Lec 01: Preparing Data for Analysis

MATH 456 - Spring 2016

Assigned Reading

Afifi: Chapters 1-5

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

On Your Own: Afifi 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.

Data wrangling, munging, recoding, editing, cleaning (*Afifi Ch 3*)

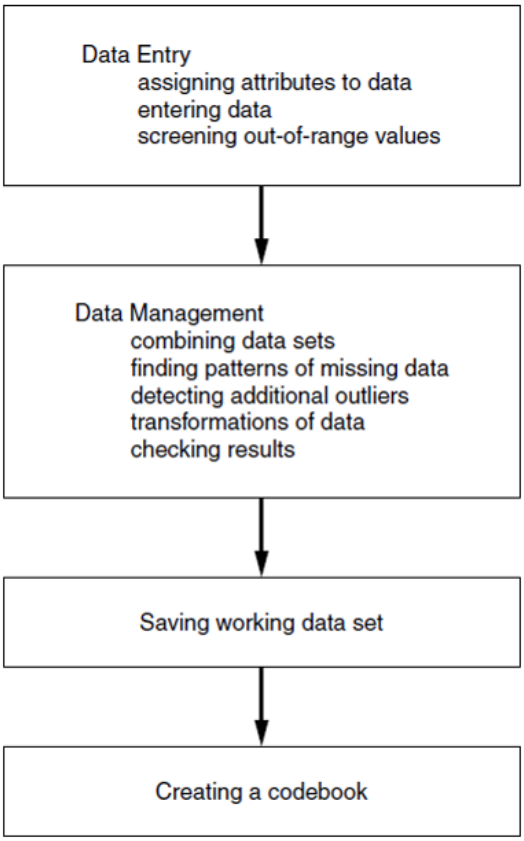


Figure 3.1: *Preparing Data for Statistical Analysis*

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.

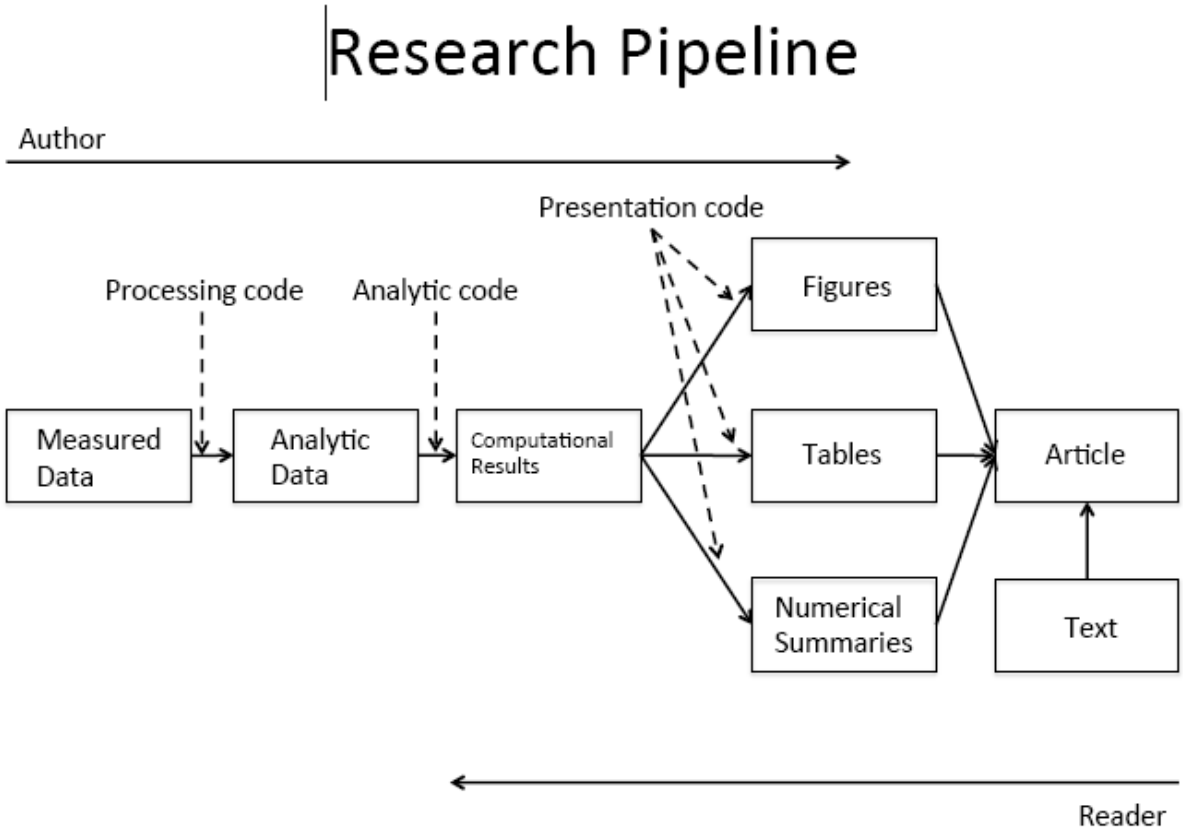


Figure Credits: Roger Peng

Practice (in class)

