STAT 8010 R Session 6: Simple Linear Regression

Whitney Huang

6/20/2023

Contents

Session Objectives

- To gain experience with R, a programming language and free software environment for statistical computing and graphics.
- To perform simple linear regression using `R`

Example: Maximum Heart Rate vs. Age

The maximum heart rate (HR_{max}) of a person is often said to be related to age (Age) by the equation:

$$\mathrm{HR}_{max} = 220 - \mathrm{Age}$$

Let's use a dataset to assess this statement.

Load the dataset

There are several ways to load a dataset into R:

- Importing Data over the Internet

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

- Read the dataset from you computer

```
dat <- read.csv('maxHeartRate.csv', header = T)
```

- If the dataset is not too big, you can type the data into R

```
age <- c(18, 23, 25, 35, 65, 54, 34, 56, 72, 19, 23, 42, 18, 39, 37)
maxHeartRate <- c(202, 186, 187, 180, 156, 169, 174, 172, 153,
                  199, 193, 174, 198, 183, 178)
dat <- data.frame(cbind(age, maxHeartRate))
```

Let's take a look at the data

```
dat
```

```
##    age maxHeartRate
## 1   18          202
## 2   23          186
## 3   25          187
## 4   35          180
## 5   65          156
## 6   54          169
## 7   34          174
## 8   56          172
## 9   72          153
## 10  19          199
## 11  23          193
## 12  42          174
## 13  18          198
## 14  39          183
## 15  37          178
```

Examine the data before fitting models

```
summary(dat)
```

```
##       age         maxHeartRate
##  Min.   :18.00   Min.   :153.0
##  1st Qu.:23.00   1st Qu.:173.0
##  Median :35.00   Median :180.0
##  Mean   :37.33   Mean   :180.3
##  3rd Qu.:48.00   3rd Qu.:190.0
##  Max.   :72.00   Max.   :202.0
```

```
var(dat$age); var(dat$maxHeartRate)
```

```
## [1] 305.8095

## [1] 214.0667
```

```
cov(dat$age, dat$maxHeartRate)
```

```
## [1] -243.9524
```

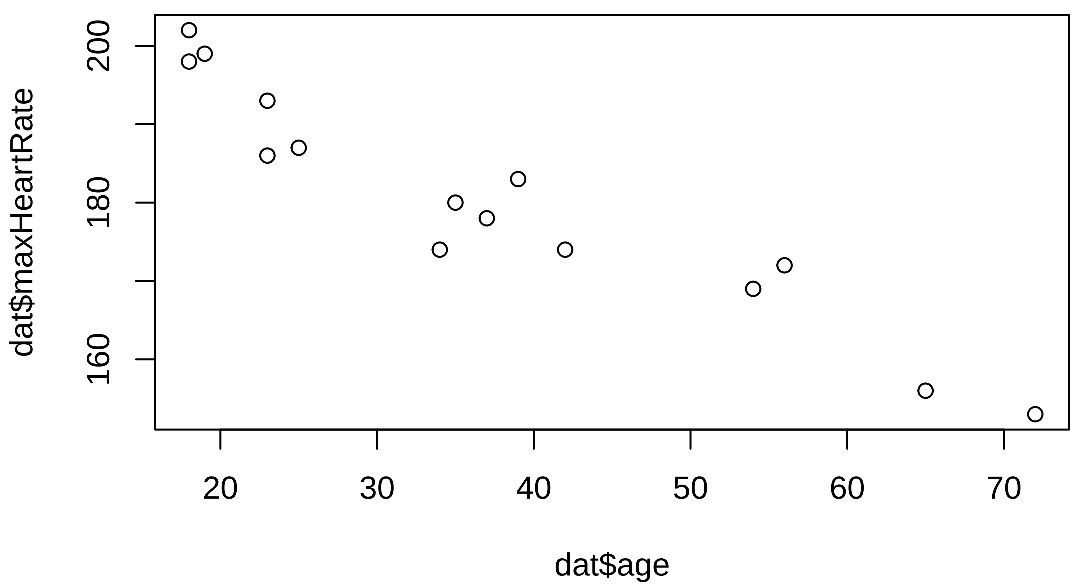
```
cor(dat$age, dat$maxHeartRate)
```

```
## [1] -0.9534656
```

