Preparing Data for Analysis

MATH 456 - Spring 2016

Week 1

Week 1 Reading

Afifi: Chapters 1-5

Characterizing Data for Analysis. (*Afifi Ch 2*)

Problem 2.5 From a field of statistical application (perhaps your own field of specialty), describe a data set and repeat the procedures described in problem 2.3. That is, classify each variable according to Steven's scale and according to whether it is discrete or continuous. Pose two possible research questions and decide on the appropriate dependent and independent variables.

Preparing Data for Analysis. (*Afifi Ch 3*)

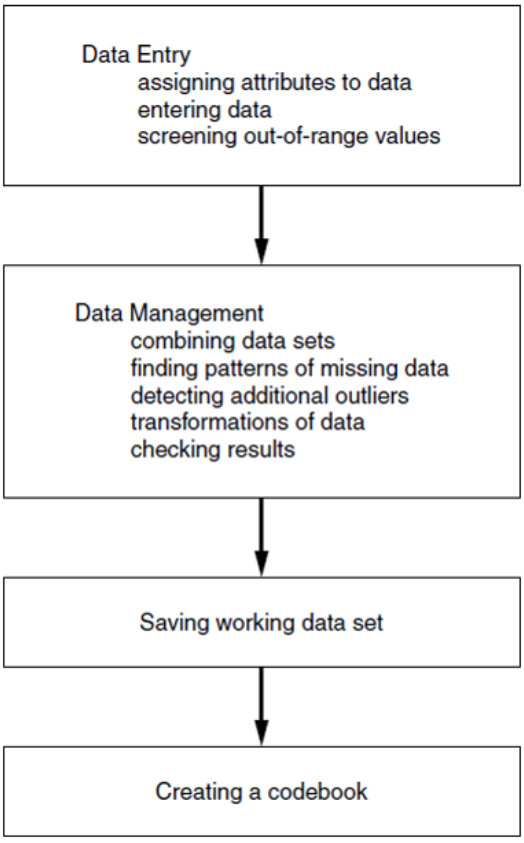


Figure 3.1: *Preparing Data for Statistical Analysis*

Reproducible Research

- You are your own collaborator 6 months from now. Make sure you will be able to understand what you were doing.
- Investing the time to do things clearly and in a reproducible manner will make your future self happy.
- Comment your code with explanations and instructions.
  - How did you get from point A to B?
  - Why did you recode this variable in this manner?
- This is reason #1 we use the Markdown language through R.