Reading data into R

- Download the Depression data set `depress` and codebook from the Google Drive Data folder.
- 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 uses the `factor()` function to take the marital status variable and convert 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"))
```

You should always confirm the recode worked. If it did not you will have to re-read in the raw data set again since the variable `SEX` was replaced.

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

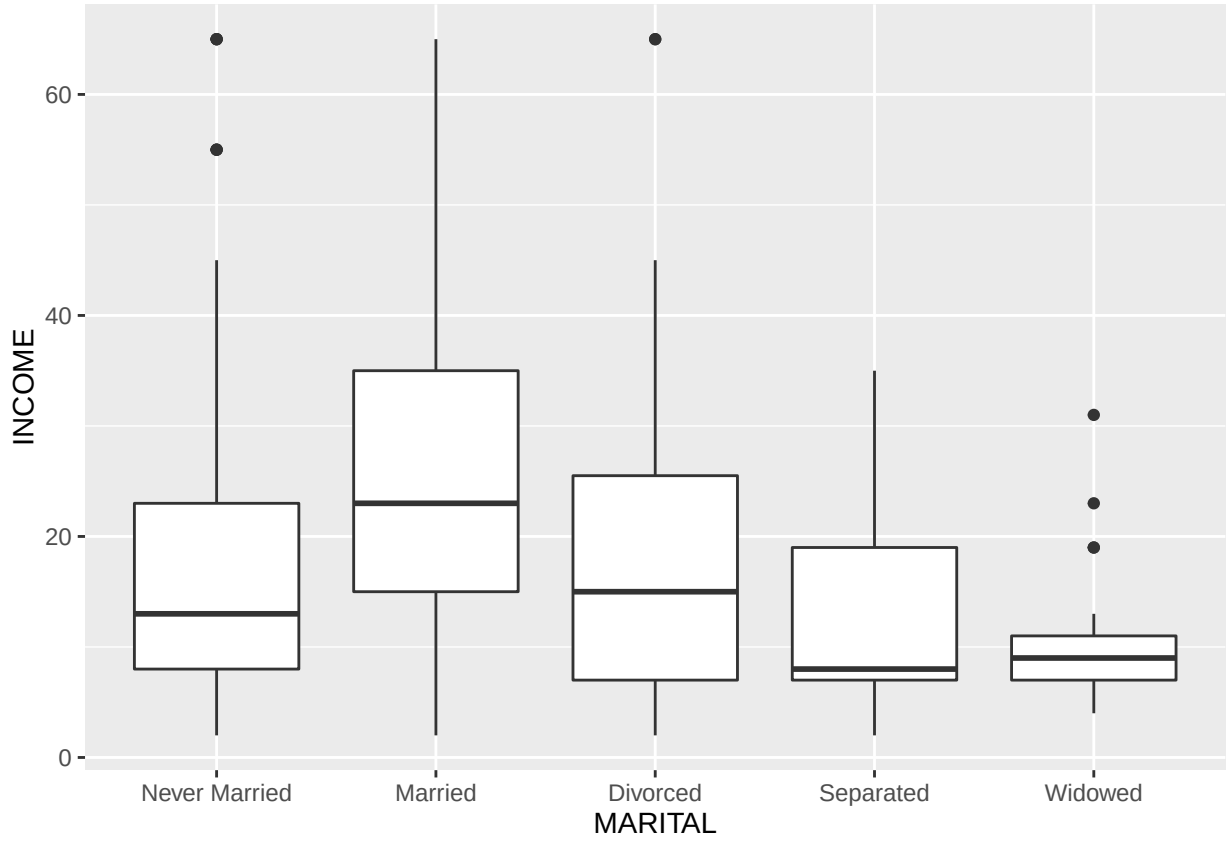
```
##
## Never Married       Married      Divorced     Separated       Widowed
##            73           127            43            13            38
```

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

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

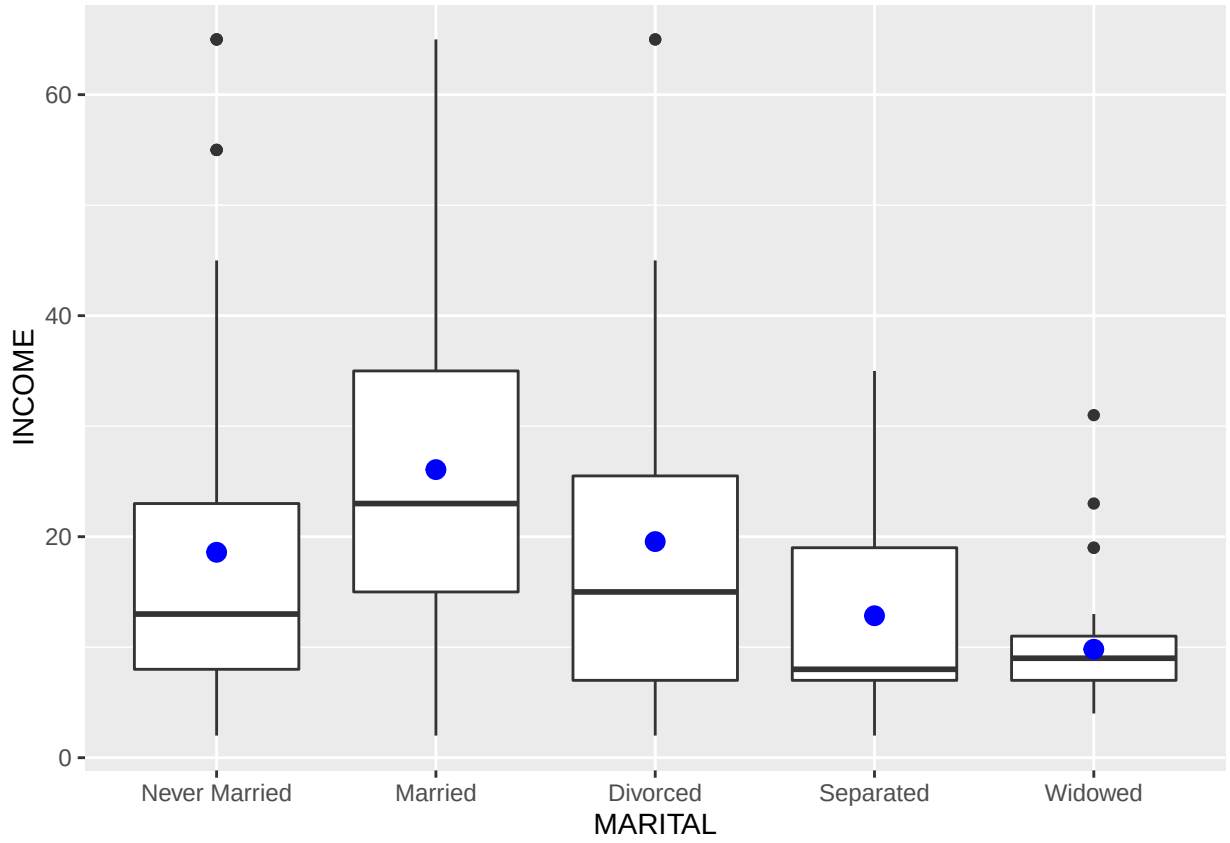
- Create a boxplot of income across marital status category.

