Data Management

Joseph Anthony, Laura Gardner, Kyle Pitzer

Outline: Messy and missing data

These slides dip a toe in the water of the vast ocean called *data management*:

1. Coding missing values
2. Understanding missing value patterns
3. Recoding continuous variables to categorical
4. Collapsing categorical variables into fewer categories
5. Identifying and correcting out-of-range values
6. Loading data from different software packages

For this workshop you will need the following packages: descr, car, tidyverse, VIM, assertr, foreign, sas7bdat

A quick review

1. Datasets

- A collection of data
- data are measurable points of information

2. Variables

- a quantity, quality, or property that can be measured
- A data set is comprised of variables

3. Observations

- A set of measurements of which a variable is comprised
- Base unit of measurement
- A data set is also comprised of observations

Messy and missing data

Definitions:

Tony Fischetti: "Messy" data =

- Many missing values
- Misspelling of variables
- Inconsistent coding
- Different unit values for the same variable
- Data entry mistakes
- Extreme outliers

Types of missing data

- Missing data can be random or systematic
- Missing completely at random: (MCAR)
  - *Reasons for missing values are not related to observed variables or other measureable characteristics*
  - *Occur entirely at random*
  - *Analysis performed on the data is unbiased*
  - *Usually an unrealistically strong assumption in practice*
- Missing at random: (MAR)
  - *Missingness is not COMPLETELY at random, but can be accounted for by observed variables*
    - For example, perhaps males were less likely to answer questions about depression on a survey
    - However, the reason males skipped the depression question had nothing to do with their level of depression
  - *MAR is impossible to verify statistically, we must rely on its reasonableness*
  - *Data with values that are MAR can be biased*
  - *There are methods you will learn in intermediate stats for estimating models that account for MAR data and give unbiased estimates*

Types of missing data (continued)

- Missing not at random: (MNAR)
  - *The information that is missing is related to the reason it is missing*
  - *This would occur if a certain group (e.g., males, older adults) failed to fill in a depression survey because their level of depression (e.g., they had higher depression than other groups)*
  - *This sort of missing results in the most bias*

Why does missing data happen?

- Intentional Coding: an event did not occur (flight was cancelled, polling place was not open, etc.)
  - *In these cases, we usually don't want to delete or omit the data, just want to code it differently*
- Survey non-response
- Data entry mistakes

Important to understand because…

- Missing values can impact analysis; they can change results so findings from a sample are not providing a true sense of what is going on in the population
- If the missing values are random, they can be dealt with in analysis and data management
- If missing values are not random, then larger bias may exist and we likely need to re-collect data or rethink our research design
- Most missing data are not completely random and possible reasons for non-randomness would be part of writing conclusions (e.g., "more men were missing the depression data, which may have biased results")

Deal with it

- How do we identify missing data in datasets?
  - *R recognizes "NA" as missing*
    - Data where missing is coded as something else should be recoded
  - *Often have to look at the data and codebook to understand what "NA" means and how it is used in each dataset*
  - *Summarize command in RStudio*
- How do we deal with missing data?
  - *Don't have any (LOL): This rarely happens*
  - *Imputation/Imputed models: Filling in the missing with a value based on other characteristics of the data (controversial)*
  - *Removal: Dropping observations or parts of observations*
  - *Recoding: Include missing as a category in the analyses*

Coding missing values

- R recognizes a data point entered as NA as a missing value
- This is not the same for all software packages and so some data sets may have missing values as blank cells or coded as "999" or some other value
- The summary function can help determine if missing values are coded as NA or something else
- For example, check the coding of variables in the smokers data set:

```
# import the data and call the data frame smokers
smokers <- read.csv("http://tinyurl.com/z2m3cgq")
```

```
# get a summary of the data
summary(smokers)
```


Examine a single variable

Start by looking at the *Income* variable more closely by just typing the name of the variable or using the freq command.

```
# print out the value of the income variable for each person
# in the data
smokers$Income
```

```
##   [1] 5-Less than $35,000 6-Less than $50,000 6-Less than $50,000
##   [4] 6-Less than $50,000 6-Less than $50,000 4-Less than $25,000
##   [7] 2-Less than $15,000 5-Less than $35,000 3-Less than $20,000
##  [10] 7-Less than $75,000 8-$75,000 of more   6-Less than $50,000
##  [13] 5-Less than $35,000 1-Less than $10,000 6-Less than $50,000
##  [16] 6-Less than $50,000 6-Less than $50,000 4-Less than $25,000
##  [19] 1-Less than $10,000 2-Less than $15,000 5-Less than $35,000
##  [22] 1-Less than $10,000 5-Less than $35,000 5-Less than $35,000
##  [25] 3-Less than $20,000 77-Don't Know       5-Less than $35,000
##  [28] 2-Less than $15,000 6-Less than $50,000 4-Less than $25,000
##  [31] 4-Less than $25,000 3-Less than $20,000 7-Less than $75,000
##  [34] 3-Less than $20,000 2-Less than $15,000 6-Less than $50,000
##  [37] 6-Less than $50,000 2-Less than $15,000 7-Less than $75,000
##  [40] 6-Less than $50,000 8-$75,000 of more   3-Less than $20,000
##  [43] 1-Less than $10,000 4-Less than $25,000 1-Less than $10,000
##  [46] 3-Less than $20,000 1-Less than $10,000 5-Less than $35,000
##  [49] 2-Less than $15,000 4-Less than $25,000 5-Less than $35,000
##  [52] 6-Less than $50,000 5-Less than $35,000 6-Less than $50,000
##  [55] 4-Less than $25,000 1-Less than $10,000 1-Less than $10,000
##  [58] 2-Less than $15,000 6-Less than $50,000 6-Less than $50,000
##  [61] 5-Less than $35,000 1-Less than $10,000 7-Less than $75,000
##  [64] 4-Less than $25,000 2-Less than $15,000 4-Less than $25,000
##  [67] 8-$75,000 of more   7-Less than $75,000 6-Less than $50,000
##  [70] 4-Less than $25,000 77-Don't Know       5-Less than $35,000
##  [73] 8-$75,000 of more   7-Less than $75,000 6-Less than $50,000
##  [76] 4-Less than $25,000 5-Less than $35,000 3-Less than $20,000
##  [79] 4-Less than $25,000 1-Less than $10,000 5-Less than $35,000
##  [82] 77-Don't Know       1-Less than $10,000 6-Less than $50,000
##  [85] 7-Less than $75,000 3-Less than $20,000 5-Less than $35,000
##  [88] 4-Less than $25,000 2-Less than $15,000 6-Less than $50,000
##  [91] 4-Less than $25,000 1-Less than $10,000 7-Less than $75,000
##  [94] 4-Less than $25,000 77-Don't Know       99-Refused
##  [97] 2-Less than $15,000 1-Less than $10,000 7-Less than $75,000
## [100] 3-Less than $20,000
## 10 Levels: 1-Less than $10,000 2-Less than $15,000 ... 99-Refused
```