Plot the data before fitting models

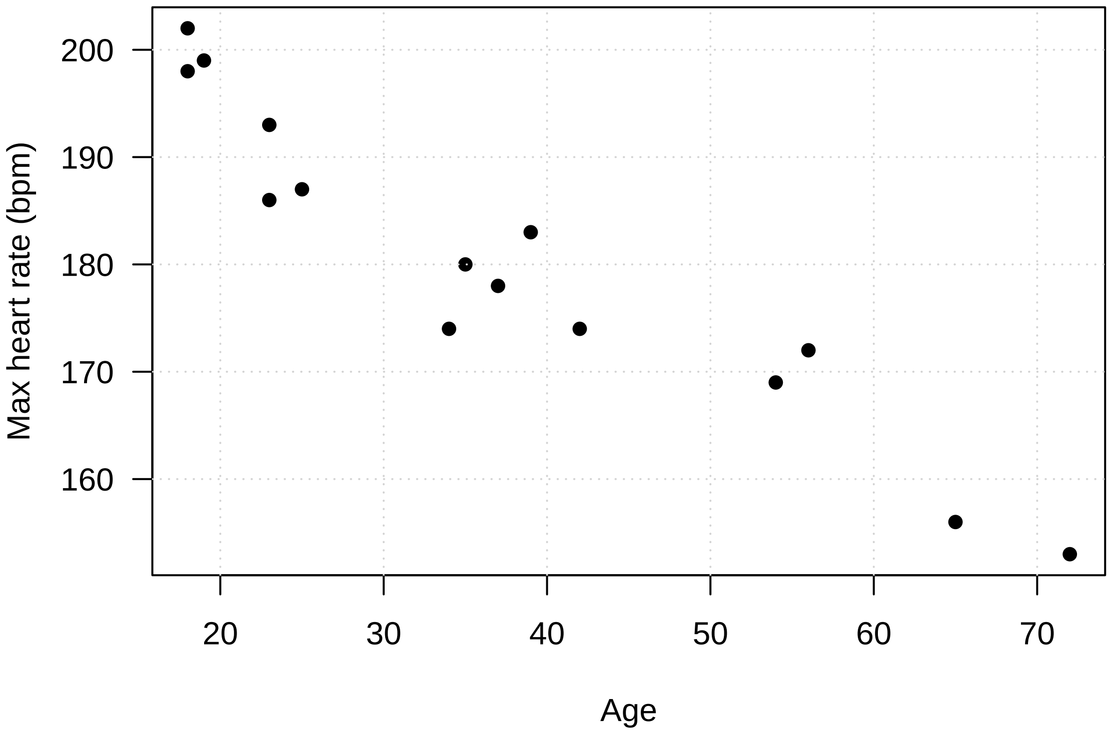
This is what the scatterplot would look like by default. Put predictor (age) to the first argument and response (maxHeartRate) to the second argument.

```
plot(dat$age, dat$maxHeartRate)
```



Let's make the plot look nicer (type ?plot to learn more).

```
par(las = 1, mar = c(4.1, 4.1, 1.1, 1.1))
plot(dat$age, dat$maxHeartRate,
     pch = 16, xlab = "Age", ylab = "Max heart rate (bpm)")
grid()
```



Question: Describe the direction, strength, and the form of the relationship.

Simple linear regression

Let's do the calculations to figure out the regression coefficients as well as the standard deviation of the random error.

- Slope: $\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$

```
X <- dat$age; Y <- dat$maxHeartRate
Y_diff <- Y - mean(Y)
X_diff <- X - mean(X)
beta_1 <- sum(Y_diff * X_diff) / sum((X_diff)^2)
beta_1
```

```
## [1] -0.7977266
```

- Intercept: $\hat{\beta}_0 = \bar{y} - \bar{x}\hat{\beta}_1$

```r
beta_0 <- mean(Y) - mean(X) * beta_1
beta_0
```

```
## [1] 210.0485
```

- Fitted values: $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$

```r
Y_hat <- beta_0 + beta_1 * X
Y_hat
```

```
##  [1] 195.6894 191.7007 190.1053 182.1280 158.1962 166.9712 182.9258 165.3758
##  [9] 152.6121 194.8917 191.7007 176.5439 195.6894 178.9371 180.5326
```

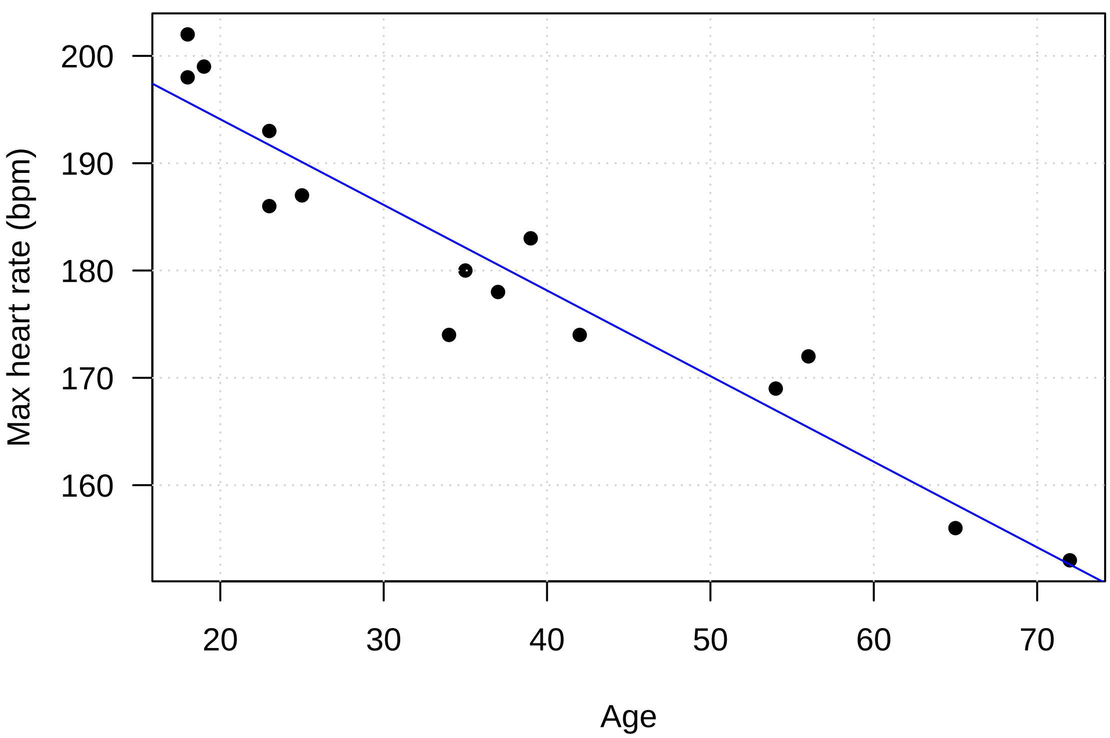
- $\hat{\sigma}$: $\hat{\sigma}^2 = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-2}$

```r
sigma2 <- sum((Y - Y_hat)^2) / (length(Y) - 2)
sqrt(sigma2)
```

```
## [1] 4.577799
```

