TOOL

Making Predictions With Multiple Linear Regression

When you analyze data from the real world, you'll often encounter situations in which a response variable is associated with more than one predictor variable. Each of these predictor variables provides different information that can improve your prediction rule. In this situation, including only one predictor variable in your prediction rule leaves useful information from the other predictor variables out of your analysis. This tool outlines how to use more than one predictor variable in a prediction rule simultaneously, which can lead to a better prediction. Then, this tool demonstrates these steps in R with an example based on the `Prestige` data set you examined in the course.

Multiple Linear Regression

Multiple linear regression allows you to combine the information from multiple predictor variables into one prediction rule. This rule takes the form

$$Y = a + b_1X_1 + b_2X_2,$$

where a is the intercept and represents the predicted value of Y when both response variables, X_1 and X_2, are zero. There are two slopes (regression coefficients), b_1 and b_2. b_1 means the average increase in Y when X_1 increases by one unit but X_2 remains constant. b_2 means the average increase in Y when X_2 increases by one unit, but X_1 remains constant. Once you have calculated the intercept and both slopes, you can use both predictor variables to make predictions about the response variable.

Building a Linear Prediction Rule With Multiple Linear Regression

Follow these steps to build a prediction rule with multiple linear regression:

1. Create a three-dimensional scatterplot.
2. Fit a multiple linear regression plane through your data set.
3. Plot the plane that represents the linear regression plane on the three-dimensional scatterplot.
4. Retrieve the intercept and slope values of the regression line, and use them to write down your linear prediction rule in the format:

$$predictedY = intercept + slope_1X_1 + slope_2X_2$$

5. Predict outcomes based on the intercept and slopes you calculated.
6. Calculate the Mean Squared Error (MSE) and R^2.

Using R With This Tool

The portions of this tool with a grey background are code text you can use to do the examples included in this tool. You can also modify them to use with your own data. In these examples,

- Commands are the lines of code that don"t begin with a pound sign (#). Type these lines into R to carry out the command.
- Commented text begins with one pound sign and explains the lines of code.
- The example code output begins with two pound signs.

Data Set Information

The **`Prestige`** data set contains information about different job types in Canada in 1971, and is part of the **`carData`** package in R. In this data set, **`education`** refers to the average number of years of education the employees that hold that job have had, and **`prestige`** refers to the Pineo-Porter prestige value which measures the pride an employee has in their job.

You can load and view the data set with the following code:

```
install.packages("carData")                  # Install the carData package
library(carData)                             # Load the carData package
data(Prestige)                               # Load the Prestige data set
Prestige = Prestige[!is.na(Prestige$type),]  # Remove rows that do not
                               # contain information on the profession type
head(Prestige)                 # View the first 6 rows of the data set

##                      education income women prestige census type
## gov.administrators       13.11  12351 11.16     68.8   1113 prof
## general.managers         12.26  25879  4.02     69.1   1130 prof
## accountants              12.77   9271 15.70     63.4   1171 prof
## purchasing.officers      11.42   8865  9.11     56.8   1175 prof
## chemists                 14.62   8403 11.68     73.5   2111 prof
## physicists               15.64  11030  5.13     77.6   2113 prof
```

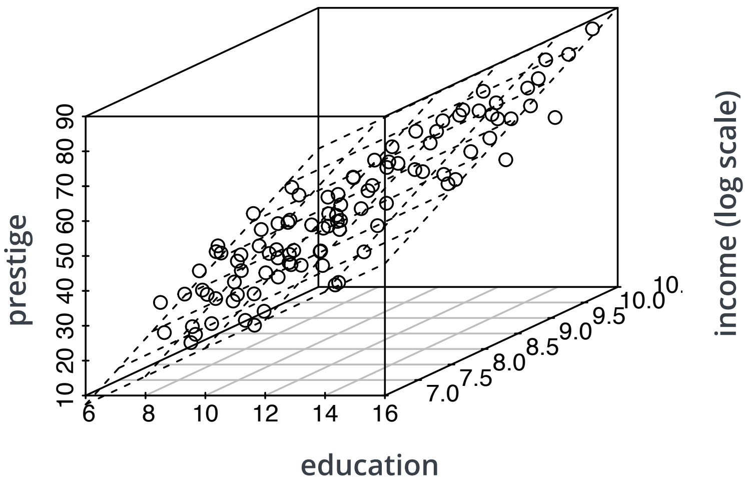
Building a Linear Prediction Rule With Multiple Linear Regression in R

Performing multiple linear regression in R is similar to performing simple linear regression. Use this part of the tool as a guide to implementing the steps for building a multiple linear prediction rule in R. You can use this code as it is presented here to build a prediction rule where prestige is the response variable Y and `education` and log`(income)` are the predictor variables X_1 and X_2 from the `Prestige` data set. You can also modify this code to build multiple linear prediction rules for your own data.

