2015 ■ Vol. 11 ■ no. 3 DOI: 10.20982/tqmp.11.3.p127

GRD 2.0: An extended SPSS extension command for generating random data

Bradley Harding[a] and Denis Cousineau[a,✉]

[a]École de psychologie, Université d'Ottawa

Abstract ■ The GRD extension command for SPSS (Harding & Cousineau, 2014) has been used in a variety of applications since its inception. Ranging from a teaching tool to demonstrate statistical analyses, to an inferential tool used to find critical values instead of looking into a z-table, GRD has been very well received. However, some users have requested other data generation components that would make GRD a more complete extension command: the possibility to add contaminants to the generated dataset as well as the ability to generate correlated variables. Another component we added is a graphical user interface (or GUI) that makes GRD accessible through the drop-down menus in the SPSS Data Editor window. This GUI allows users to generate a simple dataset by entering parameters in dedicated fields rather than writing out the full script. Finally, we devised a small series of exercises to help users get acquainted with the new sub-commands and GUI.

Keywords ■ GRD, Graphical User Interface, Contaminants, Correlation, Sampling

✉ denis.cousineau@uottawa.ca

Introduction

Statistics are often seen as the black sheep of an undergraduate degree in Psychology with students sometimes developing anxiety towards the subject matter. For this reason, teachers often give students a series of datasets to practice concepts learnt in class on their own time. However, once the available datasets have been exhausted, novelty is thwarted as the students already know the answers. It is for this reason that we have developed an extension command for SPSS that generates random data, GRD (standing for Generator of Random Data; Harding and Cousineau (2014)). This extension command uses simple sub-commands and parameters written directly in SPSS Syntax and generates a completely different dataset every time the command is run. We have used this command extensively as a teaching tool in an introductory undergraduate statistics class, eliciting a strong positive reception. Some students even used the command to generate data similar to their homework's dataset to infer whether or not their answer could be correct. However, there are some limitations to the command as pointed out by several students and other users. GRD has no option to simulate multimodality and outliers short of inserting/modifying the data manually. In addition, GRD could not generate variables that are correlated with one another. We therefore added these options in this new version of GRD.

Another aspect on which we have received many comments on is the fact that the syntax language of SPSS can have a steep learning curve. In addition to learning a large amount of statistical concepts, learning the syntax of it all can be overwhelming for students - especially if all that is required is a simple generated dataset. Therefore we have added a graphical user interface (GUI hereafter) that makes the command more accessible to users that are not well versed in SPSS syntax. While syntax allows one to save the script and gives more control to the user, a GUI allows for a quick generation of a simple dataset. This new GUI is accessible through the drop-down menus.

The following tutorial is organized in three main sections. The first section shows how to use the two new options using SPSS syntax, as introduced in (Harding & Cousineau, 2014). The second section is an overview of the GUI for GRD, describing the input fields. The third section shows three examples of classroom-ready exercises promoting the teaching of sampling and the importance of random data to understand certain statistical concepts. Following the body of the tutorial, we give in Appendix A information on the /PRINT sub-commands of GRD. This optional sub-command allows the user to delve deeper in the GRD command and see its inner workings firsthand; it is also accessible via the GRD GUI.

Except for minor changes to the /PRINT sub-command, GRD 2.0 expands on the first version and is fully backwards compatible; any command written for the first version of GRD also works with GRD 2.0. For installation

details of the GRD command as well as a review of the possible sub-commands, see Harding and Cousineau (2014). Finally the graphs presented here were all generated using random seed 473 with the syntax command:

```
SET MTINDEX = 473.
```

Execute this command immediately prior to the GRD command if you want to exactly duplicate the figures presented here.

The two new options

Generating Contaminated Data

The teaching of contaminated data such as outliers and multimodality allows students to have a better understanding of inferential statistics and the normality assumption. However, in the previous version of the GRD command, multimodal distributions were only possible by copying previously saved data into a newly generated dataset. In addition, the only possible way to simulate outliers was by inserting or modifying the dataset manually. These methods are inefficient as they are not automatized. We therefore developed the /CONTAMINANTS sub-command.

In GRD, a theoretical population is used to sample data. This population could be called the regular population. With the new /CONTAMINANTS sub-command, it is possible to define an alternate population, that can be called the abnormal population, the outlier population, or more neutrally, the contaminant population. The proportion of contaminants should logically be smaller than 50%, although it is possible to sample more contaminants than observations from the regular population.

There are two methods for generating contaminated data with GRD, one being slightly simpler. In both of these approaches, one must specify the proportion of contaminants that are desired in relation to the entire sample size. This is done by writing the proportion parameter. This parameter is required and must be written following the /CONTAMINANTS sub-command. The number of contaminant is set by writing a proportion between 0 and 1 of the generated sample's size. For example, a generated sample size of 200 with a /CONTAMINANTS proportion of 0.1 will have 180 simulated subjects generated as usual and 20 simulated subjects that are generated using the parameters of the /CONTAMINANTS sub-command (10% of 200 is 20). When the proportion parameter is written with no other /CONTAMINANTS parameter, the contaminated data are generated using the default values noted below.

The simple method revolves around sampling contaminant subjects from a normal distribution specifying values for its parameters. These parameters are:

- Mean - sets the mean of the contaminants' population. If this parameter is not written, the default value is 0;
- stddev - sets the standard deviation of the contaminants' population. If this parameter is not written, the default value is 1;
- rho - sets the correlation between variables of the contaminants' population. This parameter will be explained in more detail in the next section. If this parameter is not written, the default value is 0.