```
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")
```



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. For example, consider marital status.

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

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

There are only 13 people who report being separated. This could potentially be too small of a group size for valid statistical analysis.

One way to deal with insufficient data within a certain category is to collapse categories. The following code uses the `recode()` function from the `car` package to create a new variable that I am calling `MARITAL2` that combines the `Divorced` and `Separated` levels.

```
library(car)
depress$MARITAL2 <- recode(depress$MARITAL, "'Divorced' = 'Sep/Div'; 'Separated' = 'Sep/Div'")
```

Always confirm your recodes.

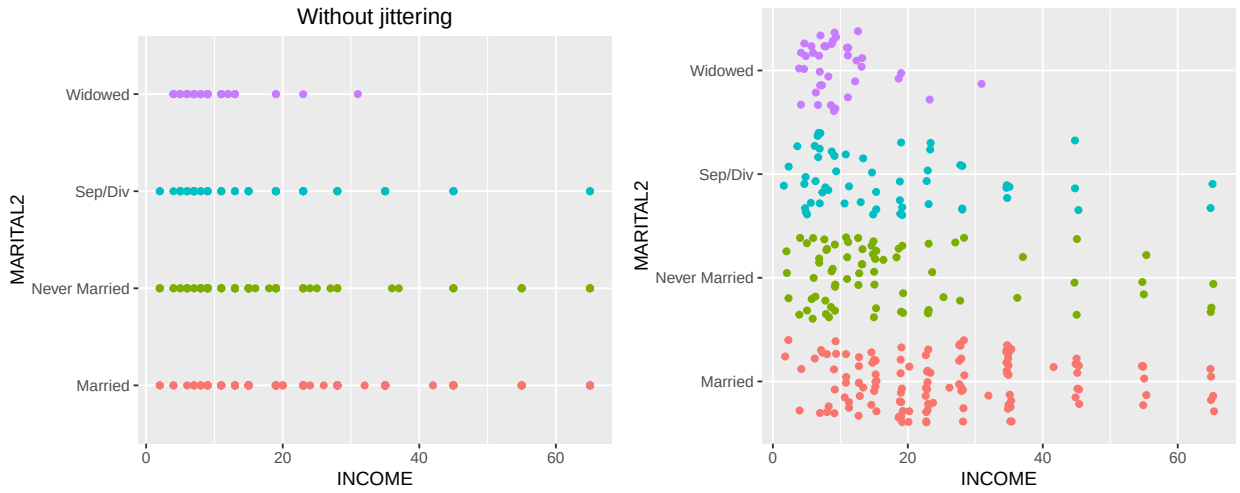
```
table(depress$MARITAL, depress$MARITAL2, useNA="always")
```

