hw03 - Data Preparation

Assignment Overview

By now you should know what variables you want to use, and you've looked over the codebook enough now that you have an idea of some potential problems that you will encounter. This assignment uses your chosen research data, and the variables that you chose in the last assignment when you created a personal research codebook. You will thoughtfully review the variables you are interested in, document which ones will need changed and how.

All raw data needs to stay raw, and all changes need to be documented. You will create a script/code file that will make all changes to the data in a programatically and reproducible way. You will create a single code file that imports your raw data, performs some data cleaning steps, and saves out an analysis ready data set that you will use throughout the semester.

You are **not** expected to have completed data management for every one of your variables under consideration by the submission date.

Instructions

> For all coding assignments: Any reference of "Math 315 folder" or "class folder" means either a folder on your laptop where you keep all files for this class, OR your workspace in R Studio cloud.

1. Create a new Rmarkdown file named `dm_dataname_rdonatello.Rmd`. (e.g. `dm_addhealth_rdonatello.Rmd` or `dm_caterpillar_rdonatello.Rmd`).
    - Save this into your class folder
2. Download your chosen analysis data set from Google Drive.
    - Put this file in a folder named `data`.
3. In the first code chunk:
    - Load the `dplyr` and `descr` libraries.
    - Read in raw data into a data frame named `raw` It should look similar to the following code below:

```
knitr::opts_chunk$set(echo = TRUE)
library(dplyr)
library(descr)
raw <- read.delim("https://norcalbiostat.netlify.com/data/depress_081217.txt",
                  sep="\t",  header=TRUE)
```

4. Click the green play button in the top right corner of the code chunk to execute the code in that chunk.

    STOP. Did you get an error message saying Error in library(dply) : there is no package called 'descr'?

If so, then you have not installed the `descr` package yet. Do this by typing `install.packages("descr")` in the console (NOT IN YOUR SCRIPT FILE). Once it is done, rerun that code chunk. The same comment goes for all packages we use in this class.

> Do you see an object called `raw` in your *Environment*? (Top right panel in R studio). If so, then proceed.

5. Make a new object called `mydata` that contains only the few variables that you have identified in your personal codebook that you may want to use.
    - Choose *at minimum* 2 categorical and 2 continuous variables.

```
mydata  <- raw %>% select(age, marital, cesd, health)
```

6. Check each variable for necessary adjustments. Complete each of the following steps for each variable.
    - First explain in english what the variable name is and what it measures.
    - Then examine the variable using the `freq` function in the `descr` package.
    - Identify the data type of the variable using `class`. Does this match with the intended data type?
    - Recode the data as necessary (See section 1.4 in the course packet)
    - Always confirm your recodes worked as intended by creating another table or summary.

(See example at bottom of page)

7. Restrict the variables to only the ones you are investigating.
    - Similar to how you trimmed down `raw` into `mydata`, now you want to only keep the variables that you want to save into your analysis data set.
    - e.g. I only want to keep the categorical version of `health`.

```
clean <- mydata %>% select(age, marital, cesd, health_cat)
```

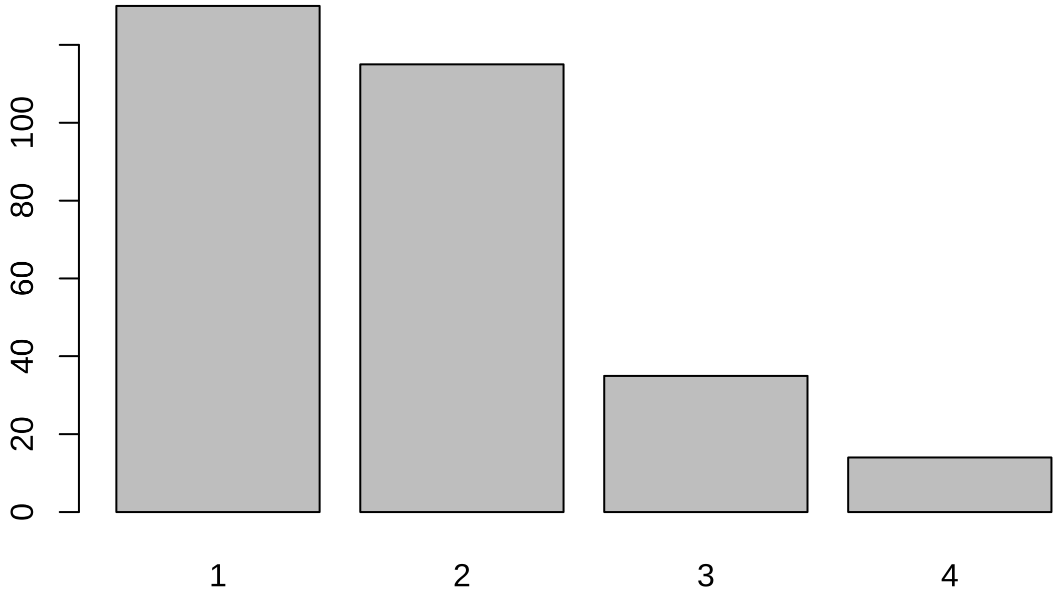
8. Save the resulting data set to your `data` folder as `datasetname_clean.Rdata` e.g. `addhealth_clean.Rdata`.
    - This will serve as your analysis data set to do all your subsequent assignments on.

```
save(clean, file="depression_clean.Rdata")
```

Example

The variable `health` records a persons perceived general health as being either Excellent, Good, Fair or Poor. This is considered an ordinal categorical variable.

```
freq(mydata$health)
```



```
## mydata$health
##       Frequency Percent
## 1           130  44.218
## 2           115  39.116
## 3            35  11.905
## 4            14   4.762
## Total       294 100.000
```

```
class(mydata$health)
```

```
## [1] "integer"
```

The variable `health` currently is an integer with numeric values 1-4, but the codebook states that this is a categorical variable where 1=Excellent, 2=Good, 3=Fair, 4=Poor. So I need to convert this numeric variable to a factor variable. There are no values outside the 1-4 range, such as a -9 that codes for missing data so I do not need to make any further adjustments (You want to code out missing before you convert variables to factors)

```
mydata$health_cat <- factor(mydata$health, labels=c("Excellent", "Good", "Fair", "Poor"))
```

I will confirm that the recode worked by making a two-way `table`

```
table(mydata$health, mydata$health_cat, useNA="always")
```

```
##
##        Excellent Good Fair Poor <NA>
##   1          130    0    0    0    0
##   2            0  115    0    0    0
##   3            0    0   35    0    0
##   4             0    0    0   14    0
##   <NA>          0    0    0    0    0
```

This shows that all 1's are now 'excellent', 4's are now 'poor' and so forth.

Submission instructions

- This is an individual project assignment
- You will compile your RMD code to create a PDF file (knit to PDF)
- Upload your PDF file to the **hw03 Data Management** folder in Google Drive.

Example

- https://norcalbiostat.netlify.com/data/dm_addhlth