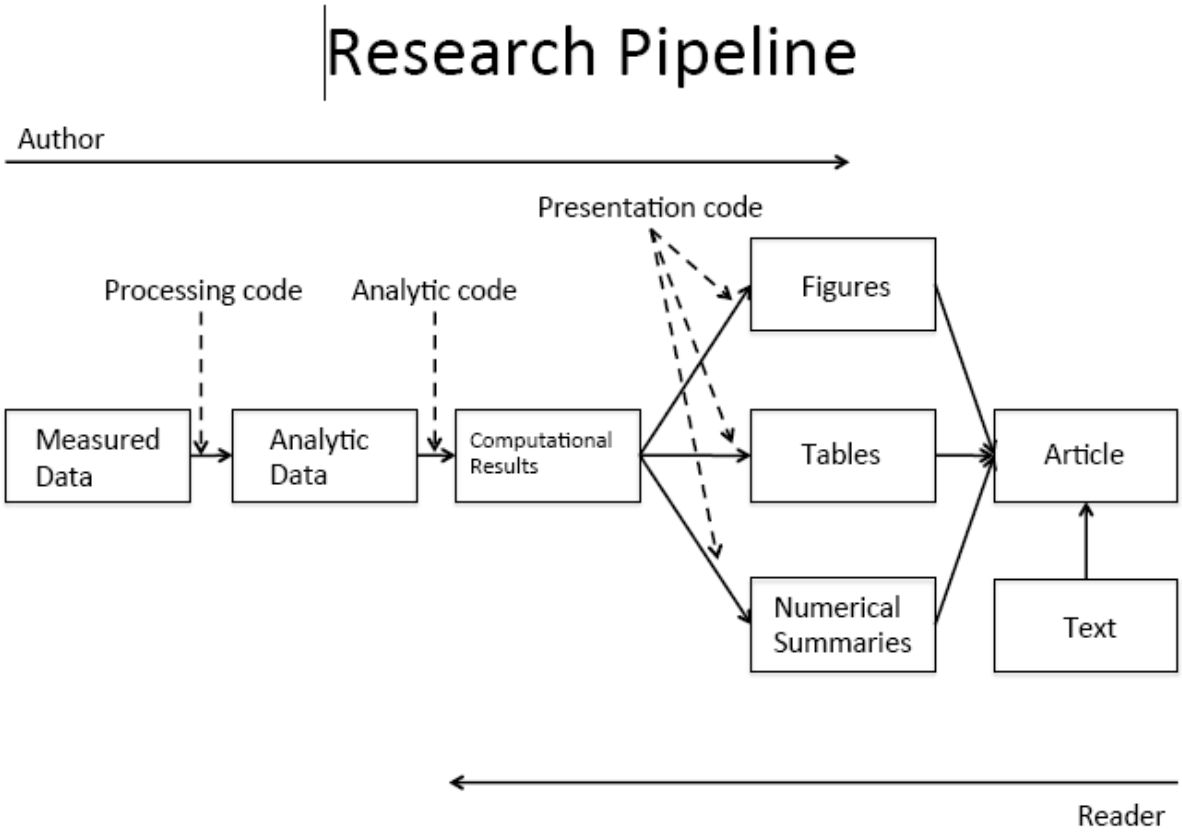


Figure Credits: Roger Peng

Practice

Reading data into R

- Download the Depression data set `depress` and codebook from the Data page of the class website.
- Save these into a **Data** sub-folder in your **MATH456** folder. This is a tab-delimited text file.

- Start a new Markdown file and in the first code chunk, read the data set into R using `read.table()`, load the `dplyr` and `ggplot2` libraries.

- Suppress the printing of messages for this code chunk by adding appropriate options to the code chunk starter line. "‘{r, message=FALSE, warning=FALSE}.

```
library(dplyr);library(ggplot2)
depress <- read.table("C:/GitHub/MATH456/data/Depress.txt",
                      sep="\t", header=TRUE)
```

Identifying variable types (and fixing them)

- Consider the variable that measures marital status What data type does the codebook say this variable is?
- What data type does R see this variable as?

```
table(depress$MARITAL)
```

```
##
##   1   2   3   4   5
##  73 127  43  13  38
```

```
str(depress$MARITAL)
```

```
##  int [1:294] 5 3 2 3 4 2 2 1 2 2 ...
```

```
is(depress$MARITAL)
```

```
## [1] "integer"             "numeric"             "vector"
## [4] "data.frameRowLabels"
```

When variables have numerical levels it is necessary to ensure that R knows it is a factor variable. The following code takes the marital status variable and turns it into a factor variable with specified labels that match the codebook.

```
depress$MARITAL <- factor(depress$MARITAL,
                          labels = c("Never Married", "Married", "Divorced", "Separated", "Widowed"))
```

Confirm the recode worked. If it did not you will have to re-read in the raw data set again since the varible SEX was replaced.

```
table(depress$MARITAL)
```