Add the fitted regression line to the scatterplot

```r
par(las = 1, mar = c(4.1, 4.1, 1.1, 1.1))
plot(dat$age, dat$maxHeartRate,
     pch = 16, xlab = "Age",
     ylab = "Max heart rate (bpm)")
grid()
abline(a = beta_0, b = beta_1,
       col = "blue")
```



Let R do all the work

```r
fit <- lm(maxHeartRate ~ age, data = dat)
summary(fit)
```

```
##
## Call:
## lm(formula = maxHeartRate ~ age, data = dat)
##
## 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
```

- Regression coefficients

```r
fit$coefficients
```

```
## (Intercept)         age
## 210.0484584  -0.7977266
```

- Fitted values

```r
fit$fitted.values
```

```
##        1        2        3        4        5        6        7        8
## 195.6894 191.7007 190.1053 182.1280 158.1962 166.9712 182.9258 165.3758
##        9       10       11       12       13       14       15
## 152.6121 194.8917 191.7007 176.5439 195.6894 178.9371 180.5326
```

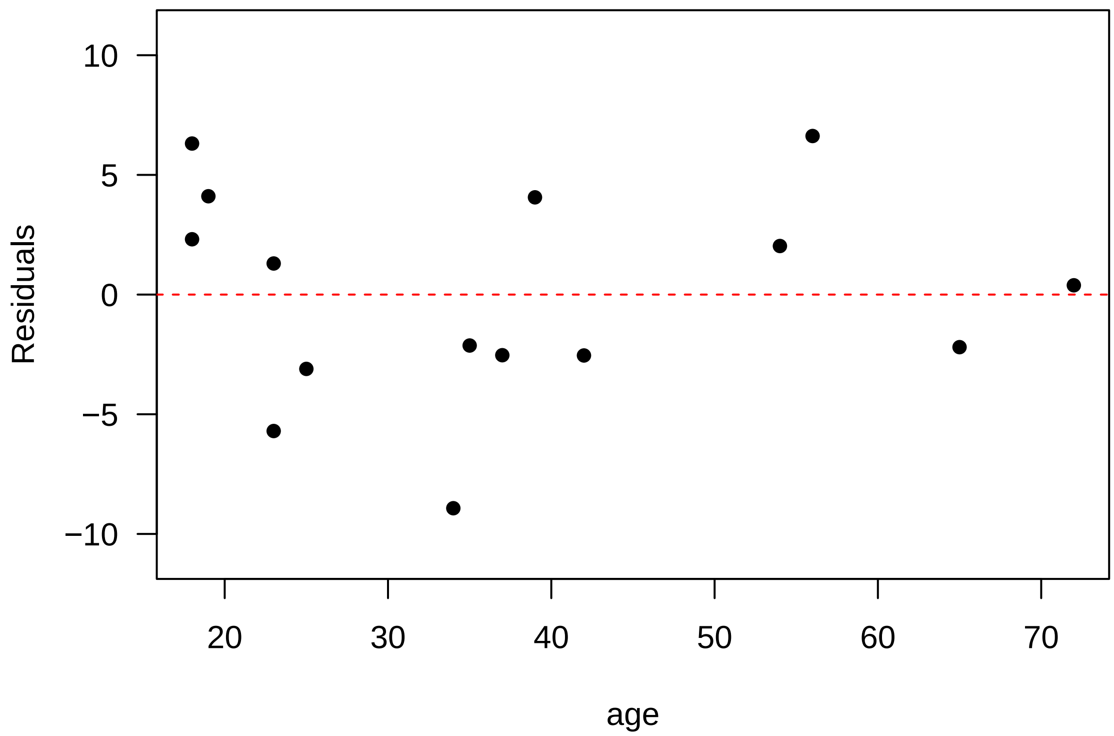
- $\hat{\sigma}$

```r
summary(fit)$sigma
```

