R review

Code

Conceptual review

Remember when we talked about tidy data? That's what we're working with when we work in R's "tidyverse". The big strength of R is that there is a package – or something pre-written for you – for almost anything you can think of that involves data analysis. The downside is that they have been written by different people, using different conventions. For example, sometimes you have to refer to a column name in quotes, sometimes not. They're also very difficult to read.

The Tidyverse was created to produce a standard grammar and structure to commands, to make them easier to read while taking advantage of the underlying program's structure.

We're going to work with a simplified version of some data you're pretty familiar with by now: the Murder Accountability Project's underlying data. You read about it in the New Yorker, and you worked with the data in the tutorial. But we're going to slow down a little and walk through everything that's happening in that dataset.

First, I want to create a smaller version with only Arizona homicides. I've copied the datasets from the original tutorial, and now I'm going to import them.

Note on "foreign" files that tripped you up

I think there was some confusion because you quite reasonably had no clue what was happening with the foreign SPSS files. SPSS, SAS and some other programs do things differently than spreadsheets and databases. In them, codes and names of columns (*fields* or *variables* from now on) can have labels. Those labels are just stored once, and appear on the screen. That's why they had to import twice.

Let's try it again. (I've already unzipped it in my data file.)

Hide

```
library(foreign)
getwd()
```

```
[1] "/Users/shcohen1/Documents/GitHub/rstats-training"
```

Hide

```
data_labels <- read.spss("SHR76_16.sav", to.data.frame=T)
```

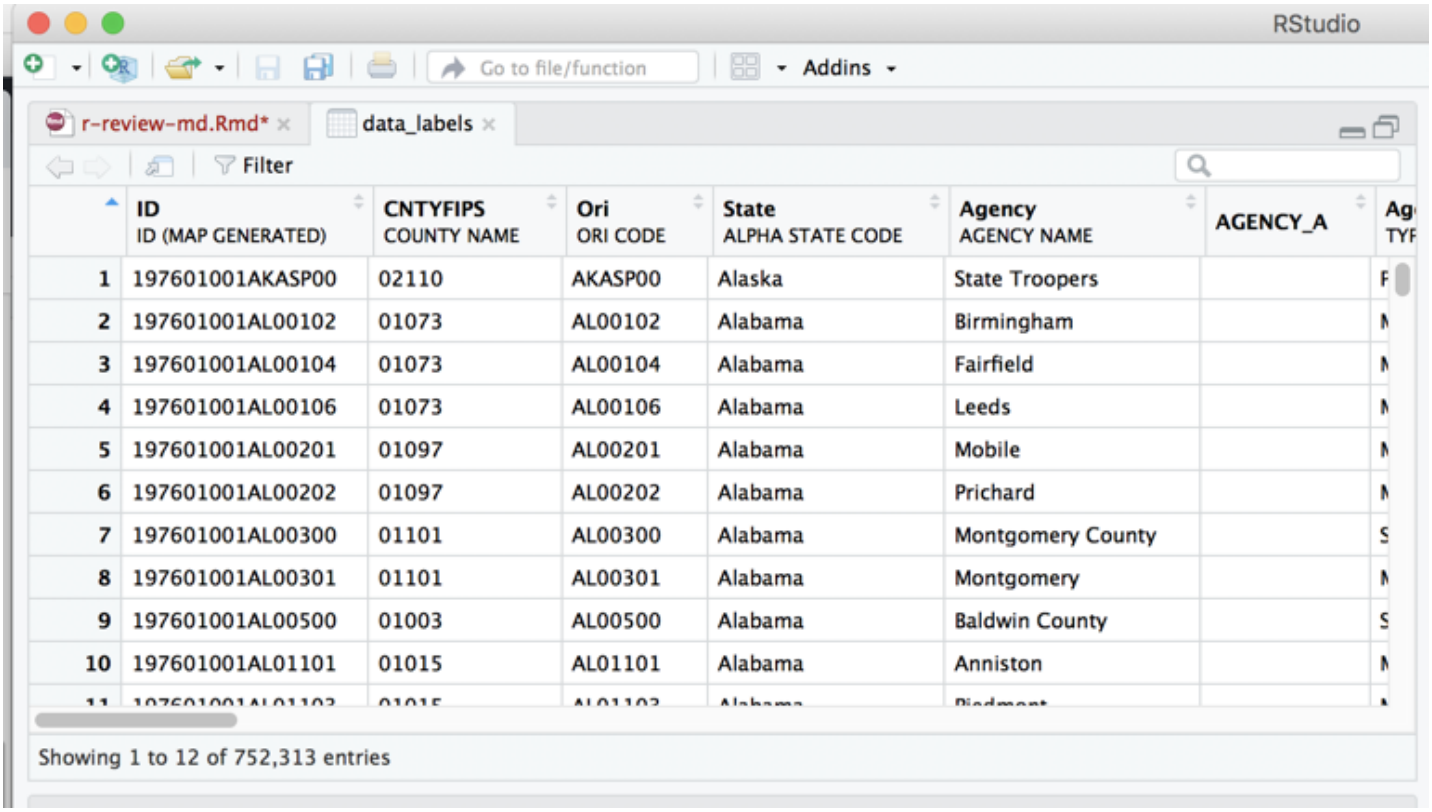
```
SHR76_16.sav: Very long string record(s) found (record type 7, subtype 14), each will be imported in consecutive separate variablesSHR76_16.sav: Long string value labels record found (record type 7, subtype 21), but ignored
```

