DSA 8020 R Session 1: Simple Linear Regression

Whitney Huang

Contents

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 following equation:

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

Let's use a dataset to assess the validity of this statement.

Load the dataset

There are several ways to load a dataset into R; for example, one could import the data over the Internet

```
dat <- read.csv('http://whitneyhuang83.github.io/STAT8010/Data/maxHeartRate.csv', header = T)
head(dat) #return the first part of the data object
```

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

Summarize the data before fitting models

```
y <- dat$MaxHeartRate; x <- dat$Age
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(x); var(y)
```

```
## [1] 305.8095
```

```
## [1] 214.0667
```

```
cov(x, y)
```

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

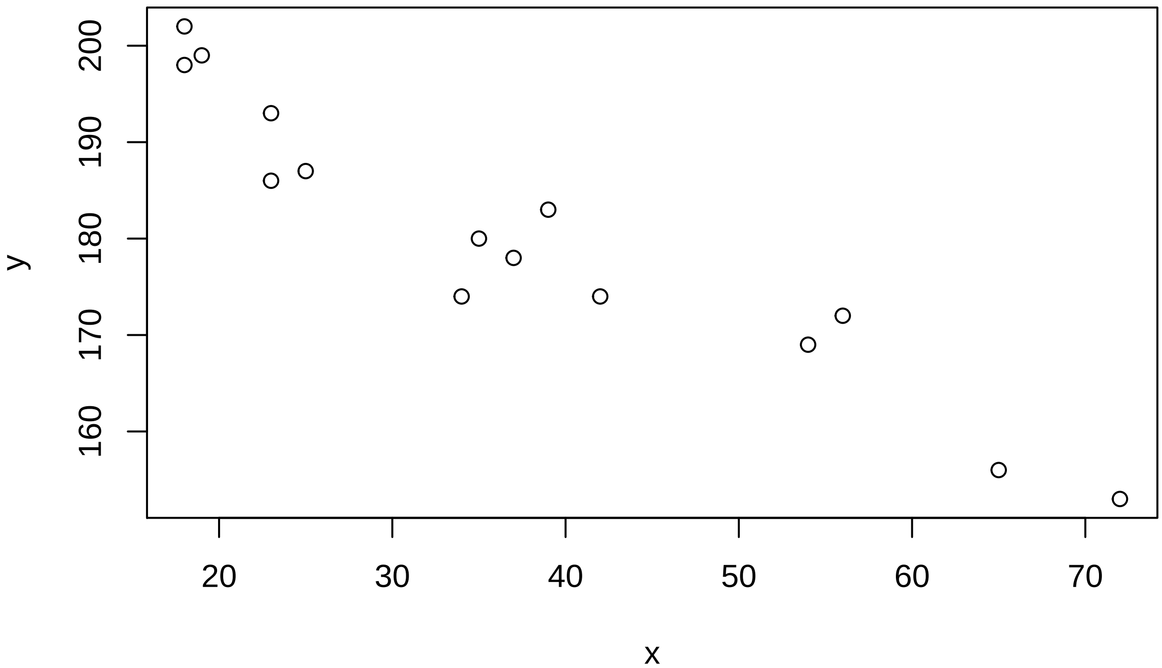
```
cor(x, y)
```

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

Plot the data before fitting models

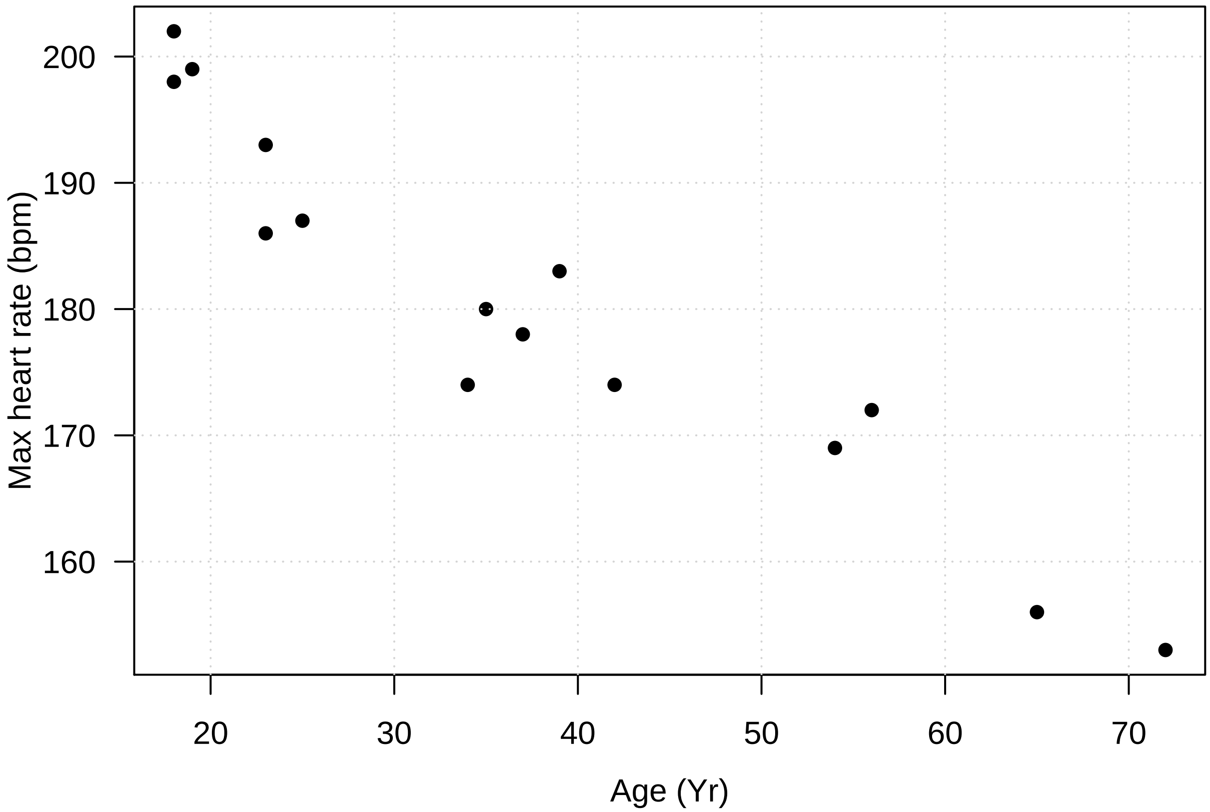