For example, consider a sample of n = 200 in which the regular population has a mean of 100 and a standard deviation of 15. 10% of this dataset is contaminated by observations from a contaminant population having a mean of 180 and a standard deviation of 5. This sample could represent a normal classroom's result to an IQ test in which a small group of students are exceptionally gifted. Using the simple method, the command should look like:

```
GRD
 /SUBJECTSPERGROUP equal = 200
 /OVERALL mean = 100 stddev = 15
 /CONTAMINANTS mean = 180 stddev = 5
    proportion = 0.1.
```

The more complex, but more flexible, way of generating contaminated data is by using the population parameter. This parameter allows users to define the population's distribution and population parameters. This parameter works exactly the same as the population parameter from the /SCORES sub-command described in Harding and Cousineau (2014). This comes in handy when a distribution other than the normal distribution, /CONTAMINANTS's default distribution, is desired. This parameter overrides the mean, stddev, and rho parameters. To generate data that follow the same example as above using the more complex method, the command should look like:

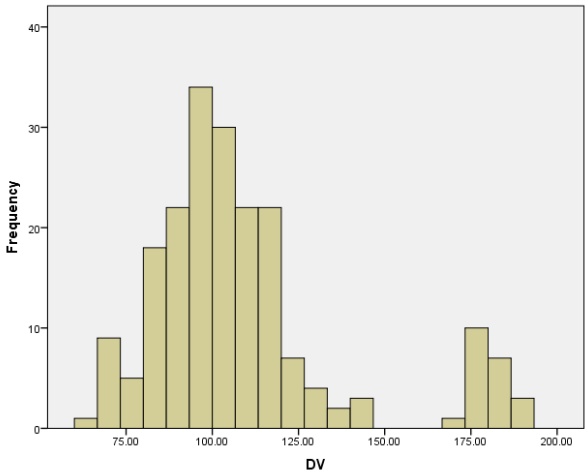
```
GRD
 /SUBJECTSPERGROUP equal = 200
 /OVERALL mean = 100 stddev = 15
 /CONTAMINANTS population = "RV.NORMAL
    (180,5)" proportion = 0.1.
```

To plot the distribution of the simulated dataset, execute the following command:

```
GRAPH
 /HISTOGRAM dv.
```

Figure 1 shows the result for one simulated dataset. As seen, the majority of the data have a score close to 100, but a small proportion of data have much higher scores, being around 180.

Figure 1 ■ Histogram of a dataset with contaminated data as generated by the GRD command. The dataset was generated with a mean of 100, a standard deviation of 15, and a sample size of 200. The contaminated data is generated with a mean of 180, a standard deviation of 5 and represent a proportion of 10% of the full dataset (20 subjects generated with the contaminants sub-command and 180 subjects generated as usual)



Generating Correlated Variables

Another aspect of GRD that was deemed missing is the ability to simulate correlated variables. A correlation is defined as the proportion of data from one variable that follows the same trend as another variable (Coladarci, Cobb, Minium, & Clarke, 2010) where the strength is interpreted by a number between 1 and -1 with 0 being a null correlation. To visualize the strength of a correlation when there are two variables, use a scatterplot in which the score of each variable is a coordinate on a grid. When all of the scores are plotted, the result is called a data cloud. This data cloud is what allows users to visualize correlation and estimate its strength and sign. Variables that are highly correlated have a data cloud that resembles a line. An upward-sloped line means that the variables are positively correlated and vice-versa. The data cloud of variables that are weakly correlated resembles a balloon with no clear trend. When there are more than two variables, it is possible to generate scatterplots for pairs of variables.

Correlated data can be sampled from multivariate distributions in which the strength of the correlation is one population parameter. Such multivariate distributions can be symmetrical or not, have thick or thin surroundings (equivalent to a high or small kurtosis in a univariate distribution), etc. However, as there is no multivariate distribution available in SPSS, we implemented one specifically for GRD. The parameters of this distribution are described below.

There are two ways to implement correlated data in GRD. The simple way is to specify the rho parameter in the /OVERALL sub-command. This approach generates correlated data in a multivariate normal distribution where both variables have the same mean and standard deviation.

For example, consider the following syntax which generates two variables with a mean of 100, a standard deviation of 15 that are correlated together with a rho of 0.9; note that the two correlated variables must be generated in a within-subject design:

```
GRD
 /SUBJECTSPERGROUP equal = 200
 /WSFACTORS Time (2)
 /OVERALL mean = 100 stddev = 15 rho =
   0.9.
```

To plot the simulated dataset in a scatterplot, write the following command:

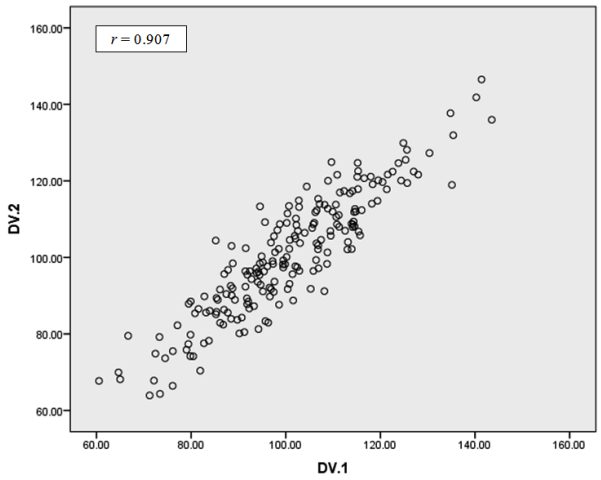
```
GRAPH
 /SCATTERPLOT dv.1 with dv.2.
```

The scatterplot in Figure 2 shows a sample composed of two variables with an observed correlation of 0.907. Correlations can be measured with the CORRELATIONS command, written as:

```
CORRELATIONS dv.1 dv.2.
```

The more complex way of simulating correlated data allows for more control of the simulated dataset by using

Figure 2 ■ Scatterplot of correlated data with identical axes using the simple method of generating correlation, the rho parameter. Both variables are generated with means of 100 and standard deviations of 15. The measured correlation is written in the top-left corner of the plot.



distinct means and distinct standard deviations for each variable. As was seen in the first version of the GRD command (Harding & Cousineau, 2014), the population parameter of the /SCORES sub-command allows users to sample data from a specific population. In GRD 2.0, it is possible to specify a multivariate normal distribution (this sub-command overrides the /OVERALL sub-command if it is present in the script) that will generate correlated variables. The multivariate normal distribution, noted RV.MVN, is written as

$$\text{RV.MVN}(\{\text{means}\},\{\text{covariance matrix}\})$$

in GRD syntax, where commas separate entries and semicolons separate lines in the matrix.