The frequency table shows missing data codes

- Notice that there are two codes that represent missing data: "77-Don't Know" and "99-Refused"
- R considers these legitimate categories of this variable since they are not NA and includes them in analyses

```
# get a table of the values of income
library(descr)
freq(smokers$Income, plot=F)
```

```
## smokers$Income
##                      Frequency Percent
## 1-Less than $10,000         13      13
## 2-Less than $15,000         10      10
## 3-Less than $20,000          9       9
## 4-Less than $25,000         15      15
## 5-Less than $35,000         15      15
## 6-Less than $50,000         20      20
## 7-Less than $75,000          9       9
## 77-Don't Know                4       4
## 8-$75,000 of more            4       4
## 99-Refused                   1       1
## Total                      100     100
```

Example of missing values in analyses

```
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |            Std Residual |
## |-------------------------|
##
## ================================================
##                       smokers$sex
## smokers$Income        1-Male   2-Female   Total
## ------------------------------------------------
## 1-Less than $10,000        2         11      13
##                        0.154      0.846   0.130
##                       -1.442      1.202
## ------------------------------------------------
## 2-Less than $15,000        5          5      10
##                        0.500      0.500   0.100
##                        0.444     -0.371
## ------------------------------------------------
## 3-Less than $20,000        3          6       9
##                        0.333      0.667   0.090
##                       -0.359      0.299
## ------------------------------------------------
## 4-Less than $25,000        6          9      15
##                        0.400      0.600   0.150
##                       -0.060      0.050
## ------------------------------------------------
## 5-Less than $35,000        7          8      15
##                        0.467      0.533   0.150
##                        0.343     -0.286
## ------------------------------------------------
## 6-Less than $50,000        9         11      20
##                        0.450      0.550   0.200
##                        0.279     -0.233
## ------------------------------------------------
## 7-Less than $75,000        2          7       9
##                        0.222      0.778   0.090
##                       -0.880      0.733
## ------------------------------------------------
## 77-Don't Know              3          1       4
##                        0.750      0.250   0.040
##                        1.062     -0.885
## ------------------------------------------------
## 8-$75,000 of more          4          0       4
##                        1.000      0.000   0.040
##                        1.843     -1.536
## ------------------------------------------------
## 99-Refused                 0          1       1
##                        0.000      1.000   0.010
```



```
##                        -0.640      0.534
## ------------------------------------------------
## Total                      41         59     100
## ================================================
##
## Statistics for All Table Factors
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 = 14.09179      d.f. = 9      p = 0.119
```


Starting to figure out missing data

- In conducting analyses, we often do not want to include a small number of missing values as an actual category of the data, so recoding these to be considered missing is one possible strategy
- To retain the original coding in case it becomes useful to know the *Refused* from the *Don't know* categories, we can create a new variable.
- In previous workshops, we have used the car package's recode command to recode variables. Here, it doesn't quite work due to the use of double and single quotes in the code (i.e. "Don't" in "Don't Know" creates a problem)
- To use car::recode, you will have to rename that category of the variable
- In tidyverse, which is a collection of commonly used packages in R, there is a package called *dplyr*
- Within dplyr, there is a function called fct_recode to accomplish this change so one can continue using car::recode

Using fct_recode to recode, then use car

```
# use the tidyverse fct_recode to get rid of the apostrophe
# fct_recode means factor recode so is useful for categorical variables
library(tidyverse)
smokers$Income.frec <- fct_recode(smokers$Income,
                                  "77-Dont Know" = "77-Don't Know")
#check with levels()
levels(smokers$Income.frec)
```

```
##  [1] "1-Less than $10,000" "2-Less than $15,000" "3-Less than $20,000"
##  [4] "4-Less than $25,000" "5-Less than $35,000" "6-Less than $50,000"
##  [7] "7-Less than $75,000" "77-Dont Know"        "8-$75,000 of more"
## [10] "99-Refused"
```

```
# use the recode command in the car package to recode the updated variable
library(car)
smokers$Income.crec <- car::recode(smokers$Income.frec,
                                   "'77-Dont Know' = NA;
                                   '99-Refused' = NA")
```