```
##
##                  Married Never Married Sep/Div Widowed <NA>
##    Never Married       0            73       0       0    0
##    Married           127             0       0       0    0
##    Divorced            0             0      43       0    0
##    Separated           0             0      13       0    0
##    Widowed             0             0       0      38    0
##    <NA>                0             0       0       0    0
```

This confirms that records where `MARITAL` (rows) is `Divorced` or `Separated` have the value of `Sep/Div` for `MARITAL2` (columns).

Now let's examine the relationship between income against marital status by creating a boxplot. This is a situation where *jittering* or *dodging* the points is helpful to avoid overplotting of points. Note that the full `ggplot` code had to be used here, not the simpler `qplot` methods. Furthermore, the `grid.arrange` function from the `gridExtra` package is used to display these plots side by side.

```
library(gridExtra)
a <- qplot(x=MARITAL2, y=INCOME, data=depress, col=MARITAL2, geom="point", main = "Without jittering") +
           coord_flip() + theme(legend.position="none")
b <- ggplot(depress, aes(x=INCOME, y=MARITAL2, color=MARITAL2), main="With jittering") +
            geom_point(position=position_jitter()) + theme(legend.position="none")
grid.arrange(a, b, ncol=2)
```



- What do you think `coord_flip()` does? Look at the difference in the X and Y values between plot a and plot b.
- What do you think `theme(legend.position="none")` does?

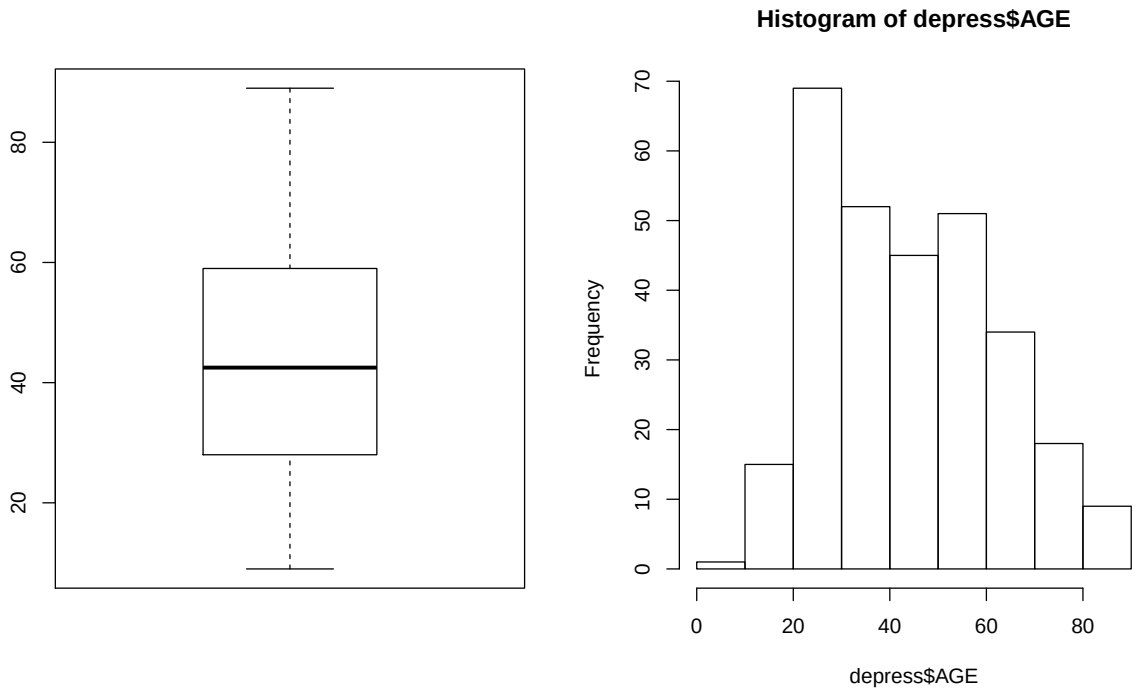
Hint: Try removing them and see what happens.

- What can you say about the relationship between Income and marital status?

Detecting and recoding Outliers and/or inconsistent data.

Let's look at the age variable in the depression data set.