We get a weird error message, but it's unclear what's going on. It appears to happen in Record 7. Let's view the dataset and see what's happening here.

Hide

```
View(data_labels)
```

(An image of what happens in RStudio)



That's a way to see the whole dataset, but how about just a few records? The first of these shows us the top of the file. The second one says, "Show me rows 6 through 8, and columns 14 - 21" (that's why there's a comma). The last one says, show me the structure of the data frame. It looks very

Hide

```
head(data_labels)
```

	ID <fctr>	CNTYF… <fctr>	Ori <fctr>	State <fctr>	Agency <fctr>	AGEN… <fctr>	Agentype <fctr>	S
1	197601001AKASP00	02110	AKASP00	Alaska	State Troopers		Primary state LE	F
2	197601001AL00102	01073	AL00102	Alabama	Birmingham		Municipal police	F
3	197601001AL00104	01073	AL00104	Alabama	Fairfield		Municipal police	F
4	197601001AL00106	01073	AL00106	Alabama	Leeds		Municipal police	F
5	197601001AL00201	01097	AL00201	Alabama	Mobile		Municipal police	F
6	197601001AL00202	01097	AL00202	Alabama	Prichard		Municipal police	F

6 rows | 1-9 of 33 columns

Hide

```
data_labels[6:8, 14:21]
```

	ActionType <fctr>	Homicide <fctr>	Situation <fctr>
6	Normal update	Murder and non-negligent manslaughter	Single victim/multiple offenders
7	Normal update	Murder and non-negligent manslaughter	Single victim/single offender
8	Normal update	Murder and non-negligent manslaughter	Single victim/single offender