Check the newly recoded variable

```
# check your work with a frequency table (uses descr package)
freq(smokers$Income.crec, plot=FALSE)
```

```
## smokers$Income.crec
##                       Frequency Percent Valid Percent
## 1-Less than $10,000          13      13        13.684
## 2-Less than $15,000          10      10        10.526
## 3-Less than $20,000           9       9         9.474
## 4-Less than $25,000          15      15        15.789
## 5-Less than $35,000          15      15        15.789
## 6-Less than $50,000          20      20        21.053
## 7-Less than $75,000           9       9         9.474
## 8-$75,000 of more             4       4         4.211
## NA's                          5       5
## Total                       100     100       100.000
```


Use dplyr for the whole process instead

- Instead of using two packages, the recode in dplyr is not subject to the apostrophe limitation
- Use dplyr::recode if you have both the car and dplyr packages loaded
- Try recoding to a new variable to preserve the original:

```
# recode income to a new variable
smokers$Income.drec <- dplyr::recode(smokers$Income,
                                     "77-Don't Know" = NA_character_,
                                     "99-Refused" = NA_character_)
freq(smokers$Income.drec, plot=FALSE)
```

```
## smokers$Income.drec
##                       Frequency Percent Valid Percent
## 1-Less than $10,000          13      13        13.684
## 2-Less than $15,000          10      10        10.526
## 3-Less than $20,000           9       9         9.474
## 4-Less than $25,000          15      15        15.789
## 5-Less than $35,000          15      15        15.789
## 6-Less than $50,000          20      20        21.053
## 7-Less than $75,000           9       9         9.474
## 8-$75,000 of more             4       4         4.211
## NA's                          5       5
## Total                       100     100       100.000
```


Examine the chi-squared again to see what changed

```
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |            Std Residual |
## |-------------------------|
##
## ================================================
##                          smokers$sex
## smokers$Income.drec     1-Male   2-Female    Total
## ------------------------------------------------
## 1-Less than $10,000          2         11       13
##                          0.154      0.846    0.137
##                         -1.403      1.146
## ------------------------------------------------
## 2-Less than $15,000          5          5       10
##                          0.500      0.500    0.105
##                          0.500     -0.408
## ------------------------------------------------
## 3-Less than $20,000          3          6        9
##                          0.333      0.667    0.095
##                         -0.316      0.258
## ------------------------------------------------
## 4-Less than $25,000          6          9       15
##                          0.400      0.600    0.158
##                          0.000      0.000
## ------------------------------------------------
## 5-Less than $35,000          7          8       15
##                          0.467      0.533    0.158
##                          0.408     -0.333
## ------------------------------------------------
## 6-Less than $50,000          9         11       20
##                          0.450      0.550    0.211
##                          0.354     -0.289
## ------------------------------------------------
## 7-Less than $75,000          2          7        9
##                          0.222      0.778    0.095
##                         -0.843      0.689
## ------------------------------------------------
## 8-$75,000 of more            4          0        4
##                          1.000      0.000    0.042
##                          1.897     -1.549
## ------------------------------------------------
## Total                       38         57       95
## ================================================
##
## Statistics for All Table Factors
##
## Pearson's Chi-squared test
```



```
## ------------------------------------------------------------
## Chi^2 = 11.53668      d.f. = 7      p = 0.117
```

Try it with the VBMI4CAT variable

Examine the BMI variable:

```
# examine the variable
freq(smokers$VBMI4CAT, plot=F)
```

```
## smokers$VBMI4CAT
##                Frequency Percent
##                        4       4
## Healthy weight        33      33
## Obese                 27      27
## Overweight            35      35
## Tall                   1       1
## Total                100     100
```

- There are 4 blank cells that are not being treated as missing
- This time it is not necessary to retain the original variable since we are just replacing blanks with NA values

Replacing the blank cells

- For replacing a single value with NA, dplyr has a handy function called na_if, like this:

```
# replace blank values with NA
smokers$VBMI4CAT <- na_if(smokers$VBMI4CAT,"")
# check your work
freq(smokers$VBMI4CAT, plot=F)
```

```
## smokers$VBMI4CAT
##                Frequency Percent Valid Percent
##                        0       0         0.000
## Healthy weight        33      33        34.375
## Obese                 27      27        28.125
## Overweight            35      35        36.458
## Tall                   1       1         1.042
## NA's                   4       4
## Total                100     100       100.000
```


Using droplevels to get rid of unused categories

- The droplevels command removes any categories with no observations:

```
# remove categories with no observations
smokers$VBMI4CAT<-droplevels(smokers$VBMI4CAT)
freq(smokers$VBMI4CAT, plot=F)
```

```
## smokers$VBMI4CAT
##                Frequency Percent Valid Percent
## Healthy weight        33      33        34.375
## Obese                 27      27        28.125
## Overweight            35      35        36.458
## Tall                   1       1         1.042
## NA's                   4       4
## Total                100     100       100.000
```

Perfecto!

You try it!