This is what the scatterplot would look like by default. Place the predictor (`age`) as the first argument and the response (`maxHeartRate`) as the second argument.

```r
plot(x, y)
```



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

```r
par(las = 1, mar = c(4.1, 4.1, 1, 0.5), mgp = c(2.5, 1, 0))
# Set Graphical Parameters
plot(x, y, pch = 16, xlab = "Age (Yr)", ylab = "Max heart rate (bpm)")
grid()
```



Simple Linear Regression

Estimation

Let's perform the calculations to determine the regression coefficients as well as the standard deviation of the random error.

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

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

```
beta_0 <- mean(y) - mean(x) * beta_1
beta_0
```

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

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

```
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
```

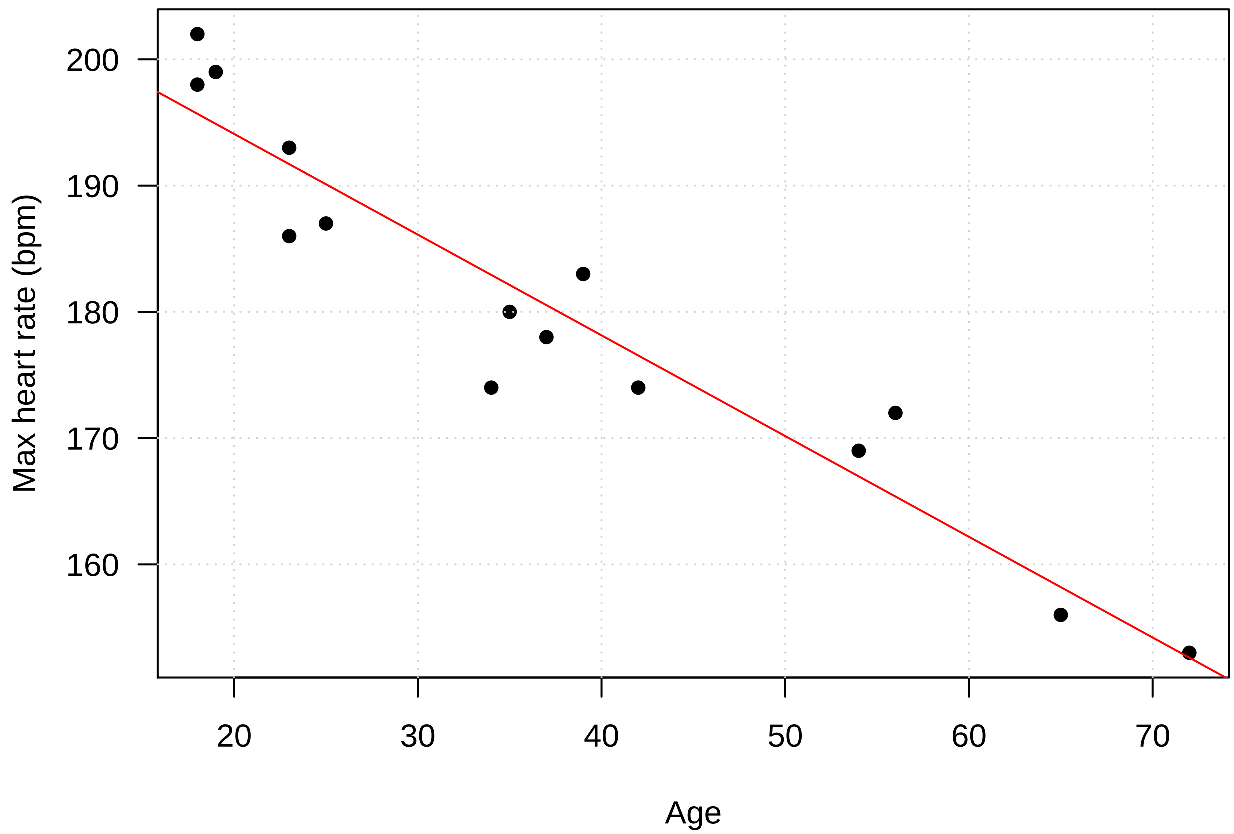
$\hat{\sigma}$: $\hat{\sigma}^2 = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-2}$

```
sigma2 <- sum((y - y_hat)^2) / (length(y) - 2)
sqrt(sigma2)
```

