5.5: Descriptive Statistics Separately for each Group

It is very commonly the case that you find yourself needing to look at descriptive statistics, broken down by some grouping variable. This is pretty easy to do in R, and there are three functions in particular that are worth knowing about: `by()`, `describeBy()` and `aggregate()`. Let's start with the `describeBy()` function, which is part of the `psych` package. The `describeBy()` function is very similar to the `describe()` function, except that it has an additional argument called `group` which specifies a grouping variable. For instance, let's say, I want to look at the descriptive statistics for the `clin.trial` data, broken down separately by `therapy` type. The command I would use here is:

```
describeBy( x=clin.trial, group=clin.trial$therapy )
```

```
##
##  Descriptive statistics by group
## group: no.therapy
##           vars n mean   sd median trimmed  mad min max range skew kurtosis
## drug*        1 9 2.00 0.87    2.0    2.00 1.48 1.0 3.0   2.0 0.00    -1.81
## therapy*     2 9 1.00 0.00    1.0    1.00 0.00 1.0 1.0   0.0  NaN      NaN
## mood.gain    3 9 0.72 0.59    0.5    0.72 0.44 0.1 1.7   1.6 0.51    -1.59
##             se
## drug*     0.29
## therapy*  0.00
## mood.gain 0.20
## ------------------------------------------------------------
## group: CBT
##           vars n mean   sd median trimmed  mad min max range  skew
## drug*        1 9 2.00 0.87    2.0    2.00 1.48 1.0 3.0   2.0  0.00
## therapy*     2 9 2.00 0.00    2.0    2.00 0.00 2.0 2.0   0.0   NaN
## mood.gain    3 9 1.04 0.45    1.1    1.04 0.44 0.3 1.8   1.5 -0.03
##           kurtosis   se
## drug*        -1.81 0.29
## therapy*       NaN 0.00
## mood.gain    -1.12 0.15
```

As you can see, the output is essentially identical to the output that the `describe()` function produce, except that the output now gives you means, standard deviations etc separately for the `CBT` group and the `no.therapy` group. Notice that, as before, the output displays asterisks for factor variables, in order to draw your attention to the fact that the descriptive statistics that it has calculated won't be very meaningful for those variables. Nevertheless, this command has given us some really useful descriptive statistics `mood.gain` variable, broken down as a function of `therapy`.

A somewhat more general solution is offered by the `by()` function. There are three arguments that you need to specify when using this function: the `data` argument specifies the data set, the `INDICES` argument specifies the grouping variable, and the `FUN` argument specifies the name of a function that you want to apply separately to each group. To give a sense of how powerful this is, you can reproduce the `describeBy()` function by using a command like this:

```
by( data=clin.trial, INDICES=clin.trial$therapy, FUN=describe )
```

```
## clin.trial$therapy: no.therapy
##           vars n mean   sd median trimmed  mad min max range skew kurtosis
## drug*        1 9 2.00 0.87    2.0    2.00 1.48 1.0 3.0   2.0 0.00    -1.81
## therapy*     2 9 1.00 0.00    1.0    1.00 0.00 1.0 1.0   0.0  NaN      NaN
## mood.gain    3 9 0.72 0.59    0.5    0.72 0.44 0.1 1.7   1.6 0.51    -1.59
##              se
## drug*     0.29
## therapy*  0.00
## mood.gain 0.20
## ------------------------------------------------------------
## clin.trial$therapy: CBT
##           vars n mean   sd median trimmed  mad min max range  skew
## drug*        1 9 2.00 0.87    2.0    2.00 1.48 1.0 3.0   2.0  0.00
## therapy*     2 9 2.00 0.00    2.0    2.00 0.00 2.0 2.0   0.0   NaN
## mood.gain    3 9 1.04 0.45    1.1    1.04 0.44 0.3 1.8   1.5 -0.03
##           kurtosis   se
## drug*        -1.81 0.29
## therapy*       NaN 0.00
## mood.gain    -1.12 0.15
```

This will produce the exact same output as the command shown earlier. However, there's nothing special about the `describe()` function. You could just as easily use the `by()` function in conjunction with the `summary()` function. For example:

```
by( data=clin.trial, INDICES=clin.trial$therapy, FUN=summary )
```

```
## clin.trial$therapy: no.therapy
##        drug         therapy    mood.gain
##  placebo :3   no.therapy:9   Min.   :0.1000
##  anxifree:3   CBT        :0   1st Qu.:0.3000
##  joyzepam:3                   Median :0.5000
##                               Mean   :0.7222
##                               3rd Qu.:1.3000
##                               Max.   :1.7000
## ------------------------------------------------------------
## clin.trial$therapy: CBT
##        drug         therapy    mood.gain
##  placebo :3   no.therapy:0   Min.   :0.300
##  anxifree:3   CBT        :9   1st Qu.:0.800
##  joyzepam:3                   Median :1.100
##                               Mean   :1.044
##                               3rd Qu.:1.300
##                               Max.   :1.800
```

Again, this output is pretty easy to interpret. It's the output of the `summary()` function, applied separately to `CBT` group and the `no.therapy` group. For the two factors (`drug` and `therapy`) it prints out a frequency table, whereas for the numeric variable (`mood.gain`) it prints out the range, interquartile range, mean and median.

What if you have multiple grouping variables? Suppose, for example, you would like to look at the average mood gain separately for all possible combinations of drug and therapy. It is actually possible to do this using the `by()` and `describeBy()` functions, but I usually find it more convenient to use the `aggregate()` function in this situation. There are again three arguments that you need to specify. The `formula` argument is used to indicate which variable you want to analyse, and which

variables are used to specify the groups. For instance, if you want to look at `mood.gain` separately for each possible combination of `drug` and `therapy`, the formula you want is `mood.gain ~ drug + therapy`. The `data` argument is used to specify the data frame containing all the data, and the `FUN` argument is used to indicate what function you want to calculate for each group (e.g., the `mean`). So, to obtain group means, use this command:

```
aggregate( formula = mood.gain ~ drug + therapy,  # mood.gain by drug/therapy combina
           data = clin.trial,                      # data is in the clin.trial data f
           FUN = mean                              # print out group means
 )
```

```
##       drug    therapy mood.gain
## 1  placebo no.therapy  0.300000
## 2 anxifree no.therapy  0.400000
## 3 joyzepam no.therapy  1.466667
## 4  placebo        CBT  0.600000
## 5 anxifree        CBT  1.033333
## 6 joyzepam        CBT  1.500000
```

or, alternatively, if you want to calculate the standard deviations for each group, you would use the following command (argument names omitted this time):

```
aggregate( mood.gain ~ drug + therapy, clin.trial, sd )
```

```
##       drug    therapy mood.gain
## 1  placebo no.therapy 0.2000000
## 2 anxifree no.therapy 0.2000000
## 3 joyzepam no.therapy 0.2081666
## 4  placebo        CBT 0.3000000
## 5 anxifree        CBT 0.2081666
## 6 joyzepam        CBT 0.2645751
```