

5.1: Data Splitting for Training and Testing

In Chapter 4 we used all of the data available in the `int00.dat` data frame to select the appropriate predictors to include in the final regression model. Because we computed the model to fit this particular data set, we cannot now use this same data set to test the model's predictive capabilities. That would be like copying exam answers from the answer key and then using that same answer key to grade your exam. Of course you would get a perfect result. Instead, we must use one set of data to *train* the model and another set of data to *test* it.

The difficulty with this train-test process is that we need separate but similar data sets. A standard way to find these two different data sets is to split the available data into two parts. We take a random portion of all the available data and call it our *training set*. We then use this portion of the data in the `lm()` function to compute the specific values of the model's coefficients. We use the remaining portion of the data as our *testing set* to see how well the model predicts the results, compared to this test data.

The following sequence of operations splits the `int00.dat` data set into the training and testing sets:

```
rows <- nrow(int00.dat)
f <- 0.5
upper_bound <- floor(f * rows)
permuted_int00.dat <- int00.dat[sample(rows), ]
train.dat <- permuted_int00.dat[1:upper_bound, ]
test.dat <- permuted_int00.dat[(upper_bound+1):rows, ]
```

The first line assigns the total number of rows in the `int00.dat` data frame to the variable `rows`. The next line assigns to the variable `f` the fraction of the entire data set we wish to use for the training set. In this case, we somewhat arbitrarily decide to use half of the data as the training set and the other half as the testing set. The `floor()` function rounds its argument value down to the nearest integer. So the line `upper_bound <- floor(f * rows)` assigns the middle row's index number to the variable `upper_bound`.

The interesting action happens in the next line. The `sample()` function returns a permutation of the integers between 1 and `n` when we give it the integer value `n` as its input argument. In this code, the expression `sample(rows)` returns a vector that is a permutation of the integers between 1 and `rows`, where `rows` is the total number of rows in the `int00.dat` data frame. Using this vector as the row index for this data frame gives a random permutation of all of the rows in the data frame, which we assign to the new data frame, `permuted_int00.dat`. The next two lines assign the lower portion of this new data frame to the training data set and the top portion to the testing data set, respectively. This randomization process ensures that we obtain a new random selection of the rows in the train-and-test data sets every time we execute this sequence of operations.

This page titled [5.1: Data Splitting for Training and Testing](#) is shared under a [CC BY-NC 4.0](#) license and was authored, remixed, and/or curated by [David Lilja \(University of Minnesota Libraries Publishing\)](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.