Understanding R^2

Chapter 6, Lab 3: Solutions

OpenIntro Biostatistics

Topics

- R^2 with simulated data
- R^2 with the PREVEND data

The correlation coefficient r measures the strength of the linear relationship between two variables. However, it is more common to measure the strength of a linear fit using r^2, which is usually written as R^2 in the context of regression.

This lab first uses simulated data to explore the idea behind the quantity R^2, then provides an example of using R^2 to assess the strength of the linear fit of a regression model.

The material in this lab corresponds to Section 6.3.2 of *OpenIntro Biostatistics*.

Introduction

The quantity R^2 describes the amount of variation in the response variable that is explained by the least squares line:

$$R^2 = \frac{\text{variance of predicted } y\text{-values}}{\text{variance of observed } y\text{-values}} = \frac{\text{Var}(\hat{y}_i)}{\text{Var}(y_i)}$$

R^2 can also be calculated using the following formula:

$$R^2 = \frac{\text{variance of observed } y\text{-values} - \text{variance of residuals}}{\text{variance of observed } y\text{-values}} = \frac{\text{Var}(y_i) - \text{Var}(e_i)}{\text{Var}(y_i)}$$

R^2 with simulated data

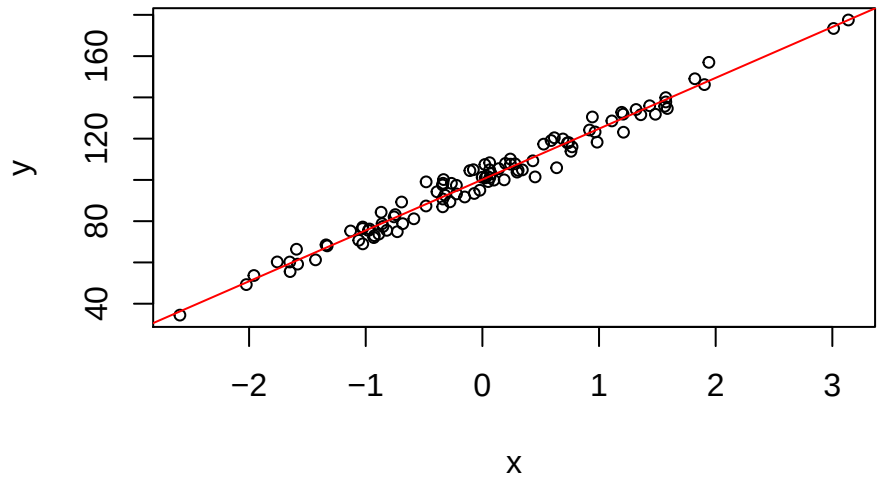
A simulation can be conducted in which y-values are sampled according to a population regression model $y = \beta_0 + \beta_1 x + \epsilon$, where the parameters β_0, β_1, and the standard deviation of ϵ are known. Recall that ϵ is a normally distributed error term with mean 0 and standard deviation σ.

1. Run the following code chunk to simulate 100 (x, y) values, where the values for x are 100 numbers randomly sampled from a standard normal distribution and the values for y are determined by the population model $y_i = 100 + 25x_i + \epsilon_i$, where $\epsilon_i \sim N(0, 5)$.

```
#set the seed
set.seed(2017)

#simulate values
x = rnorm(100)
error = rnorm(100, 0, 5)
y = 100 + 25*x + error

#create a scatterplot with the line of best fit
plot(y ~ x, cex = 0.75)
abline(lm(y ~ x), col = "red")
```




```
#print coefficients of the regression line
coef(lm(y ~ x))
```

```
## (Intercept)           x
##   100.17046    24.64955
```

   a) Create a scatterplot of y versus x and add the line of best fit to the plot.

      i. Does the line appear to be a good fit to the data?

Yes, the line appears to be a good fit to the data. The line passes through the center of the spread of points, without areas where it seems the line is either overestimating or underestimating the data.

ii. Why do the data points not fall exactly on a line, even though the data are simulated according to a known linear relationship between x and y?