Fill in the blanks to recode the Refused category in marital_status to NA.

```
#fill in the blank to look at frequencies of the variable
________(smokers$marital_status)
# fill in the blank to add the command to change "9-Refused" to NA
smokers$marital_status <-  ________(smokers$marital_status,"9-Refused")

# fill in the blank to drop the unused categories in marital status
smokers$marital_status<-_________________(smokers$marital_status)

# check the work
freq(smokers$marital_status, plot=F)
```

You try it correct code

```
#look at the variable
freq(smokers$marital_status, plot=F)
```

```
## smokers$marital_status
##                                      Frequency Percent
## 1-Married                                   48      48
## 2-Divorced                                  21      21
## 3-Widowed                                   10      10
## 4-Separated                                  2       2
## 5-Never married                             12      12
## 6-A member of an unmarried couple            6       6
## 9-Refused                                    1       1
## Total                                      100     100
```

```
#change "9-Refused" to NA
smokers$marital_status <-  na_if(smokers$marital_status,"9-Refused")
#drop the unused categories in marital status
smokers$marital_status<-droplevels(smokers$marital_status)
#check your work
freq(smokers$marital_status, plot=F)
```

```
## smokers$marital_status
##                                      Frequency Percent Valid Percent
## 1-Married                                   48      48        48.485
## 2-Divorced                                  21      21        21.212
## 3-Widowed                                   10      10        10.101
## 4-Separated                                  2       2         2.020
## 5-Never married                             12      12        12.121
## 6-A member of an unmarried couple            6       6         6.061
## NA's                                         1       1
## Total                                      100     100       100.000
```


Understanding missing value patterns

- When building statistical models with methods like linear and logistic regression, sometimes you find that the number of observations your model is based on is far fewer than the number of observations in your data set
- This often means that there is one variable or a couple of variables with a lot of missing data
- In most analyses, observations with missing data are dropped
- If you do have one variable in a regression model that is missing a lot of data, it might be useful to drop the variable from the model so that the model is based on more data
- For example, often people do not give income on surveys
- If your income variable were missing 50% of the values, you might use something else in your model like education or insurance status to represent socioeconomic status
- Missing 10% or more is a good threshold to start paying attention to missing

Using VIM to figure out which variables are missing data

- To figure out which variables might be problematic, there is a package called VIM
- Open VIM and run the aggr command to check the patterns of missing data in the smokers data set
- To get the number of missing values for each variable, start by using the plot=FALSE option

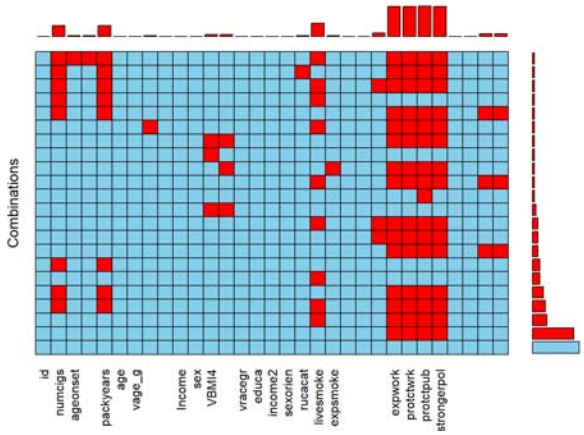
```
# load VIM and examine the number of missing values
library(VIM)
aggr(smokers, plot=FALSE)
```

```
##
##  Missings in variables:
##         Variable Count
##           numcigs    22
##         ageonset     1
##     yearssmoking     1
##        packyears    22
##   marital_status     1
##             VBMI4     4
##          VBMI4CAT     4
##           rucacat     1
##        livesmoke    27
##          expsmoke     1
##       nosmokecar     7
##           expwork    61
##         protctwrk    61
##         protctpub    62
##      strongerpol    61
##      Income.crec     5
##      Income.drec     5
```


Visualizing the missing data

Use the *combined=TRUE* option to visualize the missing data.

```
# use combined = TRUE to get a visual
aggr(smokers, combined=TRUE)
```



Interpreting the VIM output

- Variables missing the most data are the four variables indicating secondhand smoke exposure in the workplace, car, and public and the support for stronger secondhand smoke policy
- Including any of these variables in a model would result in more than 60 of the 100 observations in the data set to be dropped before the model was estimated
- Other possibly problematic values are numcigs, packyears, and livesmoke. When using these in analyses, nearly one-third of observations would be dropped
- This is important to consider, especially if people missing this information might be somehow different than people with complete data, which can bias results (MNAR = Missing Not At Random)

Strategy for identifying Missing Not At Random (MNAR)

- One method for figuring out if data has Missing Not At Random (MNAR) is to compare observations with missing values to observations without missing values on characteristics that are important to your research question/analysis
- For example, in a study to determine differences in heaviness of smoking for males and females, we might want to know if the numcigs and packyears variables are missing significantly more data for males compared to females (or vice-versa)
- We can test this using this procedure:
  - *recode numcigs into a new variable that is 'missing' for missing and 'not missing' for not missing using an if_else function in dplyr*
  - *compare the new variable to the sex variable using chi-squared*

Example of identifying MNAR

```
# recode numcigs to a missingNumcigs variable
smokers$missingNumcigs <- if_else(smokers$numcigs > 0,
                                  "not missing", "",
                                  missing = "missing")
#check the variable
freq(smokers$missingNumcigs, plot=F)
```

```
## smokers$missingNumcigs
##              Frequency Percent
## missing             22      22
## not missing         78      78
## Total              100     100
```