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

Add the fitted regression line to the scatterplot

```
par(las = 1, mar = c(4.1, 4.1, 1, 0.5))
plot(x, y, pch = 16, xlab = "Age", ylab = "Max heart rate (bpm)")
grid()
abline(a = beta_0, b = beta_1, col = "red")
```



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
```

Model Checking: Residual Analysis

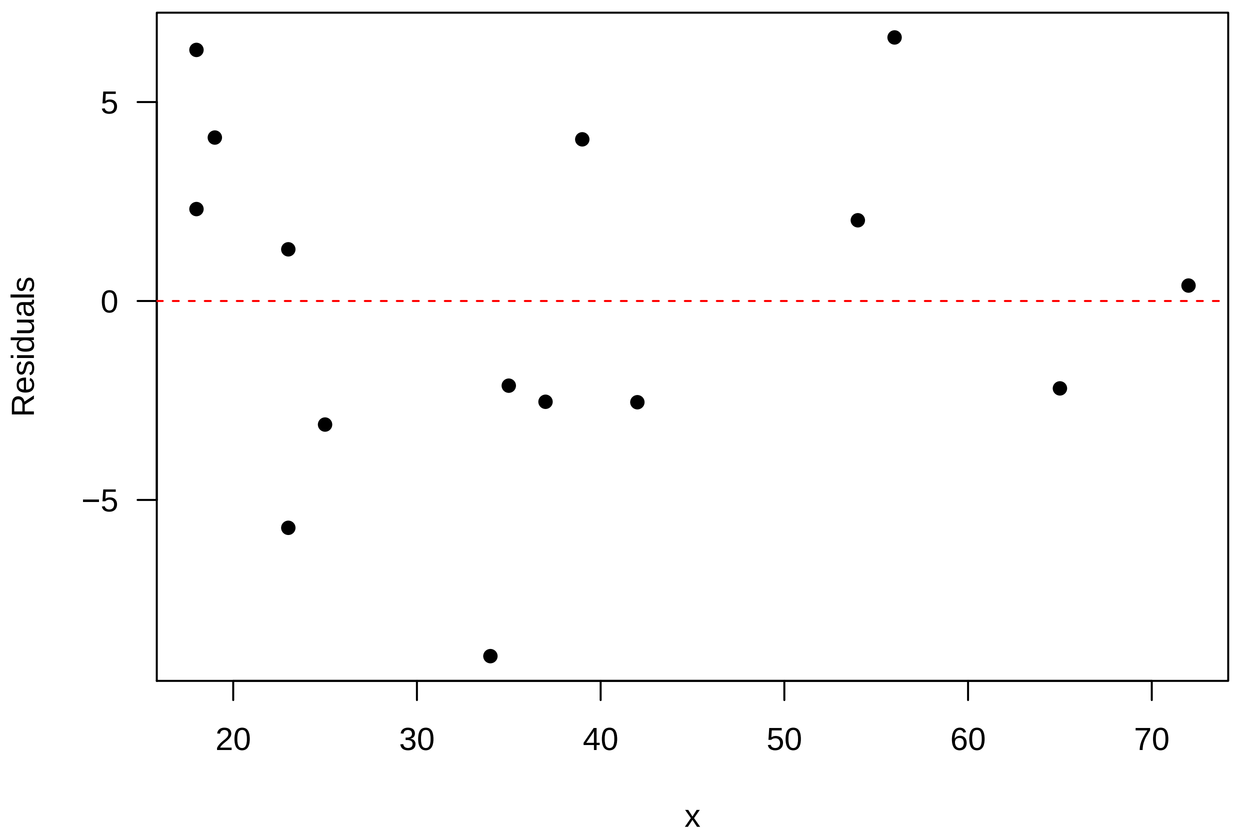
Assumptions on error ε:

- $\mathrm{E}[\varepsilon_i] = 0$
- $\mathrm{Var}[\varepsilon_i] = \sigma^2$
- $\varepsilon_i \overset{iid}{\sim} \mathrm{N}(0, \sigma^2)$

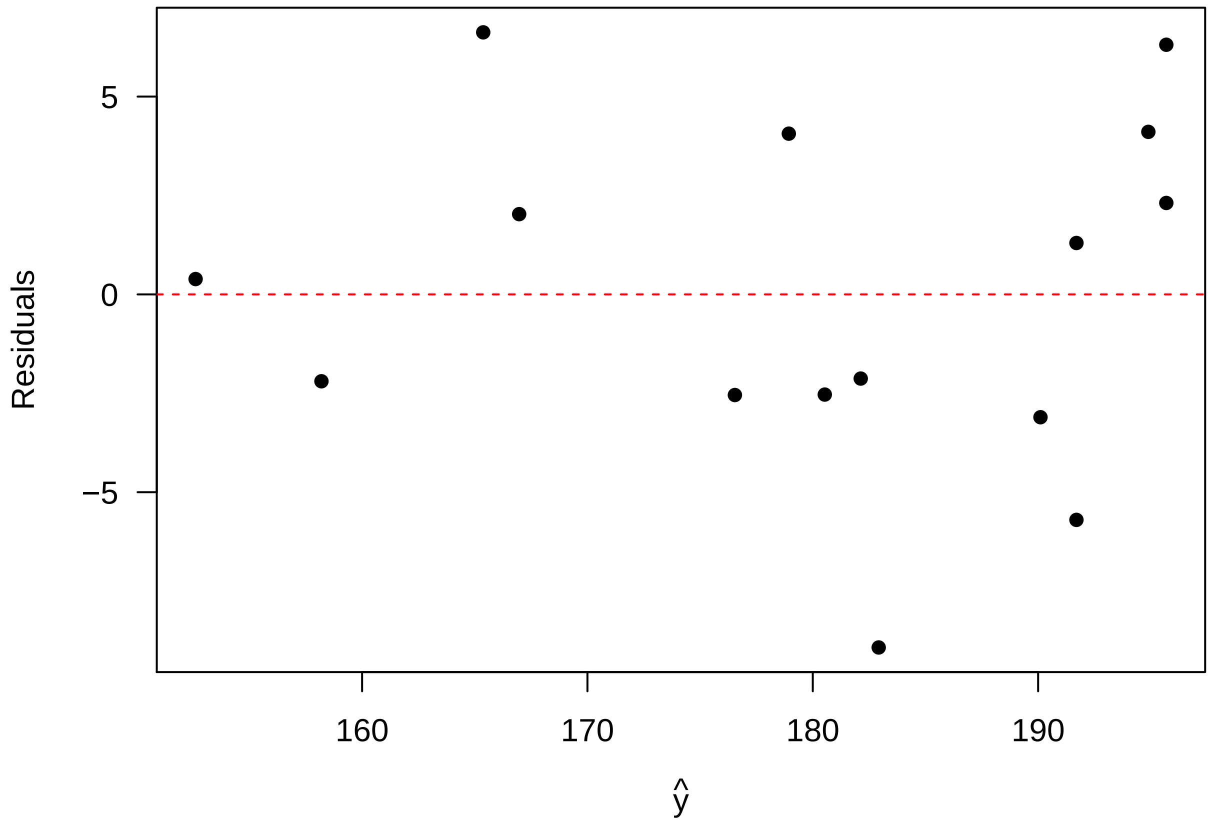
We use $e_i = y_i - \hat{y}_i$, where $i = 1, \cdots, n$ to assess these model assumptions.

Residual plots