The y-values are not completely determined by the linear relationship; the error term adds some noise to the y-value, where the noise is normally distributed with mean 0 and standard deviation 5. The points would only fall exactly on the line if the error term were not included (or equivalently, the error term had standard deviation of 0).

iii. How well does the regression line estimate the population parameters β_0 and β_1?

The regression line has coefficients $b_0 = 100.17$ and $b_1 = 24.65$, which are very close to the population parameters, $\beta_0 = 100$ and $\beta_1 = 25$.

b) From a visual inspection, does it seem that the R^2 for this linear fit is relatively high or relatively low?

From a visual inspection, the two variables look to have a very strong linear association; this suggests a relatively high R^2.

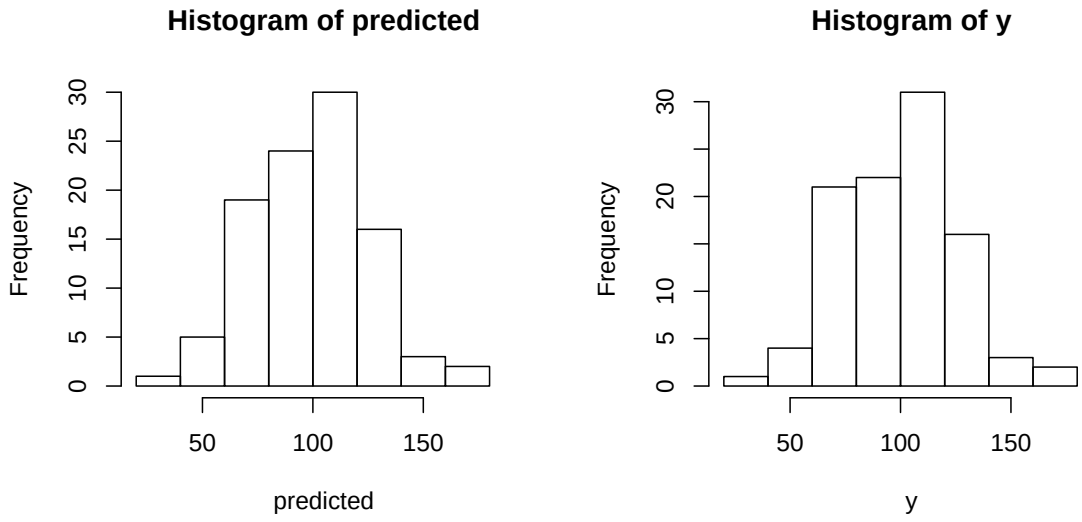
c) Run the code chunk shown in the template to create two histograms, one of the predicted y-values and one of the observed (i.e., simulated) y-values. Visually compare the variances of the two sets of values; do the predicted and observed y-values seem to have similar spread?

The predicted and observed y-values have similar spread; no one set of values looks more variable than the other.

```
par(mfrow = c(1, 2))

#histogram of predicted y-values
predicted = predict(lm(y ~ x))
hist(predicted, xlim = c(20, 190))    #bounds x-axis between 20 and 190

#histogram of observed y-values
hist(y, xlim = c(20, 190))
```



d) Run the code chunk shown in the template to calculate R^2 from the following formula. What is the R^2 for this model?

$$R^2 = \frac{\text{variance of predicted } y\text{-values}}{\text{variance of observed } y\text{-values}} = \frac{\text{Var}(\hat{y}_i)}{\text{Var}(y_i)}$$

The R^2 for this model is 0.975.

```
#store predicted y-values
predicted = predict(lm(y ~ x))

#observed y-values are the simulated y values
observed = y

#calculate R-squared
R.squared = var(predicted)/var(observed)
R.squared
```

```
## [1] 0.9753583
```

e) Calculate the R^2 for the model using the following formula. Confirm that the value is the same as from using the formula in part d).

$$R^2 = \frac{\text{variance of observed } y\text{-values} - \text{variance of residuals}}{\text{variance of observed } y\text{-values}} = \frac{\text{Var}(y_i) - \text{Var}(e_i)}{\text{Var}(y_i)}$$

Using this formula results in $R^2 = 0.975$, which is the same as the answer from part d).

```
#store residual values
residuals = resid(lm(y ~ x))

#calculate R-squared
```

```
R.squared = (var(observed) - var(residuals))/var(observed)
R.squared
```

f) To have R print the R^2 of a linear model, use the `summary(lm( ))` function as shown in the template. Confirm that this value matches the ones from the previous calculations.

