Handling Non-Applicable Survey Values

POL 501

2024-11-26

Table of Contents

### **Defining Non-Applicable Values in the Survey Context**

In survey data, **non-applicable values** are responses that do not provide meaningful information for analysis. These values typically represent cases where a question was skipped, the respondent didn’t understand or chose not to answer, or the response is irrelevant due to the survey’s flow logic.

Here are common types of non-applicable values in a survey context:

#### **1. Skipped Responses**

* Represented by negative integers or large numbers.
* Examples:
  + -1: Legitimate skip (e.g., the question did not apply to the respondent due to routing logic).
  + -7: Refusal to answer.
  + -9: Missing data for unknown reasons.

#### **2. Placeholder Codes for Missing or Irrelevant Responses**

* Large numbers are often used as placeholders.
* Examples:
  + 97: Not applicable.
  + 98: Don’t know.
  + 99: Prefer not to say.

#### **3. Irrelevant Responses**

* These may include codes used for respondents whose answers are logically unrelated to the question.
* For example, a respondent who answered “No” to a question about being employed may skip a question about job satisfaction.

### **Why Transform Non-Applicable Values to NA?**

* Non-applicable values are **not numerical data** but placeholders.
* Leaving them as-is can distort analysis:
  + Inflates means or sums.
  + Causes errors in functions like regression or ANOVA.
* Replacing with NA allows R to ignore these values (e.g., using na.rm = TRUE).

### **Typical Handling in R**

1. Identify the non-applicable values specific to the survey (e.g., -1, 97, etc.).
2. Replace them with NA for compatibility with analytical functions.

## Mock Dataset

We simulate a survey dataset with non-applicable values such as -1, -7, 97, 98, and 99.

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

# Create a mock dataset  
set.seed(123)  
survey\_data <- data.frame(  
 respondent\_id = 1:20,  
 support\_abortion = sample(c(1, 2, 3, 5, -1, -7, 98), 20, replace = TRUE),  
 religion\_importance = sample(c(1, 2, 3, 4, 5, 99), 20, replace = TRUE),  
 gender = sample(c("Male", "Female"), 20, replace = TRUE),  
 vote\_intent = sample(c("Democrat", "Republican", "Independent", "Other", NA), 20, replace = TRUE)  
)  
  
# View unique values in columns  
table(survey\_data$support\_abortion)

##   
## -7 -1 1 2 3 5 98   
## 4 2 2 3 6 1 2

table(survey\_data$religion\_importance)

##   
## 1 2 3 4 5 99   
## 6 3 3 3 3 2

## Replace Non-Applicable Values with NA

We transform non-applicable values (-1, -7, 97, 98, 99) into NA.

# Define non-applicable values  
non\_applicable <- c(-1, -7, 97, 98, 99)  
  
# Replace non-applicable values with NA  
survey\_data\_clean <- survey\_data %>%  
 mutate(  
 support\_abortion = ifelse(support\_abortion %in% non\_applicable, NA, support\_abortion),  
 religion\_importance = ifelse(religion\_importance %in% non\_applicable, NA, religion\_importance)  
 )  
  
# View cleaned data  
table(is.na(survey\_data\_clean$support\_abortion))

##   
## FALSE TRUE   
## 12 8

table(is.na(survey\_data\_clean$religion\_importance))

##   
## FALSE TRUE   
## 18 2

## Handling NA in Analysis

With non-applicable values converted to NA, functions like mean() and sum() can now exclude them using na.rm = TRUE.

# Example: Mean of support\_abortion  
mean\_support\_abortion <- mean(survey\_data\_clean$support\_abortion, na.rm = TRUE)  
mean\_support\_abortion

## [1] 2.583333

# Example: Mean of religion\_importance  
mean\_religion\_importance <- mean(survey\_data\_clean$religion\_importance, na.rm = TRUE)  
mean\_religion\_importance

## [1] 2.666667

## Explanation of mutate() Logic:

* **mutate()**: Adds or modifies columns in a data frame.
* **ifelse()**: Replaces values matching the condition (%in% non\_applicable) with NA. Values not matching remain unchanged.
* **%in%**: Checks if a value exists in the non\_applicable list.

With this setup, non-applicable values are now consistently handled as NA, making them compatible with R’s analysis functions.