Linear Regression: Women

Karen Mazidi

This example looks at the built-in data set **women** as an introductory example of linear regression in R.

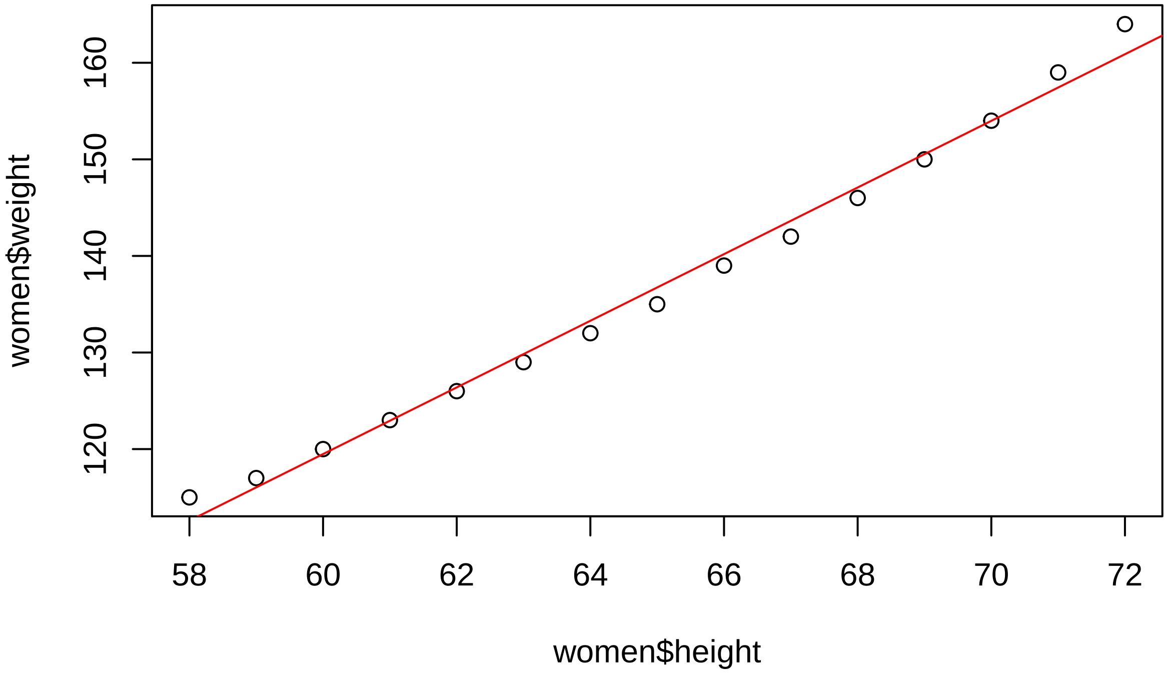
Data Exploration

First, load the data, look at its structure and plot the points.

```
data(women)
str(women)
```

```
## 'data.frame':    15 obs. of  2 variables:
##  $ height: num  58 59 60 61 62 63 64 65 66 67 ...
##  $ weight: num  115 117 120 123 126 129 132 135 139 142 ...
```

```
plot(women$weight~women$height)
abline(lm(women$weight~women$height), col="red")
```



Build a linear regression model

The first argument to the lm() function is the formula, in this case weight as a function of height. The second argument specifies what data to use.

When we type the model name, R returns basic information about the model.

```
lm1 <- lm(weight~height, data=women)
lm1
```

```
##
## Call:
## lm(formula = weight ~ height, data = women)
##
```

```
## Coefficients:
## (Intercept)       height
##      -87.52         3.45
```

The summary() function

Output more about the model with summary(). This gives statistics on the residuals, the coefficients, and the model itself.

```
summary(lm1)
```

```
##
## Call:
## lm(formula = weight ~ height, data = women)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7333 -1.1333 -0.3833  0.7417  3.1167
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -87.51667    5.93694  -14.74 1.71e-09 ***
## height        3.45000    0.09114   37.85 1.09e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.525 on 13 degrees of freedom
## Multiple R-squared:  0.991,  Adjusted R-squared:  0.9903
## F-statistic:  1433 on 1 and 13 DF,  p-value: 1.091e-14
```

Test and train

Normally we will divide a data set into at least two portions. The larger portion, usually more than 50% of the data, will be used for training the model. The smaller portion is a test set that will be used to evaluate how well the model does on data it has not seen before. In this case we are just getting to know linear regression on a tiny data set so we are just going to make up some test data.

Let's hallucinate some test data.

```
test <- women[c(5, 9, 11),]
test[1, 2] <- 135
test[2, 2] <- 118
test[3, 2] <- 156
test
```

```
##    height weight
## 5      62    135
## 9      66    118
## 11     68    156
```

The predict() function

Once we have a model of our data, lm1, we can use this model to predict target values for new data with the predict() function. The first argument to predict() is our model, the second specifies the new data. The output is a vector of predicted values. It's always a good idea to look at your output at each stage so that simple errors don't propagate forward. You can look at pred by typing "pred" at the console, or by looking in the Environment pane at the upper right of the RStudio screen. You should see that it is a vector with 3 values.

