Keeping a Project Together

Paul E. Johnson[1] [2]

[1] Department of Political Science

[2] Center for Research Methods and Data Analysis, University of Kansas

August 16, 2017

Overview

1. Workflow
2. Separate places for separate things
3. Example with Dog Data
4. Take-Aways

This presentation is part of a folder created for training purposes in CRMDA. The data and output files to which it refers are available in the zip package, “projects.zip” that includes this presentation as well as the data and output folders. Its attached to the Event listing on http://crmda.ku.edu/events

Outline

1 Workflow

2 Separate places for separate things

3 Example with Dog Data

4 Take-Aways

Understanding each other now and in the future

Replication is the priority

- Replication across teammates (mutually understandable projects)
- Replication across time; can understand & repeat work in future
- Can repeat work from top to bottom **EXACTLY**

Clear directory and file names!

- Directories separate work into understandable pieces
- Even Apple now recommends against spaces and special characters in file names!
  Cross-platform filename best practices and conventions
- Names are chosen by experience and testing

Outline

1. Workflow
2. Separate places for separate things
3. Example with Dog Data
4. Take-Aways

Novices

- A novice throws all files, for all of their projects, into “My Documents”. (Or similar, such as “/users/your-name-here/Documents”).
- Smarter novices will create subdirectories for courses

  ```
  Documents/hist101
  Documents/eng101
  ```

  But everything for hist101 will fall into that one (possibly giant) folder.

Forward-Looking Novice

- Separate folders for separate projects (better!)

```
Documents/hist101/exam_1
Documents/hist101/exam_2
Documents/hist101/termpaper_1
Documents/hist101/final
Documents/eng101/bookreport_1
Documents/eng101/bookreport_2
Documents/eng101/bookreport_3
```

Not a Novice? Folders Within Projects

A project has subdirectories

data: Data in “fresh” “pristine” “unaltered state”. Never altered. Read Only

workingdata: “Recoded”, “Cleaned”, “Subsetted”

output: Graphs and tables

lit: reading material

writeup: things we write

tmp: trash

The “working directory” is usually a software-specific folder, as in

R: R code files, usually suffixed “*.R”

Stata: Stata files, usually suffixed “*.do”

SAS: usually suffixed “*.sas”

Mplus: “*.inp”

Outline

1 Workflow

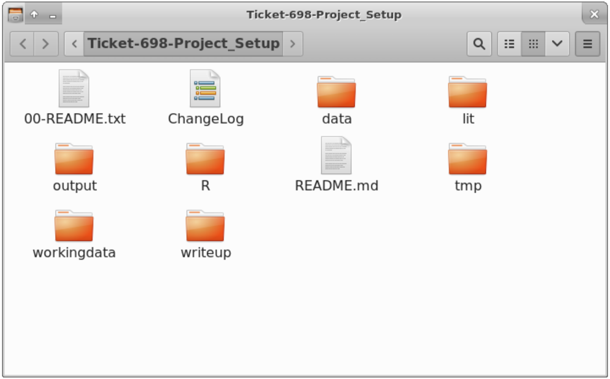
2 Separate places for separate things

3 Example with Dog Data

4 Take-Aways

Standard Project folder

- Here is the project directory



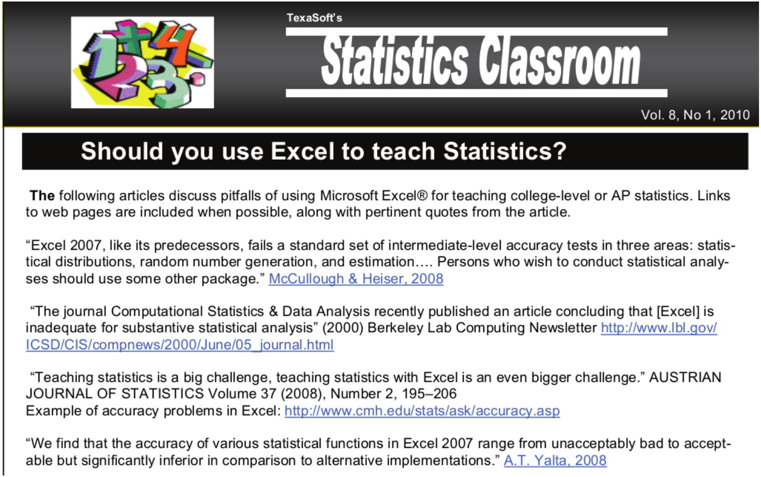
- I wrote a little program that creates a set of directories automatically. (The kutils package for R)

