## CS 524 Task 4: Monte Carlo Sensitivity Analysis [UPDATE 1]

### Description

The task focuses on a basic deductive, abductive, and inductive approach to sensitivity analysis by combining brute force with lightweight Monte Carlo methodology to tease out the landscape of hidden contributing factors for all combinations of inputs. The premise is to evaluate the performance of a single type of notional projectile with respect to its mass, velocity, drag, and stability characteristics out to 1,000 yards. Performance is measured in terms of the distance between the actual and expected impact points. (The target and aiming process are not considered.)

The projectile is subject to external influences that may effect its baseline performance based on probability. As such, some experiments may require multiple runs to tease out the true performance characteristics. Use basic statistics to make sense of the results.

### Setup

The ballistics model `PerformanceMapper` is defined in package `cs524task4` in jar file `cs524task4.jar`. In Eclipse, add it as an external jar with the Build Path→Confgure Build Path→Libraries option. The process is similar in other IDEs.

To access the model, create one or more performance mappers with the desired test configuration. For any distance input, the output will be the targeting error resulting from a single shot. Combining multiple performance measures, calculating and exporting the results, and visualizing and analyzing them are up to you.

The following example fires one shot at 10 yards and measures the effect of stability performance given certain hidden external influences:

```
PerformanceMapper mapper =
  new PerformanceMapper(PerformanceMapper.E_Data.STABILITY,
                        PerformanceMapper.E_PerturbationMode.MYSTERY_1);

double distanceYards = 10;

double errorInches = mapper.map(distanceYards);
```

### Requirements

For all graphs, the independent axis (or axes) is distance from 0 to 999 yards, and the dependent axis is error in inches. Plot both the average error and standard deviation. Be sure to explain how to interpret the results. For each graph, indicate the "sweet spot," if there is one.

In all parts, M, V, D, and S respectively correspond to `MASS`, `VELOCITY`, `DRAG`, and `STABILITY` in `E_Data`.

For Parts 1 through 4 use perturbation mode `MYSTERY_1`.

Part 1.A

Plot the individual performance of M, V, D, and S in four graphs. The sample size is 1.

Part 1.B

Plot the individual performance of M, V, D, and S in four graphs. The sample size is 10.

Part 1.C

Plot the individual performance of M, V, D, and S in four graphs. The sample size is 1000.

Compare and discuss the results from A, B, and C. What can you conclude about the hidden external influences?

Part 2

Plot the paired performance of MV, MD, MS, VD, VS, and DS in six graphs. The sample size is 1000.

For each pairing, indicate the "sweet spot" of minimum error with respect to distance, if there is one. What does the sweet spot mean?

Part 3

Plot the grouped performance of MVD, MVS, MDS, and VDS in four graphs. The sample size is 1000.

Part 4

Plot the grouped performance of MVDS in a graph. The sample size is 1000.

Parts 5.A, 5.B, 5.C, 6, 7, and 8

Repeat Parts 1, 2, 3, and 4 respectively with perturbation mode `MYSTERY_2`. What can you conclude about the hidden external influences?

Parts 9.A, 9.B, 9.C, 10, 11, and 12

Repeat Parts 1, 2, 3, and 4 respectively with perturbation mode `MYSTERY_3`. What can you conclude about the hidden external influences?

Part 13

Why are we not able to analyze accuracy and precision?

## Deliverables

Submit your source code and a professional PDF report with meaningful, labeled graphs that addresses all the parts as specified.