Introduction to Statistical Inference (QTM 100 Lab)

Lecture 3: Data Cleaning and Manipulation

Justin Eloriaga — Emory University

Fall 2024

Gameplan

Data Preliminaries

Creating New Variables

Numerical to Categorical

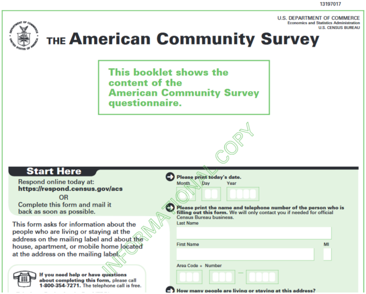
Categorical to Categorical

Data Preliminaries

Why clean data?

- Data we often recieve are often messy... ehem... more like garbage sometimes.
- Need a lot of preparation and care to data even before you begin the analysis
- Sadly, this is probably the *longest* step but also arguably the most crucial. It is what separates good analysis from trash!

American Community Survey (ACS)


13197017

U.S. DEPARTMENT OF COMMERCE
Economics and Statistics Administration
U.S. CENSUS BUREAU

THE American Community Survey

This booklet shows the content of the American Community Survey questionnaire.

INFORMATIONAL COPY

Start Here

Respond online today at:
https://respond.census.gov/acs
OR
Complete this form and mail it back as soon as possible.

This form asks for information about the people who are living or staying at the address on the mailing label and about the house, apartment, or mobile home located at the address on the mailing label.

If you need help or have questions about completing this form, please call **1-800-354-7271.** The telephone call is free.

Please print today's date.
Month Day Year

Please print the name and telephone number of the person who is filling out this form. We will only contact you if needed for official Census Bureau business.
Last Name
First Name MI
Area Code + Number

How many people are living or staying at this address?

- Detailed info on income, benefits, health insurance, education
- "Pretty big" Data (approx 3.5 million households surveyed annually)
- For class, we just take 1000 observations and 10 variables

Variables under Study

Variable	Description
`Sex`	gender
`Age`	age in years
`MarStat`	marital status
`Income`	annual income (in \$1,000s)
`HoursWk`	hours of work per week
`Race`	Asian, Black, White, or Other
`US Citizen`	citizen versus non-citizen
`HealthInsurance`	yes=have health insurance, no = no health insurance
`Language`	native English speaker versus other

Importing the Dataset (again)

- Like before, we can use point-and-click or the working directory

```
setwd("YourFilePath")

acs <- read.csv("acs.csv", header = TRUE)
```

- Let's also examine the structure and give an overview of the dataset

```
str(acs)

summary(acs)
```

Creating New Variables

Looking at `Age`

Suppose you want to look at the distribution of `Age`. One way to do that is through a table

```
table(acs$Age)
```

```
Console   Terminal ×   Background Jobs ×
R 4.4.1 · ~/
> table(acs$Age)

  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29 
 11   7   7  12  15  13  15   9  14  15  15  11  10  10  11  13  17   8  13  14  14  15  10   5  14  15   8  14  10  13 
 30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59 
 12   8  15  11   9  16  12  11  17   8  11  23  12  18   9  16  13  14  13  23  21  12  11  19  16   7  17   8  13  17 
 60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89 
  8  21  14  14  13   6  16  10   9  10   8   5   6   5  10   7   6   8   9   5   8   2   7   3   2   3   2   3   2   1 
 90  93  94 130 
  2   2   2   1 
```

Look at the values? Do you see anything peculiar?

Looking at `Age`

Suppose you want to look at the distribution of `Age`. One way to do that is through a table

```
table(acs$Age)
```

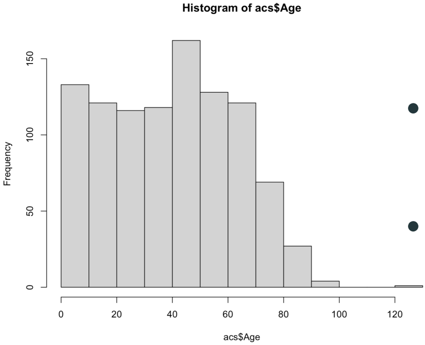
```
Console   Terminal ×   Background Jobs ×
R 4.4.1 · ~/
> table(acs$Age)

  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29 
 11   7   7  12  15  13  15   9  14  15  15  11  10  10  11  13  17   8  13  14  14  15  10   5  14  15   8  14  10  13 
 30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59 
 12   8  15  11   9  16  12  11  17   8  11  23  12  18   9  16  13  14  13  23  21  12  11  19  16   7  17   8  13  17 
 60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89 
  8  21  14  14  13   6  16  10   9  10   8   5   6   5  10   7   6   8   9   5   8   2   7   3   2   3   2   3   2   1 
 90  93  94 130 
  2   2   2   1 
```

Look at the values? Do you see anything peculiar?