Copy in the Doggie Data File

- We are provided with this original data file by the “client”: WorkingDogDataCleaned.xlsx
- Probably created by MS Excel
- Copy that file into the directory named

```
project
    data
```

I'm Luke Warm on Excel



TexaSoft's

Statistics Classroom

Vol. 8, No 1, 2010

Should you use Excel to teach Statistics?

The following articles discuss pitfalls of using Microsoft Excel® for teaching college-level or AP statistics. Links to web pages are included when possible, along with pertinent quotes from the article.

"Excel 2007, like its predecessors, fails a standard set of intermediate-level accuracy tests in three areas: statistical distributions, random number generation, and estimation…. Persons who wish to conduct statistical analyses should use some other package." McCullough & Heiser, 2008

"The journal Computational Statistics & Data Analysis recently published an article concluding that [Excel] is inadequate for substantive statistical analysis" (2000) Berkeley Lab Computing Newsletter http://www.lbl.gov/ICSD/CIS/compnews/2000/June/05_journal.html

"Teaching statistics is a big challenge, teaching statistics with Excel is an even bigger challenge." AUSTRIAN JOURNAL OF STATISTICS Volume 37 (2008), Number 2, 195–206
Example of accuracy problems in Excel: http://www.cmh.edu/stats/ask/accuracy.asp

"We find that the accuracy of various statistical functions in Excel 2007 range from unacceptably bad to acceptable but significantly inferior in comparison to alternative implementations." A.T. Yalta, 2008

I Like R because . . .

- R is free and open source (www.r-project.org)
- Worldwide community seems to thrive on idea that R is the “lingua franca of statistics”
- 1000s of contributed addon packages, including rockchalk and kutils from KU!
- CRMDA offers Summer “Stats Camp” workshops on R, Stata, and other computer software (http://crmda.ku.edu/statscamp)

Use R code to Import the XLSX file

- We have great luck lately with the R addon package named `openxlsx` by Alexander Walker.
- The R commands to import

```
library(openxlsx)
fn <- "../data/WorkingDogDataCleaned.xlsx" ## The
    filename
dogs <- read.xlsx(fn, sheet = 1)
owners <- read.xlsx(fn, sheet = 2)
```

Whoa. What's that "../" thing?

- It is a Relative Path. It works on all operating systems.
- "data" and "R" folders are on same level, within the project

```
project
    data
    R
```

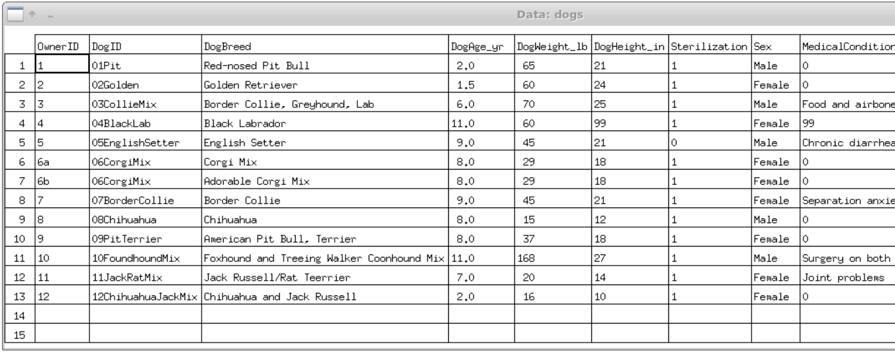
- Code inside the R folder wants files in the data folder, so it asks for "../data/WorkingDogDataCleaned.xslx".
- Similar logic, can write output to "../output" folder.