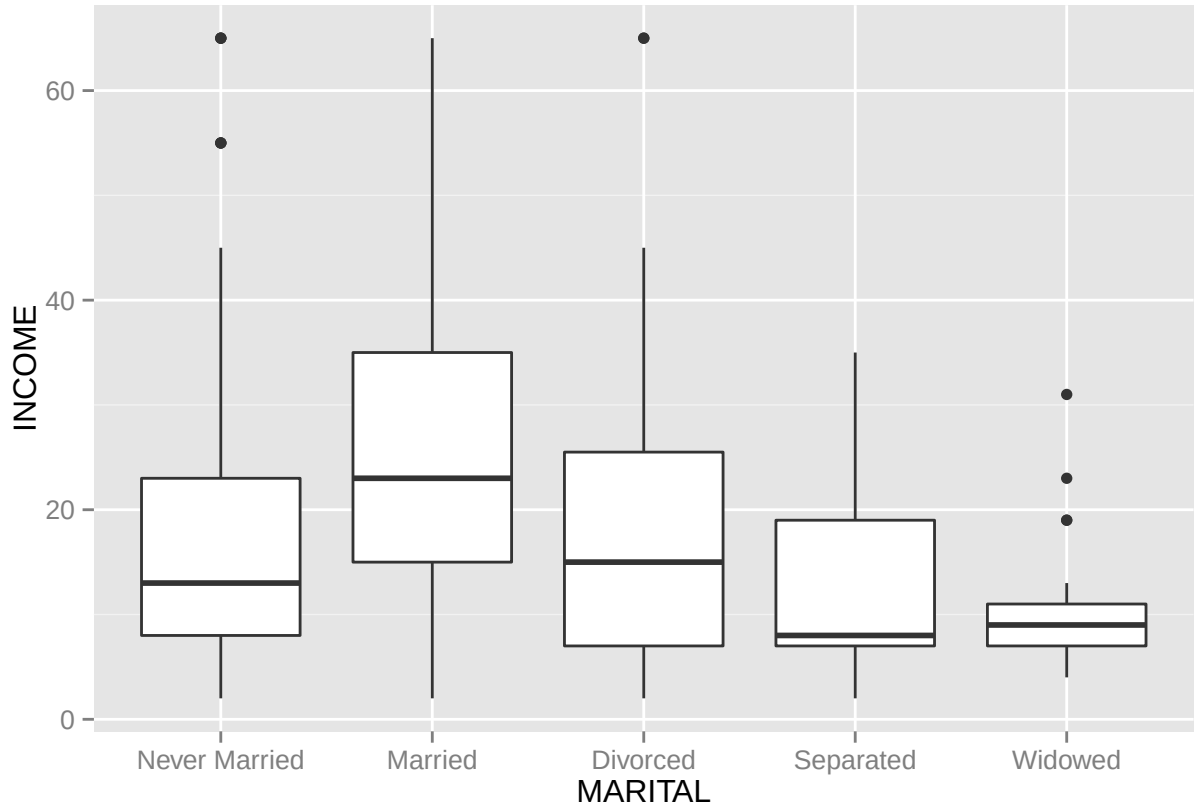
```
##
## Never Married       Married      Divorced     Separated       Widowed
##            73           127            43            13            38
```

```
is(depress$MARITAL)
```

```
## [1] "factor"              "integer"             "oldClass"
## [4] "numeric"             "vector"              "data.frameRowLabels"
```

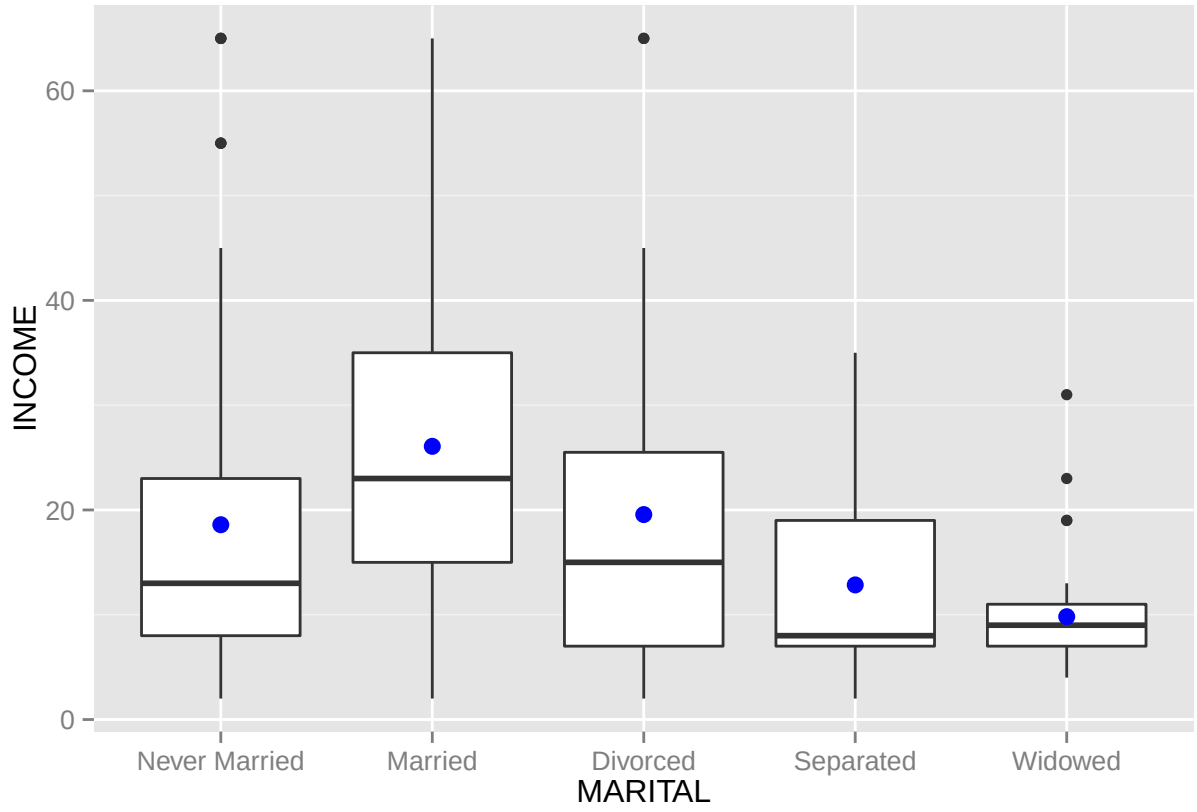
- Create a boxplot of income across marital status category.

```
qplot(y=INCOME, x=MARITAL, data=depress, geom="boxplot")
```



