

# Visibility Metrics

## References

- [Visibility Metrics and the Importance of Hiding Things](#)
- [Calculate metrics](#)
- [Controlling Access to Members of a Class](#)
- [Neo4j Python Driver](#)

## Relative Visibility Of Types

A Java class or interface may be declared with the modifier `public`, in which case it is visible to all classes everywhere. If a class or interface has no modifier (the default, also known as package-private), it is visible only within its own package.

The relative visibility is the number of inner components that are visible outside (public) divided by the number of all types:

$$relativevisibility = \frac{public\ types}{all\ types}$$

Using package protected types is one of many ways to improve encapsulation and implementation detail hiding.

## How to apply the results

The relative visibility is between zero (all types are package protected) and one (all types are public). A value lower than one means that there are types that are declared package protected. The lower the value is, the better implementation details are hidden.

Non public classes can't be accessed from another package so they can be changed without affecting code in other packages. They clearly indicate functionality that only belongs to one package. This also motivates to use more classes and to split up code into smaller pieces with a single responsibility and reason to change.

Table 1a - Top 40 artifacts with lowest median of package protection

# encapsulation

This table shows the relative visibility statistics aggregated for all packages per artifact and focusses on artifacts with many packages and hardly any package protected types (lowest median, high visibility). Package protected types would help to improve encapsulation.

Only the top 40 entries are shown. The whole table can be found in the following CSV report:

Global\_relative\_visibility\_statistics\_for\_types

| artifact | all | public | min | max | average | percentile25 | percentile50 | percentile75 | percentile90 |
|----------|-----|--------|-----|-----|---------|--------------|--------------|--------------|--------------|
|----------|-----|--------|-----|-----|---------|--------------|--------------|--------------|--------------|

## Table 1b - Top 40 artifacts with highest median of package protection encapsulation

This table shows the relative visibility statistics aggregated for all packages per artifact and focusses on artifacts with many packages and the highest median of package protected types (low visibility). Package protected types help to improve encapsulation.

Only the top 40 entries are shown. The whole table can be found in the following CSV report:

Global\_relative\_visibility\_statistics\_for\_types

| artifact | all | public | min | max | average | percentile25 | percentile50 | percentile75 | percentile90 |
|----------|-----|--------|-----|-----|---------|--------------|--------------|--------------|--------------|
|----------|-----|--------|-----|-----|---------|--------------|--------------|--------------|--------------|

## Table 1 Chart 1 - Relative visibility in artifacts

```
/home/runner/miniconda3/envs/codegraph/lib/python3.11/site-packages/pandas/plotting/_matplotlib/core.py:1259: UserWarning: No data for colormapping provided via 'c'. Parameters 'cmap' will be ignored
    scatter = ax.scatter(
<Figure size 640x480 with 0 Axes>
```