What has to Happen

- Inspect each column
- Apply corrections
- save result into the "../workingdata" folder.

When the Data is Open in the R session, we investigate

- Many R functions exist to “find out what we have”.
- View(dogs) gives a spread-sheet style view



	OwnerID	DogID	DogBreed	DogAge_yr	DogWeight_lb	DogHeight_in	Sterilization	Sex	MedicalConditior
1	1	01Pit	Red-nosed Pit Bull	2.0	65	21	1	Male	0
2	2	02Golden	Golden Retriever	1.5	60	24	1	Female	0
3	3	03CollieMix	Border Collie, Greyhound, Lab	6.0	70	25	1	Male	Food and airbone
4	4	04BlackLab	Black Labrador	11.0	60	99	1	Female	99
5	5	05EnglishSetter	English Setter	9.0	45	21	0	Male	Chronic diarrhea
6	6a	06CorgiMix	Corgi Mix	8.0	29	18	1	Female	0
7	6b	06CorgiMix	Adorable Corgi Mix	8.0	29	18	1	Female	0
8	7	07BorderCollie	Border Collie	9.0	45	21	1	Female	Separation anxie
9	8	08Chihuahua	Chihuahua	8.0	15	12	1	Male	0
10	9	09PitTerrier	American Pit Bull, Terrier	8.0	37	18	1	Female	0
11	10	10FoundhoundMix	Foxhound and Treeing Walker Coonhound Mix	11.0	168	27	1	Male	Surgery on both
12	11	11JackRatMix	Jack Russell/Rat Teerrier	7.0	20	14	1	Female	Joint problems
13	12	12ChihuahuaJackMix	Chihuahua and Jack Russell	2.0	16	10	1	Female	0
14									
15									

- I wrote a new one called peek (in kutils) that gives a quick look at all of the variables.

When the Data is Open in the R session, we investigate ...

- Check the output folder that comes with this presentation!

I wrote a new function called ”peek” in July, 2016

```
library(kutils)
peek(dogs, file = "../output/peek_dog.pdf", freq
    = TRUE, sort = FALSE, height = 5, width = 8)
```

```
[1] "OwnerID"              "DogID"                "DogBreed"
     "DogAge_yr"            "DogWeight_lb"         "DogHeight_in"
     "Sterilization"
[8] "Sex"                  "MedicalConditions"
```

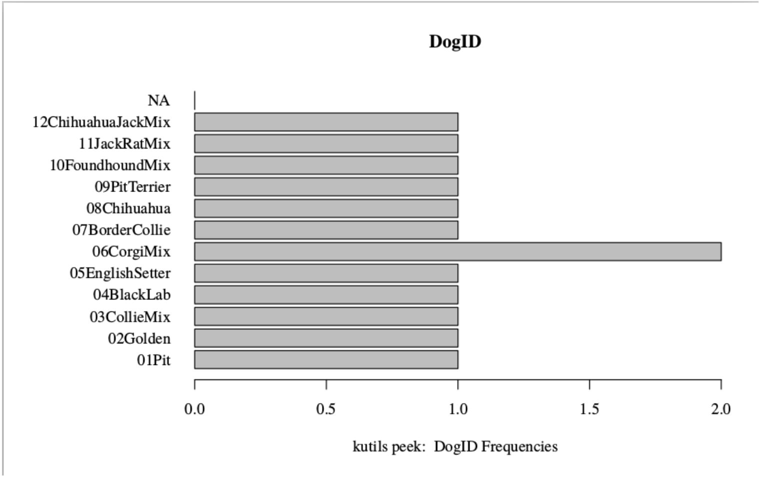
- Puts output “over there” in “../output/peek_dog.pdf” file.
- Check in the output directory, look at peek_dog.pdf

