density", xlim = c(-3.75, 3.75))
polygon(c(x_grid[x_grid < -t_star], rev(x_grid[x_grid < -t_star])),
        c(y_grid[x_grid < -t_star], rep(0, length(y_grid[x_grid < -t_star]))), col = "skyblue")

polygon(c(x_grid[x_grid > t_star], rev(x_grid[x_grid > t_star])),
        c(y_grid[x_grid > t_star], rep(0, length(y_grid[x_grid > t_star]))), col = "skyblue")
abline(v = t_star, lty = 2)
abline(v = -t_star, lty = 2)
abline(h = 0)
```



ANOVA

```r
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: MaxHeartRate
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## Age        1 2724.50 2724.50  130.01 3.848e-08 ***
## Residuals 13  272.43   20.96
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```