```r
## res vs. x
par(las = 1, mar = c(4.1, 4.1, 1, 0.5))
plot(x, fit$residuals, pch = 16, ylab = "Residuals")
abline(h = 0, col = "red", lty = 2)
```

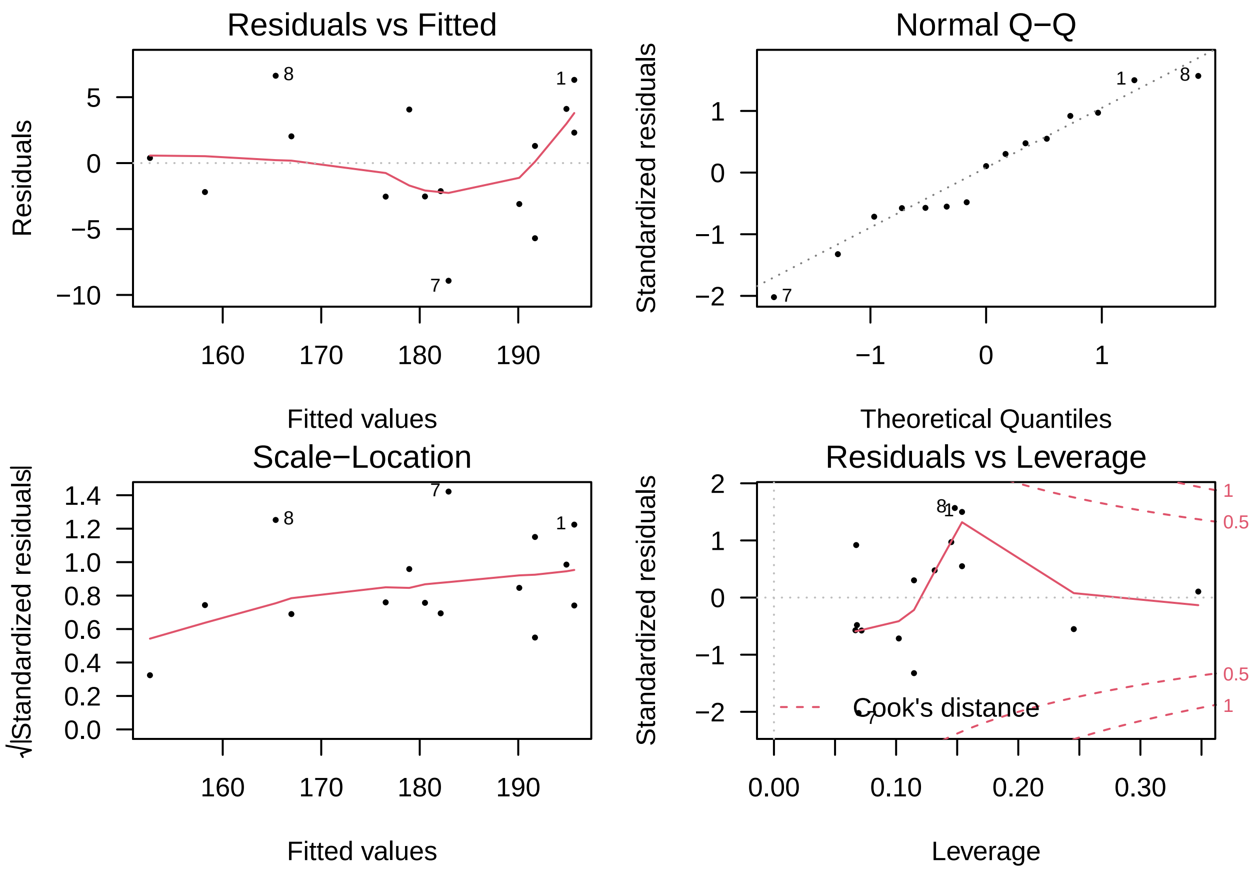



```r
## res vs. yhat
par(las = 1, mar = c(4.1, 4.1, 1.1, 1.1))