This value matches the answers from parts d) and e).

```
summary(lm(y ~ x))$r.squared
```

```
## [1] 0.9753583
```

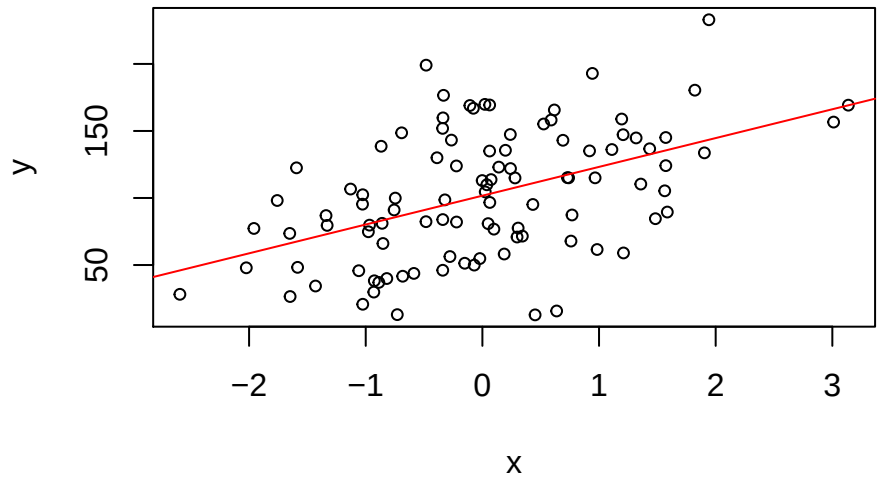
2. Simulate 100 new (x, y) values. Like before, the x values are 100 numbers randomly sampled from a standard normal distribution and the y values are determined by the population model $y_i = 100 + 25x_i + \epsilon_i$. For these data, however, the error term is distributed $N(0, 50)$.

```
#clear the workspace
rm(list = ls())

#set the seed
set.seed(2017)

x = rnorm(100)
error = rnorm(100, 0, 50)
y = 100 + 25*x + error

#create a scatterplot with the line of best fit
plot(y ~ x, cex = 0.75)
abline(lm(y ~ x), col = "red")
```



```r
#print coefficients of the regression line
coef(lm(y ~ x))
```

```
## (Intercept)           x
##   101.70460    21.49548
```

a) Create a scatterplot of y versus x and add the line of best fit to the plot. Does the line appear to be a good fit to the data? How well does the regression line estimate the population parameters β_0 and β_1?

Even though these data exhibit more spread, the line seems like a reasonable fit. A positive trend is visible in the upward slope of the cloud of points, with larger values of x generally corresponding to larger values of y. The line goes through the center of the cloud of points.

The regression line has coefficients $b_0 = 101.70$ and $b_1 = 21.50$, which are close estimates of $\beta_0 = 100$ and $\beta_1 = 25$, but not as close as in Problem 1 when the error term had a smaller σ.

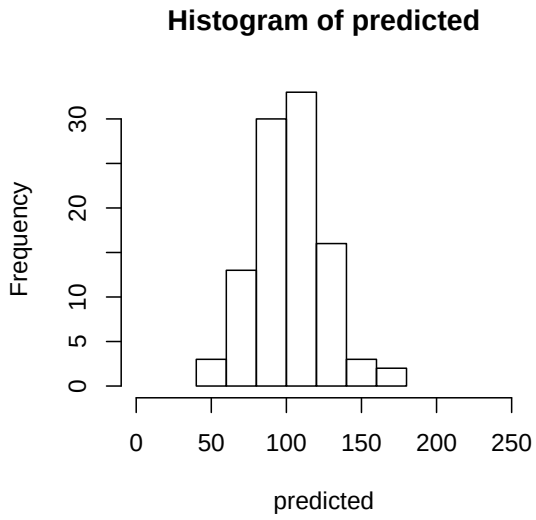
b) Run the code chunk shown in the template to create two histograms, one of the predicted y-values and one of the observed (i.e., simulated) y-values. Visually compare the variances of the two sets of values; do the predicted and observed y-values seem to have similar spread?

