Cohen d’ (one-sample) (es_cohen_d_os)

Author: P. Stikker
Website: https://peterStatistics.com
YouTube: https://www.youtube.com/stikpet
Version: 0.1 (2023-01-31)

Introduction

The *es_cohen_d_os* function calculates a Cohen d’ effect size, used for one-sample situations.

This document contains the details on how to use the functions, and formulas used in them.

1 About the Function

1.1 Input parameters:

- **data**
  - Excel: a specific range with the numeric scores
  - Python: a pandas series with the numeric scores
  - R: a vector with the numeric scores

- *Optional parameters*
  - **mu**
    the hypothesized mean. If not specified the midrange will be used.
  - **qual**
    the rule-of-thumb method to use for the classification (see *th_cohen_d* for details)

1.2 Output

- **d’**
  the effect size measure
- **classification**
  classification/qualification of the effect size

Note for *Excel:*

the array function *es_cohen_d_os_arr* will require 2 rows and 2 columns.

1.3 Dependencies

- **Excel**
  - *th_cohen_d* from *thumb_cohen_d* for the qualification
  - Optional you can run the **es_cohen_d_os_addHelp** macro so that the function will be available with some help in the ‘User Defined’ category in the functions overview.
- **Python**
  The following libraries are needed:
  - pandas is needed for data entry and showing the results
  - *th_cohen_d* from *thumb_cohen_d* for the qualification
- **R**
  - *th_cohen_d* from *thumb_cohen_d* for the qualification

2 Examples

2.1 Excel

	A	B	C	D	E	F	G
1	Over_Grade						
2	20						
3	50		hyp. Mean.	70			
4	80						
5	15		Cohen d'				
6	40		value	-0,59534	=es_cohen_d_os(A2:A21;D3)		
7	85		qual	large	=es_cohen_d_os(A2:A21;D3;;C7)		
8	30						
9	45		Cohen d'	Qualification			
10	70		-0,0253334	negligible			
11	60						
12	90		C9:D10 =>	=es_cohen_d_os_arr(A2:A21)			
13	25						
14	40						
15	70						
16	65						
17	70						
18	98						
19	40						
20	65						
21	60						

2.2 Python

```
[1]: from eff_size_cohen_d_os import es_cohen_d_os
     import pandas as pd

     dataList = [20, 50, 80, 15, 40, 85, 30, 45, 70, 60, 90, 25, 40, 70, 65, 70, 98, 40, 65, 60]
     data = pd.Series(dataList)

     es_cohen_d_os(data)
```

[1]:

	d'	Classification
0	0.025333	negligible

```
[2]: es_cohen_d_os(data, mu=70)
```

[2]:

	d'	Classification
0	0.595336	very large

2.3 R

```
> source("thumb_cohen_d.R")
> source("eff_size_cohen_d_os.R")
>
> data <- c(20, 50, 80, 15, 40, 85, 30, 45, 70, 60, 90, 25, 40, 70, 65, 70, 98, 40, 65, 60)
> es_cohen_d_os(data)
          d' qualification
1 0.02533345    negligible
> es_cohen_d_os(data, mu=70)
        d' qualification
1 0.595336         large
```

3 Details of Calculations

$$d' = \frac{|\bar{x} - \mu_{H0}|}{s}$$

To convert this to a regular Cohen's d for interpretation use:

$$d = d' \times \sqrt{2}$$

Symbols:

- $\bar{x}$ the sample mean
- μ_{H0} the hypothesized mean in the population
- s the unbiased sample standard deviation

4 Sources

For the one-sample case (Case 3), we define

(2.3.3) $$\mathbf{d}_3' = \frac{\mathbf{m} - \mathbf{c}}{\sigma}$$

ever, the latter is readily compensated for. For the one-sample case, use the power tables with **n** and

(2.3.4) $$\mathbf{d} = \mathbf{d}_3'\sqrt{2}.$$

(Cohen, 1988, p. 46)

References

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). L. Erlbaum Associates.