3 rows | 1-5 of 8 columns

Hide

```
str(data_labels)
```

```
'data.frame':   752313 obs. of  33 variables:
 $ ID          : Factor w/ 719011 levels "197601001AKASP00",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ CNTYFIPS    : Factor w/ 3064 levels "01001          ",..: 75 37 37 37 49 49 51 51 2 8
...
 $ Ori         : Factor w/ 12616 levels "AK00101","AK00102",..: 31 35 37 39 58 59 68 69
75 108 ...
 $ State       : Factor w/ 54 levels "Alaska","Alabama",..: 1 2 2 2 2 2 2 2 2 2 ...
 $ Agency      : Factor w/ 9356 levels "24th Jud Cir Drug & Vctf
                                                                "| __truncated__,..: 7997
633 2491 4252 4988 6258 5045 5043 359 176 ...
 $ AGENCY_A    : Factor w/ 1 level "
                                                              "| __truncated__: 1 1 1 1 1 1
1 1 1 1 ...
 $ Agentype    : Factor w/ 8 levels "Sheriff","County police",..: 4 3 3 3 3 3 1 3 1 3
...
 $ Source      : Factor w/ 2 levels "MAP","FBI": 2 2 2 2 2 2 2 2 2 2 ...
 $ Solved      : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
 $ Year        : num  1976 1976 1976 1976 1976 ...
 $ StateName   : Factor w/ 52 levels "ALA   ","ALASKA",..: 2 1 1 1 1 1 1 1 1 1 ...
 $ Month       : Factor w/ 12 levels "January","February",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ Incident    : num  1 1 1 1 1 1 1 1 1 1 ...
 $ ActionType  : Factor w/ 2 levels "Normal update",..: 1 1 1 1 2 1 1 1 1 1 ...
 $ Homicide    : Factor w/ 2 levels "Murder and non-negligent manslaughter",..: 1 1 1 1
1 1 1 1 1 1 ...
 $ Situation   : Factor w/ 6 levels "Single victim/single offender",..: 1 1 1 1 1 3 1 1
1 1 ...
 $ VicAge      : num  48 65 45 43 35 25 27 42 41 50 ...
 $ VicSex      : Factor w/ 3 levels "Female","Male",..: 2 2 1 2 2 2 1 1 2 2 ...
 $ VicRace     : Factor w/ 5 levels "Asian or Pacific Islander",..: 3 2 2 2 5 2 2 2 5 5
...
 $ VicEthnic   : Factor w/ 3 levels "Hispanic origin",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ OffAge      : num  55 67 53 35 25 26 29 19 30 42 ...
 $ OffSex      : Factor w/ 3 levels "Female","Male",..: 1 2 2 1 1 2 2 2 1 2 ...
 $ OffRace     : Factor w/ 5 levels "Asian or Pacific Islander",..: 3 2 2 2 5 2 2 2 5 5
...
 $ OffEthnic   : Factor w/ 3 levels "Hispanic origin",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ Weapon      : Factor w/ 17 levels "Firearm, type not stated",..: 6 4 4 6 15 3 2 6 4 2
...
 $ Relationship: Factor w/ 29 levels "Acquaintance",..: 13 1 27 3 1 10 27 26 13 3 ...
 $ Circumstance: Factor w/ 32 levels "Rape","Robbery",..: 18 30 28 18 32 18 18 32 18 18
...
 $ Subcircum   : Factor w/ 7 levels "Felon attacked police officer",..: NA 5 NA NA NA NA
NA NA NA NA ...
 $ VicCount    : num  0 0 0 0 0 0 0 0 0 0 ...
 $ OffCount    : num  0 0 0 0 0 2 0 0 0 0 ...
 $ FileDate    : Factor w/ 5346 levels "010181","010191",..: 1015 1015 1015 1015 1015 10
15 1015 1015 1015 1015 ...
 $ fstate      : Factor w/ 55 levels "Alabama","Alaska",..: 2 1 1 1 1 1 1 1 1 1 ...
 $ MSA         : Factor w/ 411 levels "Abilene, TX",..: 365 36 36 36 219 219 223 223 364
15 ...
 - attr(*, "variable.labels")= Named chr  "ID (MAP GENERATED)" "COUNTY NAME" "ORI CODE"
"ALPHA STATE CODE" ...
```

```
  ..- attr(*, "names")= chr  "ID" "CNTYFIPS" "Ori" "State" ...
 - attr(*, "codepage")= int 65001
```

This shows us everything “labeled”. That is, the underlying data in SPSS has values that are masked by formats, or labels. It’s kind of like in Excel, when you formatted a number as a date – it appeared different than the underlying data was.

Cut the data down to size so you can see it a little better.

We’re not going to bother getting the UN-labeled data for us. And we’re going to get rid of some of the columns and rows. This is when the “tidyverse” comes in.

Select columns

We’re going to “select” some variables to play with. Think of this as filtering vertically.

In Excel, it would be the equivalent of copying your whole dataset, then deleting some of the columns so you could see it more clearly.

