LATERITE DATA QUALITY ANALYST

LANGAT ERICK

2023-12-30

#### DATA QUALITY ANALYST ASSESSEMENT

**Question:1**

1. Using the variables names and values labels provided in Table 1 below, please label:
2. all variables in the main and roster datasets. (10 marks)
3. the values of all categorical variables in both the main and roster datasets. (5marks)

### ***Main Dataset:***

**Variable Labels:**

* province\_id: Province code
* province\_name: Province of residence
* district\_id: District code
* district\_name: District of residence
* sector\_id: Sector code
* sector\_name: Sector of residence
* cell\_id: Cell code
* cell\_name: Cell of residence
* village\_id: Village code
* village\_name: Village of residence
* list\_avail: Ubudehe list availability
* list\_source: Source of the ubudehe list
* ubudehe\_contact: Ubudehe list contact person
* contact\_role: Role of the ubudehe list contact person
* contact\_phone: Phone number of the ubudehe list contact person
* num\_households: Number of households in ubudehe category 1
* hh\_head\_position: Position of household head on ubudehe list
* name\_hh\_head: Name of the household head
* nid\_hh\_head: National ID number of the household head
* phone\_hh\_head: Phone number of the household head
* gps: GPS coordinates of where the list was obtained
* parent\_key: Unique ID
* **Categorical variables**
* ***list\_avail:***
* 0: No
* 1: Yes
* ***list\_source:***
* 1: Sector Executive Officer
* 2: Cell Executive Officer
* 3: Village Leader
* ***contact\_role:***
* 1: Village Leader
* 2: Village in-charge of security
* 3: Village Community Health Worker

**Roster Dataset:**

**Variable Labels:**

* hh\_head\_position: Position of household head on ubudehe list
* name\_hh\_head: Name of the household head
* nid\_hh\_head: National ID number of the household head
* phone\_hh\_head: Phone number of the household head

**Categorical Variable Values:**

There are no specific categorical variables in the roster dataset based on the information provided.

**Question:2**

1. Using the values and value labels in Table 1 (Variable description) below, create a single variable for each of the location variables (province, district, sector, cell, and village) that displays the value label but also has the value embedded. (15 marks)

#Load Packages  
suppressPackageStartupMessages(require(tidyverse))

#Load data  
#main dataset  
main <- read.csv("C:/Users/langa/OneDrive/Desktop/DATA CLEANING R PROGRAMMING/main\_dataset.csv")  
colnames(main)#column names

## [1] "province\_id" "province\_name" "district\_id" "district\_name"   
## [5] "sector\_id" "sector\_name" "cell\_id" "cell\_name"   
## [9] "village\_id" "village\_name" "list\_available" "list\_source"   
## [13] "ubudehe\_contact" "contact\_role" "contact\_phone" "num\_households"   
## [17] "gpslatitude" "gpslongitude" "gpsaltitude" "gpsaccuracy"   
## [21] "parent\_key"

#single variable for each of the location variables (province, district, sector, cell, and  
# village) that displays the value label but also has the value embedded.  
  
# For Province  
main$province\_name <- paste(main$province\_name, "(", main$province\_id, ")", sep = " ")  
  
# For District  
main$district\_name <- paste(main$district\_name, "(", main$district\_id, ")", sep = " ")  
  
# For Sector  
main$sector\_name <- paste(main$sector\_name, "(", main$sector\_id, ")", sep = " ")  
  
# For Cell (replace 'cell' with the actual column name)  
main$cell\_name <- paste(main$cell\_name, "(", main$cell\_id, ")", sep = " ")  
  
# For Village (replace 'village' with the actual column name)  
main$village\_name <- paste(main$village\_name, "(", main$village\_id, ")", sep = " ")  
  
# Display the updated data frame  
# print(main)  
# view(main)

#Roster dataset  
roster <- read.csv("C:/Users/langa/OneDrive/Desktop/DATA CLEANING R PROGRAMMING/roster\_dataset.csv")# importing dataset  
colnames(roster)# These are the column names

## [1] "hh\_head\_position" "name\_hh\_head" "nid\_hh\_head" "phone\_hh\_head"   
## [5] "parent\_key"

**Question:3**

**3.** Please create a dataset combining the main and roster dataset so that we have one observation per village with all household heads as well as their personal information contained in the roster. Save this dataset as “merged\_yourname\_yyyymmdd”.