plot(fit$fitted.values, fit$residuals, pch = 16, ylab = "Residuals", xlab = expression(hat(y)))
abline(h = 0, col = "red", lty = 2)
```



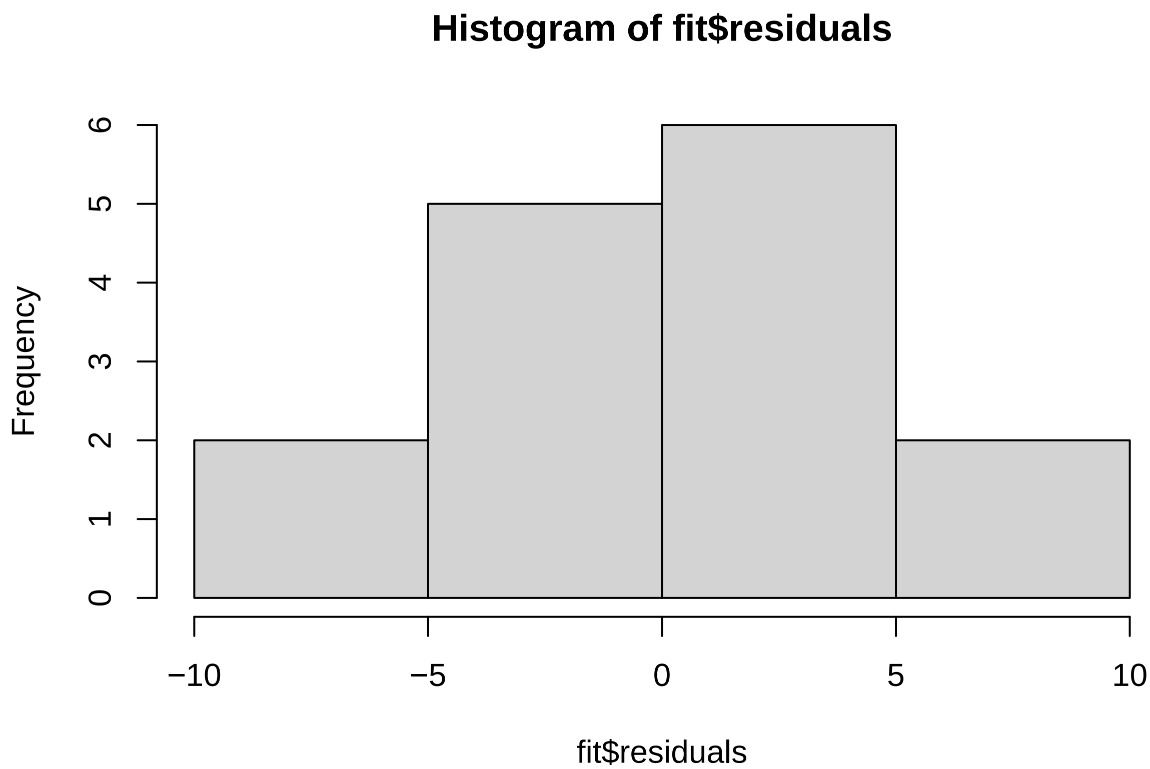
Plot Diagnostics for an `lm` Object