We’re also introducing the %<% operator. This is only used in the Tidyverse, and it means “do what I asked before this, then go do the next thing without stopping”

Hide

```
library(tidyverse)
```

```
[30m── [1mAttaching packages[22m ─────────────────────────────────────── tidyvers
e 1.2.1 ──[39m
[30m[32m✔[30m [34mggplot2[30m 3.0.0     [32m✔[30m [34mpurrr  [30m 0.2.5
[32m✔[30m [34mtibble [30m 1.4.2     [32m✔[30m [34mdplyr  [30m 0.7.6
[32m✔[30m [34mtidyr  [30m 0.8.1     [32m✔[30m [34mstringr[30m 1.3.1
[32m✔[30m [34mreadr  [30m 1.1.1     [32m✔[30m [34mforcats[30m 0.3.0[39m
[30m── [1mConflicts[22m ────────────────────────────────────────── tidyverse_conf
licts() ──
[31m✖[30m [34mdplyr[30m::[32mfilter()[30m masks [34mstats[30m::filter()
[31m✖[30m [34mdplyr[30m::[32mlag()[30m    masks [34mstats[30m::lag()[39m
```

Hide

```
murder_data <-
  data_labels %>%  #start with the data frame called "data labels", then...
  select( ID, CNTYFIPS, Ori, State, Agency, AGENCY_A, Source, Solved, Year,
          StateName, Month, Situation, VicAge, VicSex, VicRace, VicEthnic, OffAge, OffSe
x,
          OffRace, OffEthnic,
          Weapon, Relationship, Circumstance)
head(murder_data)
```

ID	CNTYF…	Ori	State	Agency	AGEN…	Sou…	Solv…	Y…
<fctr>	<fctr>	<fctr>	<fctr>	<fctr>	<fctr>	<fctr>	<fctr>	<dbl>

	ID <fctr>	CNTYF… <fctr>	Ori <fctr>	State <fctr>	Agency <fctr>	AGEN… <fctr>	Sou… <fctr>	Solv… <fctr>	Y… <dbl>
1	197601001AKASP00	02110	AKASP00	Alaska	State Troopers		FBI	Yes	1976
2	197601001AL00102	01073	AL00102	Alabama	Birmingham		FBI	Yes	1976
3	197601001AL00104	01073	AL00104	Alabama	Fairfield		FBI	Yes	1976
4	197601001AL00106	01073	AL00106	Alabama	Leeds		FBI	Yes	1976
5	197601001AL00201	01097	AL00201	Alabama	Mobile		FBI	Yes	1976
6	197601001AL00202	01097	AL00202	Alabama	Prichard		FBI	Yes	1976

6 rows | 1-10 of 23 columns

Filter rows

But we want to only keep the Arizona data, so now we need to **filter**, just the way you filtered in Excel. This is the equivalent of finding the records you want to keep using your filter in Excel, then copying them to another sheet. We could have done this whole thing in one step, but sometimes it's easier to just keep them separate. We're not going to create another whole data frame – we'll just replace the one we're working with.

Hide

```
murder_data <-
  murder_data %>%
    filter ( State == "Arizona")
head(murder_data)
```

	ID <fctr>	CNTYF… <fctr>	Ori <fctr>	State <fctr>	Agency <fctr>	AGEN… <fctr>	Sou… <fctr>	Solv… <fctr>	Y… <dbl>
1	197601001AZ00301	04005	AZ00301	Arizona	Flagstaff		FBI	Yes	1976
2	197601001AZ00700	04013	AZ00700	Arizona	Maricopa County		FBI	No	1976
3	197601001AZ00705	04013	AZ00705	Arizona	Chandler		FBI	Yes	1976
4	197601001AZ00707	04013	AZ00707	Arizona	El Mirage		FBI	Yes	1976
5	197601001AZ00713	04013	AZ00713	Arizona	Glendale		FBI	Yes	1976
6	197601001AZ00717	04013	AZ00717	Arizona	Mesa		FBI	Yes	1976

6 rows | 1-10 of 23 columns

Save it for later use

Now I'm just going to save this data frame for you so you can use it directly without any of the trouble I just went through.

Don’t worry about the first command here. What happened, if you look at your data in View(murder_data) is that some labels for the variables are wrong. I’m just getting rid of them. (They would have been right if I had not removed some of the variables.)

Hide

```
attributes(murder_data)$variable.labels <- NULL
save (murder_data, file="murder_data.Rda")
```

From here on out, we can work together on the dataset, but we’ll get different answers from the tutorial and work with a much smaller set of data.

Next page (r-wrangle.nb.html)