Examine missingness by sex using chi-squared

```
# examine the relationship of missingNumcigs to
# the sex variable to see if missingNumcigs differs by sex
CrossTable(smokers$sex, smokers$missingNumcigs, chisq=TRUE, prop.c=F,
           prop.t=F, prop.chisq=F, sresid=T)
```

```
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |            Std Residual |
## |-------------------------|
##
## ============================================
##                 smokers$missingNumcigs
## smokers$sex    missing   not missing   Total
## --------------------------------------------
## 1-Male              11            30      41
##                  0.268         0.732   0.410
##                  0.659        -0.350
## --------------------------------------------
## 2-Female            11            48      59
##                  0.186         0.814   0.590
##                 -0.550         0.292
## --------------------------------------------
## Total               22            78     100
## ============================================
##
## Statistics for All Table Factors
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 = 0.9444462      d.f. = 1      p = 0.331
##
## Pearson's Chi-squared test with Yates' continuity correction
## ------------------------------------------------------------
## Chi^2 = 0.5276796      d.f. = 1      p = 0.468
```

Interpreting the MNAR test

- The chi-squared value is not significant, so missing values and sex are not associated
  - *Numcigs data is equally likely to be missing or non-missing for males and females*
- Use a t-test if the predictor of interest is continuous
- If you do find significant results, this should be noted when you interpret your results as a possible source of bias in the data
  - *Example of noting this: "Missing values were statistically significantly higher on the xxxx variable for males compared to females."*

Other types of missing data (not MNAR)

- There are a number of ways to deal with missing data as long as it is not MNAR. Several are described in the *Fischetti* text.
- Nearly all of these methods have major limitations.
- It is usually best to analyze the complete cases and describe the missingness well so that your reader understands your data.
- This is the default in most statistical programs, including R.

You try it!

Fill in the blanks to complete the code for adding a new variable differentiating missing and not missing for packyears and determining if there is a significant association between missing packyears data and sex.

```
# recode packyrs to a missingPackyrs variable
smokers$missingPackyrs <- _________(smokers$packyears > 0,
                                   "not missing", "",
                                   ________ = "missing")
#check the variable
________(smokers$missingPackyrs, plot=F)

# check the recoding and association with sex
CrossTable(____________, _____________, chisq=TRUE, prop.c=F,
           prop.t=F, prop.chisq=F, sresid=T)
```


Correct code for you try it

```
# recode packyrs to a missingPackyrs variable
smokers$missingPackyrs <- if_else(smokers$packyears > 0,
                                  "not missing", "",
                                  missing = "missing")
#check the variable
freq(smokers$missingPackyrs, plot=F)
```

```
## smokers$missingPackyrs
##              Frequency Percent
## missing             22      22
## not missing         78      78
## Total              100     100
```

```
# check the recoding and association with sex
CrossTable(smokers$sex, smokers$missingPackyrs,
           chisq=TRUE, prop.c=F,
           prop.t=F, prop.chisq=F, sresid=T)
```

```
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |            Std Residual |
## |-------------------------|
##
## ============================================
##                 smokers$missingPackyrs
## smokers$sex    missing   not missing   Total
## --------------------------------------------
## 1-Male              11            30      41
##                  0.268         0.732   0.410
##                  0.659        -0.350
## --------------------------------------------
## 2-Female            11            48      59
##                  0.186         0.814   0.590
##                 -0.550         0.292
## --------------------------------------------
## Total               22            78     100
## ============================================
##
## Statistics for All Table Factors
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 = 0.9444462      d.f. = 1      p = 0.331
##
## Pearson's Chi-squared test with Yates' continuity correction
```

```
## ------------------------------------------------------------
## Chi^2 = 0.5276796      d.f. = 1      p = 0.468
```


Recoding continuous variables to categorical

- Research has demonstrated that heavy smokers have different health problems than light or moderate smokers
- Say you were interested in examining the characteristics significantly associated with heavy smoking, where heavy smoking is 1 or more packs of cigarettes per day (20 or more cigarettes)
- We do not have this variable in the smokers data set, but we can create a new variable called *heavy* from the numcigs variable using the same if_else function

```
#recode numcigs variable to a heavy smoker variable, yes for a pack or more, no for less
smokers$heavy <- if_else(smokers$numcigs >= 20, "yes", "no")

#check your work
freq(smokers$heavy, plot=F)
```

```
## smokers$heavy
##       Frequency Percent Valid Percent
## no           29      29         37.18
## yes          49      49         62.82
## NA's         22      22
## Total       100     100        100.00
```


Checking new variable for correct recode

- When recoding, it is important to check your new variable to make sure it is coded correctly
- The new variable, *heavy*, should have values of *yes* when numcigs is at least 20 and should have values of *no* when numcigs is less than 20
- Check it using the by command with the summary command

```
#get the summary for numcigs for the two heavy categories
by(smokers$numcigs, smokers$heavy, summary)
```

```
## smokers$heavy: no
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5.00   10.00   10.00   10.48   10.00   18.00
## ------------------------------------------------------------
## smokers$heavy: yes
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   20.00   20.00   20.00   23.37   25.00   40.00
```

Looks good!

