Martin Package Metrics

Pedro Arruda 60663

# Metrics:

* Abstractness (A)
* Afferent couplings (Ca)
* Efferent Couplings (Ce)
* Distance from the main sequence (D)
* Instability (I)

# Introduction

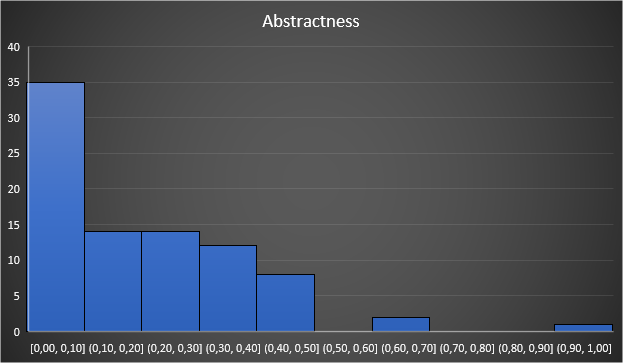
In this report I will talk about the set of metrics that I thought would be useful to find code smells. This set of metrics is called the Martin Package Metrics and it is commonly used to assess the relationship between packages and identify poorly designed ones.

# Abstractness (A)

Abstractness is the ratio of the number of abstract classes and interfaces in the package to the total number of all classes. This metric is used to measure the degree of abstraction of the package.

Preferred values for the metric A should take extreme values close to 0 or 1:

* Having A = 0 means it is completely concrete.
* Having A = 1 means it is completely abstract.



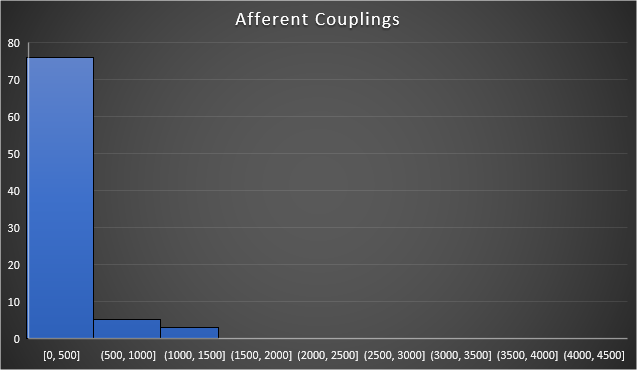
*Abstractness Metric Histogram*

Looking at the metric histogram, it is possible to see that the most common values are between 0 and 0.10, although, there are also a considerable number of packages with values between 0.10 and 0.50, which might not be a very good thing.

# Afferent Couplings (Ca)

This metric calculates the number of Afferent Couplings for each package. An Afferent Coupling is a reference from a class or interface external to the package to a class or interface internal to the package.

* Having high values for Ca means a lot of classes outside the package depend on it.
* Having low values for Ca means few classes outside the package depend on it.



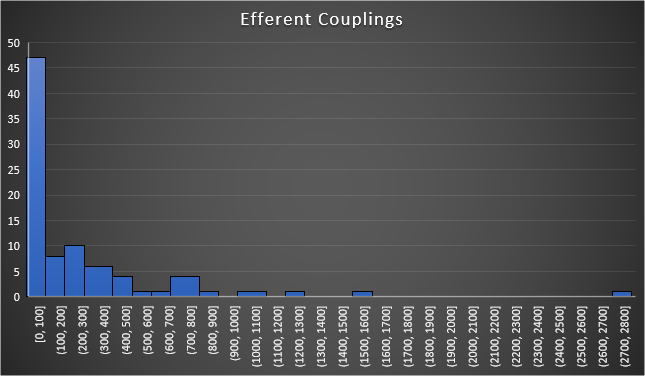
*Afferent Couplings Metric Histogram*

Looking at the metric histogram, it is possible to see that the most common values are between 0 and 500, which most likely means a good thing since the values are low.

# Efferent Couplings (Ce)

Contrary to the Ca metric, Efferent Coupling is the number of classes outside the package on which the package depends upon.