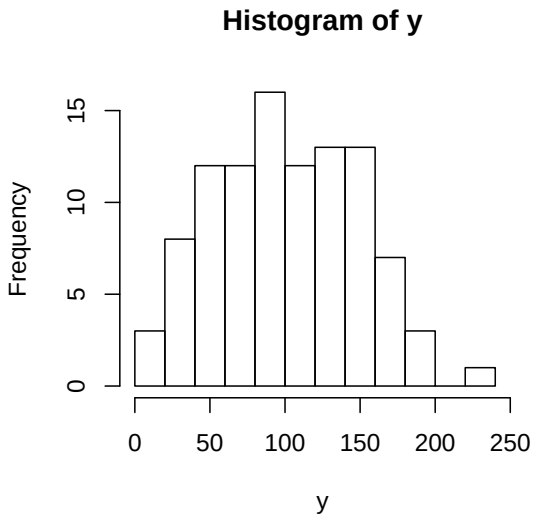
The predicted and observed y-values do not have similar spread. The predicted y-values are much less variable than the observed y-values.

```r
par(mfrow = c(1, 2))

#histogram of predicted y-values
predicted = predict(lm(y ~ x))
hist(predicted, xlim = c(0, 250))    #bounds x-axis between 0 and 250

#histogram of observed y-values
hist(y, xlim = c(0, 250))
```





c) Based on the answers to parts a) and b), does it seem that the R^2 for this linear model is relatively high or relatively low?

The R^2 for this linear model should be relatively low. The plot shows that the data do not closely follow the line of best fit; mathematically, since the numerator will be much smaller than the denominator, R^2 will not be close to 1.

d) Use any method to calculate R^2 for the linear model.

The R^2 for the linear model is 0.232.

```
#print the value from summary(lm( ))
summary(lm(y ~ x))$r.squared
```

```
## [1] 0.2313617
```

```
#use first formula
predicted = predict(lm(y ~ x))
var(predicted)/var(y)
```

```
## [1] 0.2313617
```

```
#use second formula
residuals = resid(lm(y ~ x))
(var(y) - var(residuals))/var(y)
```

```
## [1] 0.2313617
```

3. Run the code chunk shown in the template to simulate 100 new (x, y) values.

```
#clear the workspace
rm(list = ls())

#set the seed
set.seed(2017)
```

```r
#simulate values
x = rnorm(100)
error = rnorm(100, 0, 5)
y = 100 + 25*x + 5*x^2 + error

#calculate R-squared
summary(lm(y ~ x))$r.squared
```

```
## [1] 0.9050362
```

a) Fit a linear model predicting y from x to the data and calculate the R^2 for the model. Based on R^2, does the model seem to fit the data well?

The R^2 is relatively high, at 0.905. Based on R^2, the model seems to fit the data well.

b) Plot the data and add the line of best fit. Evaluate whether the linear model is a good fit to the data; how does viewing the data change the conclusion from part a)?

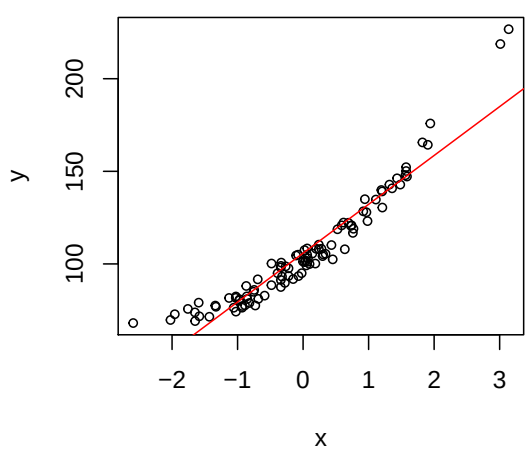
Plotting the data reveals that the linear model is not a good fit to the data. The data show curvature and cannot be modeled well with a straight line relationship; the best fit line systematically underpredicts data in the center of the range and overpredicts data at the extremes. The curvature is more obvious in a residual plot.