# Merge the main and roster datasets based on the common column "parent\_key"  
merged\_data <- inner\_join(main, roster, by = "parent\_key")# joining the 2-data using a common ID  
  
# The inner\_join() function from the 'dplyr' package is used to merge the main and roster datasets based on the common column 'parent\_key'. This function performs an inner join, which means it keeps only the rows that have matching values in both   
  
# Save the merged dataset as a CSV file  
write.csv(merged\_data, file = "merged\_ericklangat\_20231229.csv",   
 row.names = FALSE)#Save the results in a CSV.\_ Format  
  
# Display the merged dataset  
 head(merged\_data,4)# check the First 4 Rows of the data

## province\_id province\_name district\_id district\_name sector\_id  
## 1 1 Kigali ( 1 ) 11 Nyarugenge ( 11 ) 1101  
## 2 1 Kigali ( 1 ) 11 Nyarugenge ( 11 ) 1101  
## 3 1 Kigali ( 1 ) 11 Nyarugenge ( 11 ) 1101  
## 4 1 Kigali ( 1 ) 11 Nyarugenge ( 11 ) 1101  
## sector\_name cell\_id cell\_name village\_id village\_name  
## 1 Gitega ( 1101 ) 110101 Akabahizi ( 110101 ) 11010102 Gihanga ( 11010102 )  
## 2 Gitega ( 1101 ) 110101 Akabahizi ( 110101 ) 11010102 Gihanga ( 11010102 )  
## 3 Gitega ( 1101 ) 110101 Akabahizi ( 110101 ) 11010102 Gihanga ( 11010102 )  
## 4 Gitega ( 1101 ) 110101 Akabahizi ( 110101 ) 11010102 Gihanga ( 11010102 )  
## list\_available list\_source ubudehe\_contact contact\_role contact\_phone  
## 1 1 3 ndakaza 2 788004184  
## 2 1 3 ndakaza 2 788004184  
## 3 1 3 ndakaza 2 788004184  
## 4 1 3 ndakaza 2 788004184  
## num\_households gpslatitude gpslongitude gpsaltitude gpsaccuracy  
## 1 5 -1.943086 30.05912 0 200  
## 2 5 -1.943086 30.05912 0 200  
## 3 5 -1.943086 30.05912 0 200  
## 4 5 -1.943086 30.05912 0 200  
## parent\_key hh\_head\_position name\_hh\_head  
## 1 uuid:7ae0a1ab-0f9f-4708-b787-5f9a9209ae23 1 mwitende  
## 2 uuid:7ae0a1ab-0f9f-4708-b787-5f9a9209ae23 2 ruhumuriza  
## 3 uuid:7ae0a1ab-0f9f-4708-b787-5f9a9209ae23 3 kabanda  
## 4 uuid:7ae0a1ab-0f9f-4708-b787-5f9a9209ae23 4 kangabe  
## nid\_hh\_head phone\_hh\_head  
## 1 1.197194e+15 797352301  
## 2 1.194808e+15 729858153  
## 3 1.196502e+15 726559120  
## 4 1.200033e+15 723251934

names(merged\_data)# Checking the names of the Merged\_dataset(Main & Roster)

## [1] "province\_id" "province\_name" "district\_id" "district\_name"   
## [5] "sector\_id" "sector\_name" "cell\_id" "cell\_name"   
## [9] "village\_id" "village\_name" "list\_available" "list\_source"   
## [13] "ubudehe\_contact" "contact\_role" "contact\_phone" "num\_households"   
## [17] "gpslatitude" "gpslongitude" "gpsaltitude" "gpsaccuracy"   
## [21] "parent\_key" "hh\_head\_position" "name\_hh\_head" "nid\_hh\_head"   
## [25] "phone\_hh\_head"

#Rename Two Variables "list\_available" to "list\_avail"& 'gpsaccuracy' to 'gps'  
merged\_data$list\_avail <- merged\_data$list\_available #Renaming the column  
merged\_data$gps <- merged\_data$gpsaccuracy# Renaming the column