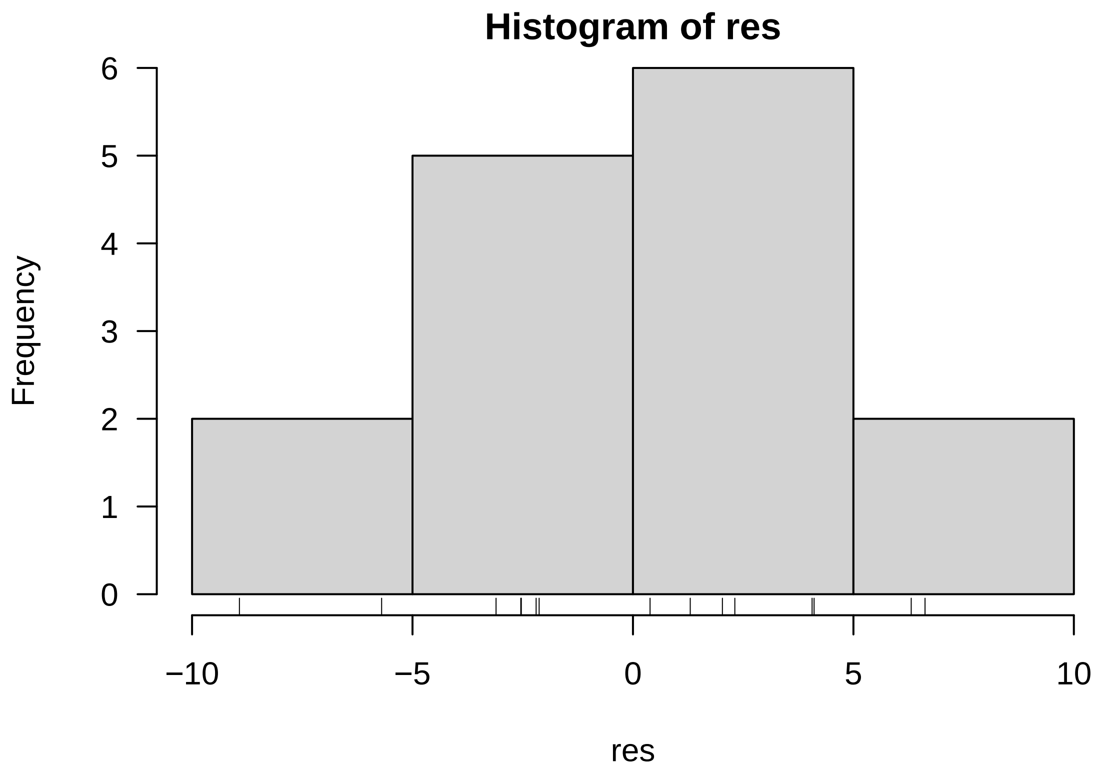
```
## [1] 4.577799
```

Residual Analysis

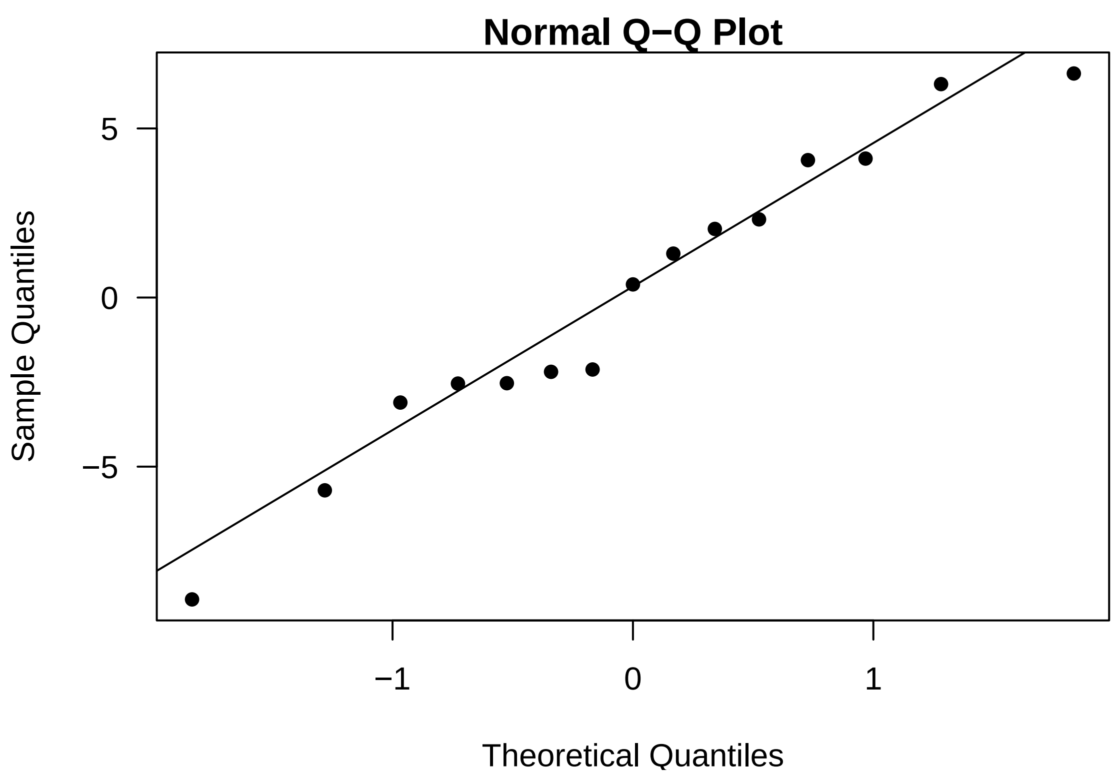
```
par(las = 1, mar = c(4.1, 4.1, 1.1, 1.1))
plot(age, fit$residuals, pch = 16, ylab = "Residuals", ylim = c(-11, 11))
abline(h = 0, col = "red", lty = 2)
```