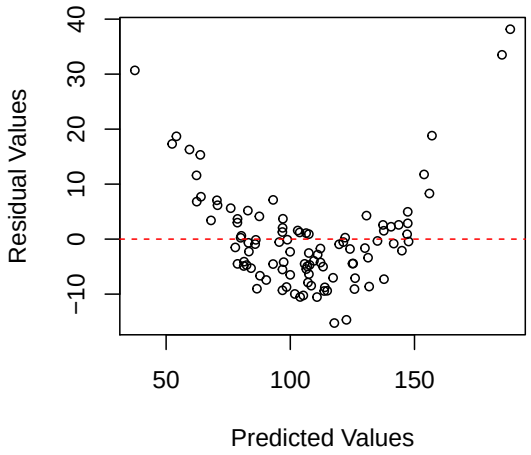
It is important to always visualize the data and not rely solely on a single metric like R^2 to determine whether a model has a good fit. As seen from this problem, only examining R^2 can be misleading; high R^2 values are not necessarily indicative of a good fit!

```r
par(mfrow = c(1, 2))

#create a scatterplot with the line of best fit
plot(y ~ x, cex = 0.75)
abline(lm(y ~ x), col = "red")

#create a residual plot
plot(resid(lm(y ~ x)) ~ fitted(lm(y ~ x)),
 xlab = "Predicted Values",
 ylab = "Residual Values",
 cex = 0.75)
abline(h = 0, col = "red", lty = 2)
```





FOR INTERESTED READERS: Polynomial regression is not covered in *Openintro Biostatistics* and is a topic for courses specializing in regression.

Linear regression is capable of handling nonlinear relationships between a response variable and a predictor variable. For example, in a polynomial regression model like $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \epsilon$, there is a quadratic term that models curvature in the data. Polynomial regression is still considered *linear* regression because the equation is linear with respect to the coefficients β_0, β_1, and β_2. The data in Problem 3 were generated according to a model with a quadratic term,

$$y_i = 100 + 25x_i + 5x_i^2 + \epsilon,$$

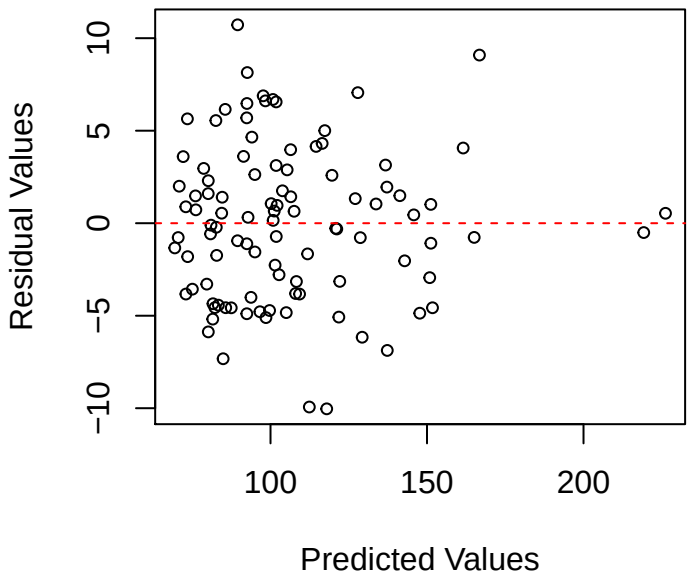
where $\epsilon \sim N(0, 5)$.