```
par(mfrow=c(1,2))
boxplot(depress$AGE)
hist(depress$AGE)
```



Just looking at the data graphically raises no red flags. The boxplot shows no outlying values and the histogram does not look wildly skewed. This is where knowledge about the data set is essential. The codebook does not provide a valid range for the data, but the description of the data starting on page 3 in the textbook clarifies that this data set is on adults. In the research world, this specifies 18 years or older.

Now look back at the graphics. See anything odd? It appears as if the data go pretty far below 20, possibly below 18. Let's check the numerical summary to get more details.

```
summary(depress$AGE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    9.00   28.00   42.50   44.38   59.00   89.00
```

The minimum value is a 9, which is outside the range of valid values for this variable. This is where you, as a statistician, data analyst or researcher goes back to the PI and asks for advice. Should this data be set to missing, or edited in a way that changes this data point into a valid piece of data.

As an example of a common data entry error, and for demonstration purposes, I went in and changed a 19 to a 9. So the correct thing to do here is to change that 9, back to a 19. This is a very good use of the `ifelse()` function.

```
depress$AGE <- ifelse(depress$AGE==9, 19, depress$AGE)
```

The logical statement is `depress$AGE==9`. Wherever this is true, replace the value of `depress$AGE` with 19, wherever this is false then keep the value of `depress$AGE` unchanged (by "replacing" the new value with the same old value).

Alternatively, you can change that one value using bracket notation. Here you are specifying that you only want the rows where `AGE==9`, and directly assign a value of 19 to those rows.

```
depress$AGE[depress$AGE==9] <- 19
```

Confirm the recode.

```
summary(depress$AGE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.00   28.00   42.50   44.41   59.00   89.00
```

Looks like it worked.

On Your Own: Data Wrangling Write your responses in a new Markdown file named *userid_ch3.rmd*.

1. Using the depression data set, 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. What can you say about the relationship between Income and Educational level?
3. Determine if any variables in the depression data set have observations that do not fall within the ranges given in the codebook. If there are any, decide what to do with those values and implement your decision.
4. Update the Parental HIV data set by creating all the subscales listed at the bottom of the codebook.
    - This is saved as an Excel file so use the `read_excel` function in the `readxl` package to import the data into R.

```
library(readxl)
hiv <- read_excel("C:/GitHub/MATH456/data/Parhiv.xlsx")
```

    - Create a new R code file for this work. This can be done by going to *File -> new file -> R script*. This will save as a `.R` file. We will come back to this code file at a later point.
    - If a scale says that a variable is reversed (*e.g. reverse(pb03)*) that means the variable is reverse-coded prior to being included in the scale.
    - Some values may be missing. For consistency, do not use `na.rm` to remove the missing data from the mean calculations. This means that if a person is missing data on any component to a scale, the value for the entire scale is missing.
    - Use the `write.table()` function to write this data set as a tab-delimited text file using the current date in the file name.