```
par(mfrow = c(2, 2), mar = c(4, 4, 1.5, 1.2), las = 1)
plot(fit, cex = 0.5, pch = 16)
```

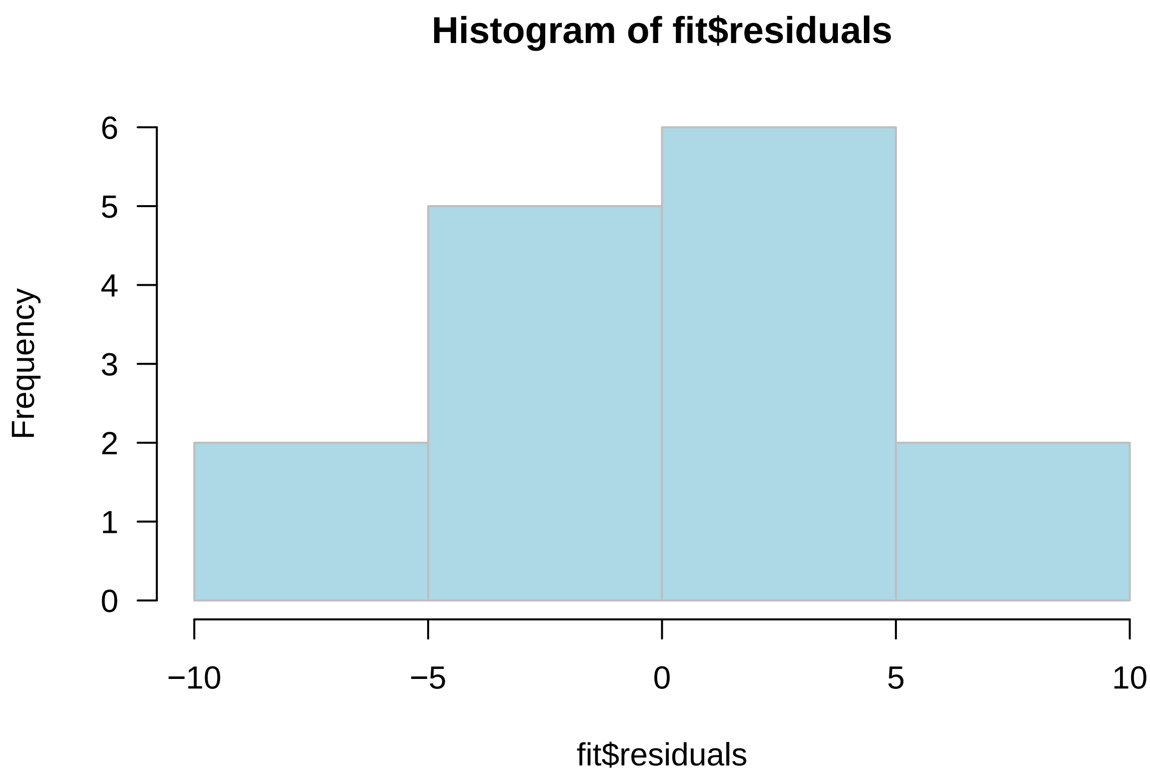


Assessing normality of random error

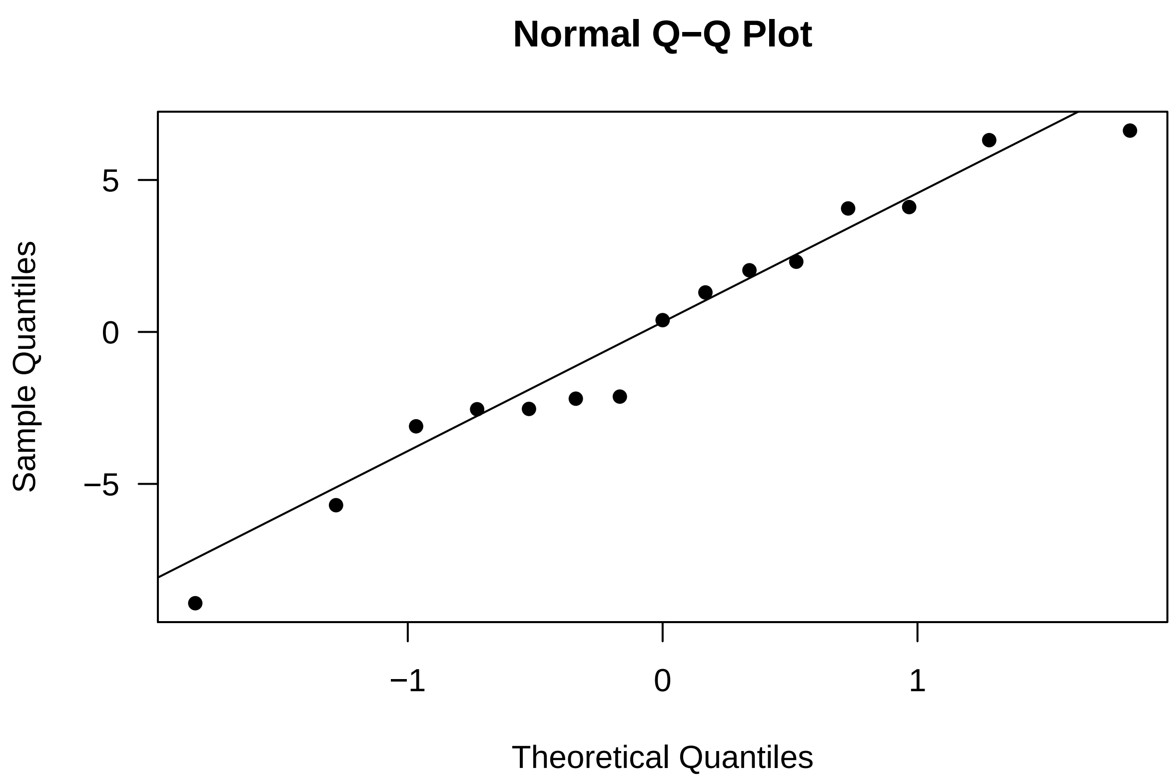
```
# histogram
hist(fit$residuals)
```