A regression model can be fit that specifies the presence of a quadratic term, using the syntax `I(x^2)`. The result is a linear model with coefficients $b_0 = 100.25$, $b_1 = 24.67$, and $b_2 = 4.93$, which are very close to the population model used to specify the data. The residual plot demonstrates that once the quadratic term is specified, the data satisfy the linearity assumption. Additionally, R^2 is now 0.98, indicating a better model fit. Note that since there is now more than one slope coefficient in the model, this is a *multiple regression* model and the data cannot be directly visualized with a scatterplot of y versus x.

```
#print coefficients of model with quadratic term
coef(lm(y ~ x + I(x^2)))
```

```
## (Intercept)           x      I(x^2)
##  100.247983   24.673313    4.931232
```

```
#create residual plot
plot(resid(lm(y ~ x + I(x^2))) ~ fitted(lm(y ~ x + I(x^2))),
 cex = 0.75,
 ylab = "Residual Values",
 xlab = "Predicted Values")
abline(h = 0, col = "red", lty = 2)
```




```
#calculate R-squared
summary(lm(y ~ x + I(x^2)))$r.squared
```

```
## [1] 0.980047
```

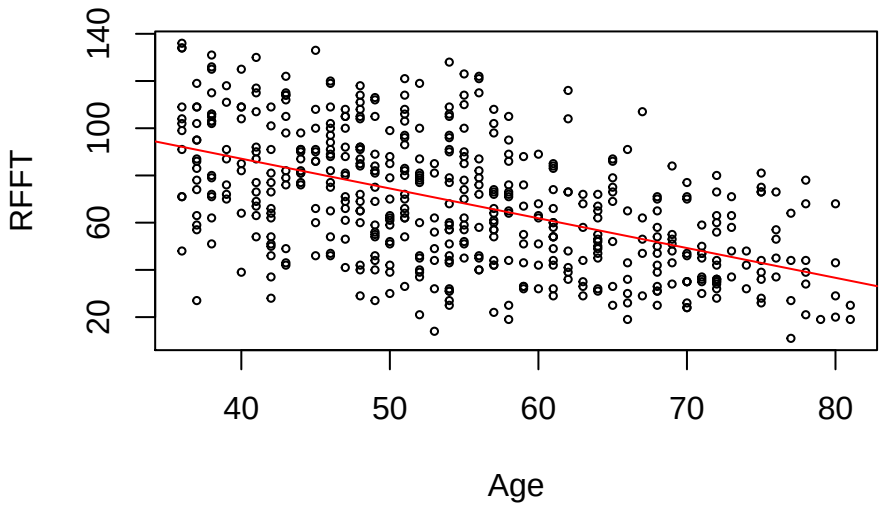
R^2 with the PREVEND data

4. Run the code chunk shown in the template to create `prevend.samp`, the random sample of 500 individuals from the PREVEND data that was used in the previous labs in this chapter.

```
#load the data
library(oibiostat)
data("prevend")

#take a random sample
set.seed(5011)
sample.size = 500
prevend.sample = prevend[sample(1:nrow(prevend), sample.size, replace = FALSE), ]

#plot RFFT scores versus age
plot(RFFT ~ Age, data = prevend.sample,
 cex = 0.5)
abline(lm(RFFT ~ Age, data = prevend.sample), col = "red")
```




```
#calculate R-squared
summary(lm(RFFT ~ Age, data = prevend.sample))$r.squared
```

```
## [1] 0.2850083
```

a) Plot RFFT scores versus age and confirm that these data seem reasonably linear.

The data follow a linear trend; on average, higher age is associated with lower RFFT score.

b) What proportion of the variability in the observed RFFT scores is explained by the linear model predicting average RFFT score from age?

0.285 of the variability in the observed RFFT scores is explained by the linear model predicting RFFT scores from age.

c) Evaluate the strength of the linear relationship between RFFT score and age. Does it seem like there might be other factors that explain the variability in RFFT score?

About 29% of the variance in observed RFFT scores are explained by the linear model with age as a predictor. Although there appears to be a linear relationship between RFFT score and age, the somewhat low R^2 value suggests that age may not be the only variable associated with RFFT score—this point will be explored in Chapter 7.

Note: There is no obvious cutoff that defines a "high" R^2 value. It is important to consider that R^2 is only one way to assess how well a linear model fits a particular set of observations. A higher R^2 value generally indicates that the data follow the linear model well enough such that the predicted y-values are very close to the observed y-values.