To explain this distribution with more details, we concentrate on the case with only two variables (a bivariate normal distribution). This distribution is written as

$$\text{RV.MVN}(\{\overline{X},\overline{Y}\},\{s_X^2,\text{cov}_{XY};\text{cov}_{XY},s_Y^2\})$$

where we use X to denote the first variable and Y to denote the second variable. This notation allows users to specify the mean for each variable in the first set of brackets (noted as and for the means of X and Y respectively). The standard deviations and correlation is found in the second set of brackets (a variance/covariance matrix). Although the variance/covariance matrix allows for more complex datasets than simply writing rho, the notation is more complex as well. The covariance matrix is equivalent to where is the variance of X, is the variance of Y and is the covariance between both variables. The correlation is found by dividing the covariance by the product of both variables' standard deviation as shown in Equation 1:

$$r_{XY} = \frac{\text{cov}_{XY}}{s_X s_Y} \tag{1}$$

where r_{XY} is the coefficient of correlation between both variables, is the covariance between X and Y and , are the standard deviations of X and Y respectively. If we isolate the covariance in (1) we get:

$$\text{cov}_{XY} = r_{XY} s_X s_Y \tag{2}$$

Therefore, the variance/covariance matrix is equivalent to

$$\{s_X{}^{**}2, r_{XY}{}^{*}s_X{}^{*}s_Y; r_{XY}{}^{*}s_X{}^{*}s_Y, s_Y{}^{**}2\} \tag{3}$$

The bivariate normal distribution then becomes:

$$\text{RV.MVN}(\{\overline{X},\overline{Y}\},\{s_X{}^{**}2, r_{XY}{}^{*}s_X{}^{*}s_Y; r_{XY}{}^{*}s_X{}^{*}s_Y, s_Y{}^{**}2\}).$$

Negative correlations are generated by using a negative covariance. In addition, in bivariate situations, it is imperative that the covariances in the matrix be equal to one another.

For example, to generate data where X has a mean of 100, a standard deviation of 15, and Y has a mean of 200, a standard deviation of 20, with both variables correlated with r = 0.9, one would write 100, 200 for the means and as the covariance matrix. The complete GRD script is thus:

```
GRD
 /SUBJECTSPERGROUP equal = 200
 /WSFACTORS Time (2)
 /SCORES population =
```

```
    "RV.MVN({100,200},
   {15**2,0.9*15*20;0.9*15*20,20**2})".
GRAPH
 /SCATTERPLOT dv.1 with dv.2.
CORRELATIONS dv.1 dv.2.
```

Note that everything within quotes must appear on a single line in the SPSS syntax editor.

One sample is shown in Figure 3. As seen, both variables have different means as well as different standard deviations. The measured correlation (r = 0.907) is shown in the top-left corner of the plot.

Contaminants can also follow a multivariate normal distribution. As aforementioned, /CONTAMINANTS has the rho parameter. When this parameter is present, the correlated contaminated data have the same proportion of outliers, the same mean, and the same standard deviation for both variables. /CONTAMINANTS can also be sampled with the population parameter. When the population parameter is written, the notation is the same as that of the /SCORES population parameter written above. For example, if one were to contaminate 20% of the data presented in Figure 3 with cases having a mean of 50 for X and 300 for Y, a standard deviation of 10 and 5 for X and Y respectively, and a correlation of rho = 0.5 between both variables, one would write the following:

```
GRD
 /SUBJECTSPERGROUP equal = 200
 /WSFACTORS Time (2)
 /SCORES population =
   "RV.MVN({100,200},
   {15**2,.9*15*20;.9*15*20,20**2})"
 /CONTAMINANTS proportion = .2
   population =
   "RV.MVN({50,300},
   {10**2,.5*10*5;.5*10*5,5**2})".
GRAPH
 /SCATTERPLOT dv.1 with dv.2.
CORRELATIONS dv.1 dv.2.
```

As seen in Figure 4, the majority of the data follows a positive correlation with a small set of data (the contaminants) present in the top-left corner of the plot. Although the data and contaminants have a strongly positive relation on X and Y, the presence of the outliers makes for an overall negative correlation (r = -0.516, noted in the top-right corner of the plot). This correlation is therefore uninterpretable as the whole dataset do not satisfy the normality assumption.

Easier access to the GRD command

Syntax can be an overwhelming endeavor to learn for new students in statistics, especially when it is the student's first contact with SPSS. Therefore, we added a GUI to GRD to facilitate the use of random number generators in the classroom. The new GUI for GRD is an intuitive alternative to the written command that allows students to generate a dataset by filling input fields rather than writing out a code. While we argue that syntax remains a tool worth learning for undergraduate students, occasional use of GRD might benefit from a more user-friendly interface (a graphical user interface, abbreviated GUI). All of the GRD sub-commands presented in this article as well as the ones presented in Harding and Cousineau (2014) are present and accessible in this GUI.