peek output has one image per variable

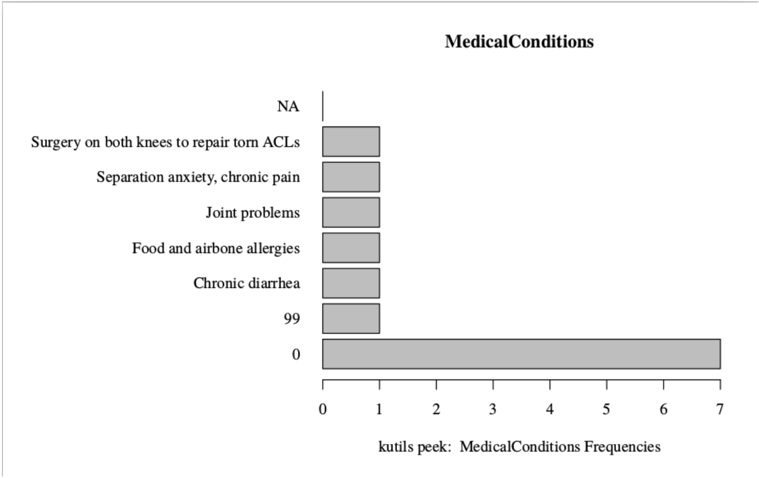
- Numeric variables: peek shows “up and down” *histogram*
- Categorical (AKA “factor”) variables: are sideways bar plots.

Your Mission: Cycle through the graphs to Spot your data recoding challenges

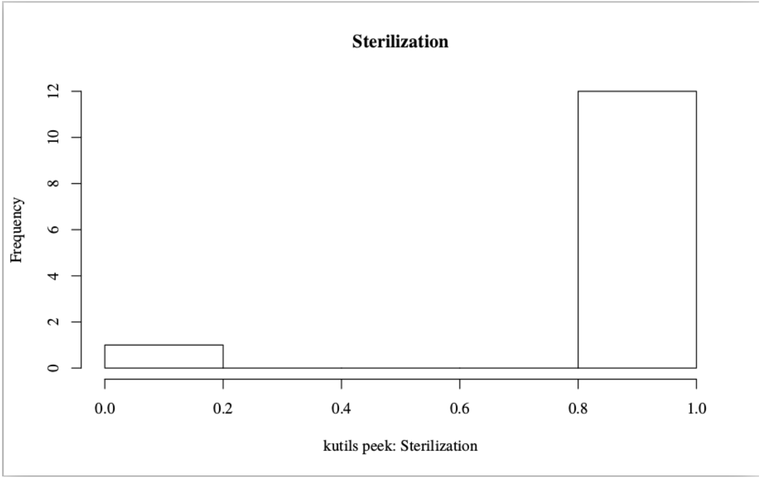
peek output has one image per variable ...



peek output has one image per variable



peek output has one image per variable



Need to Systematically Check for

- Data entry errors
- "Missing" value indicators that need to be turned into missing values
- The sterilization variable should be "No", "Yes" rather than a numeric 0, 1
- Example code in the R folder shows how I might have done some of that work.

When Recoding Work is Done

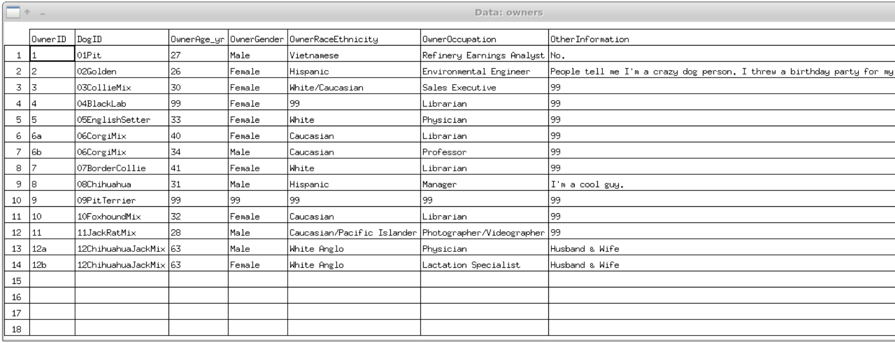
- Save the output of the recoding program in the folder “../workingdata”.

```
saveRDS(dogs, file = "../workingdata/dogs.rds")
```

- The analysis step begins with a new R program that begins, for example, with

```
dogdat <- readRDS("../workingdata/dogs.rds")
```