Recoding into three categories

- For three categories, you can use multiple ifelse statements with *mutate* in dplyr.
- The *mutate* function is used to create new variables in a data set with flexibility in defining the new variable
- This one is slightly different because it uses the base r ifelse function
- Here, we will recode the ageonset variable into three categories: under 18, 18, over 18

```
#create a new variable called agecat with three categories in the smokers data
smokers <- mutate(smokers, ageOnsetCat = ifelse(ageonset < 18,
                                                "Under 18",
                ifelse(ageonset == 18, "18",
                ifelse(ageonset > 18, "Over 18", NA))))
#check the variable
freq(smokers$ageOnsetCat, plot = F)
```

```
## smokers$ageOnsetCat
##          Frequency Percent Valid Percent
## 18               9       9         9.091
## Over 18         36      36        36.364
## Under 18        54      54        54.545
## NA's             1       1
## Total          100     100       100.000
```

```
#check your work
by(smokers$ageonset, smokers$ageOnsetCat, summary)
```

```
## smokers$ageOnsetCat: 18
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      18      18      18      18      18      18
## ------------------------------------------------------------
## smokers$ageOnsetCat: Over 18
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   19.00   20.00   21.50   23.69   25.00   45.00
## ------------------------------------------------------------
## smokers$ageOnsetCat: Under 18
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   11.00   13.25   15.00   14.74   16.00   17.00
```


4. Recoding categorical variables to fewer categories

- Sometimes there are too few people in a category for analysis to be useful
- Basing statistical estimates and results on 1 or 2 people is unreliable at best
- Instead, categories with small numbers of people in them are often combined before analysis
- Examine the employment variable from the smokers data set

```
##              1-Employed for wages                 2-Self-employed
##                                47                               8
## 3-Out of work for more than 1 year 4-Out of work for less than 1 year
##                                 5                               4
##                     5-A homemaker                    6-A student
##                                 5                               5
##                         7-Retired                  8-Unable to work
##                                10                              16
```


Recoding employment into two categories

- Several of the categories represent people not employed outside the home for various reasons
- While there may be important differences between students, homemakers, retirees, and others, categories with 4 or 5 people in them are not useful for analyses
- Instead, combine those not working for compensation into a single category
- At the same time, combine employed for wages and self-employed into one category

```
#recode employment into a variable with two categories
smokers$employed <- dplyr::recode(smokers$employment,
                                  "1-Employed for wages" = "yes",
                                  "2-Self-employed" = "yes",
                                  "3-Out of work for more than 1 year" = "no",
                                  "4-Out of work for less than 1 year" = "no",
                                  "5-A homemaker" = "no",
                                  "6-A student" = "no",
                                  "7-Retired" = "no",
                                  "8-Unable to work" = "no")
#check your work
freq(smokers$employed, plot=F)
```

```
## smokers$employed
##       Frequency Percent
## yes          55      55
## no           45      45
## Total       100     100
```


Checking the new variable against the old variable

One option to check the recoding when the old and new variables are both categorical is using a table with the old variable and the new variable:

```
#check the recode against the old variable
table(smokers$employment, smokers$employed)
```

```
##
##                                    yes no
##   1-Employed for wages              47  0
##   2-Self-employed                    8  0
##   3-Out of work for more than 1 year 0  5
##   4-Out of work for less than 1 year 0  4
##   5-A homemaker                      0  5
##   6-A student                        0  5
##   7-Retired                          0 10
##   8-Unable to work                   0 16
```

Looks good!

You try it!

Fill in the blanks to complete the code for recoding the marital status variable into two categories, one for married and one for not married. Check your recoding using the table command.

```
## smokers$marital_status
##                                    Frequency Percent Valid Percent
## 1-Married                                 48      48        48.485
## 2-Divorced                                21      21        21.212
## 3-Widowed                                 10      10        10.101
## 4-Separated                                2       2         2.020
## 5-Never married                           12      12        12.121
## 6-A member of an unmarried couple          6       6         6.061
## NA's                                       1       1
## Total                                    100     100       100.000
```

```
#recode the marital status variable in a variable with two categories
smokers$married <- ____________(smokers$marital_status,
                                "_________" = "yes",
                                "_________" = "no",
                                "_________" = "no",
                                "_________" = "no",
                                "_________" = "no",
                                "_________" = "no")
#check your work
____________(smokers$married, plot=F)
#check your new variable against the old variable
____________(smokers$marital_status, smokers$married)
```


Correct code for you try it

```
#recode the marital status variable in a variable with two categories
smokers$married <- dplyr::recode(smokers$marital_status,
                                 "1-Married" = "yes",
                                 "2-Divorced" = "no",
                                 "3-Widowed" = "no",
                                 "4-Separated" = "no",
                                 "5-Never married" = "no",
                                 "6-A member of an unmarried couple" = "no")
#check your work
freq(smokers$married, plot=F)
```

```
## smokers$married
##        Frequency Percent Valid Percent
## yes           48      48         48.48
## no            51      51         51.52
## NA's           1       1
## Total        100     100        100.00
```

```
#check your new variable against the old variable
table(smokers$marital_status, smokers$married)
```

```
##
##                                    yes no
##   1-Married                         48  0
##   2-Divorced                         0 21
##   3-Widowed                          0 10
##   4-Separated                        0  2
##   5-Never married                    0 12
##   6-A member of an unmarried couple  0  6
```


Identifying and correcting out-of-range values and incorrect data types

- Occasionally you will come across a data entry error or miscoded value in your data set
- Sometimes the value can be corrected, and other times the value will have to be replaced with NA
- To find out-of-range values you can use the summary command and review the values for nonsense or, if you suspect an out-of-range value, you can use the assertr package to check
- For example, the age variable in the smokers data should not go below 18 or above 100.
- Check to see if all the age values are in this range

```
#loading assertr package
library(assertr)
#check to see if any values in age fall outside 18-100
assert(smokers, within_bounds(18,100), age)
```