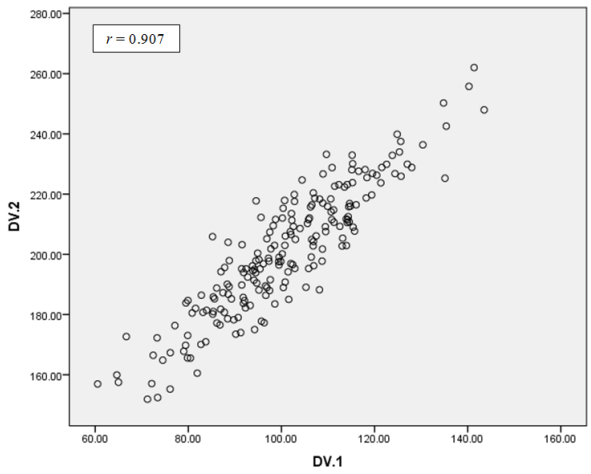
The GUI is installed automatically with the GRD bundle. Follow the steps given in Appendix A from Harding and Cousineau (2014) with the new bundle freely available on this journal's web site. After a program restart, go to the "Utilities" drop-down menu as is seen in Figure 5 and GRD 2.0 is then ready for use.

Figure 6 shows the opened GUI window of the GRD command.

When the command window is opened, one can see the following fields and buttons:

1. Sample Size: In this field one sets how many subjects are generated in each independent group. By default the groups are equal; to generate unequal groups, uncheck the "all groups equal" field and write the desired number of subjects per independent group separated by colons. This sub-command is the only required information for GRD to run - when the field is left empty, the "Ok" button is disabled.
2. The four buttons at the bottom: At the bottom of the GRD window there are four buttons.
   (a) The "OK" button generates the sample when it is pressed; the default sample has a mean of 0 and a standard deviation of 1 unless otherwise specified by the "Population Parameters" field explained in 3.
   (b) The "Paste" button takes the parameters that have been filled in and copies them to the nearest Syntax window (if none are open, this button opens a new Syntax window). This button is useful to modify GRD parameters and create more complex datasets.
   (c) The "Reset" button returns all of the GUI fields to their default blank position.
   (d) The "Cancel" button cancels the command and exits to the Data Editor window - all changes are lost.
3. Population Parameters: In these fields, one sets the parameters for the population that the generated dataset is sampled from. Here, one sets the mean, the standard deviation as well as the correlation coefficient when there is a within-subject experimental design (as explained in 5). GRD generates samples from a normal distribution by default. For other distributions,

Figure 3 ■ Scatterplot of correlated data with identical axes using the more complex method of generating correlation, the population parameter. Variable 1 is generated with a mean of 100 and a standard deviation of 15 and variable 2 is generated with a mean of 200 and a standard deviation of 20. The measured correlation is shown in the top-left corner of the plot.



press the "Optional" button explained in 6 or press the "Paste" button in the bottom of the window and adjust manually in the syntax.

4. Between-Subject Factors: This field allows users to generate a between-subject experimental design. Users can specify a name as well as how many groups each factor has. In the GUI, users can create up to two different between-subject factors. If more factors are needed, press the "Paste" button and add more manually in the syntax. However, GRD can only generate a maximum of 4 different factors (regardless of if they are between- or within-subject factors)
5. Within-Subject Factors: This field allows users to generate a within-subject experimental design. Users can specify a name as well as how many repeated measures each factor has. In the GUI, users can create up to two different within-subject factors. If more factors are needed, likewise use the "Paste" button. Again, GRD can only generate a maximum of 4 different factors (regardless of if they are between- or within-subject factors)
6. Optional Buttons on the right: In this dialog window, users have access to:
   (a) The "Random Seed" field where one can choose a specific seed to be generated.
   (b) The "Population Function" field allows users to manually change the population's shape. When this field is filled, it overrides the "Population Parameters" fields from 3.
   (c) The "Print Information" checkboxes allow users to get information on the "Debug information", the "GRD version", and the "Full Instruction Generated". These technical commands are new in GRD 2.0 and are explained further in Appendix A. There is also the option to print the "Effect Size Table" which allows users to print the population effect size when effects are defined (see 8).
7. Contaminants Button: This button on the main window brings users to a dialog window where one can add contaminants do the data. Here one sets the proportion of contaminants, the mean, the standard deviation, and the correlation coefficient for the contaminant population. Again, when the rho field is filled out, there needs to be a within-subject experimental design. There is also the option to set the contaminants' population's distribution. When this parameter is filled out, it overrides the other three contaminants' population shape parameters.
8. Define Effects: This button on the main window brings users to a dialog window where it is possible to add an effect to a factor. In the "Factor Name" field, one writes the name of the between- or within-subject factor where an effect is desired. The factor's name must be written exactly the same as it is in GRD's main window. In the "Effect-Type" drop down menu one sets what type of effect is desired. In the "Value" entry field, one sets the value of the factor's effect. It is possible to add a total of two effects on the four possible generated factors. To add more than two effects, modify the GRD syntax manually. For more information on the ef-

Figure 4 ■ Scatterplot of a dataset generated with contaminated data. In the top-left of the data cloud, there is contaminated data that was generated using the population parameter. As is seen, the presence of the contaminated data heavily skews the correlation (in the top-left corner of the plot) of an otherwise well-correlated dataset.

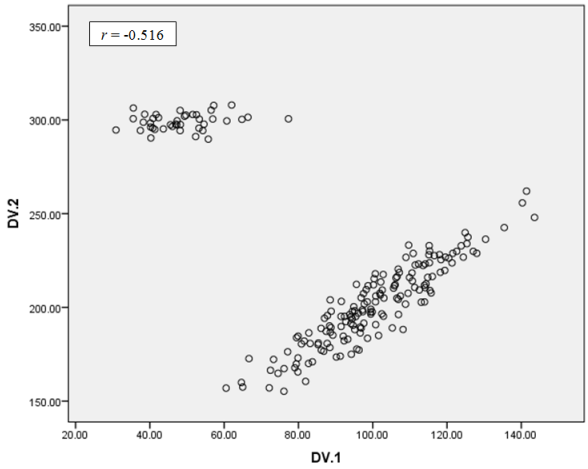


Figure 5 ■ Where GRD is nested in the drop-down menu.