**Question:4**

**4.** Organize the data set.

a. Order the variables as per Table 1 (Variable description) below. (5 marks)

b. Make sure all variables are labeled properly as per Table 1 (Variable description) below. Save this dataset as “clean\_yourname\_yyyymmdd”. (10 marks)

# Reorder the variables as per description provided  
ordered\_variables <- c( "province\_id", "province\_name", "district\_id", "district\_name", "sector\_id", "sector\_name",   
 "cell\_id", "cell\_name", "village\_id", "village\_name", "list\_avail", "list\_source",   
 "ubudehe\_contact", "contact\_role", "contact\_phone", "num\_households",   
 "hh\_head\_position", "name\_hh\_head", "nid\_hh\_head", "phone\_hh\_head", "gps", "parent\_key")  
  
ordered\_data <- merged\_data %>% select(ordered\_variables)# Selecting the required columns only and ordering them as per the description

## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.  
## ℹ Please use `all\_of()` or `any\_of()` instead.  
## # Was:  
## data %>% select(ordered\_variables)  
##   
## # Now:  
## data %>% select(all\_of(ordered\_variables))  
##   
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

# Save the ordered dataset to #Clean data as a CSV file  
write.csv(ordered\_data, file = "clean\_ericklangat\_20231230.csv", row.names = FALSE)#Save the results in a csv-format

**Question:5**

**5.** To aid the field team in drafting a data collection field plan, please extract lists of households per district in .xlsx format, with each sector in each of the 3 districts on a separate worksheet. Save your do-file as “yourname\_yyymmdd”. (25 marks)

dim(ordered\_data)#check number of rows and columns before Removing NA'S

## [1] 480 22

order\_d= ordered\_data  
order\_d <- na.omit(order\_d) %>% distinct()#Remove Missing Vlues and the duplicates  
dim(order\_d)# Check the dimension of the new data after Removing NA'S

## [1] 471 22

#We Group by district and sector  
Gasabo\_data <- order\_d %>%   
 select(district\_name, sector\_name, num\_households) %>%   
 filter(district\_name=="Gasabo") %>%   
 group\_by(district\_name, sector\_name)   
  
# Summarize the data as per the description provided  
summary\_data1 <- Gasabo\_data %>%   
 summarise(num\_households = sum(num\_households))

## `summarise()` has grouped output by 'district\_name'. You can override using the  
## `.groups` argument.

summary\_data1 %>% head()

## # A tibble: 0 × 3  
## # Groups: district\_name [0]  
## # ℹ 3 variables: district\_name <chr>, sector\_name <chr>, num\_households <int>

#Group by district and sector  
Nyarugenge\_data <- order\_d %>% select(district\_name, sector\_name, num\_households) %>%   
 filter(district\_name=="Nyarugenge") %>%   
 group\_by(district\_name, sector\_name)   
  
# Summarize the data as needed  
summary\_data2 <- Nyarugenge\_data %>%  
 summarise(num\_households = sum(num\_households))

## `summarise()` has grouped output by 'district\_name'. You can override using the  
## `.groups` argument.

summary\_data2 %>% head()

## # A tibble: 0 × 3  
## # Groups: district\_name [0]  
## # ℹ 3 variables: district\_name <chr>, sector\_name <chr>, num\_households <int>

#We Group by district and sector  
Kicukiro\_data <- order\_d %>% select(district\_name, sector\_name, num\_households) %>%   
 filter(district\_name=="Kicukiro") %>%   
 group\_by(district\_name, sector\_name)  
  
# Summarize the data as needed  
summary\_data3 <- Kicukiro\_data %>%  
 summarise(num\_households = sum(num\_households))

## `summarise()` has grouped output by 'district\_name'. You can override using the  
## `.groups` argument.

summary\_data3 %>% head()

## # A tibble: 0 × 3  
## # Groups: district\_name [0]  
## # ℹ 3 variables: district\_name <chr>, sector\_name <chr>, num\_households <int>

# Save the dataset to # a CSV file the convert to xlsx format  
write.csv(summary\_data1, file = "ericklangat0\_20231230.csv", row.names = FALSE)  
write.csv(summary\_data2, file = "ericklangat1\_20231230.csv", row.names = FALSE)  
write.csv(summary\_data3, file = "ericklangat2\_20231230.csv", row.names = FALSE)