**Martin Packaging Metric José Morgado 59457**

**Summary:**

* Efferent Coupling (Ce): This metric quantifies the number of classes outside the module that directly depend on classes within the module. It measures the outgoing dependencies from a module.
* Afferent Coupling (Ca): Afferent Coupling counts the number of classes outside the module that depend directly on classes within the module. It measures the incoming dependencies to a module.
* Instability (I): Instability is a metric that represents the module's tendency to change. It's calculated as the ratio of efferent coupling to the sum of efferent and afferent couplings (Ce / (Ce + Ca)). A higher instability suggests the module is more prone to change.
* Abstractness (A): Abstractness measures the ratio of the number of abstract classes/interfaces in a module to the total number of classes/interfaces in that module. It indicates how abstract or concrete a module is.
* Normalized Distance from Main Sequence (D): This metric determines how far a module is from the optimal balance between abstraction and stability, represented by the Main Sequence. It's calculated as |A + I - 1|, where A is abstractness and I is instability. Modules closer to the Main Sequence are considered more ideal in terms of design.

Data Visualization -

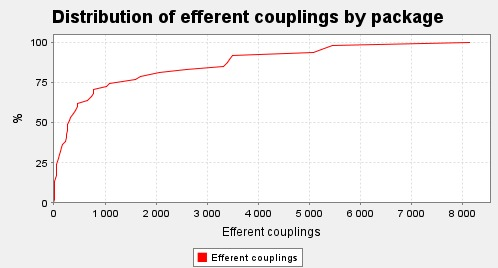


Fig. 1 - Efferent Coupling (Ce)

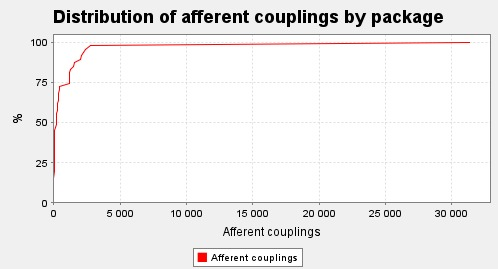


Fig. 2 - Afferent Coupling (Ca)

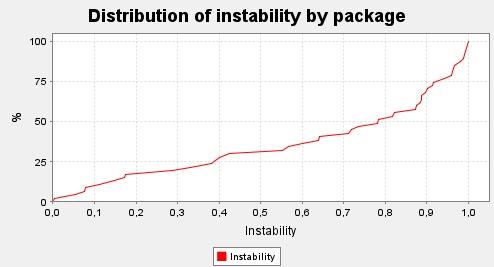


Fig. 3 - Instability (I)

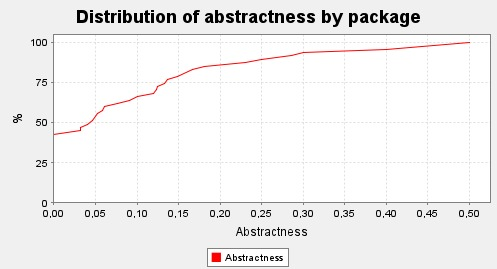


Fig. 4 - Abstractness (A)

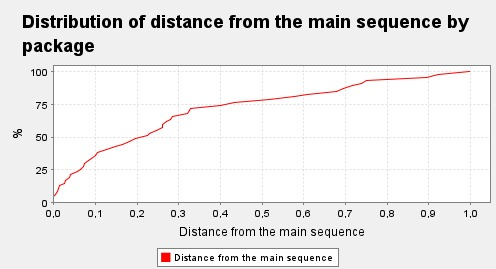


Fig. 5 - Instability (I)

**Discussion:**

**Average Efferent Couplings (ce = 1178.09):**

The average efferent couplings suggest that, on average, the classes in the project depend on many external classes. This indicates a high degree of class dependency on other parts of the system.

**Average Afferent Couplings (ca = 1178.09):**

The average afferent couplings indicate that, on average, the classes in the project have many external dependencies. This might suggest an architecture where various parts of the system heavily rely on a central set of classes.

**Average Instability (i = 0.51):**

The average instability indicates that, on average, the classes in the project are moderately unstable, with a moderate propensity for changes. A value close to 0.5 suggests a reasonable balance between stability and instability.

**Average Abstractness (a = 0.08):**

The average abstractness suggests that, on average, the classes in the project are relatively non-abstract, with more concrete implementations than abstract methods.

**Average Distance (d = 0.29):**

The average distance suggests that, on average, the classes in the project are in a reasonable balance between abstraction and coupling, although there might be room for improvements.