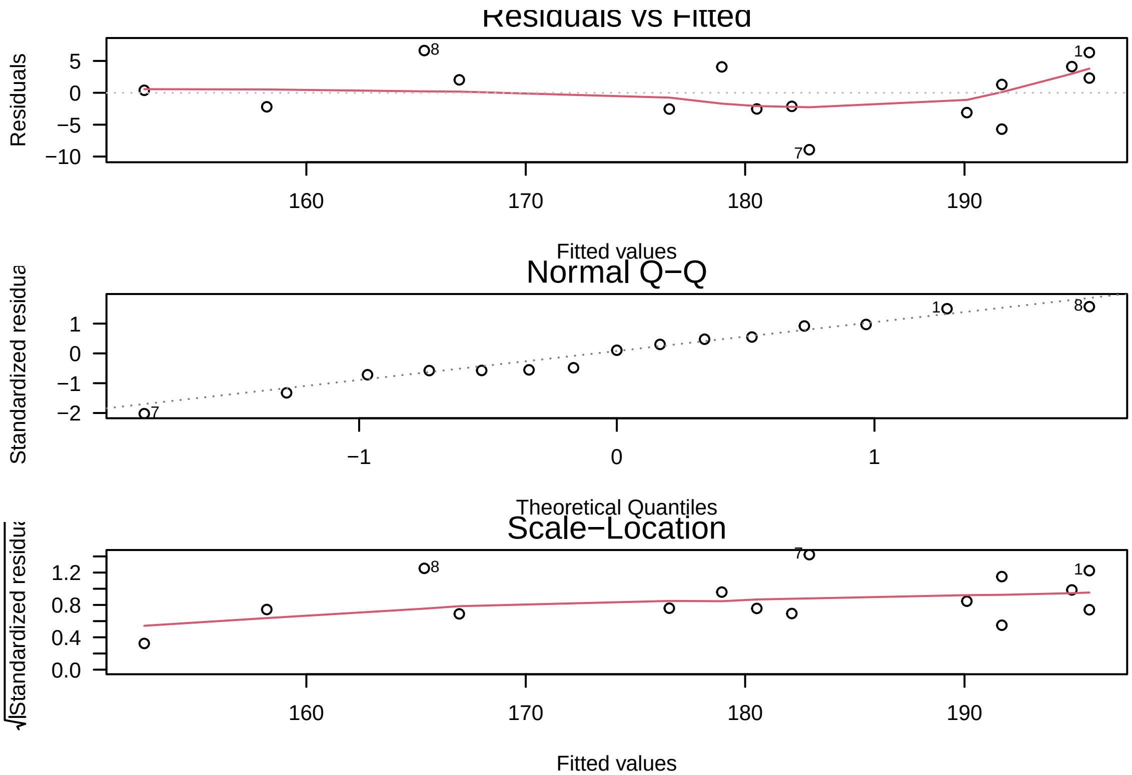

```
res <- fit$residuals
# histogram
hist(res, las = 1)
rug(res)
```




```r
# QQ plot
qqnorm(res, pch = 16, las = 1)
qqline(res)
```



```r
par(mfrow = c(3, 1))
plot(fit, which = 1:3)
```



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. Here we set $\beta_0 = 3$, $\beta_1 = 1.5$, $\sigma^2 = 1$, $n = 30$, $N = 100$.