Boxplots are nice because they clearly show the range where 50% of the data lie and any potential outliers. Boxplots can also indicate skewness, but sometimes it is helpful to visualize the location of the mean as well as the median. `ggplot2` has a nice `stat_summary` layer that will calculate and add the means to the current plot.

```
qplot(y=INCOME, x=MARITAL, data=depress, geom="boxplot") +
   stat_summary(fun.y=mean, colour="blue", size=3, geom="point", show_guide = FALSE)
```



Recoding variables

For unbiased and accurate results of a statistical analysis, sufficient data has to be present. Often times once you start slicing and dicing the data to only look at certain groups, or if you are interested in the behavior of certain variables across levels of another variable, sometimes you start to run into small sample size problems. Take Marital status as an example, there are only 13 people who report being separated.

One way to deal with insufficienet data within a certain category is to collapse categories. The following code creates a new variable that I am calling `MARITAL2` that combines the `Divorced` and `Separated` levels.

** Option 1:** Using `ifelse()` function is probably the easiest way to recode variables when you are only changing one or two levels. Read `?ifelse` for the syntax for this variable. Here I am saying if marital status is either *Separated* or *Divorced* then set the value of `MARITAL2` to *Sep/Div*, otherwise don't change the value. This is accomplished by setting the new value equal to the old.

```
depress$MARITAL2 <- ifelse(depress$MARITAL %in% c('Separated', 'Divorced'), "Sep/Div", depress$MARITAL)
```

Always confirm your recodes.

```
table(depress$MARITAL, depress$MARITAL2, useNA="always")
```

```
##
##                    1   2   5 Sep/Div <NA>
##   Never Married   73   0   0       0    0
##   Married          0 127   0       0    0
```

```
##   Divorced        0   0   0      43    0
##   Separated       0   0   0      13    0
##   Widowed         0   0  38       0    0
##   <NA>            0   0   0       0    0
```

This confirms that values with `Divorced` or `Separated` on the old variable

6. Create a dotplot or strip chart of income against employment status. *jittering* or *dodging* the points may be helpful to avoid overplotting of points. Are there any adult whose income is unusual considering their employment stauts?

On your own

1. Create a new variable that collapses the first three education levels. Confirm your recode by displaying a contingency table of the old variable `EDUCAT` against your new variable. Be sure to use the `useNA="always"` argument in the `table()` statement.
2. Determine if any variables have observations that do not fall witin the ranges given in the codebook. If there are any, decide what to do with those values and implment your decision.

Data screening and transformations (*Afifi Ch 4*)

- Describe the distribution of `INCOME`. Be sure to write out your description in paragraph form and discuss the location (measures of center), spread (measures of variance) and shape (normality or skewness) of the distribution using an appropriate plot and summary statistics as evidence. Connect your text to specific features of the plot and/or summary statistics, do not just say "as you can see in the plot…". Make sure the plot is fully annotated with an appropriate title and axes labels.
- Assess the need to transform the income variable to induce normality.