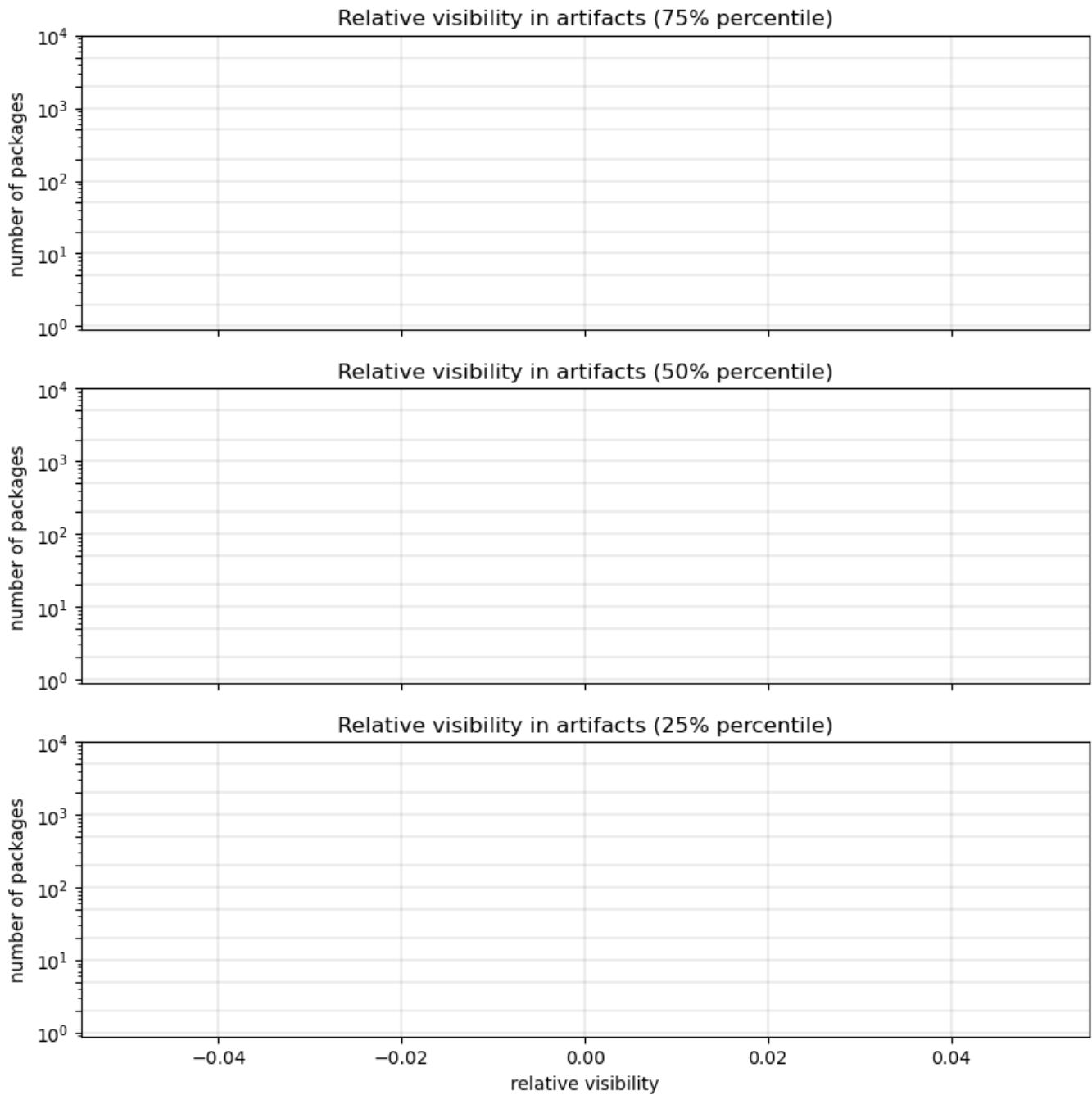


Table 2a - Top 40 packages with the highest visibility and lowest encapsulation

This table shows the relative visibility statistics per packages and artifact and focusses on packages with many types, hardly any package protected ones and therefore the highest relative visibility (lowest encapsulation). Package protected types would help to improve encapsulation.

Only the top 40 entries are shown. The whole table can be found in the following CSV report:

`Relative_visibility_public_types_to_all_types_per_package`

| artifactName | fullQualifiedPackageName | packageName | publicTypes | allTypes | relativeVisibility |
|--------------|--------------------------|-------------|-------------|----------|--------------------|
|--------------|--------------------------|-------------|-------------|----------|--------------------|

# Table 2b - Top 40 packages with the lowest visibility and highest encapsulation

This table shows the relative visibility statistics per packages and artifact and focusses on packages with many types, many package protected ones and therefore the lowest relative visibility (highest encapsulation). Package protected types help to improve encapsulation. Zero percent visibility and therefore packages with no public visible type are suspicious to be dead code.

Only the top 40 entries are shown. The whole table can be found in the following CSV report:

Relative\_visibility\_public\_types\_to\_all\_types\_per\_package

| artifactName | fullQualifiedPackageName | packageName | publicTypes | allTypes | relativeVisibility |
|--------------|--------------------------|-------------|-------------|----------|--------------------|
|--------------|--------------------------|-------------|-------------|----------|--------------------|

## Table 2 Chart 1 - Relative visibility of packages

```
/home/runner/miniconda3/envs/codegraph/lib/python3.11/site-packages/pandas/plotting/_matplotlib/core.py:1259: UserWarning: No data for colormapping provided via 'c'. Parameters 'cmap' will be ignored
    scatter = ax.scatter(
<Figure size 640x480 with 0 Axes>
```

