QM1: Creating a Dataset

Dr. Niccole M. Pamphilis

Working with Data

Before we can start to clean our data or run any statistical tests, we need to put data in to a data set. Sometimes, our data will come already formatted in a dataset, other times we will need to find our own data from different sources and put it together.

There are different approaches you can use to create a dataset, some are more straight forward than others. As this course assumes no prior coding knowledge, we are going to start with a very simple approach to putting your dataset together as well as talk about a few more advanced techniques you may want to use later one as your become more confident in your coding and data handling skills.

Excel

One of the easiest ways create a dataset is to start with excel. Using an excel spreadsheet you can create your observations in the first column and use subsequent columns for your variables. This will allow you to paste in data from various sources, move columns around, and arrange things very quickly.

Keep in mind that that variables go in to the columns and the rows represent data on each individual observation.

Once this is done you can save your excel sheet as a .csv file, which R can read. Once you load this in to R you can proceed with adding data values to variables, creating new variables, or changing variable names as you see fit.

To open a csv file in R use the following command:

```
##Read in data
data<- read.csv(file.choose(), header=T)

## The header=T commend tells R the first row corresponds to variable names
```

Inputting data directly in to R

You CAN input data directly in to R, but it is not recommended. If you are going to enter data directly in to R you will need to make sure you enter each data value in the correct order to correspond to the observation. Once you create each variable you can attach them together and create a data frame to use in R. While this is possible, it is also a high risk approach to adding user error to your data.

Below is an example of how this can be done. I will create individual "vectors" in R. Once each vector corresponding to a variable is created I can add them all to another object will become my dataset.

```
##Create an observation variable

ID<-c(1,2,3,4,5)
  #Above, I created a vector called ID with 5 values representing obs. ids
```

```
##Create variables

age<-c(20,19,32,45,NA)
  #Above is a variable called age, the order of the ages correspond to the order of IDs
  #above in variable ID

opinion <-c(-2,1,0,1,1)

##Now I will combine all three vectors into a set together

data1<-cbind(ID,age,opinion)
  #the cbind command combines columns. If we just used the "c()" command it would just
  ##stack the three variable on top of each other create one long string

data1<-as.data.frame(data1)
```

Merging two files

Sometimes we have a dataset we are working with and we find new data that is related to our observations that we want to add. We can do this using a merge command. What this requires is that we have the same set of observations for each dataset and that the observations are identified using the same variable name and the same values.

For example, we have EU countries in our dataset and EU countries in another data. In both data sets the observations are identified with a variable called country and for observations on this variable it uses the full country name. We can use the merge command here.

However, if we have two data set and one has countries referred to by their three letter abbreviations and the other full country names, this command will not work.

It is also important to note that you cannot or should not combine data sets this way that are for different observations.

Here now, I am going to add a variable to our example dataset we built above called "data".

```
##Create second data set

ID<-c(1,2,3,4,5)
  #We are going to assume the five observations here are the same as the 5 from the
  #earlier data set. If this is not the case, we should not merge the data

apples<-c(0,1,1,0,1)

data2<-cbind(ID,apples)

data2<-as.data.frame(data2)

 ##This is new data on the five observations from a new set of questions that was put in a
  ##new data set, however, since it is about the same 5 observations we were already
  ##studying it might be usful for our analyses so we want to add it in.
```

```
#Notice data has ID, age and opnion; while data1 has ID and apples.
  #The head command allows you to look at the first few observations in a data set
```

```r
head(data1)
```

```
##   ID age opinion
## 1  1  20      -2
## 2  2  19       1
## 3  3  32       0
## 4  4  45       1
## 5  5  NA       1
```

```r
head(data2)
```

```
##   ID apples
## 1  1      0
## 2  2      1
## 3  3      1
## 4  4      0
## 5  5      1
```

```r
#The merge command allows you to put two datasets togethe, but you need to identify what
#it observation variable is so that variables' data are allocated to the correct
#observation (datasets may not be ordered in the same way.)

combined_data <- merge(data1, data2, by="ID")

head(combined_data)
```

```
##   ID age opinion apples
## 1  1  20      -2      0
## 2  2  19       1      1
## 3  3  32       0      1
## 4  4  45       1      0
## 5  5  NA       1      1
```