Homework
(this was taken and adapted from Homework 1 from Stats I...)

due by November 16, 2015

NOTES

- You can find the SPSS file REGRESS.SAV on BlackBoard (Course Documents --> Week 1: Introduction --> Homework)
- On the very last page of this document here, there are some hints and tips, in case you get stuck. However, try always to do it on your own and use the help function and/or google, before you peak at the hints in this document!
- Just as a reminder, I sometimes left the old SPSS instructions in this document, but of course you should do all this in R (or, if you prefer, RStudio)!
- For now, you need only to do **Steps 0, 1, and 2** (the rest comes later in the course).
- To hand in your homework:
    - Save your R script with all your commands that you used for this homework.
    - Put your answers to the questions (including figures) in a word document.
    - To make things easy, paste your R script commands in the word document with the answers to the questions and figures, print the whole thing out, and bring it to the next class.

This assignment involves three general steps that should be initially undertaken prior to conducting multivariate analyses in order to better understand the distributional properties of individual variables and bivariate associations. The data set for this assignment is REGRESS.SAV, which contains these variables:

ID	participant id number
STRESS	a weighted score reflecting number and importance of life changes
PHYHEAL	frequency count of physical health problems
MENHEAL	frequency count of mental health problems
TIMEDRS	frequency count of visits to physical and mental health professionals (doctors)

STEP 0: Import SPSS data file REGRESS.SAV into R

Some hints how to do this:

There are at least 2 different ways how to do this:
(1) Directly open the SPSS file in R.
(2) Open the SPSS file in SPSS, save it as a new file in .csv (comma separated values) format, open the csv file in R.
It's a good idea if you are familiar with both of these ways, so try both.

We start with the first option here:
To import SPSS files directly into R, there are (at least) two packages which can do that, library(Hmisc) and library(foreign). I prefer library(foreign), but feel free to use any other that works for you.

1. Probably, you first have to install that package.
2. Then load that package.
3. Set the working directory to the folder where you have saved the SPSS file.
4. Import the data file into R, creating a data frame called regress_1. Hint 1: If you directly import the SPSS file, then don't forget the argument `to.data.frame=TRUE`.
5. Check the first few and last few rows of data of your new data frame called `regress_1`

Some comments:

(a) The variable names look somewhat differently, did you notice? Please note that in R, A and a are not the same (i.e., capitalization matters).

(b) The variable names in R will the the variable names in SPSS, not SPSS' "variable labels"
You can still get the variable labels, via this command:

```
attr(regress_1, "variable.labels")
```

In contrast, you can see the variable names with this command:
`names(regress_1)`

Thus, if you wanted to, you could use the variable labels to replace the variable names. Beware, though: In R, variable names must NOT contain spaces! (thus `Participant number` is not ok; `Participant_number` is ok.) I would just use the variable names and not replace them, so feel free to do the same.

(b) Sometimes, when importing data from a different program, things can go wrong. Therefore, it is extremely important to always thoroughly check the imported data!
There are different commands with which you can check the data, I like to use `head()` and `tail()` for quick checks (or output the whole data frame in the console to see ALL rows of data).

When I imported the SPSS file, I got a warning message (that happens sometimes, no reason to worry; also "warning messages" are not the same as "error messages:" If you get a "warning" R usually still did something; if you get an error message, it ususally means that R was not able to execute your command.)

```
Warning message:
Unrecognized record type 7, subtype 18 encountered in system file
```

When I then checked the imported data, the last row of data was non-sense, i.e., it contained only NA entries. Therefore I created a new data frame regress_2 that did not include the very last row.

How would you do that?
Hint: You can get the number of rows in the data frame and use that information to create a new data frame regress_2 that omits the last row.