```
write.table(hiv, "Your Path/PARHIV_012616", sep="\t", row.names=FALSE, col.names=FALSE)
```

- Edit the codebook to add the variable name you choose to use for each subscale.

SOLUTIONS [HTML] [RMD]

Data screening and transformations (*Afifi Ch 4*)

Recall the aim of data preparation, screening, wrangling, or transforming is to

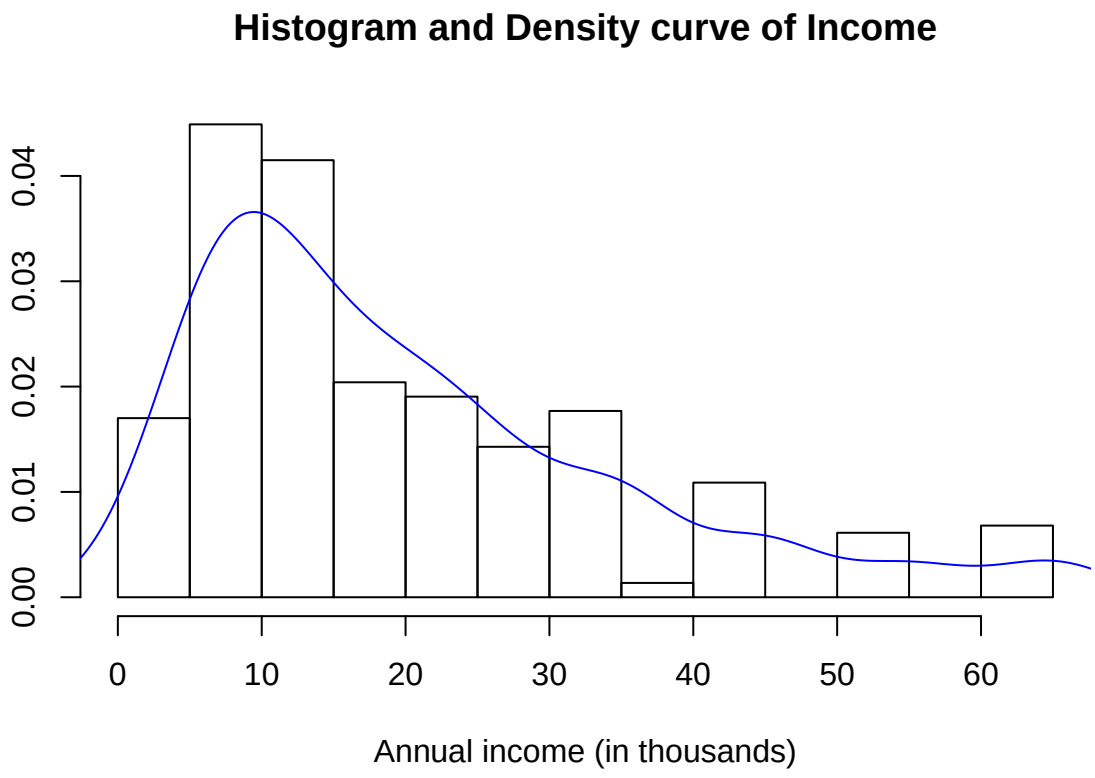
- Identify outliers and inconsistent values
- Assess normality of the distribution
- Assess independence of observations
- Explore data transformations to aid description, inference.

In the previous section we looked at outliers and inconsistent values. Now let's look at normality and independence using the cleaned depression data set.

```
rm(depress) # remove the current version that was used in the previous part of this markdown file
depress <- read.table("C:/GitHub/MATH456/data/Depress_020116.txt", sep="\t", header=TRUE)
```

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.

```
hist(depress$INCOME, prob=TRUE, xlab="Annual income (in thousands)",
     main="Histogram and Density curve of Income", ylab="")
lines(density(depress$INCOME), col="blue")
```