Now predict on our made-up test data.

```
pred <- predict(lm1, newdata=test)
```

Evaluate the results

We expect to get poor results since our test data was purposely chosen to be far from the regression line for illustration purposes. And our expectations are met.

```
correlation <- cor(pred, test$weight)
print(paste("correlation: ", correlation))
```

```
## [1] "correlation:  0.38404402702441"
```

```
mse <- mean((pred - test$weight)^2)
print(paste("mse: ", mse))
```

```
## [1] "mse:  215.284722222223"
```

```
rmse <- sqrt(mse)
print(paste("rmse: ", rmse))
```

```
## [1] "rmse:  14.6725840335717"
```

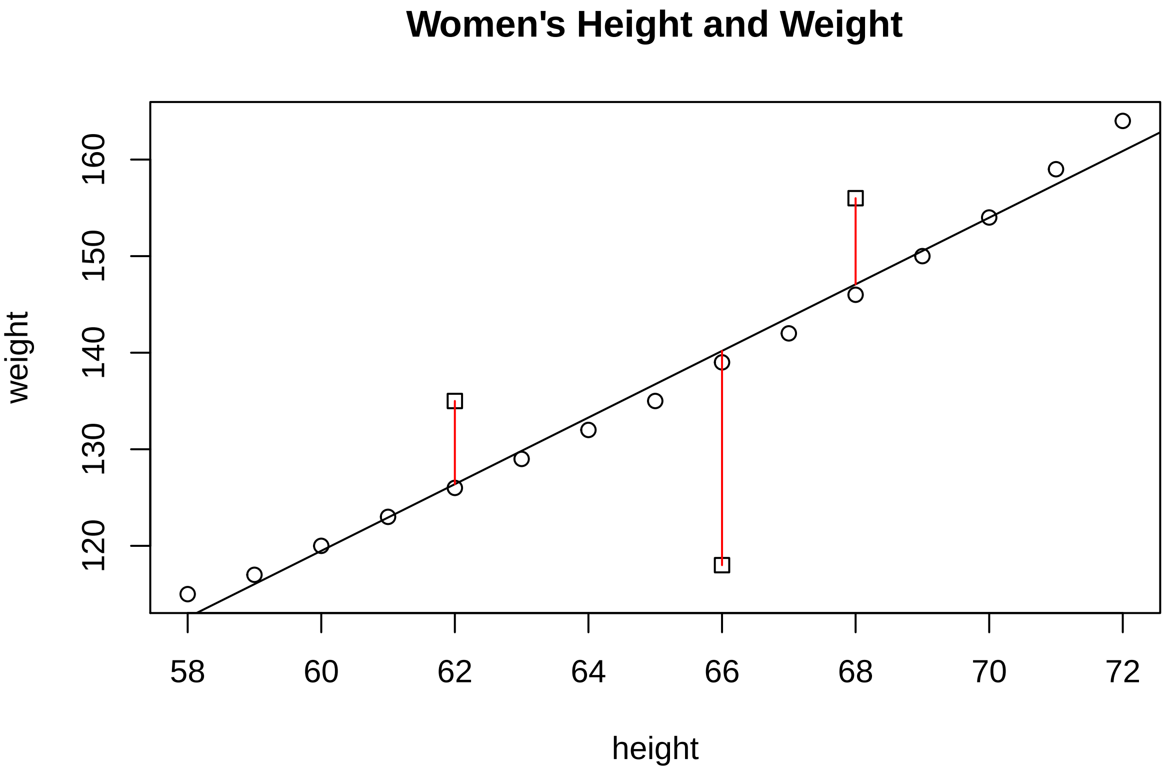
The correlation is not great, numbers closer to +/- 1 are better. The mse is hard to interpret in isolation, it is most helpful in comparing models. The rmse tells us that our test data was off by an average of almost 15 pounds.

Residuals

Residuals are errors in our predictions. They quantify how far off from the regression line (the predicted values) our actual values are. In the diagram the residuals are drawn with red lines.

Let's plot.

```
plot(women$height, women$weight, main="Women's Height and Weight",
     xlab="height", ylab="weight")
abline(lm1)
points(test$height, test$weight, pch=0)
segments(test$height, test$weight, test$height, pred, col="red")
```



Coefficient Estimates

Proving to ourselves that the coefficients match the equations in the text. Notice that these equations provide the same coefficients as lm1.

```r
x <- women$height
y <- women$weight
x_mean <- mean(women$height)
y_mean <- mean(women$weight)

w_hat <- sum((x-x_mean)*(y-y_mean)) / sum((x-x_mean)^2)
b_hat <- y_mean - w_hat * x_mean
print(paste("w and b estimates = ", w_hat, b_hat))
```

```
## [1] "w and b estimates =  3.45 -87.5166666666667"
```