Reformating the data before analysis

Surakshya Dhakal

11/30/2021

Load libraries

```
if(!require("pacman")) install.packages("pacman")
pacman::p_load(dplyr, tidyr, pander, RColorBrewer)
```

Read in data

In this case, the data was collected for a project on the use of mobile money services and the experiences of mobile money customers in three districts of a country in Africa.

```
# Read in mobilemoney_data.csv.
mm_df <- read.csv("../data/mobilemoney_data.csv", na.strings=c("","NA"))

# Check the data
# str(mm_df)   # too long to show
# head(mm_df)
```

Subset and format the data

Since the data is not in the format required for analysis, it needs to be cleaned.

Format the data so that there is one observation per participant.

```
# Convert data from long-form to wide-form
# Accounts are listed by Household ID (hhid), i.e, the participant.
# The first two columns aren't necessary for the analysis. Remove.
# Since there are 6 unique account_types,...
# ... the conversion will add 6 columns at the end of the data frame.
# Ignore account_num.
# References https://tidyr.tidyverse.org/reference/pivot_wider.html

mm_df1 <- mm_df[3:29]

mm_wide <- mm_df1 %>% pivot_wider(id_cols = c(1, 4:27),
                                  names_from = account_type, values_from = account_type)
```

```r
# Make a copy of mm_wide
mm_wide_copy <- mm_wide
```

Split character columns into multiple columns

```r
# Separate district column
# Example of an entry: District_A

sep_1 <- data.frame(do.call("rbind", strsplit(mm_wide$district, "_", fixed = TRUE)))
names(sep_1) <- c("admin_level", "district_code")
mm_wide <- cbind(mm_wide, sep_1)[-c(2:3)]
rm(sep_1)

head(mm_wide[30:31])   # The new columns are put at the end of the dataframe
```

```
##   admin_level district_code
## 1    District             A
## 2    District             B
## 3    District             A
## 4    District             A
## 5    District             C
## 6    District             B
```

```r
# Separate highest_grade_completed column
# Example of an entry: primary 6
# Instead of using do.call and strsplit, use "separate" function

mm_wide <- separate(mm_wide, 6, into = c("education", "highest_grade"), sep = " ")
head(mm_wide[6:7], 6)   # New columns are created replacing and next to the old
```

```
##   education highest_grade
## 1   primary             6
## 2   primary             3
## 3 secondary             6
## 4   primary             6
## 5   primary             6
## 6   primary             3
```

```r
# Another contains multiple company names in the same column
head(mm_wide[11])
```

```
##      mm_account_telco
## 1 Company_A Company_B
## 2                <NA>
## 3           Company_A
## 4           Company_A
## 5           Company_B
## 6                <NA>
```

```r
# Extract the companies and put each in their own column
z <- separate(mm_wide, 11, into = c("tmp1", "tmp2", "tmp3"), sep = " ")
mm_wide$Company_A <- ifelse(z$tmp1 == "Company_A", 1, NA)
mm_wide$Company_B <- ifelse(z$tmp1 == "Company_B" | z$tmp2 == "Company_B", 1, NA)
mm_wide$Company_C <- ifelse(z$tmp1 == "Company_C" |
                              z$tmp2 == "Company_C" |
                              z$tmp3 == "Company_C", 1, NA)

mm_wide <- mm_wide[-11]   # Remove the old column
rm(z)   # Remove the temporary dataframe
head(mm_wide[32:34])
```

```
##   Company_A Company_B Company_C
## 1         1         1        NA
## 2        NA        NA        NA
## 3         1        NA        NA
## 4         1        NA        NA
## 5        NA         1        NA
## 6        NA        NA        NA
```

The dataset is now ready for statistical analysis.