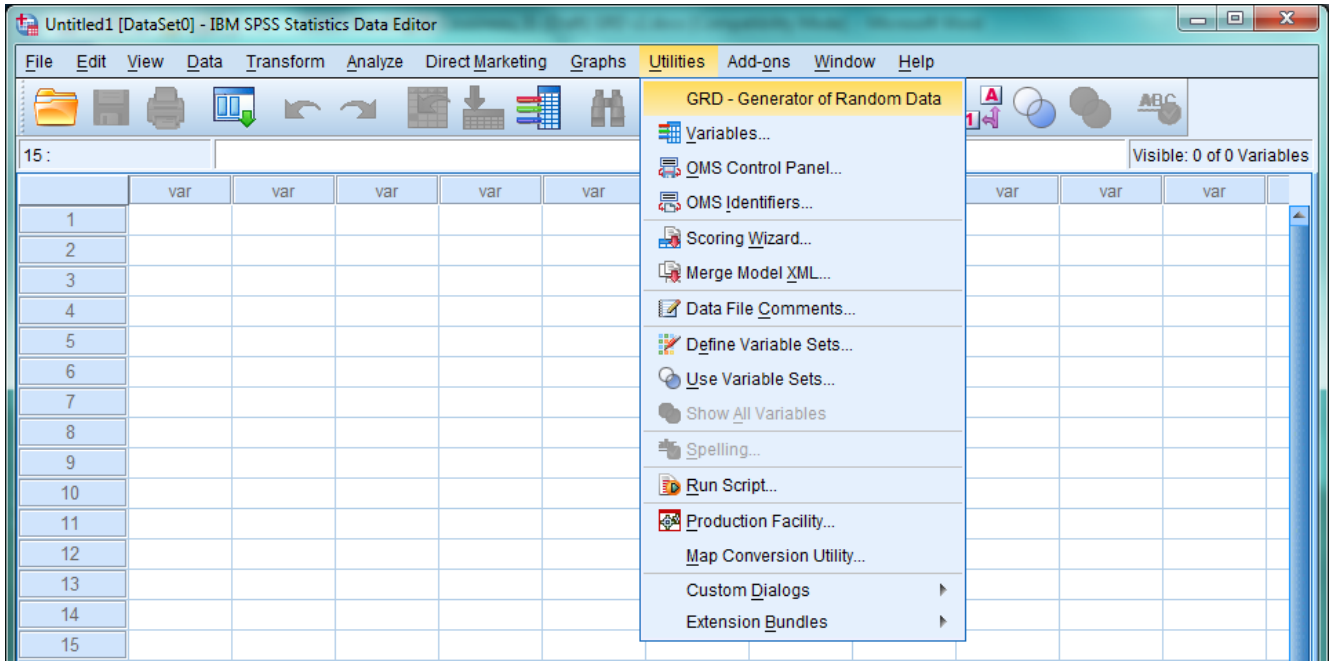
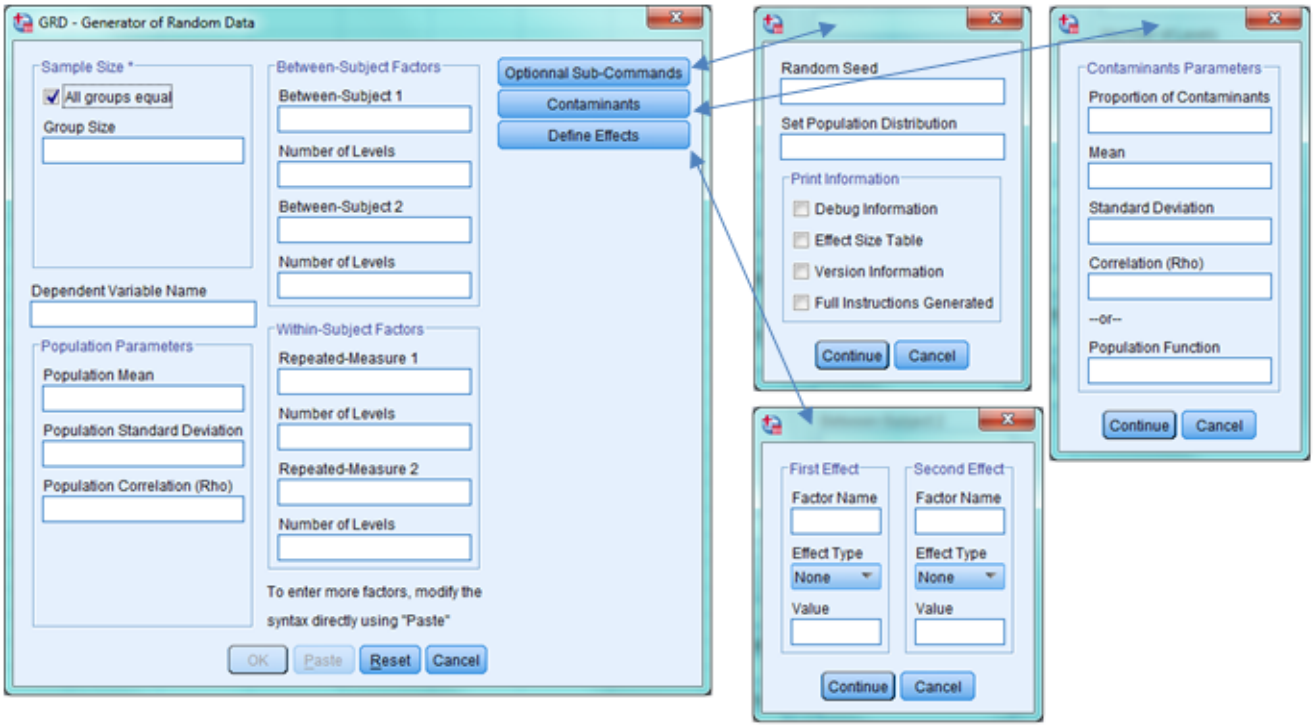


Figure 6 ■ GRD GUI window as well as the dialog windows associated with each button.



fect types and their characteristics, consult Harding and Cousineau (2014).