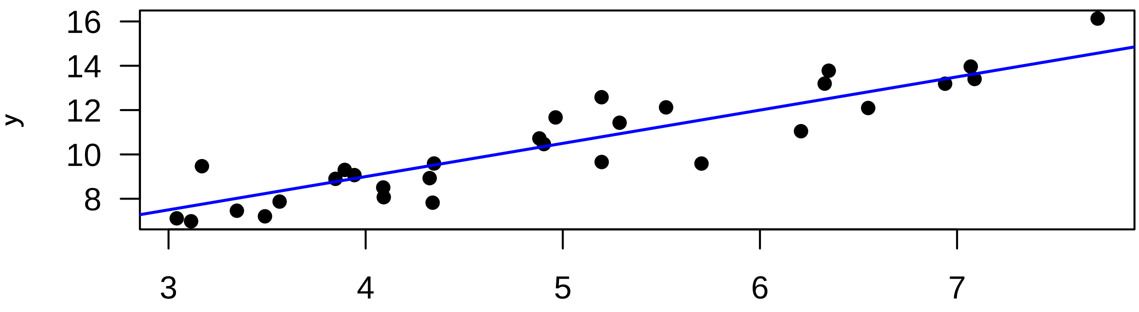
Generate data in R

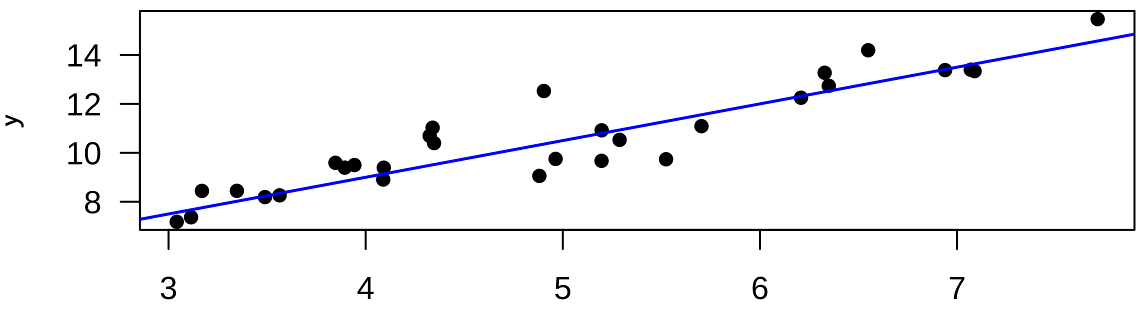
```r
set.seed(12)
n = 30; beta0 = 3; beta1 = 1.5; N = 100; sigma2 = 1
x <- 3 + 5 * runif(n)
set.seed(123)
y <- replicate(N, beta0 + beta1 * x + rnorm(n, mean = 0, sd = sqrt(sigma2)))
dim(y)
```

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

Plot the first few simulated datasets

```r
par(mfrow = c(2, 1), mar = c(3.5, 3.5, 0.8, 0.6))
for (i in 1:2){
  plot(x, y[, i], pch = 16, las = 1, ylab = "y")
  abline(3, 1.5, col = "blue", lwd = 1.5)
}
```



