REACH Data Unit Officer Test

Idrissa Dabo

2022-11-22

## REACH Data Unit Officer Test

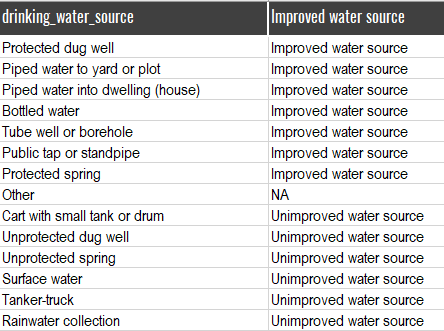
This test includes three parts, 1. General Knowledge, 2. Data Processing and 3. Coding and dashboard creation. Use the spreadsheets Annex 1 to help you answer the questions below. The test has been designed to take 2 hours and 30 minutes and examines several competencies regarding data analysis. Read all questions before you begin and note that the three parts can be completed in any order. Aim to attempt every section, but note that we would prefer high quality answers to a few questions (at least one per part), rather than a rushed attempt at all of them. When sending back your answers, please share all scripts, code, files etc. that you used to solve the exercises. Please also list all websites/external sources you used to answer the questions. All answers can be noted directly on this answer sheet unless otherwise specified. Please return this document with you answers, together with Annex 1 by email.

### Part 1: General Knowledge

1. Explain p-values in layman terms. Feel free to use analogies or examples. Keep it simple, but make sure to stay technically accurate.
2. Why are they important?
3. How can they be interpreted?
4. What are some common pitfalls/misunderstandings in their use and interpretation?
5. When would a Mosaic plot be an appropriate visualization?
6. What is personally identifiable information (PII)? Provide an example. When is it ok to collect PII?

### Part 2: Data processing

In the spreadsheet Annex 1, you will find a raw dataset from a recent data collection exercise that was carried out by your team.

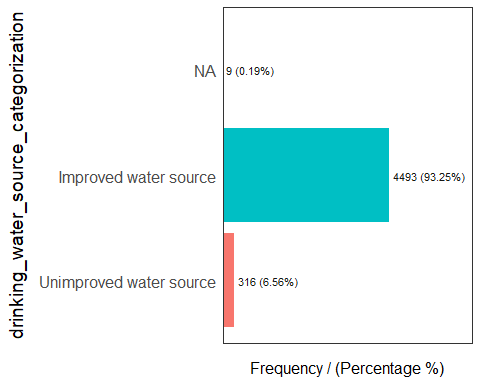
1. There are errors in the dataset. Please identify at least four errors by highlighting them in yellow in the excel sheets. In the cleaning log tab, report the cell IDs, variable name and a small explanation on why do you think this value can be an error in the comment column.
2. Using the programming language of R, create a new variable characterizing the household drinking water source into improved / unimproved source following the classification below. Paste the code / function you used below. 

dataset <- read.csv("C:/Users/USER MSI/Documents/R Project/Test\_impact/Data/Annex 1 - REACH Assessment Test Database\_DataAnalyst\_v2.csv")  
  
# check if the total household number is ok  
dataset <- dataset %>% mutate(  
 total\_calculated = Number.household.member.boy.under5.years.old +  
 Number.household.member.\_girl\_under5.years.old +   
 Number.household.member.boy\_5\_17.years.old +  
 number.adult.household.members.years.old +  
 household\_girl\_5\_17  
)  
  
dataset <- dataset %>% mutate(  
 total\_check = case\_when(  
 total\_calculated == Total.household.number ~ "ok",  
 TRUE ~ "not ok"  
 )  
)  
  
# extract row with error in the total number of household  
  
total\_different <- dataset %>% filter(  
 total\_check == "not ok"  
)  
# write\_xlsx(total\_different, "Data/total\_different.xlsx")  
  
# check household with no source of drinking water  
no\_source <- dataset %>% filter(is.na(drinking\_water\_source))  
# write\_xlsx(no\_source, "Data/no\_source.xlsx")

* in the file total different you will see households with a error in the total and also negative value in some column such as (number.adult.household.members.years.old)
* it is impossible to have households without water sources to drink. So I’m going to remove those households (in the no\_source excel sheet) and continue with those who have this information.

# dataset <- read.csv("C:/Users/USER MSI/Documents/R Project/Test\_impact/Data/Annex 1 - REACH Assessment Test Database\_DataAnalyst\_v2.csv")  
# names(dataset)  
dataset <- dataset[-c(is.na(dataset$drinking\_water\_source)),]  
dataset <- dataset %>% mutate(  
 drinking\_water\_source\_categorization = case\_when(  
 drinking\_water\_source == "Cart with small tank or drum" ~ "Unimproved water source",  
 drinking\_water\_source == "Protected dug well" ~ "Improved water source",  
 drinking\_water\_source == "Piped water to yard or plot" ~ "Improved water source",  
 drinking\_water\_source == "Piped water into dwelling (house)" ~ "Improved water source",  
 drinking\_water\_source == "Bottled water" ~ "Improved water source",  
 drinking\_water\_source == "Tube well or borehole" ~ "Improved water source",  
 drinking\_water\_source == "Public tap or standpipe" ~ "Improved water source",  
 drinking\_water\_source == "Protected spring" ~ "Improved water source",  
 drinking\_water\_source == "Unprotected dug well" ~ "Unimproved water source",  
 drinking\_water\_source == "Unprotected spring" ~ "Unimproved water source",  
 drinking\_water\_source == "Other" ~ "Unimproved water source",  
 drinking\_water\_source == "Surface water" ~ "Unimproved water source",  
 drinking\_water\_source == "Tanker-truck" ~ "Unimproved water source",  
 drinking\_water\_source == "Rainwater collection" ~ "Unimproved water source"  
  
 )  
)  
  
funModeling::freq(dataset, "drinking\_water\_source\_categorization")

## Registered S3 methods overwritten by 'Hmisc':  
## method from   
## [.labelled expss  
## print.labelled expss  
## as.data.frame.labelled expss



## drinking\_water\_source\_categorization frequency percentage cumulative\_perc  
## 1 Improved water source 4493 93.25 93.25  
## 2 Unimproved water source 316 6.56 99.81  
## 3 <NA> 9 0.19 100.00