Exercises

In this section we present a series of exercises for GRD that can be implemented in a classroom setting. We first provide context to why the exercise is important followed by the Syntax and GUI inputs for each (both options can be simply copied and pasted into their respective SPSS window). In these exercises, the parameters we propose are guidelines and there exist a variety of strategies to implement each. In addition, we encourage teachers to create their own exercises and encourage an active learning classroom, as is recommended by the GAISE College Report (American Statistical Association et al., 2005, recommendation 4).

How contaminated data affect descriptive statistics

An exercise to show how contaminants affect the descriptive statistics of a dataset is to have students generate datasets with varying levels of imperfection; encouraging students to see that every decision made on a sample must be made with a critical eye. For example, samples that do not respect the normality assumption cannot be studied with standard descriptive statistics. In this exercise, we encourage students to generate datasets with varying proportions of contaminants (10%, 25%, and 50%) and note the mean of the sample. The mean is supposed to show the representative score of a dataset, the score that is most likely to be sampled (Watier, Lamontagne, & Chartier, 2011). When outliers are present, the measured mean does not reflect the score that is most likely to be sampled as the mean is not a robust statistic (Harding, Tremblay, & Cousineau, 2014); especially when the outliers are far away from the central peak of the distribution. Students should be able to see the change in the mean as the proportion of contaminants increases and come to the conclusion that the mean is not accurately portraying the central tendency. Students should also be able to come to the conclusion that the normality assumption is essential for the interpretation of descriptive statistics.

```
GRD
 /SUBJECTSPERGROUP equal = 200
 /OVERALL mean = 100 stddev = 15
 /CONTAMINANTS mean = 180 stddev = 5
    proportion = 0.1.
GRAPH
 /HISTOGRAM dv.
```

where the mean is measured with:

```
MEANS dv.
```

GUI fields:

- Sample Size: 200
- Population Parameters;
  - Mean: 100
  - Standard deviation: 15
- Contaminants;
  - Contaminants Proportion: 0.1 (Change this value)
  - Population parameters;
    - Mean: 180
    - Standard deviation: 5

As seen in Figure 7, when the proportion of contaminants increases, the mean of the sample (noted in the top-right of each histogram) does not represent an accurate portrayal of the most-likely sampled score.

How varying levels of correlations influences the data cloud

Correlations can be a source of anxiety for students. A way to alleviate this anxiety is to have students repeatedly generate datasets with varying levels of correlations. In this exercise we encourage students to generate datasets with varying degrees of correlation (both positive and negative). The student should be able to come to the conclusion that the correlation is proportional to the elongation of the data cloud; the more the scatterplot resembles a line the higher the correlation is; conversely, the more the scatterplot resembles a circle, the lower the correlation is. Students should also come to the conclusion that small correlations (rho = 0.10), regardless of polarity (+ or -) are hard to interpret and differentiate from one another.

```
GRD
 /SUBJECTSPERGROUP equal = 200
 /WSFACTORS X (2)
 /OVERALL mean = 100 stddev = 15 rho =
   0.10.
GRAPH
 /SCATTERPLOT dv.1 with dv.2.
CORRELATIONS dv.1 dv.2.
```

GUI Entry Fields:

- Sample Size: 200
- Population Parameters;
  - Mean: 100
  - Standard Deviation: 15
  - Rho: 0.1 (Change this value)
- Within-Subject Factor;
  - Name 1: X
  - Number of levels 1: 2

As seen in Figure 8, panel A and B, when the correlation is low (rho = 0.1) between variables the correlation is hard to interpret and difficult to see. When rho = 0.5, the correlation is easier to distinguish and differentiate between signs (panel C and D). When the correlation is high between variables (rho = 0.9) the scatterplot resembles a line (panel E and F); these correlations are easy to differentiate between one another.

Bivariate outliers

In this demonstration we invite students to generate multivariate outliers. Univariate outliers are easier to detect because they have extremely low or high scores. Thus, sorting the scores is a simple and effective way to locate univariate outliers. Multivariate outliers are more subtle. When looking at each individual variable's distribution, none has extreme scores yet when they are plotted together; the multivariate outliers do not fit the trend of the majority of the data. In the example presented here, we generated data with multivariate outliers and plotted each variable individually and in a scatterplot. Students should be able to come to the conclusion that while the means of the outlier population is moved away from the mean of the regular population, the correlation coefficient becomes uninterpretable. This demonstration can therefore be used as a teaching tool for students to see how a correlation can relate to each individual variable's distribution.

Syntax:

```
GRD
 /SUBJECTSPERGROUP equal = 1000
 /WSFACTORS X (2)
 /SCORES population =
   "RV.MVN
    ({0,0},{5**2,.99*5*5;.99*5*5,5**2})"
 /CONTAMINANTS population =
   "RV.MVN
    ({-5,5},{1**2,-.01*1*1;-.01*1*1,1**2})
    " PROPORTION = .5.
GRAPH
 /SCATTERPLOT dv.1 with dv.2.
CORRELATIONS dv.1 dv.2.
```

GUI Entry Fields:

- Sample Size: 1000
- Within-Subject Factor;
  - Name 1: X
  - Number of Levels 1: 2
- Optional;
  - Population Function: RV.MVN(0,0, 5**2, .99*5*5; .99*5*5, 5**2)
- Contaminants;
  - Contaminants Proportion: 0.5
  - Population Function: RV.MVN(-5,5, 1**2, -.01*1*1; -.01*1*1, 1**2)

In the syntax, change the position of the contaminant population from being the same as the regular population

Figure 7 ■ Contaminated data exercise with the measured mean of each generated dataset located in the top-right of each histogram. The dataset is generated with a mean of 100, a standard deviation of 15, and a sample size of 200. Each dataset has generated contaminated data with a mean of 180, a standard deviation of 5, and an increasing proportion of contaminated data, where A) has a proportion of 0.1, B) has a proportion of 0.25, and C) has a proportion of 0.5.

