Preparation questions for Data Management

2017-09-04

You know what variables you want to use, and you've looked over the codebook enough now that you have an idea of some potential data issues that you will have to address.

Here are some questions for you to think about in preparation for doing data management on your own set of variables.

Answer the following questions in prepartion to learn how to tackle problems such as missing data and recoding variables in the next week. Some of these items guide you to updating your personal codebook. For each variable that you have to modify, write a note in your codebook what you plan to do and why. You will include these notes into your data management script file as a record of your changes.

1. **Do you understand what each variable is actually measuring?**

- In your codebook, give labels to All your variables. This includes adding in a note for each variable identifying if it is Quantitative or Categorical, and Level of Measurement (i.e., Nominal, Ordinal, Interval, and Ratio).
- Does your SPC recognize these variables as such: If not, note down that you'll have to change this later.

```
depress <- read.table("https://norcalbiostat.netlify.com/data/Depress.txt", sep="\t", header=TRUE)
names(depress) <- tolower(names(depress)) # turn all variable names lower case
```

2. **Do you need to code out missing data?**

- Go through your codebook for each variable and treat all variables for missing data. If there are no missing values then write that.

3. **Are you going to only look at a subset of individuals?**

- What value of which variable specifically are you going to filter on?

```
female_only <- subset(depress, depress$sex ==2)
```

4. **Do you need to code out skip patterns?** Are you looking at questions that only pertain to a specific subpopulation? (i.e. number of packs smoked per week only applies to smokers). Responses to the # of packs question should be set to missing for non-smokers.

5. **Do you need to edit data?**

Fix a typo in AGE. A 9 was recorded when it should have been 19.

```
depress$age[depress$age==9] <- 19
min(depress$age)
```

```
## [1] 18
```

6. **Do you need to make response codes more logical?** Some examples include:

- Think about how the "yes"" and "no"" variables are coded
  - Does NO = 0 and YES = 1?

```
depress$female <- depress$sex -1
```

- Think about how the "strongly agree" to "strongly disagree" variables are coded
  - Do the numbers make sense?

- Consider recoding a quantitative variable into a categorical variable

```
depress$female_cat <- factor(depress$female, labels=c("Male", "Female"))
table(depress$female, depress$female_cat, useNA="always")
```

```
##
##        Male Female <NA>
##   0     111      0    0
##   1       0    183    0
##   <NA>    0      0    0
```

```
depress$educat = factor(depress$educat,
                        labels = c("<HS", "Some HS", "HS Grad", "Some college", "BS", "MS", "PhD"))
```

- Consider collapsing across categories
  - maybe going from 5 categories for strongly agree, agree, neutral, disagree, strongly disagree to 3 categories that represent strongly agree and agree as one category, disagree and strongly disagree as another, then neutral still in the middle.

```
depress$postHS <- ifelse(depress$educat %in%
                           c("<HS", "Some HS", "HS Grad"), "Up to HS", "Above HS")
table(depress$postHS, depress$educat, useNA="always")
```

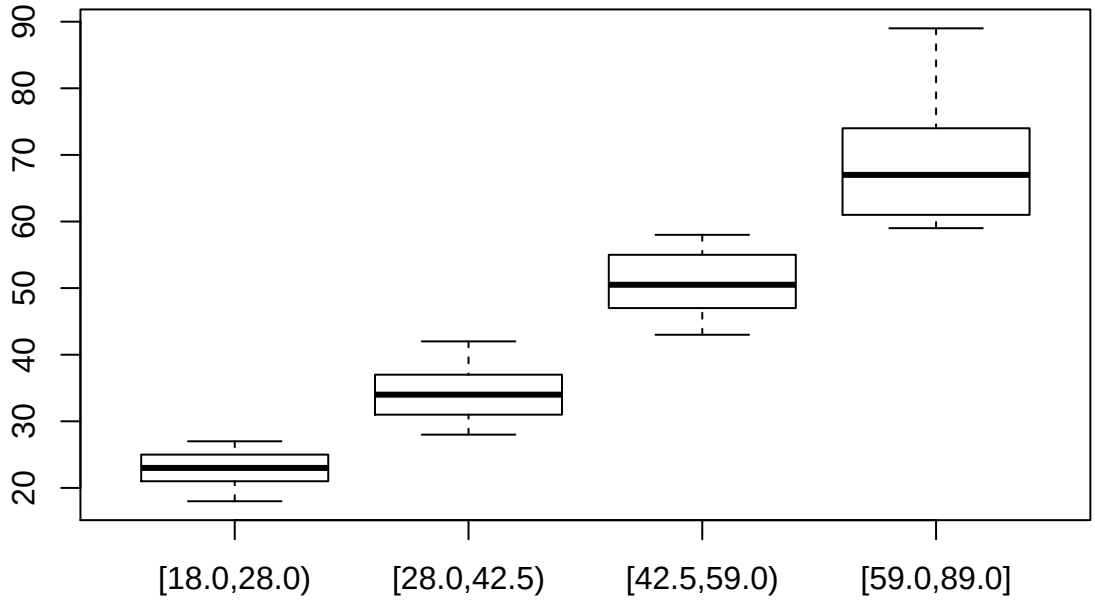
```
##
##            <HS Some HS HS Grad Some college  BS  MS PhD <NA>
##   Above HS   0       0       0           48  43  14   9    0
##   Up to HS   5      61     114            0   0   0   0    0
##   <NA>       0       0       0            0   0   0   0    0
```

- Consider collapsing a quantitative variable into categories based on percentages of the data you find after examining the frequency table.
  - (i.e. BMI: <16: underweight, 16-18.5 normal, 18.5-25 overweight, 30+ obese)

```
cuts <- quantile(depress$age, c(.25, .5, .75))
cuts
```

```
##  25%  50%  75%
## 28.0 42.5 59.0
```

```
depress$agecat <- Hmisc::cut2(depress$age, cuts=cuts)
boxplot(depress$age ~ depress$agecat)
```



7. **Do you need to create secondary variables?** If necessary, create secondary variables from continuous variables. If you are working with a number of items that represent a single construct, it may be useful to create a composite variable/score.
    - For example, I want to use a list of nicotine dependence symptoms meant to address the presence or absence of nicotine dependence (i.e., tolerance, withdrawal, craving, etc.). Rather than using a dichotomous variable (i.e., nicotine dependence present/absent), I want to examine the construct as a dimensional scale (i.e., number of nicotine dependence symptoms). In this case, I would want to recode each symptom variable so that YES = 1 and NO = 0 and then sum the items so that they represent one composite score. In the code below, the nd_sum is the new variable I am creating and the variables after of are the variables I am totaling up.
    - nd_sum=sum (of nd_symptom1 nd_symptom2 nd_symptom3 nd_symptom4); (_ this is SAS code, but the idea is similar_)

- Don't forget when creating secondary variables, you need to visually check your new variables to ensure what you thought you did actually did what you wanted to.