TOOL

Using Formulas Within Functions

All functions use arguments to determine what operations they should carry out, but some functions can also use formulas. A formula is a special type of object in R. Using formulas within a function can make it easier to write and adjust your code. Functions that can use formulas include **`plot(),`** **`aggregate(), barchart(),`** and **`boxplot()`**.

The examples below are based on the **`titanic`** data set you examined in this course. You can use the **`titanic`** data set to ask many different questions that have varying levels of complexity.

Using R With This Tool

The portions of this tool with a gray background are code text that you can use to do the examples included in this tool or modify to work with your own data. To use these examples, type the lines of code that don't begin with a pound sign (#) into R to carry out the command. Commented text begins with one pound sign (#) and explains the lines of code. The code output begins with two pound signs (##).

Data Set Information

The **`titanic`** data set contains demographic information of passengers on the RMS Titanic, which sank in the Atlantic Ocean in 1912. The **`titanic`** data set has data on each passenger in the rows and on passenger characteristics in the columns. To use the **`titanic`** data set with this tool, download the data set, set your working directory to the location of the data set, and run the following code:

```r
titanic <- read.table("titanic.txt", header = TRUE) # Read in the data

# Change the Survived variable to a factor with the factor() function by
# telling R to replace 0 with No and 1 with Yes, then replacing
# titanic$Survived with SurvivedFactor:

SurvivedFactor <- factor(titanic$Survived, levels = c("0", "1"),
                            labels = c("No", "Yes"))
titanic$Survived <- SurvivedFactor

# Create the variable SurvBin in which 1 indicates a passenger
# survived and 0 indicates a passenger did not survive:

titanic$SurvBin = ifelse(titanic$Survived == "Yes", 1, 0)

head(titanic) # display the first 6 rows of data
```

```
   Name                                          PClass    Age  Sex     Survived  SurvBin
   <fctr>                                        <fctr>  <dbl>  <fctr>  <fctr>      <dbl>
1  Allen, Miss Elisabeth Walton                  1st     29.00  female  Yes             1
2  Allison, Miss Helen Loraine                   1st      2.00  female  No              0
3  Allison, Mr Hudson Joshua Creighton           1st     30.00  male    No              0
4  Allison, Mrs Hudson JC (Bessie Waldo Daniels) 1st     25.00  female  No              0
5  Allison, Master Hudson Trevor                 1st      0.92  male    Yes             1
6  Anderson, Mr Harry                            1st     47.00  male    Yes             1
6 rows
```

Writing a Formula

When you answer a bivariate question about the association between two variables, you'll need to visualize or summarize a variable for different values of one or more other variables. For example, if you want to understand the association between a passenger's sex and their survival, `SurvBin` will be grouped by `Sex`. In this scenario, your formula is:

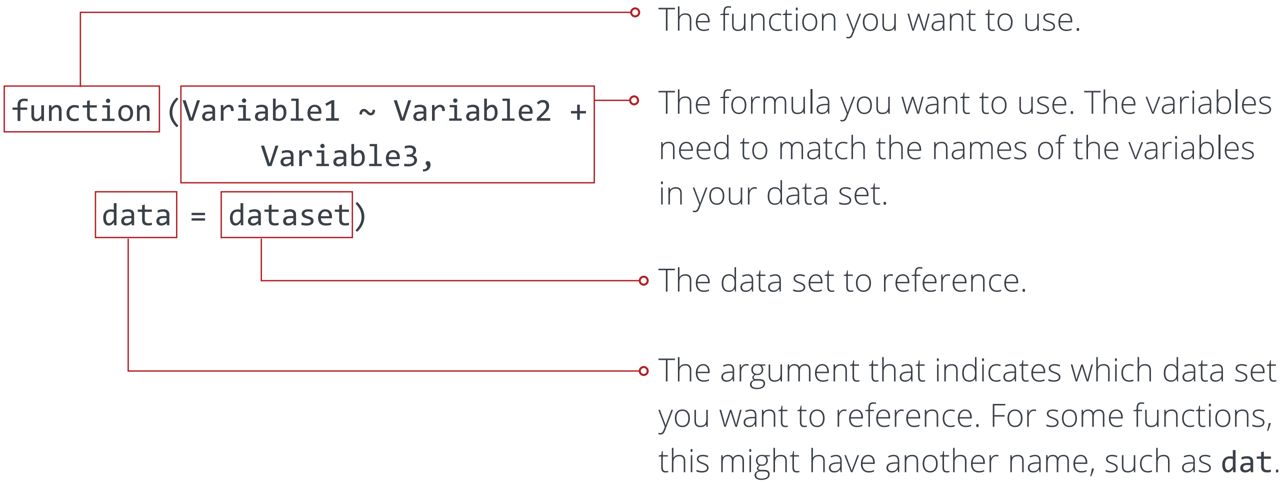
`SurvBin ~ Sex`.

When you answer a multivariate question, you are assessing one variable in terms of multiple other variables. For example, if you want to understand how a passenger's class influences the association of `SurvBin` and `Sex`, all you need to do is change the formula to add the variable `PClass:`

`SurvBin ~ Sex + PClass`.

Specifying Your Command

Once you’ve determined the formula you should use, a function that contains a formula usually looks like this:



```
function (Variable1 ~ Variable2 +
              Variable3,
          data = dataset)
```

The function you want to use.

The formula you want to use. The variables need to match the names of the variables in your data set.

The data set to reference.

The argument that indicates which data set you want to reference. For some functions, this might have another name, such as `dat`.

Example: `aggregate()`

You can use the **`aggregate()`** command to summarize passenger survival (**`SurvBin`**) across different passenger classes (**`PClass`**). This function uses the argument **`FUN`** to indicate which summary statistic you want to calculate. Examples of summary statistics you can calculate include **`mean`**, **`median`**, or standard deviation (**`sd`**).

The command below on the left creates a table, **`prop1`**, that calculates the mean survival of Titanic passengers by their sex. The command below on the right creates a table, **`prop2`**, that calculates the mean survival of Titanic passengers by both sex and passenger class.

```
prop1 <- aggregate(SurvBin ~ Sex,
    data = titanic,
    FUN = mean)
prop1

##        Sex   SurvBin
## [1] female 0.6666667
## [2]   male 0.1668625
```

```
prop2 <- aggregate(SurvBin ~ Sex + PClass,
    data = titanic,
    FUN = mean)
prop2

##        Sex PClass   SurvBin
## [1] female    1st 0.9370629
## [2]   male    1st 0.3296089
## [3] female    2nd 0.8785047
## [4]   male    2nd 0.1445087
## [5] female    3rd 0.3773585
## [6]   male    3rd 0.1162325
```