```
summary(depress$INCOME)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.00    9.00   15.00   20.57   28.00   65.00
```

The distribution of annual income is slightly skewed right with a mean of $20.5k per year and a median of $15k per year income. The range of values goes from $2k to $65k. Reported income above $40k appear to have been rounded to the nearest $10k, because there are noticeable peaks at $40k, $50k, and $60k.

In general, transformations are more effective when the the standard deviation is large relative to the mean. One rule of thumb is if the sd/mean ratio is less than 1/4, a transformation may not be necessary.

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

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

Alternatively Hoaglin, Mosteller and Tukey (1985) showed that if the largest observation divided by the smallest observation is over 2, then the data may not be sufficiently variable for the transformation to be decisive.

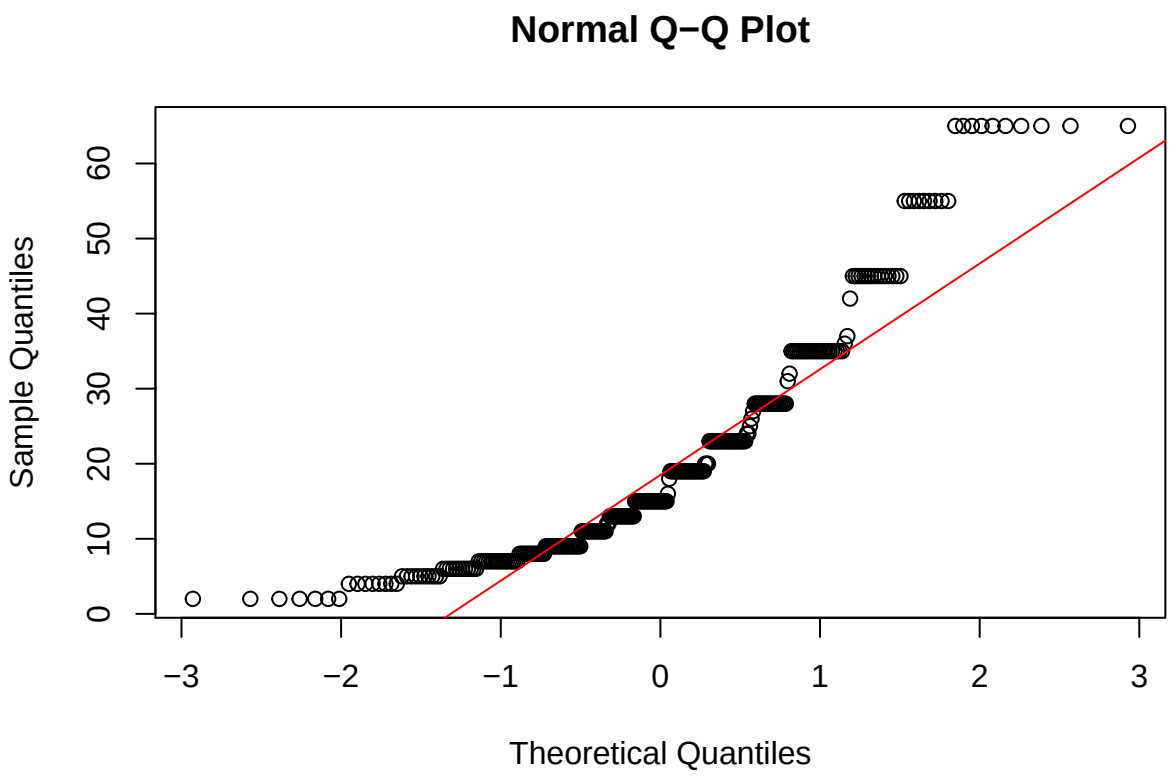
```
max(depress$INCOME) / (min(depress$INCOME)+.1)
```

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

Note these rules are not meaningful for data without a natural zero. The book goes into more detail about options for interval data.

Another common method of assessing normality is to create a normal probability (or normal quantile) plot.

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



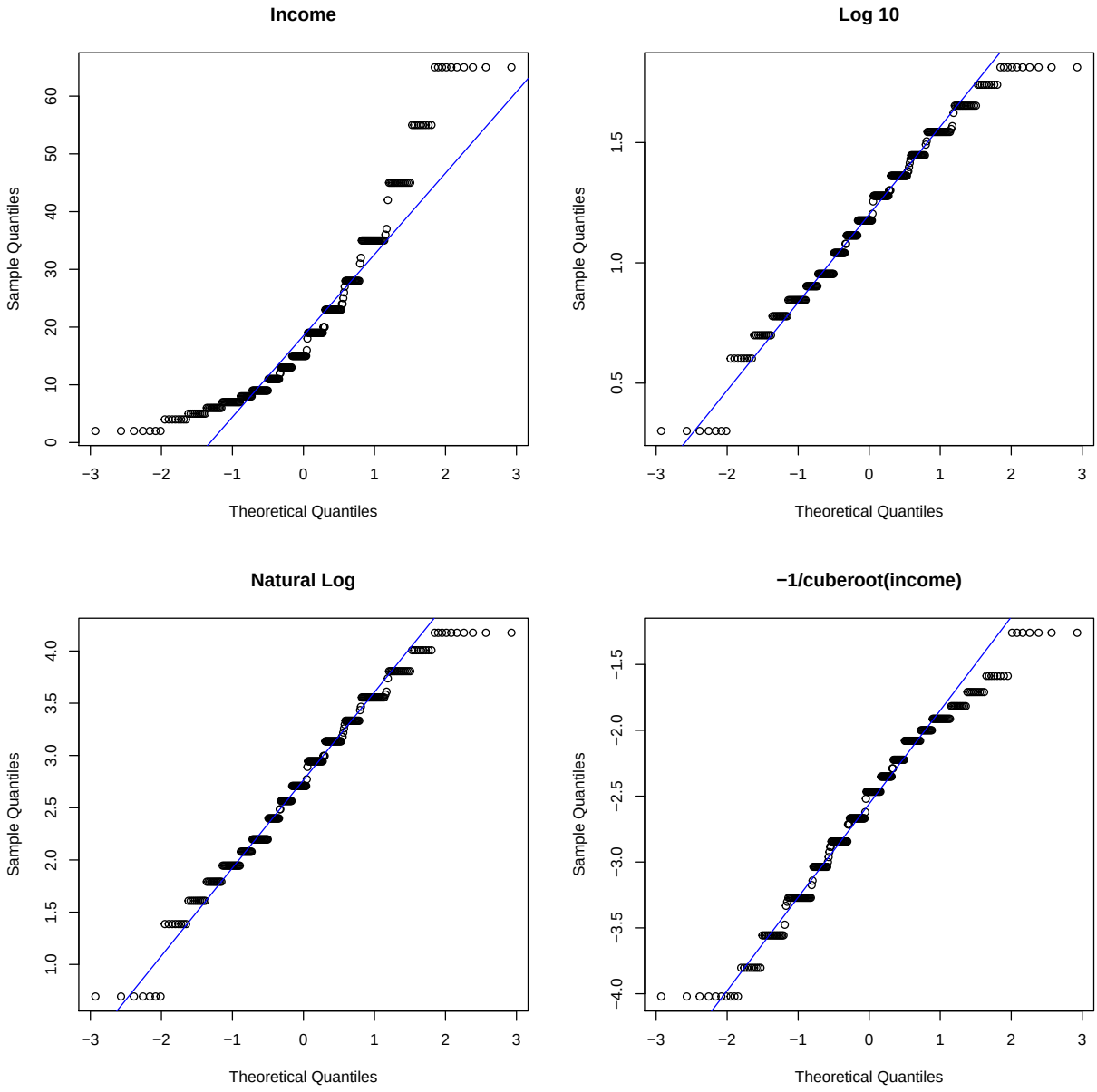
The points on the normal probability plot do not follow the red reference line very well. The dots show a more curved, or `U` shaped form rather than following a linear line. This is another indication that the data is skewed and a transformation for normality should be created.

- 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.