OK, now let's import the data again, this time by using method (2):
1. Open the SPSS file in SPSS.
2. Save it in csv format (you could give it a name like For_RImport_regress.csv or something like that).
3. In R, use the command read.csv to load the csv file (e.g., in a data frame called regress_1b).
4. Use head() and tail() to see whether the imported data look ok.
5. If they look weird (like the entries are not nicely in separate columns as they should be), use the command read.csv2 to try again. Does it look ok now?
6. If it's still not ok, you might have to use either the command read.csv or read.csv2 with an argument like sep = "," or something similar (sep might be also ";" or something else; have a look at the data you imported in steps 4. and 5. to choose the character that separates the entries). Thus for example, the command might look something like:

```
regress_1b <- read.csv2("For_RImport_regress.csv", sep = ",")
```

You may have to try out different things until it works fine.

STEP 1: Check Univariate Distributions

Answer the following questions that apply to the variables TIMEDRS, PHYHEAL, MENHEAL, and STRESS.
1. What is the mean, standard deviation, and range?
2. Is the distribution skewed? If so, is it positively or negatively skewed? [Comment: Do not worry about testing the *significance* of skewness, just report how strongly it is skewed.]
3. Is the distribution kurtosed? If so, is it leptokurtic or platykurtic? [Comment: Same here, do not worry about the *significance* of kurtosis, just report how strongly it is kurtosed.]
4. Are there outliers? If so, how many? [Comment: For now, just use the plot functions below to get a rough idea whether there are outliers and approx. how many; no need to give the exact number. The exact number depends on the criterion used to define a data point as an outlier, and different people use different criteria...]

Hints

- In R, the package `psych()` (--> install and download it!) has nice functions called `describe()` and `describeBy()` that give means, medians, and lots of other descriptive statistics. One simple way to answer the questions above is to use one of these functions.
- To get the actual values, you can either use the `describe()` command or functions such as `mean()`, `median()`, `sd()`, `min()`, `max()`, `skew()`, and `kurtosi()`.
- Use `densityplot()` (from the package `lattice`) to plot one variable at a time; save them as jpg (or some other format) and paste them in your word document with the answers; other simple plots to look at distributions are `boxplot()` and `hist()`.

STEP 2: Determine the Utility of Linear Transformations

OLD SPSS instructions (you don't have to do that again in SPSS, but I left it in there for your information):
When the distribution of a variable is significantly skewed or kurtosed, linear transformations can be used to alter the shape of the distribution. Use the *Compute* command in SPSS (in the *Transform* menu) to transform all variables with significantly skewed or kurtosed distributions (see Tukey Ladders of Transformation.pdf). Re-examine the distributions of the raw and transformed scores (using the *Frequencies* or *Explore* function in SPSS) and answer the following question:

***NOTE: For variables that include zeroes (such as TIMEDRS, MENHEAL, and STRESS), a constant of 1 should be added when calculating logarithmic transformations.

R instructions:
In R, this might be even easier: Create **new** variables and add them to your existing data frame; these new variables are to be the transformed versions of the old variables, for example using the commands `log()` or `sqrt()`.
Call the new transformed variables things like `log_menheal` to indicate that this is the log-transformed variant of the original variable.
Look at the new/transformed variables using `describe()` (or `skew()` etc) and create densityplots for them (as before, do not worry about whether things are significantly or non-significantly skewed or kurtosed.).
Save the data frame that contains both the original and the newly created variables as a csv file called regress_plusTransforms.csv

Some hints and tips how to do these things

Step 0
download and install a package: either use the R menu item (e.g., on the Mac: Tools --> Install Packages...) or use the command `install.packages("NameOfTheLibrary")`. Bill and I recommend that you use the command `install.packages().`

loading a package: `library()`

setting the working directory: `setwd()`

loading the data and saving it in a data frame called regress_1:
```
regress_1 <- read.spss('REGRESS.SAV', to.data.frame=TRUE)
```

get rid of last row of data:
```
regress_2 <- regress_1[-nrow(regress_1),]
```

Step 1
for example:
```
describe(regress_2$phyheal)
skew(regress_2$stress)
```

Step 2
for example:
```
regress_2$log_stress <- log(regress_2$log_stress)
```

save data frame to your working directory:
```
write.csv(regress_2, file = "regress_plusTransforms.csv")
```