Homework 2: Data Manipulation and Contingency Tables

Updated: February, 2016

Table of Contents

Data Manipulation

A central component of data analysis is the manipulation of those data to meet certain analytical requirements. For example, data may be miscoded, or mistyped in the sense that they are stored as characters when in fact they are integers. Data are often missing, or extraneous data may be present. In formal database manipulations, the process of “extract-transform-load” or ETL is performed to create useable subsets of data. For this homework, you will need to identify erroneous items, recode items, create new values, subset data by frequency of occurrence, and generally prepare the data for analysis.

Null, Erroneous, and Superfluous Values

Null values generally indicate data that is unknown, not applicable, or that the data will be added later [1]. They are often represented by a designated character set or value, or occasionally missing altogether (look up `is.na()` in R for more info). For designated values, a number that is outside the normal range of the data it’s representing is generally used. For example, a dataset that measures length would never contain a number less than 0, so negative number could be used to represent null values. Occasionally missing values or NAs are “errors of omission” which can be handled by masking (`na.rm=TRUE` or removing from analysis), or through more sophisticated statistical techniques for missing values, such as mean replacement. Erroneous values often include “errors of commission”, which are most often transcription errors such as typos or misspellings. These can often remedied through manipulation. Occasionally superfluous or unneeded data are present as either columns with extraneous data unrelated to the research, or row values that are outside the scope of interest.

Homework Exercise (4 Points total)

These exercises rely on the lab data found in the resource section. Download `riparian_survey.csv` from UCMCROPS and save to your R workspace.

Your document should have the following sections, and provide written explanations formatted in RMarkdown that explains your code, output and graphics in the following format:

NAME
CLASS
DATE

Homework Assignment 2

Objective Statement: [What are you trying to accomplish?]

Methods: [In general terms, what analyses are you doing?]

Data: [What are the data and where did they come from?]

Code: [In specific terms, what is the code that was used to conduct the analysis?]

Results: [What do the results show? Numerical evidence and graphic evidence are required.]

Discussion: [What do the results mean?]

Limitations: [What are the limitations, caveats, and assumptions of the analysis?]

OBJECTIVE

You have been asked to analyze a dataset of field measurements and observations taken at various project sites throughout northern California that are intended to estimate aboveground carbon stocks in riparian areas. Some of these project sites are likely more productive than others, and you are being asked to determine which sites have more carbon stocks than the others. As a quick back of the envelope exercise you decided that you could develop a report showing the location of the project sites testing the assumption that the project sites are independent of each other in the frequency of trees present in the sample plots. There are a bunch of different species, and so you decided that a subsample of the most frequently occurring genera (*look this term up if you don’t know what it means*) will suffice for this preliminary assessment.

Step 1 - Data Examination, Correction, and Summary

Load `riparian_survey.csv.` Rename to a more suitable data frame object name for quicker analysis, such as `ripdata` (eg. `ripdata <- read.csv(“riparian_survey.csv”)`). Examine columns (variables) using `str(ripdata)` and moments using `summary(ripdata)`. Which of the columns appear relevant to your analysis? There is a ProjectID that looks like the name, but not a project code. There also appear to be some geographic coordinates, species identifiers, and some measurements.

Since the analysis will be by “Project”, create a new column that has a unique project code for each identifier. Using `levels(ripdata$ProjectID)`, use the following code list for each corresponding project site in turn: `c("COSRP", "HEROW","NAPSO", "SACTO").`

```
ripdata$ProjCode <- to something #what would you use? Use the
following as a guide:
```

```
#one way is to go value by value …
ripdata$ProjCode[which(ripdata$ProjectID==”Napa_Sonoma”)] <- “NAPSO”

#or use “match”
oldvals <- levels(ripdata$ProjectID)
newvals <- factor(c("COSRP", "HEROW","NAPSO", "SACTO"))
ripdata$ProjCode <- newvals[match(ripdata$ProjectID, oldvals)]

#for another _easy_ way, load the “plyr” package and run mapvalues
ProjCode <- mapvalues(ProjectID, from=levels(ProjectID), to=codelist)
```

Are there null, erroneous, or superfluous data? Let’s handle each in turn by using `summary()` and `levels()` to examine the data. Remove rows that contain null, erroneous, or superfluous values.

Also examine the species codes and name columns. Delete rows that have values similar to “unknown”, “dead wood”, and “not recorded” (There may be more than what is listed here!).

Create summaries by project site code using the `aggregate()` function. For example,

```
ProjLoc <- aggregate(cbind(Longitude,Latitude) ~ ProjCode,
data=ripdata, mean)
```

Step 2 - Species Selection and Enumeration

For this study, we want to concentrate on the top 5 most frequent genera, but our data are listed only by species. Create a new column that takes the first “word” or epithet from the `ripdata$SpeciesVarietalName` and saves as `ripdata$Genus`. Use the following as a guide.

```
#Split SpeciesVarietalName column as a string and extract just the
first word. Save to a new column
ripdata$Genus =
sapply(strsplit(as.character(ripdata$SpeciesVarietalName), " "),
"[[", 1)
```

Create a frequency table for species (using the genus values you just created) (what kind of table is this?). Use this table to decide which species to keep.

Take a subset of the new dataframe you found from step 1, keeping only the 5 most frequent species.

Step 3 - Test for Independence

Test the subset for trees species independence based on observation counts within genera by project site (`Genus ~ ProjCode`). Create a two-way table for the new subset and test for independence using the Chi-square test `chisq.test()`.

Step 4 - Create a Map of Project Sites

Plot the mean coordinate locations of the project sites (Longitude (x), Latitude(y)) with different symbols for each site. Advanced users are encouraged to display the State of California in the map.

```
# for example, plot(ProjLoc$x, ProjLoc$y)

# though you can make it look much nicer, like:

# library(OpenStreetMap)
# library(rgdal)
# map <- openmap(c(max(y)+1,min(x)-1), c(min(y)-1,max(x)+1))
# plot(map)

# How would you plot the points on the map? Try sapply(1:nrow(),…)
```

Step 5 - Save dataframe

Save your new dataframe to a new csv file. Similar to the `read.csv` we also have a `write.csv` function (see ?write.csv for its usage).

Step 6 - Commit to your git repository

Resources

Data

- `riparian_survey.csv` on CATCOURSES (HOMEWORK→HOMEWORK @)

References

- Kabacoff, Robert. "Chapter 9: Analysis of variance." *R in Action: Data Analysis and Graphics with R*. Shelter Island, NY: Manning ;, 2011. 219-245. Print.

Citations

[1] http://technet.microsoft.com/en-us/library/ms191504(v=sql.105).aspx

Keywords `aggregate, plyr, mapvalues, xtabs, chisq.test`