```
log10inc <- log10(depress$INCOME)
loginc   <- log(depress$INCOME)
xincome  <- -1/(depress$INCOME)^(-1/3)
```

- 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$INCOME, main="Income"); qqline(depress$INCOME,col="blue")
qqnorm(log10inc, main="Log 10"); qqline(log10inc, col="blue")
qqnorm(loginc, main = "Natural Log"); qqline(loginc, col="blue")
qqnorm(xincome, main="-1/cuberoot(income)"); qqline(xincome, col="blue")
```



On Your Own: Data Transformations Write your responses in a new Markdown file named *userid_ch4.rmd*.

1. Take the logarithm of the CESD score plus 1 and compare the histograms of CESD and log(CESD+1). Describe the distribution of each.

2. Why was the +1 added to CESD prior to taking the log?

3. Create a new variable that categorizes income into the following ranges: <30, [30, 40), [40,50), [50, 60), 60+.
   Hint: Two options include using several `ifelse()` *functions, or the* `cut2()` *function in the* `Hmisc` *package.*

4. Replace the placeholder values in the code below with the real variable names to conduct an ANOVA to determine if the mean transformed `CESD` (from #1) differs across income category (from #3).

```
summary(aov(continuous variable ~  categorical variable, data=_____))
```

5. Using the Parental 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 (*Hint: You already did this so use the updated data set!*). Log transform the BSI score. Obtain a normal probability plot for the log transformed variable. Does the log-transformed variable seem to be normally distributed? As you might 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 normality of the transformed variable? How could this be avoided?

Selecting Appropriate Analysis (*Afifi Ch 5*)

Considerations:

- Purpose of analysis.
- Types of variables in data set.
- Data used in analysis.
- Assumptions needed; satisfied?
- Choice of analyses is often arbitrary: consider several

Example:

5 independent variables: 3 interval, 1 ordinal, 1 nominal

1 dependent variable: interval

Analysis options

- Multiple regression: pretend independent ordinal variable is an interval variable use dummy (0 /1) variables for nominal variables
- Analysis of variance: categorize all independent variables
- Analysis of covariance: leave variables as is, check assumptions
- Logistic regression: Categorize dependent variable: high, low
- Survival analysis: IF dependent variable is time to an event

Unsure? Do several and compare results.

On Your Own Write your responses in a new Markdown file named *userid_ch5.rmd*.

Using Afifi Table 5.2 (p75) to help your decision making, work with a partner to answer problems 5.1-5.7, 5.10-13, 5.15. One submission per pair, include both names in the Author field.

Session Info

This document was compiled on 2016-02-01 16:39:45 and with the following system information:

```
sessionInfo()
```

```
## R version 3.2.2 (2015-08-14)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 7 x64 (build 7601) Service Pack 1
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] readxl_0.1.0    gridExtra_2.0.0 car_2.1-1       ggplot2_2.0.0
## [5] dplyr_0.4.3     rmarkdown_0.8.1 knitr_1.11
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.1         formatR_1.2.1       nloptr_1.0.4
##  [4] plyr_1.8.3          tools_3.2.2         digest_0.6.8
##  [7] lme4_1.1-10         evaluate_0.8        gtable_0.1.2
## [10] nlme_3.1-121        lattice_0.20-33     mgcv_1.8-7
## [13] Matrix_1.2-2        DBI_0.3.1           yaml_2.1.13
## [16] parallel_3.2.2      SparseM_1.7         stringr_1.0.0
## [19] MatrixModels_0.4-1 grid_3.2.2          nnet_7.3-10
## [22] R6_2.1.1            minqa_1.2.4         magrittr_1.5
## [25] scales_0.3.0        htmltools_0.2.6     MASS_7.3-43
## [28] splines_3.2.2       assertthat_0.1      pbkrtest_0.4-4
## [31] colorspace_1.2-6    labeling_0.3        quantreg_5.19
## [34] stringi_0.5-5       munsell_0.4.2
```