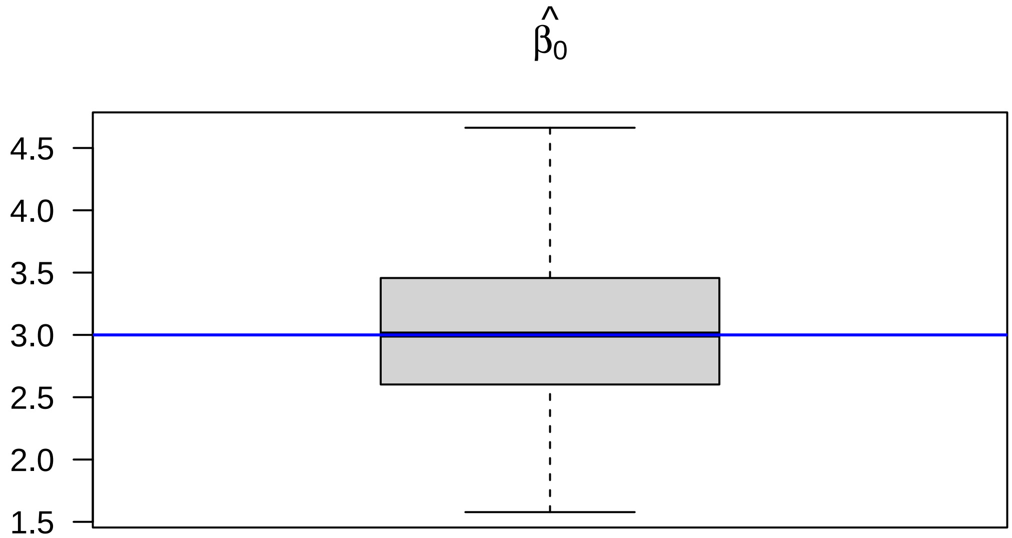


Estimate the β_0, β_1, and σ^2 for each simulated dataset

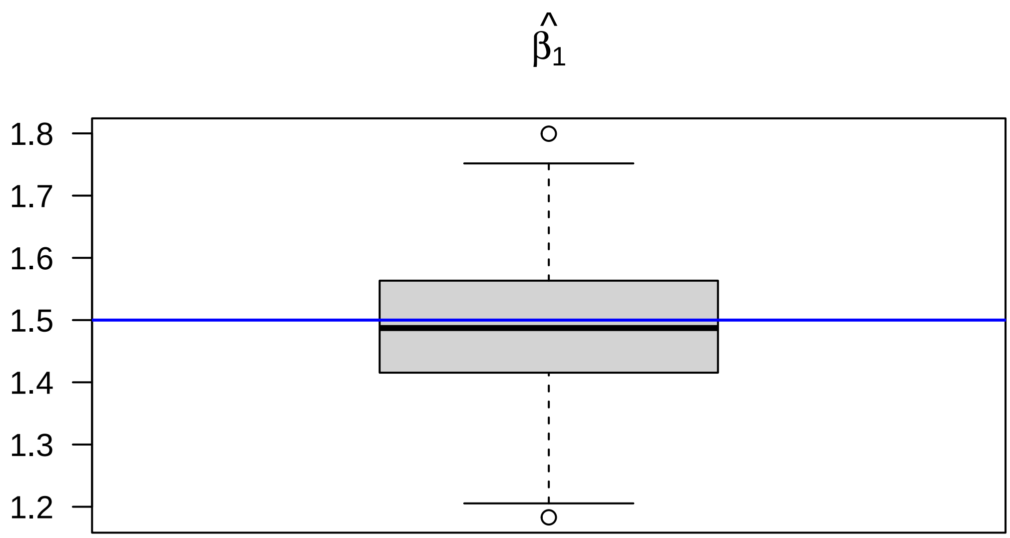
```r
beta0_hat <- beta1_hat <- sigma2_hat <- se_beta1 <- numeric(N)
for (i in 1:100){
  fit <- lm(lm(y[, i] ~ x))
  beta0_hat[i] <- summary(fit)[["coefficients"]][, 1][1]
  beta1_hat[i] <- summary(fit)[["coefficients"]][, 1][2]
  se_beta1[i] <- summary(fit)[["coefficients"]][, 2][2]
  sigma2_hat[i] 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

Assess the estimation perfromance

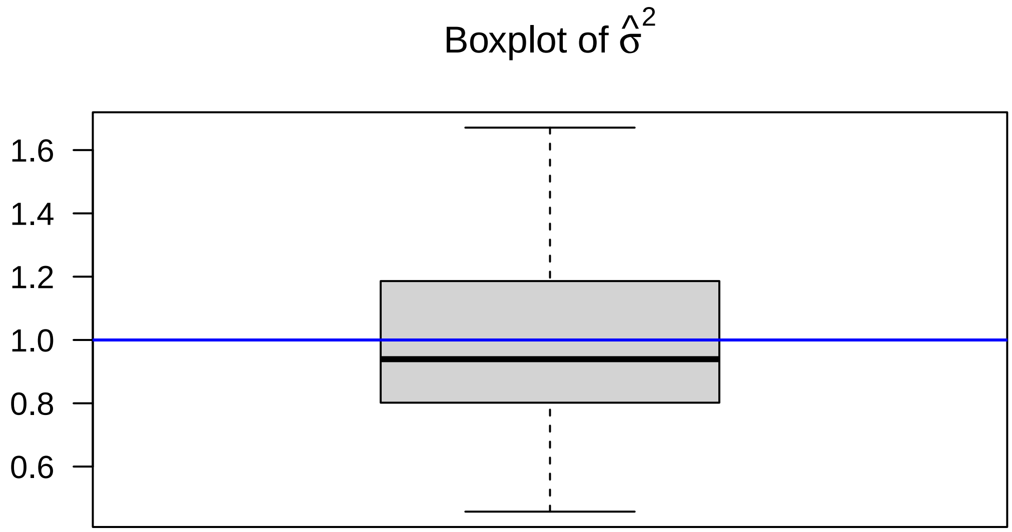
```r
boxplot(beta0_hat, las = 1, main = expression(hat(beta[0])))
abline(h = beta0, col = "blue", lwd = 1.5)
```




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

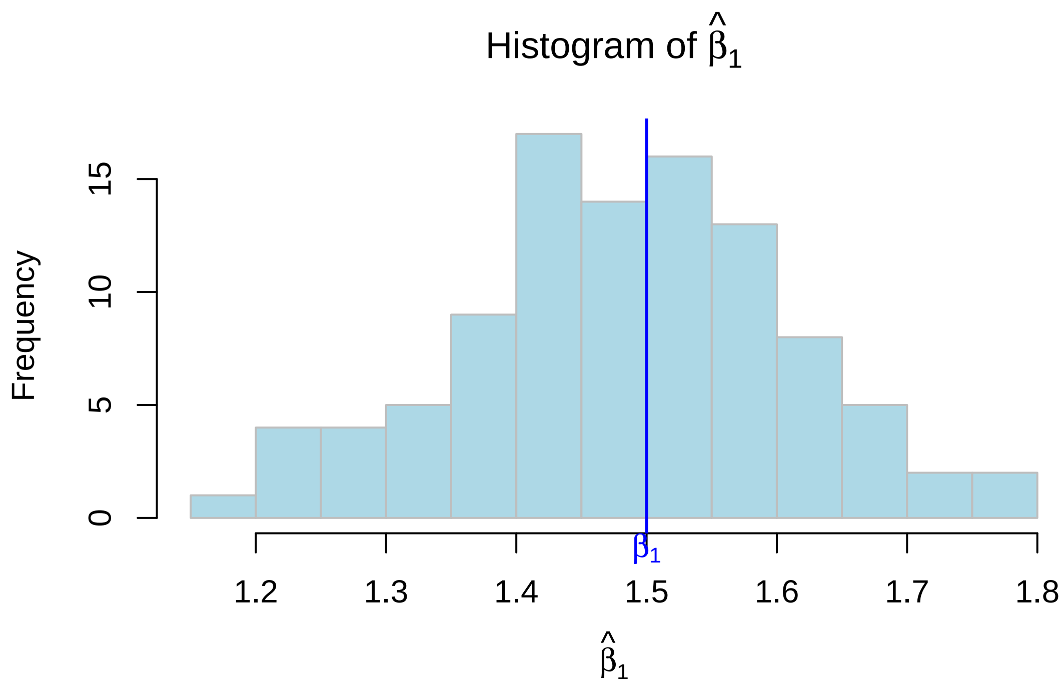



