

How do I create dummy variables in R step-by-step?

Authored by
stats writer

April 25, 2024

RECOMMENDED CITATION

stats writer (2024). *How do I create dummy variables in R step-by-step?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=139080>

Creating dummy variables in R is a process of converting categorical variables into numerical variables, where 0 represents the absence of a category and 1 represents the presence of a category. This is a useful technique in data analysis and statistical modeling. To create dummy variables in R, follow these steps:

1. **Understand the categorical variable:** Before creating dummy variables, it is essential to understand the categorical variable and its categories. This includes the number of categories, the naming convention, and the specific category that will serve as the reference group.
2. **Load the necessary packages:** To create dummy variables, you will need to load the "dummies" package in R. This can be done by using the command: `library(dummies)`.
3. **Create the dummy variables:** Once the package is loaded, you can use the `dummy.data.frame()` function to create the dummy variables. This function takes two arguments: the data frame and the column name of the categorical variable. It will automatically create dummy variables for each category and add them as new columns to the data frame.
4. **Rename the dummy variables:** By default, the dummy variables will be named after the categories with a prefix of "dummy_". It is recommended to rename these variables to make them more meaningful and easier to work with.
5. **Remove the reference group:** The reference group, or the category that was chosen as the reference, will not have a corresponding dummy variable. Therefore, it is necessary to remove this column from the data frame to avoid multicollinearity.
6. **Use the dummy variables in your analysis:** The dummy variables can now be used in your analysis or modeling process. They can be treated as numerical variables and included in regression or other statistical models.

Overall, creating dummy variables in R is a simple process that involves loading a package, using a function, and renaming the variables. It is a useful technique for converting categorical data into numerical data and is commonly used in data analysis.

Create Dummy Variables in R (Step-by-Step)

A is a type of variable that we create in regression analysis so that we can represent a categorical variable as a numerical variable that takes on one of two values:

zero or one.

For example, suppose we have the following dataset and we would like to use *age* and *marital status* to predict *income*:

Income	Age	Marital Status
\$45,000	23	Single
\$48,000	25	Single
\$54,000	24	Single
\$57,000	29	Single
\$65,000	38	Married
\$69,000	36	Single
\$78,000	40	Married
\$83,000	59	Divorced
\$98,000	56	Divorced
\$104,000	64	Married
\$107,000	53	Married

To use *marital status* as a predictor variable in a regression model, we must convert it into a dummy variable.

Since it is currently a categorical variable that can take on three different values ("Single", "Married", or "Divorced"), we need to create $k-1 = 3-1 = 2$ dummy variables.

To create this dummy variable, we can let "Single" be our baseline value since it occurs most often. Thus, here's how we would convert *marital status* into dummy variables:

Income	Age	Marital Status	Income	Age	Married	Divorced
\$45,000	23	Single	\$45,000	23	0	0
\$48,000	25	Single	\$48,000	25	0	0
\$54,000	24	Single	\$54,000	24	0	0
\$57,000	29	Single	\$57,000	29	0	0
\$65,000	38	Married	\$65,000	38	1	0
\$69,000	36	Single	\$69,000	36	0	0
\$78,000	40	Married	\$78,000	40	1	0
\$83,000	59	Divorced	\$83,000	59	0	1
\$98,000	56	Divorced	\$98,000	56	0	1
\$104,000	64	Married	\$104,000	64	1	0
\$107,000	53	Married	\$107,000	53	1	0

This tutorial provides a step-by-step example of how to create dummy variables for this exact dataset in R and then perform regression analysis using these dummy variables as predictors.

Step 1: Create the Data

First, let's create the dataset in R:

```
#create data frame
```

```
df <- data.frame(income=c(45000, 48000, 54000, 57000,
```

```
65000, 69000,  
78000, 83000, 98000, 104000, 107000),  
age=c(23, 25, 24, 29, 38, 36, 40, 59, 56, 64, 53),  
status=c('Single', 'Single', 'Single', 'Single',  
'Married', 'Single', 'Married', 'Divorced',  
'Divorced', 'Married', 'Married'))
```

```
#view data frame  
df
```

```
income age status
```

```
1 45000 23 Single
```

```
2 48000 25 Single
```

```
3 54000 24 Single
```

```
4 57000 29 Single
```

```
5 65000 38 Married
```

```
6 69000 36 Single
```

```
7 78000 40 Married
```

```
8 83000 59 Divorced
```

```
9 98000 56 Divorced
```

```
10 104000 64 Married
```

```
11 107000 53 Married
```

Step 2: Create the Dummy Variables

Next, we can use the `ifelse()` function in R to define dummy variables and then define the final data frame we'd like to use to build the regression model:

```
#create dummy variables
```

```
married <- ifelse(df$status == 'Married', 1, 0)
```

```
divorced <- ifelse(df$status == 'Divorced', 1, 0)
```

```
#create data frame to use for regression
```

```
df_reg <- data.frame(income = df$income,
```

```
age = df$age,
```

```
married = married,
```

```
divorced = divorced)
```

```
#view data frame
```

```
df_reg
```

```
income age married divorced
```

```
1 45000 23 0 0
```

```
2 48000 25 0 0
```

```
3 54000 24 0 0
```

```
4 57000 29 0 0
```

```
5 65000 38 1 0
```

```
6 69000 36 0 0
```

```
7 78000 40 1 0
```

```
8 83000 59 0 1
9 98000 56 0 1
10 104000 64 1 0
11 107000 53 1 0
```

Step 3: Perform Linear Regression

Lastly, we can use the `lm()` function to fit a multiple linear regression model:

```
#create regression model
```

```
model <- lm(income ~ age + married + divorced,
data=df_reg)
```

```
#view regression model output
```

```
summary(model)
```

Call:

```
lm(formula = income ~ age + married + divorced, data =
df_reg)
```

Residuals:

```
Min 1Q Median 3Q Max
```

```
-9707.5 -5033.8 45.3 3390.4 12245.4
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	14276.1	10411.5	1.371	0.21266
age	1471.7	354.4	4.152	0.00428 **
married	2479.7	9431.3	0.263	0.80018
divorced	-8397.4	12771.4	-0.658	0.53187

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8391 on 7 degrees of freedom

Multiple R-squared: 0.9008, Adjusted R-squared: 0.8584

F-statistic: 21.2 on 3 and 7 DF, p-value: 0.0006865

Income = 14,276.1 + 1,471.7*(age) + 2,479.7*(married) - 8,397.4*(divorced)

We can use this equation to find the estimated income for an individual based on their age and marital status. For example, an individual who is 35 years old and married is estimated to have an income of \$68,264:

Income = 14,276.2 + 1,471.7*(35) + 2,479.7*(1) - 8,397.4*(0) = \$68,264

Here is how to interpret the regression coefficients from the table:

Intercept: The intercept represents the average income for a single individual who is zero years old. Obviously you can't be zero years old, so it doesn't make sense to interpret the intercept by itself in this particular regression model.

Age: Each one year increase in age is associated with an average increase of \$1,471.70 in income. Since the p-value (.004) is less than .05, age is a statistically significant predictor of income.

Married: A married individual, on average, earns \$2,479.70 more than a single individual. Since the p-value (0.800) is not less than .05, this difference is not statistically significant.

Divorced: A divorced individual, on average, earns \$8,397.40 less than a single individual. Since the p-value (0.532) is not less than .05, this difference is not statistically significant.

Since both dummy variables were not statistically significant, we could drop *marital status* as a predictor from the model because it doesn't appear to add any predictive value for income.