

Mean, Median and Mode

Objectives

- Produce a program following the criteria (provided below) which reliably calculates the correct results and outputs them in the expected format.
- Demonstrate your ability to solve the given problem proficiently in PHP.
- Implement an efficient sort algorithm in native PHP code.
- Create code which is sufficiently documented and readable to someone unfamiliar with your solution.

Explanation of task

Consider the sample data below:

Record	Sale Amount
1	144.00
2	197.00
3	76.00
4	151.00
5	233.00
6	229.00
7	70.00
8	60.00
9	71.00
10	233.00

From the sample data above, it is possible to produce the following statistics:

Total value of sales:	£1,464.00
Average (mean) value of sales:	£ 146.40
Sales values that occur most often:	£ 233.00
These sales values occur:	2 times
Median value of sales:	£ 147.50

Requirements and constraints

All submissions are required to follow these rules:

- Must be self-contained, no external dependencies.
- Must execute on PHP 7.1, your solution may additionally support other PHP versions.
- We will be testing against PHP with a 4GB memory limit set.
- Must contain a script named `tech-test.php` which will be the script we execute, you may include other scripts.
- Must read data in from a file called `testdata.csv` from the same directory as itself.
- Must output *to the screen* a valid JSON document matching the below example:

```
{
  "total": 3141.59,
  "mean": 271.82,
  "modal": [161.80],
  "frequency": 42,
  "median": 299.79
}
```

Please note the output does not have to be “pretty printed”, but the outputted JSON must match this structure.

- Must round floating point values to two decimal places.
- Must not use any built-in sort functions.
- Must not use any built-in statistical functions.
- Must not use any external application, for example MySQL or Excel.
- Must ignore the first row *if* it contains column headers.

Prohibited functions

A non-exhaustive list of prohibited functions:

- `array_count_values()`
- `array_multisort()`
- `array_product()`
- `array_sum()`
- `array_unique()`
- `arsort()`
- `asort()`
- `krsort()`
- `ksort()`
- `max()`
- `min()`
- `natcasesort()`
- `natsort()`
- `rsort()`
- `sort()`
- `uasort()`
- `uksort()`
- `usort()`

Resources

We will test your submissions against several data files, we have provided two of these for you to aid you in developing your program. Remember your program *must* read data from a file named `testdata.csv`