```
boxplot(sigma2_hat, las = 1, main = expression(paste("Boxplot of ", hat(sigma)^2)))
abline(h = sigma2, col = "blue", lwd = 1.5)
```



Sampling distribution

```
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")
```

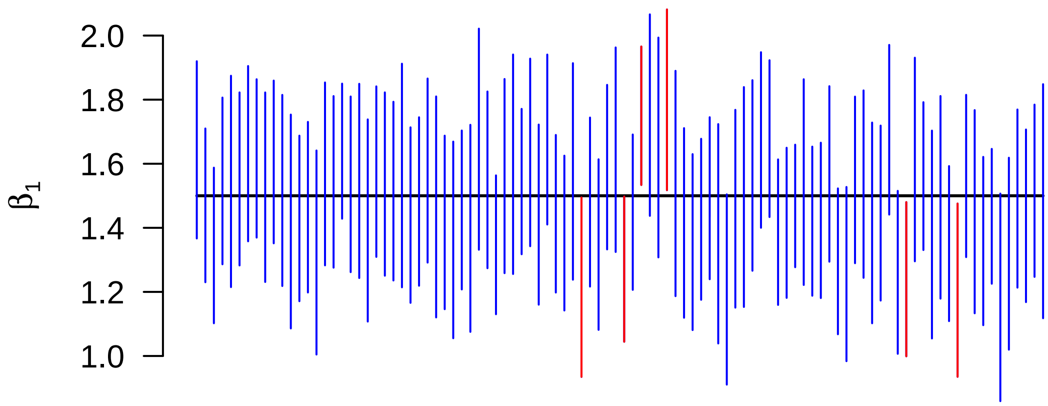


CI's for all the simulated datasets

```r
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")
}
```



Confidence Intervals for Maximum Heart Rate Example

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
```

β_1

```r
beta1_hat <- summary(fit)[["coefficients"]][, 1][2]
se_beta1 <- summary(fit)[["coefficients"]][, 2][2]
alpha = 0.05
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
```

```r
confint(fit)
```

```
##                   2.5 %      97.5 %
## (Intercept) 203.854813 216.2421034
## Age          -0.948872  -0.6465811
```

$Y_h | X_h = 40$

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

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

```r
predict(fit, Age_new, interval = "confidence", level = 0.9)
```

```
##        fit      lwr      upr
## 1 178.1394 176.0203 180.2585
```

```r
predict(fit, Age_new, interval = "predict", level = 0.9)
```

```
##        fit    lwr      upr
## 1 178.1394 169.76 186.5188
```

Check

```r
sd <- sqrt((sum(fit$residuals^2) / 13))
ME <- qt(1 - 0.1 / 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)
##    169.7600    186.5188
```

Hypothesis Tests for β_1

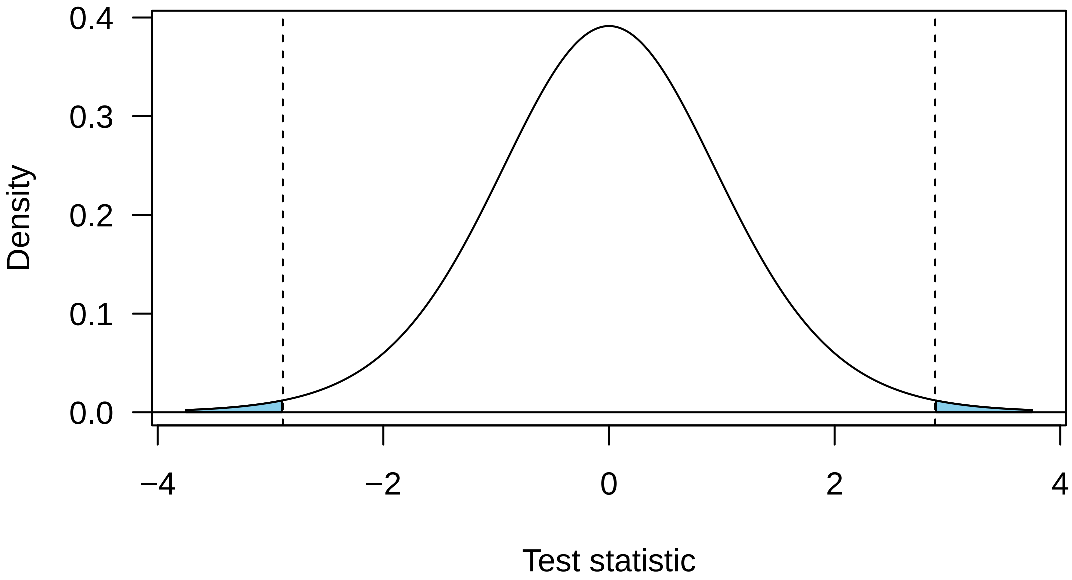
$H_0 : \beta_1 = -1$ vs. $H_a : \beta_1 \neq -1$ with $\alpha = 0.05$

```r
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
```

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

Fitting a simple linear regression

```
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
```

```
R.sq <- summary(fit)[["r.squared"]]
r <- cor(dat$Age, dat$MaxHeartRate)
r^2; R.sq
```

```
## [1] 0.9090967
```

```
## [1] 0.9090967
```

ANOVA Table

```
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
```