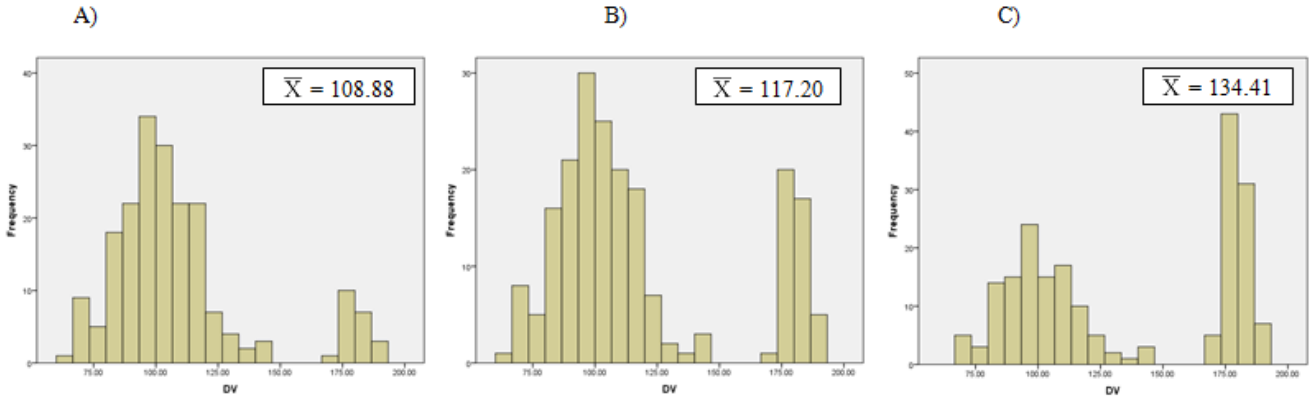


Figure 8 ■ Correlated data exercise where students must visually interpret the correlation's strength and direction. The datasets have variables generated with a mean of 100, a standard deviation of 15 and a correlation of A) 0.1, B) -0.1, C) 0.5, D) -0.5, E) 0.9, and F) -0.9. The measured correlations are written in the top-left corner of each plot.

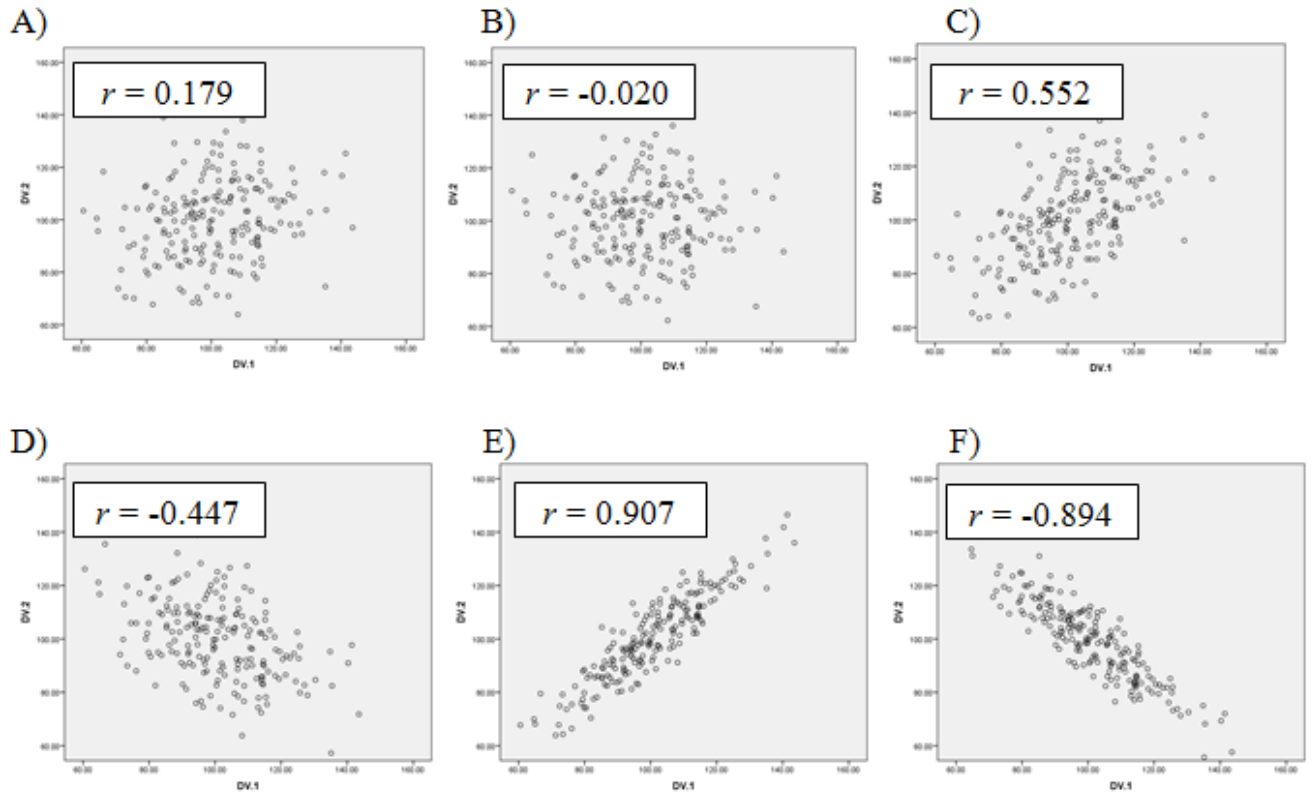
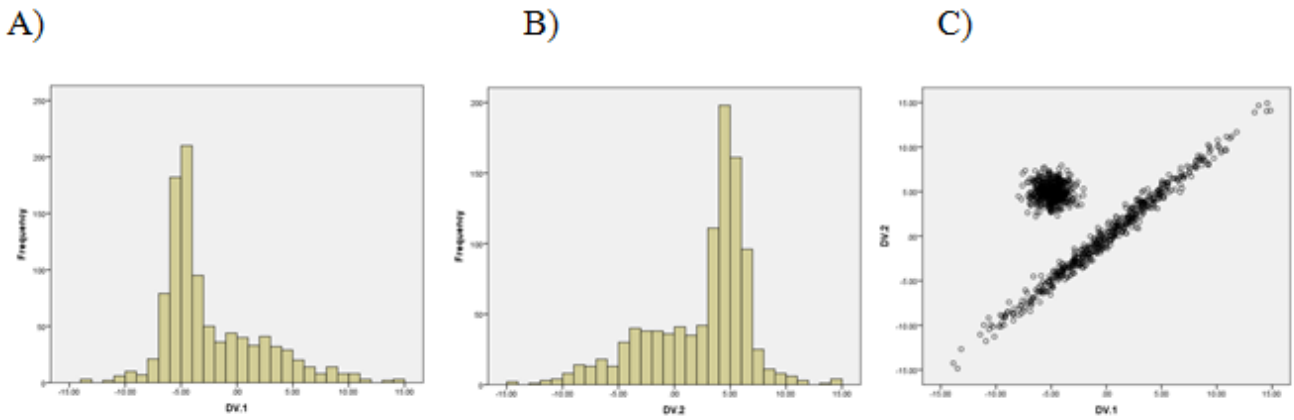