* Having high values for Ce means the package depends on a lot of classes outside of it.
* Having low values for Ce means the package depends on few classes outside of it.



*Efferent Couplings Metric Histogram*

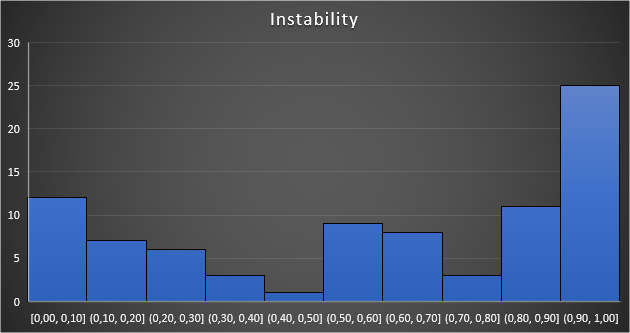
Looking at the metric histogram, it is possible to see that the most common values are between 0 and 100, which are low values, meaning a good thing. We can also see that more packages have “worse” values when compared to the Ca metric though. It might be worth to pay attention to those packages as to not let those values rise more.

# Instability (I)

This metric is used to measure the relative susceptibility of class to changes. According to the definition instability is the ratio of outgoing dependencies to all package dependencies.

Preferred values for the metric I are the same as for the A metric:

* Having I = 0 indicates a stable package.
* Having I = 1 indicates a maximally instable package.

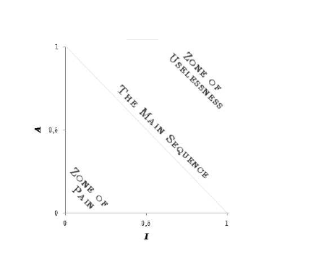


*Instability Metric Histogram*

Looking at the metric histogram, it is possible to see that the values are very distributed. The fact that there are a decent number of values in the middle might alert us to some poorly designed packages.

# **[Normalized Distance from Main Sequence](https://kariera.future-processing.pl/wp-content/uploads/2015/06/Normalized-Distance-from-Main-Sequence.jpg)**Distance from the main sequence (D)

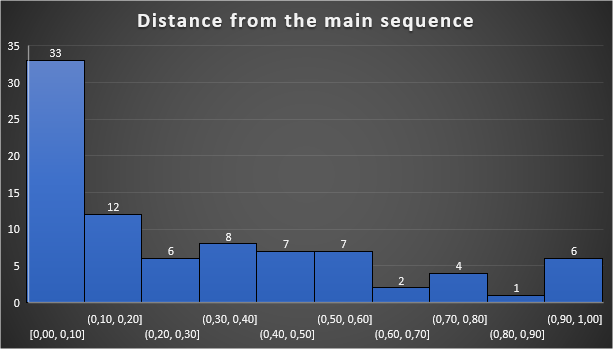
This metric can be considered the most important one, since it calculates the balance between stability and abstractness.





The value of D can be interpreted as the distance to the main sequence (which is the line that connects the points A=1; I=0 and A=0; I=1) so its value should be as low as possible.

A D closer to 1 means that the package is poorly designed.



*Distance from the main sequence Metric Histogram*

# Conclusion

When analyzing the last histogram (related to the D metric), we can see that most of the packages seem to be very well designed.

Nevertheless, the number of packages close to the value of 1 led me to investigate these packages further to find out what could be poorly designed.

Uma imagem com mesa

Descrição gerada automaticamente



When looking at the packages with very high values of D (> 0.600) we can see that all of them also have very low values of both abstractness and instability, meaning that they are all either close or located in the “Zone of Pain” – refer to the IA-graph in the section above – meaning that these packages are very rigid. They are difficult to extend because of their lack of abstractness and hard to change because of their responsibilities.

This could be considered a code smell and should be investigated by either turning the packages less rigid or by reconsidering if this packages really are necessary.

# Bibliography

* “A Validation of Martin’s Metric” by Sami Hyrynsalmi and Ville Lepp¨anen;
* Future processing blog – “Object-oriented metrics by Robert Martin”;
* Wikipedia – “Software package metrics”