> Maybe 0, maybe 130?

Looking at `Age`



- There are 11 obs with a value of zero. But seems okay, there are a lot of young children in the ACS.
- However, the value of 130 is clearly impossible and needs to be re-coded to missing!

```
hist(acs$Age)
```

Cleaning `Age`

Let's create a new variable which will be the cleaned version of `Age`, called `Age2`

```
acs$Age2 <- acs$Age
```

We need to replace the value of `Age2` of 130 with `NA` (missing). But maybe there are other implausible observations in age.

```
acs$Age2 == 130
```

```
acs$Age2 > 100
```

Any of these commands would show that the 157th entry contains a problematic age.

Indexing to Access

We can use square brackets to access (also called **indexing**) this observation and recode it to missing. I.O.W., brackets = where!

```
acs$Age2[157]
```

Recoding that entry of `Age2` to `NA`

```
acs$Age2[157] <- NA
```

To verify that we recoded correctly, use the summary command

```
summary(acs$Age2)
```

Maximum age is now 94 (previously 130), and there is now one `NA` (previously none)

More on Indexing

You could examine the rows of the dataset too using indexing. For example, print the entry in the 157th row and the second column

```
acs[157,2]
```

We could also see if all rows or columns satisfy a certain condition. For example, let us print all columns where `Age2` is greater than 100

```
acs[acs$Age2>100,]
```

Pesky `NA`s

When data entries take `NA`'s, it may need an extra argument for some `R` commands to run. For example, consider taking the mean

```
mean(acs$Age2)
```

It doesn't run properly because of the missing value. We need to specify an option to `R` to ignore the missing value

```
mean(acs$Age2, na.rm= T)
```

Numerical to Categorical

Having `Age` groups

Consider classifying people by age in such a way that 0 -18 are children, 19-55 are adults, and >55 are senior citizens. To do this, let us first create an `AgeCategory` variable which is a `factor` variable

```
acs$AgeCategory <- factor(NA,levels=c("child","adult","senior citizen"))
```

Then, we need to assign values of each age category

```
acs$AgeCategory[acs$Age2<=18] <- "child"
acs$AgeCategory[acs$Age2>18 & acs$Age2 <=55] <- "adult"
acs$AgeCategory[acs$Age2>55] <- "senior citizen"
```

Checking the Dataset Again

acs × Untitled1* ×

Filter

32.00	Sex	Age	MarStat	Income	HoursWk	Race	USCitizen	HealthInsurance	Language	Age2	AgeCategory
1	female	31	not married	60.00	40	white	citizen	yes	other	31	adult
2	male	31	not married	0.36	12	black	citizen	yes	native English	31	adult
3	male	75	not married	0.00	NA	white	citizen	yes	native English	75	senior citizen
4	female	80	not married	0.00	NA	white	citizen	yes	native English	80	senior citizen
5	male	64	married	0.00	NA	white	citizen	yes	native English	64	senior citizen
6	male	14	not married	NA	NA	white	citizen	yes	native English	14	child
7	male	78	married	0.00	NA	white	citizen	yes	native English	78	senior citizen
8	male	35	not married	87.00	40	white	citizen	yes	other	35	adult
9	female	70	married	0.00	1	white	citizen	yes	native English	70	senior citizen
10	female	18	not married	0.00	NA	white	citizen	yes	native English	18	child
11	male	48	married	32.00	48	white	citizen	no	native English	48	adult
12	female	61	married	0.00	20	white	citizen	no	native English	61	senior citizen
13	female	52	not married	48.00	40	white	citizen	yes	native English	52	adult
14	female	36	married	2.00	40	white	citizen	yes	native English	36	adult
15	female	20	not married	0.50	2	white	citizen	yes	native English	20	adult
16	male	67	not married	0.00	NA	white	citizen	yes	native English	67	senior citizen
17	male	8	not married	NA	NA	white	citizen	yes	native English	8	child
18	male	60	married	0.00	NA	white	citizen	yes	native English	60	senior citizen
19	male	75	married	21.60	6	white	citizen	yes	native English	75	senior citizen
20	female	25	married	0.00	NA	other	citizen	yes	native English	25	adult

Categorical to Categorical

Recoding `Race`

As of now, `Race` has four categories. Suppose you want to classify individuals as "white" and "non-white" (to fit, say, a binomial distribution). We can do that by creating a new variable

```
acs$RaceNew <- factor(NA,levels=c("white","non-white"))
```

Then, we can re-assign the values of the new race variable

```
acs$RaceNew[acs$Race == "white"] <- "white"

acs$RaceNew[acs$Race == "asian" | acs$Race == "black"| acs$Race == "other"] <- "non-white"
```

Verify that you recoded it properly by using a `table()` command or just look at the dataset.