Correcting out-of-range values

- The code above results in an error message:

```
## Column 'age' violates assertion 'within_bounds(18, 100)' 2 times
##      verb redux_fn              predicate column index value
## 1 assert       NA within_bounds(18, 100)    age    82   199
## 2 assert       NA within_bounds(18, 100)    age   100   620
```

```
## Error: assertr stopped execution
```

- The error message is actually useful this time, telling us that the variable goes outside that range 2 times and gives us the violations of 199 and 620
- Once this has an error, we can review the data set to find these out-of-range values for age: 620 and 199

Making the corrections

- Correct these values to NA since we don't have the original data set to compare to

```
#replace 199 with NA
smokers$age <- na_if(smokers$age, 199)
#replace 620 with NA
smokers$age <- na_if(smokers$age, 620)
#check your work
summary(smokers$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   18.00   34.25   46.50   46.83   56.00   84.00       2
```

In the past we have used subset and dropped these observations from the data set, which means any statistics we do on the data will not include anything for these two individuals on any variable in the data set. Changing the values to NA allows you to retain more of your data.

Checking for incorrect categories listed for a factor

- Checking for incorrect categories listed for a factor is also done with the assert command
- For example, the VBMI4CAT variable should have three categories: *Healthy weight*, *Obese*, *Overweight*
- Using assert we can see if there are any additional incorrect categories

```
#check to see if there are any unwanted categories in VBMI4CAT
assert(smokers, in_set("Healthy weight","Obese",
                       "Overweight"), VBMI4CAT)
```


Removing incorrect categories

The incorrect category, *Tall* can be removed using the same na_if command as above:

```
#replace Tall with NA
smokers$VBMI4CAT <- na_if(smokers$VBMI4CAT, "Tall")
#check your work
summary(smokers$VBMI4CAT)
```

```
## Healthy weight          Obese     Overweight           Tall           NA's
##             33             27             35              0              5
```

Whoops, forgot to delete unused categories:

```
# remove categories with no observations
smokers$VBMI4CAT<-droplevels(smokers$VBMI4CAT)
#check your work
summary(smokers$VBMI4CAT)
```

```
## Healthy weight          Obese     Overweight           NA's
##             33             27             35              5
```


Changing data types

- Sometimes data types that should be characters or numeric variables are assigned as factors by R
- Checking a data type uses the *class* command and changing the data type requires one of several *as* commands
- For example, numcigs would best be described as integer or numeric
- Use the class command and as commands to check and change data types

```
#check class of numcigs
class(smokers$numcigs)
```

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

```
#change numcigs to a factor
smokers$numcigs<-as.factor(smokers$numcigs)
#check the change
class(smokers$numcigs)
```

```
## [1] "factor"
```

```
#change numcigs to a numeric
smokers$numcigs<-as.numeric(smokers$numcigs)
#check the change
class(smokers$numcigs)
```

```
## [1] "numeric"
```

You try it!

Fill in the blanks to check VBMI4 and nosmokecar for out-of-range values and variable class. Fix what you find.

Note: * BMI for adults ranges from 18 to 50, with a small percent of adults below or above this. * The categories for nosmokecar should be "Allowed" and "Not allowed"

```
#check class and look at the VBMI4 variable
_______(smokers$VBMI4)
smokers$VBMI4
#use assert to determine any out-of-range values
assert(smokers, ___________, VBMI4)
```

```
#replace the out-of-range value(s) with NA
smokers$VBMI4 <- na_if(smokers$VBMI4, ______)
#check your work
summary(smokers$VBMI4)
```

```
#check class and look at the nosmokecar variable
_________(smokers$nosmokecar)
smokers$nosmokecar
#use assert to determine any out-of-range values
assert(smokers, ___________, nosmokecar)
```

```
#replace Tall with NA
smokers$nosmokecar <- na_if(smokers$nosmokecar, _______)
#check your work
summary(smokers$nosmokecar)
# remove categories with no observations
smokers$nosmokecar<-_________(smokers$nosmokecar)
#check your work
summary(smokers$nosmokecar)
```


Correct code for you try it

```
#check class and look at the VBMI4 variable
class(smokers$VBMI4)
smokers$VBMI4
#use assert to determine any out-of-range values
assert(smokers, within_bounds(0,100), VBMI4)
```

```
#replace the out-of-range value(s) with NA
smokers$VBMI4 <- na_if(smokers$VBMI4, 99999)
#check your work
summary(smokers$VBMI4)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   15.66   23.25   26.58   27.58   30.25   56.23       5
```

```
#check class and look at the nosmokecar variable
class(smokers$nosmokecar)
smokers$nosmokecar
#use assert to determine any out-of-range values
assert(smokers, in_set("Not allowed","Allowed"), nosmokecar)
```

```
#replace yn with NA
smokers$nosmokecar <- na_if(smokers$nosmokecar, "yn")
#check your work
summary(smokers$nosmokecar)
```

```
##     Allowed Not allowed          yn        NA's
##          82          10           0           8
```

```
# remove categories with no observations
smokers$nosmokecar<-droplevels(smokers$nosmokecar)
#check your work
summary(smokers$nosmokecar)
```

```
##     Allowed Not allowed        NA's
##          82          10           8
```


Loading data from other software packages

The most common types of data files you will encounter in public health and social work are:

- R data (.Rdata)
- Excel data (.xls, .xlsx)
- Stata data (.dta)
- SPSS data (.sav)
- SAS data (.sas7bdat)
- Comma separated values (.csv)

Some of these are easy to load, like *Rdata*, which requires just the load command:

- dataset <- load("nameOfDataSet.Rdata")

Excel files require the *readxl* package to be installed and opened:

- library(readxl)
- dataset <- read_xls("nameOfDataSet.xls",header=TRUE)}

Loading data from Stata, SPSS, and SAS

- Stata and SPSS data formats require the *foreign* package
- SAS data format requires the *sas7bdat* package
- Install them to try loading
- Note: The *haven* package in tidyverse can also accomplish this goal in a very similar way
- Google "tidyverse haven package" for more information for how this package works

Stata data

```
# load foreign package
library(foreign)
# read the Stata data file
schools<-read.dta("https://stats.idre.ucla.edu/stat/stata/examples/ara/anscombe.dta")
# get summary of the data
summary(schools)
```

```
##     state              educspnd         income         prop18
##  Length:51          Min.   :112.0   Min.   :2081   Min.   :326.2
##  Class :character   1st Qu.:165.0   1st Qu.:2786   1st Qu.:342.1
##  Mode  :character   Median :192.0   Median :3257   Median :354.1
##                     Mean   :196.3   Mean   :3225   Mean   :358.9
##                     3rd Qu.:228.5   3rd Qu.:3612   3rd Qu.:369.1
##                     Max.   :372.0   Max.   :4425   Max.   :439.7
##     propurb
##  Min.   : 322.0
##  1st Qu.: 552.5
##  Median : 664.0
##  Mean   : 664.5
##  3rd Qu.: 790.5
##  Max.   :1000.0
```


SPSS data

```
# read the SPSS data file
right2work<-read.spss("https://stats.idre.ucla.edu/wp-content/uploads/2016/02/p005.sav", to.data
# get summary of the data
summary(right2work)
```

```
##                  CITY          COL              PD
##  Atlanta           : 1   Min.   : 99.0   Min.   :  43.0
##  Austin            : 1   1st Qu.:170.8   1st Qu.: 302.0
##  Bakersfield       : 1   Median :205.5   Median : 400.0
##  Baltimore         : 1   Mean   :223.6   Mean   : 780.2
##  Baton Rouge       : 1   3rd Qu.:266.5   3rd Qu.: 963.8
##  Boston            : 1   Max.   :381.0   Max.   :6908.0
##  (Other)           :32
##      URATE             POP              TAXES          INCOME
##  Min.   : 6.50   Min.   : 162304   Min.   :3965   Min.   : 782
##  1st Qu.:17.82   1st Qu.: 497050   1st Qu.:4620   1st Qu.:3110
##  Median :24.05   Median :1408054   Median :4858   Median :4865
##  Mean   :24.22   Mean   :2040736   Mean   :4903   Mean   :4709
##  3rd Qu.:30.00   3rd Qu.:2355462   3rd Qu.:5166   3rd Qu.:6082
##  Max.   :39.20   Max.   :9561089   Max.   :6404   Max.   :8392
##
##       RTWL
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.2632
##  3rd Qu.:0.7500
##  Max.   :1.0000
##
```


SAS data

```
# load the sas7bdat package
library(sas7bdat)
# read the SAS data file
alcohol<-read.sas7bdat("http://www.principlesofeconometrics.com/sas/alcohol.sas7bdat")
# get summary of the data
summary(alcohol)
```

```
##      ADULTS           KIDS           INCOME          CONSUME
##  Min.   :1.000   Min.   :0.000   Min.   :  12.0   Min.   :0.000
##  1st Qu.:2.000   1st Qu.:0.000   1st Qu.: 295.0   1st Qu.:1.000
##  Median :2.000   Median :0.000   Median : 562.5   Median :1.000
##  Mean   :2.012   Mean   :0.722   Mean   : 649.5   Mean   :0.766
##  3rd Qu.:2.000   3rd Qu.:1.000   3rd Qu.: 887.5   3rd Qu.:1.000
##  Max.   :6.000   Max.   :5.000   Max.   :3846.0   Max.   :1.000
```


Practice activity

There is no challenge for this week, but these skills will come in handy on the final exam and in real life. To practice, read the scenario below and try the tasks below it:

You are examining health department expenditures predicted by a number of variables included in a small data set. A colleague of yours was working with the data set when her cat walked across the keyboard changing several values and then saving the data set.

- Use foreign to import the SPSS data set saved at: https://tinyurl.com/y84489qw
- Examine the codebook for the data saved here: https://drive.google.com/file/d/0B9UP9lGaNL2hYnd4NXlnb09FV0U/view?usp=sharing
- Conduct the tasks on the next slide

Tasks

- Identify and correct any value(s) that appear to be out-of-range or representing a missing value for any of the variables in the data set so that they come up as missing in analyses. Drop levels to clean up any factor variables if needed (Hint: R only recognizes NA as missing).
- You are thinking about studying health departments in categories of people served. Recode the numserved into a categorical variable called servedCat that has 4 levels: <25000; 25000-99999; 100000-499999; 500000+.
- Create a variable that has the value of 1 for missing and 0 for not missing on the cancerscreen variable. Use the appropriate bivariate test to compare mean expenditures for health departments missing and not missing cancer screen data. Interpret your results including an assessment of whether the cancerscreen variable is MNAR or not.
- Create a variable that has the value of 1 for missing and 0 for not missing on the revenues variable. Use the appropriate bivariate test to compare mean expenditures for health departments missing and not missing revenues data. Interpret your results including an assessment of whether the revenues variable is MNAR or not.