Do Same Cleanup with Owners



Data: owners

	OwnerID	DogID	OwnerAge_yr	OwnerGender	OwnerRaceEthnicity	OwnerOccupation	OtherInformation
1	1	01Pit	27	Male	Vietnamese	Refinery Earnings Analyst	No.
2	2	02Golden	26	Female	Hispanic	Environmental Engineer	People tell me I'm a crazy dog person. I threw a birthday party for my
3	3	03CollieMix	30	Female	White/Caucasian	Sales Executive	99
4	4	04BlackLab	99	Female	99	Librarian	99
5	5	05EnglishSetter	33	Female	White	Physician	99
6	6a	06CorgiMix	40	Female	Caucasian	Librarian	99
7	6b	06CorgiMix	34	Male	Caucasian	Professor	99
8	7	07BorderCollie	41	Female	White	Librarian	99
9	8	08Chihuahua	31	Male	Hispanic	Manager	I'm a cool guy.
10	9	09PitTerrier	99	99	99	99	99
11	10	10FoxhoundMix	32	Female	Caucasian	Librarian	99
12	11	11JackRatMix	28	Male	Caucasian/Pacific Islander	Photographer/Videographer	99
13	12a	12ChihuahuaJackMix	63	Male	White Anglo	Physician	Husband & Wife
14	12b	12ChihuahuaJackMix	63	Female	White Anglo	Lactation Specialist	Husband & Wife
15							
16							
17							
18							

Ethnicity looks especially tricky in the owner data

	OwnerRaceEthnicity (count)
99	2
Caucasian	3
Caucasian/Pacific Islander	1
Hispanic	2
Vietnamese	1
White	2
White Anglo	2
White/Caucasian	1

Ethnicity looks especially tricky in the owner data

- Appears we need to relabel “White/Caucasian”, “White Anglo”, “White” and “Caucasian” as the same thing.
- Looks complicated? (I’d teach you up in the summer stats camp. . .).

```
## use `mapvalues` from the plyr package
owners$OwnerRaceEthnicity <-
    plyr::mapvalues(owners$OwnerRaceEthnicity, from =
    c("White/Caucasian", "White Anglo", "White", "Caucasian"), to
    = "Caucasian")
```

- Find all rows from OwnerRaceEthnicity that match (%in%) the target values, and then reassign them (“<-”) as “Caucasian”

After Re-grouping Categories

	OwnerRaceEthnicity
99	2
Caucasian	8
Caucasian/Pacific Islander	1
Hispanic	2
Vietnamese	1

Then we'd need to do more work

1. Convert the 99's to the "missing value" symbol NA
2. Wrestle with the question of how to deal with the other non-Caucasian categories

Other Software, Same Story

- Many stats programs can import XLSX well enough.
- They don't have a super cool function like `peek` to snoop through columns (but now that we've let the cat out of the bag, they probably will).
- If there is trouble with importing, use a spread sheet program to "save as" CSV (comma-separated-variable) file.

Outline

1 Workflow

2 Separate places for separate things

3 Example with Dog Data

4 Take-Aways

Follow the Cross-Platform name conventions

- Use directory and file names that don't have spaces or other "reserved symbols" like like `!` , `*` , `&` , or `@` .
- Names that have clear, intuitive, don't confuse many people.
- Don't be afraid to cultivate habits and consistency among your efforts. Don't think every project deserves a completely different directory and file naming system.
- Short lower case names make me happy ☺

Use sub-folders for projects

- I strongly prefer to keep input, output in separate folders

  data workingdata output lit
  writeup admin R Stata
  Mplus SAS

- I find it bizarre that some people don't want to separate data from code from output, but am resigned to fact that people are free to disagree (no matter how wrong they are).

Consider in the Future

- Develop good Backup Habits
  - keep copies on a server, not in a USB stick that you put through the laundry
- Using a “Version Management” scheme
  - We use Git , a combination of “snapshot backup” and “project management”
  - Guide for that at `http://crmda.ku.edu/guides`.
- In the kutils package, we have developed a simple notation system that will help to better-organize the recoding process. This is called the Variable Key framework and we have an essay about it distributed with the package.