1. Create a three-dimensional scatterplot of your data. In this example, `prestige` scores of different occupations are plotted against both `education` and log`(income)`.

```
install.packages("scatterplot3d")         # Install the scatterplot3d package
library(scatterplot3d)                    # Load the scatterplot3d package
Prestige$logincome <- log(Prestige$income)# Create the logincome variable
x1 <- Prestige$education                  # Designate education as x1
x2 <- Prestige$logincome                  # Designate logincome as x2
y <- Prestige$prestige                    # Designate prestige as y

s3d <- scatterplot3d(x1, x2, y,           # Create the plot
        xlab = "Education",               # x-axis label
        ylab = "Income (log scale)",      # y-axis label
        zlab = "Prestige Score")          # z-axis label
s3d
```

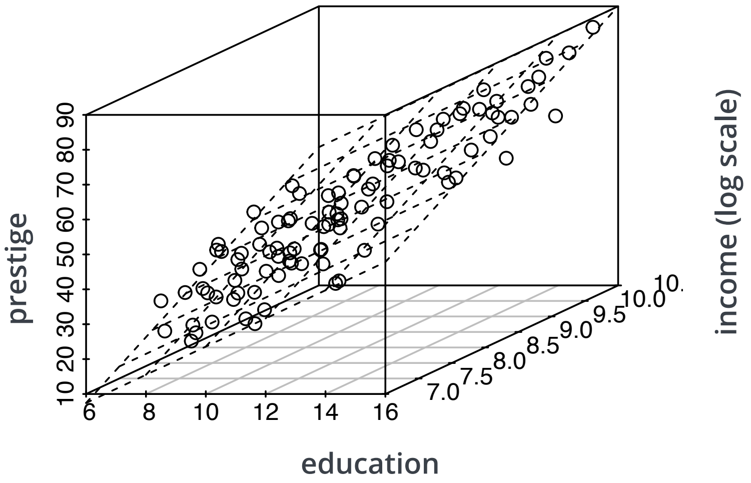


2. Use multiple linear regression to build a linear prediction rule that fits a plane through your data set. In this example, the code to do this looks like this:

```
fit.edu.inc <- lm(prestige ~ education + logincome, data = Prestige)
```

3. Plot the plane through your 3D scatterplot so you can see where it passes through your data.

```
s3d$plane3d(fit.edu.inc)
```



4. Retrieve the intercept and slopes of your regression so that you can write down your linear prediction rule and use it to make predictions.

```
fit.edu.inc

##
## Call:
## lm(formula = prestige ~ education + logincome, data = Prestige)
##
## Coefficients:
## (Intercept)    education    logincome
##    -101.188        4.038       12.056

fit.edu.inc$coefficients[2] # slope of education

## education
##  4.037525

fit.edu.inc$coefficients[3] # slope of logincome

## logincome
##  12.056
```

Using these values, you can write your prediction rule as:

$$PredictedPrestige = -101.188 + 4.038 \times education + 12.056 \times log(income)$$

This means the average prestige score increases by 4.038 units for every additional year of education among the employees in a particular occupation when average income of employees is constant.

5. Use the `predict()` command to make predictions based on your prediction rule. In the example below, `prestige` is predicted based on both `education` and log`(income)`.

```
# Write out your prediction rule:
predict.edu.8.loginc.10 = -101.188 + 4.038*8 + 12.056*10

# Create a data frame that contains values of predictor variables for which
# to calculate response variables:

new.dat = data.frame(education = c(8,8,8), logincome = c(8, 9, 10))
new.dat
##   education logincome
## 1         8         8
## 2         8         9
## 3         8        10

# Use the predict() function with the data frame you created:
predict(fit.edu.inc, newdata = new.dat)
##        1        2        3
## 27.56324 39.61964 51.67604
```

6. You can calculate MSE and R^2 of your prediction rule by retrieving the residuals stored in the vector `fit.edu.inc$residuals`, then use these values to compare the fits of different prediction rules.

```
# Calculate MSE:
MSE = mean(fit.edu.inc$residuals^2)
MSE

# Calculate variance:
y = Prestige$prestige
var = mean((y - mean(y))^2)

# Calculate R-squared:
R_squared = 1 - MSE/var
R_squared
```