```
hist(fit$residuals, col = "lightblue", border = "gray", las = 1)
```



```r
# qqplot
qqnorm(fit$residuals, pch = 16, las = 1)
qqline(fit$residuals)
```



Statistical Inference

Confidence Intervals for β_0 and β_1

```r
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
```

```r
# use the `confint` built-in function in R to calculate confidence intervals
confint(fit)
```

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

Confidence and prediction intervals for $\mathrm{E}[Y_{new}|x_{new} = 40]$

```
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", level = 0.95)
```

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

```
predict(fit, Age_new, interval = "predict", level = 0.95)
```

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

Check

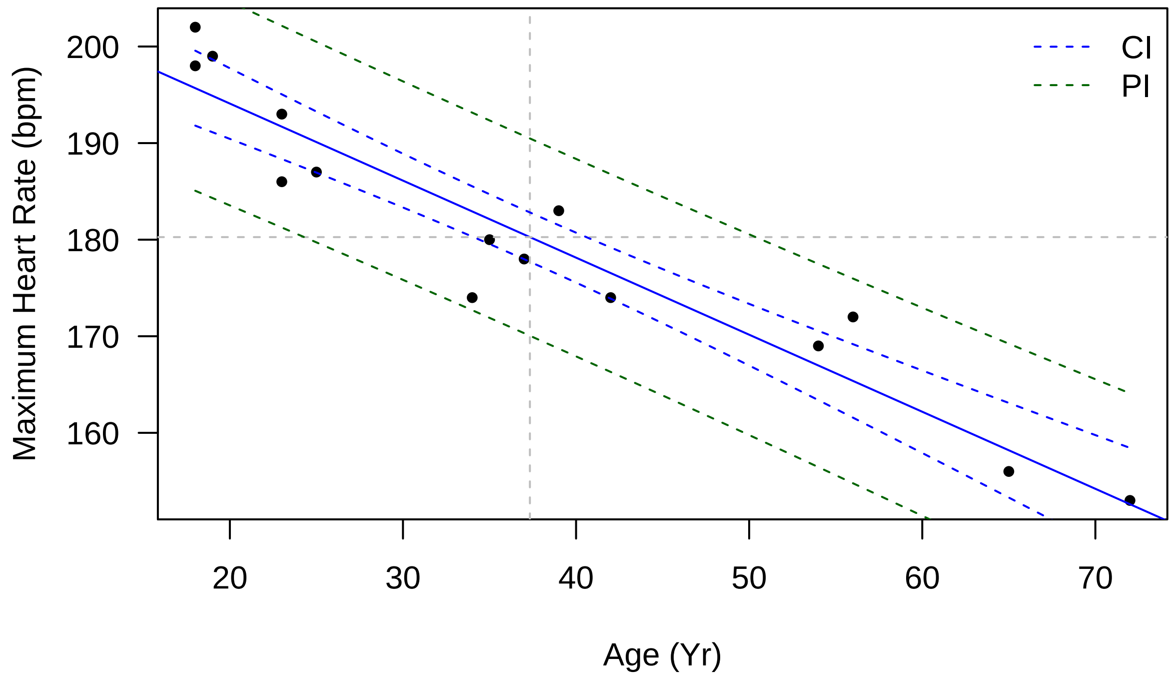
```
sd <- sqrt((sum(fit$residuals^2) / 13))
ME <- qt(1 - alpha / 2, 13) * sd * sqrt(1 + 1 / 15 + (40 - mean(x))^(2) / sum((x - mean(x))^2))
c(hat_Y - ME, hat_Y + ME)
```

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

Constrcuting pointwise CIs/PIs

```
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

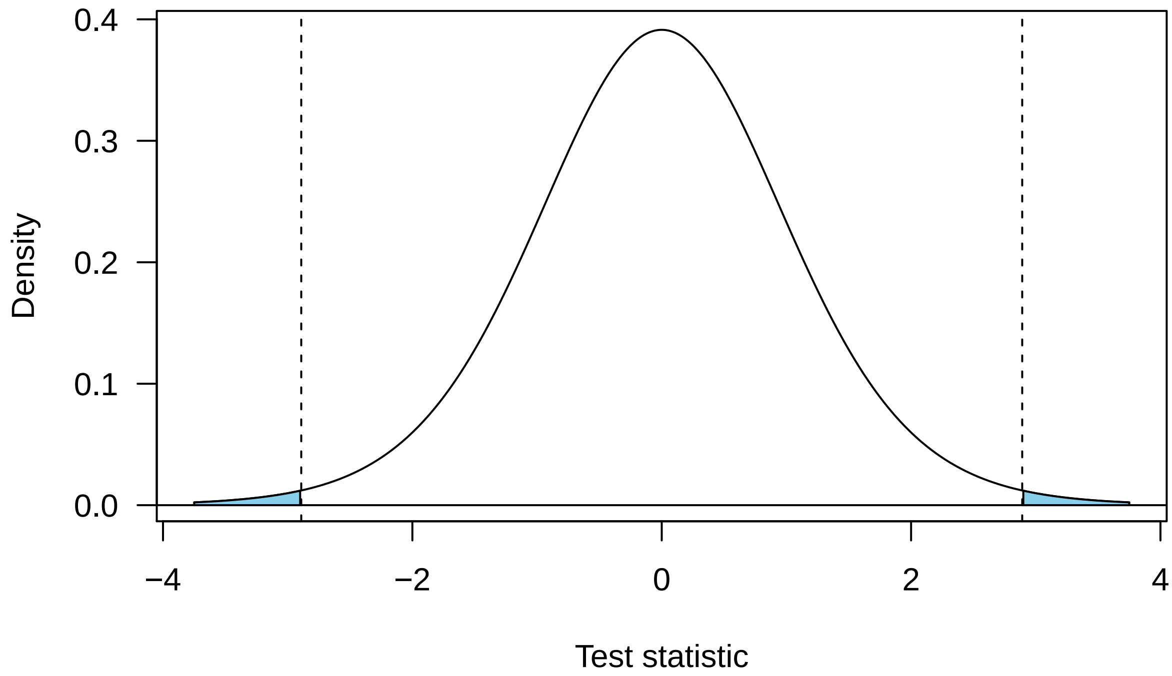
$H_0 : \beta_1 = -1$ vs. $H_a : \beta_1 \neq -1$ with $\alpha = 0.05$

```
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)
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)
```


0.4
0.3
0.2
0.1
0.0
Density
−4
−2
0
2
4
Test statistic