Figure 9 ■ Figure 9. Example of multivariate outliers where each individual variable respects the normality assumption yet go against the homogeneity of variance assumption when paired. Panel A shows the histogram for the first variable, B) the histogram for the second variable, and C) the scatterplot for both.



0,0 to much different -5,5 by increment of 1 (i.e., 0,0, -1,1, -2,2, etc.). At what point do the multivariate outliers influence markedly the correlation? Are they easily detected on a scatter plot?

Conclusion

Here we presented two new options to GRD as well as an overview of a GUI. This update to the GRD command addresses comments made by users of the first version of GRD (Harding & Cousineau, 2014). We also provided a series of exercises that aim to inspire teachers and students alike to think of statistics as a series of constructs rather than simply as equations written on a page.

References

American Statistical Association, Aliaga, M., Cobb, G., Cuff, C., Garfield, J., Gould, R., ... Witmer, J. (2005). *Guidelines for assessment and instruction in statistics education [gaise] college report*. Alexandria, VA.: American Statistical Association.

Coladarci, T., Cobb, C. D., Minium, E. W., & Clarke, R. C. (2010). *Fundamentals of statistical reasoning in education* (3rd ed.). Hoboken, NJ: John Wiley & Sons.

Harding, B. & Cousineau, D. (2014). GRD: an SPSS extension command for generating random data. *The Quantitative Methods for Psychology, 10(2)*, 80–94.

Harding, B., Tremblay, C., & Cousineau, D. (2014). Standard errors: a review and evaluation of standard error estimators using monte carlo simulations. *The Quantitative Methods for Psychology, 10(2)*, 107–123.

Watier, N., Lamontagne, C., & Chartier, S. (2011). What does the mean mean? *Journal of Statistics Education, 19(2)*, 1–20.

Appendix A: Technical sub-command for GRD

Here we show how to use technical sub-commands for GRD that allow users to delve deeper in the command's inner workings. In this section we give a short description of these sub-commands as well as their Syntax - they are all nested under the /PRINT sub-command. These commands are also available in the "Optional Sub-commands" side window of the GUI as checkboxes as explained in the second section of this tutorial.

PRINT version

Here one can check which version of GRD is installed. This command comes in handy when one wants to verify that the command is installed properly. For users of the first version of GRD, this command is useful to make sure that the new version of GRD is installed. It also displays copyright information and how to cite the command if necessary. The sub-command should be added in SPSS Syntax like this:

```
...
/PRINT version
...
```

PRINT debug

Here one can check if the sub-commands are running properly. When this command is present, the Output window for SPSS shows each of the individual sub-commands and if they have successfully parsed. This command could be useful if one has made a mistake in generation of a complex dataset and cannot find where the error has occurred.

```
...
/PRINT debug
...
```

PRINT fullinstructions

Here one can see the steps that GRD goes through to generate a dataset. When this command is written, the SPSS Output file shows a series of code that is the backbone of GRD. This sub-command is useful for users who wish to see the inner workings of the command and be inspired to create extension commands of their own.

```
...
/PRINT fullinstructions
...
```

PRINT es

Here one can see the effect size table of a dataset generated by GRD. For this sub-command to run, it is imperative that an effect has been requested on at least one of the experimental design's factors; if no effects are requested, this command does not return any results. When an effect is specified, the generated table shows the synopsis of each group's change in mean in relation the population parameters set in the command. For example, a factor generated with a mean of 100 and an effect size of 10 signifies that that factor's group has a generated mean of 110 (100+10). All factors are noted in the table along with each factor's group's effect size is separated by a comma.

```
...
/PRINT es
...
```

Citation

Harding, B., & Cousineau, D. (2015) GRD 2.0: An extended SPSS extension command for generating random data. *The Quantitative Methods for Psychology, 11*(*3*), 127-138.

Received: 20/07/2015