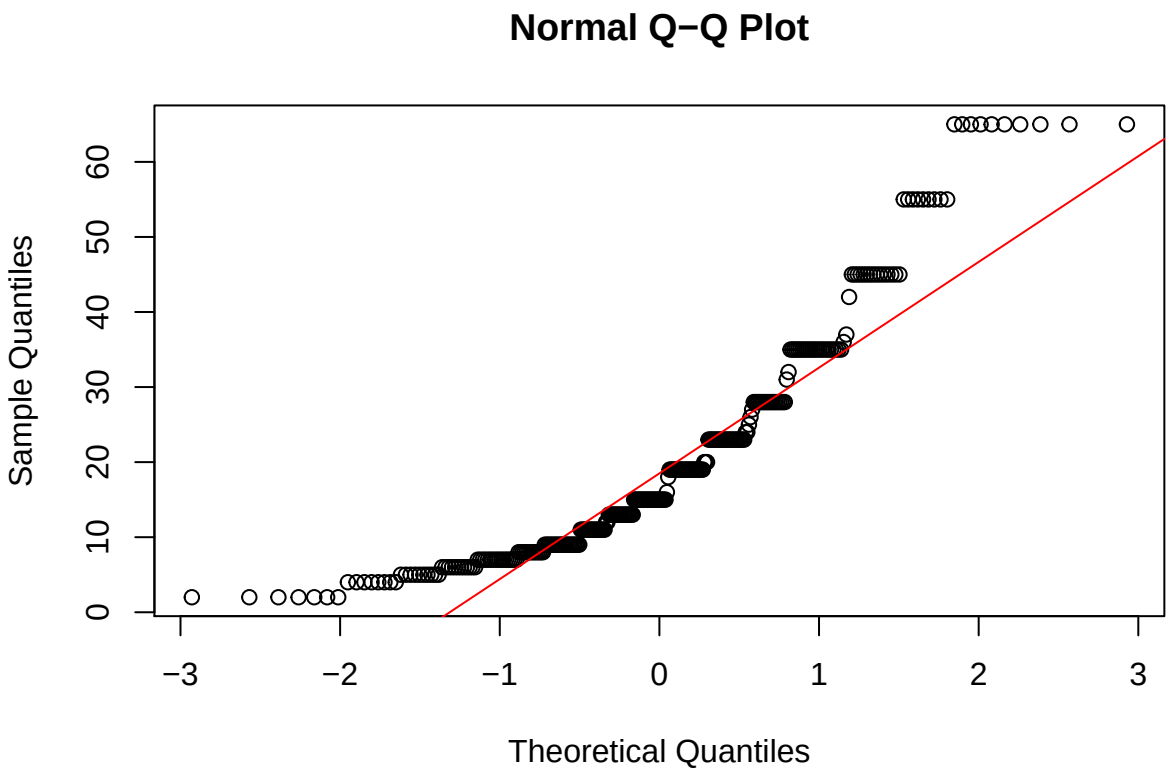
```
sd(depress$INCOME) / mean(depress$INCOME)
```

```
## [1] 0.743147
```

```
max(depress$INCOME) / (min(depress$INCOME)+.1)
```

```
## [1] 30.95238
```

```
qqnorm(depress$INCOME);qqline(depress$INCOME, col="red")
```



- Create three new variables: `log10inc` as the log base 10 of Income, `loginc` as the natural log of Income, and `xincome` which is equal to the negative of one divided by the cubic root of income.
- Create a single plot that display normal probability plots for the original, and each of the three transformations of income. Use the base graphics grid organizer `par(mfrow=c(r,c))` where `r` is the number of rows and `c` is the number of columns. Which transformation does a better job of normalizing the distribution of Income?

```
par(mfrow=c(2,2)) # Try (4,1) and (1,4) to see how this works.
qqnorm(depress$INCOME); qqline(depress$INCOME,col="blue")
qqnorm(log10inc); qqline(log10inc, col="blue")
qqnorm(loginc); qqline(loginc, col="blue")
qqnorm(xincome); qqline(xincome, col="blue")
```

Above plot not showing? Make sure you have removed the `eval=FALSE` *argument in the R code chunk.*

- Take the logarithm of the CESD score plus 1 and compare the histograms of CESD and log(CESD+1). Describe the distribution of each.
- Why was the +1 added to CESD prior to taking the log?

5. Using the Parentla HIV data set, plot a histogram, boxplot, and a normal probability plot for the variable `AGESMOKE`. this variable is the age in years when the respondent started smoking. If the respondent did not start smoking, AGESMOKE was assigned to a value of zero. Decide what to do about the zero values and if a transformation should be used for this variable if the assumption of normality is made when it is used in a statistical analysis.

6. Using the Parental HIV data calculate an overall Brief Symptom Inventory (BSI) score of each adolescent (See the codebook for details). Log transform the BSI score. Obtain a normal probabilty plot for the log transformed variable. Does the log-transformed variable seem to be normally distributed? As you migth notice, the number of adolescents with a missing value on the overall BSI score and the log-transformed BSI score are different. Why is this the case? Could this influence our conclusion regarding the normalit of the transformed variable? How oculd this be avoided?

References

- http://rprogramming.net
- http://www.cookbook-r.com/
- http://www.uni-kiel.de/psychologie/rexrepos/index.html
- http://norcalbiostat.github.io/R-Bootcamp/labs/Data_Visualization_Tutorial_Full.html