Visibility Metrics for Typescript

References

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

Relative Visibility Of Elements

A Typescript element (variable, function, class, ...) may be exported in which case it is visible and can be imported everywhere (if there are no other rules defined). If there is no "export" keyword and the element (variable, function, class, ...) is only declared, then it is only visible within the file or module.

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

$$relativevisibility = \frac{exported\ elements}{all\ declared\ elements}$$

Using directories with an index file as a module and exporting only the elements (variables, function, classes, ...) that the caller of the module should use is a good way to improve encapsulation and implementation detail hiding.

How to apply the results

The relative visibility is between zero (no element is exported) and one (all elements are exported). A value lower than one means that there are elements that are not exported. The lower the value is, the better the encapsulation and the better the implementation details are hidden.

Non exported elements can't be accessed from another modules so they can be changed without affecting code in other modules. They clearly indicate functionality that only belongs to one modules. This also motivates to split up code into smaller pieces with a dedicated reason to change (single responsibility).

Table 1a - Top 40 projects with lowest median of module encapsulation

This table shows the relative visibility statistics aggregated for all modules per project and focusses on projects with many modules and hardly any non-exported elements (lowest median, high visibility). Module directories with an index file and intentional exporting helps 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_elements_for_Typescript`

	projectPath	all	exported	min	max	average	percentile25	percentile50	percentile75	percentile90	percentile95	percentile99
0	/home/runner/work/code-graph-analysis-pipeline...	6	6	1.000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1	/home/runner/work/code-graph-analysis-pipeline...	15	12	0.800	0.800000	0.800000	0.800000	0.800000	0.800000	0.800000	0.800000	0.800000
2	/home/runner/work/code-graph-analysis-pipeline...	85	41	0.375	0.507246	0.441123	0.408062	0.441123	0.474185	0.494022	0.500634	0.505924

Table 1b - Top 40 projects with highest median of module encapsulation

This table shows the relative visibility statistics aggregated for all modules per project and focusses on project with many modules and the highest median of non-exported elements (variables, functions, classes, ...) (low visibility). Module directories with an index file and intentional exporting helps 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_elements_for_Typescript`

	projectPath	all	exported	min	max	average	percentile25	percentile50	percentile75	percentile90	percentile95	percentile99
0	/home/runner/work/code-graph-analysis-pipeline...	85	41	0.375	0.507246	0.441123	0.408062	0.441123	0.474185	0.494022	0.500634	0.505924
1	/home/runner/work/code-graph-analysis-pipeline...	15	12	0.800	0.800000	0.800000	0.800000	0.800000	0.800000	0.800000	0.800000	0.800000
2	/home/runner/work/code-graph-analysis-pipeline...	6	6	1.000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Table 1 Chart 1 - Relative visibility in projects

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

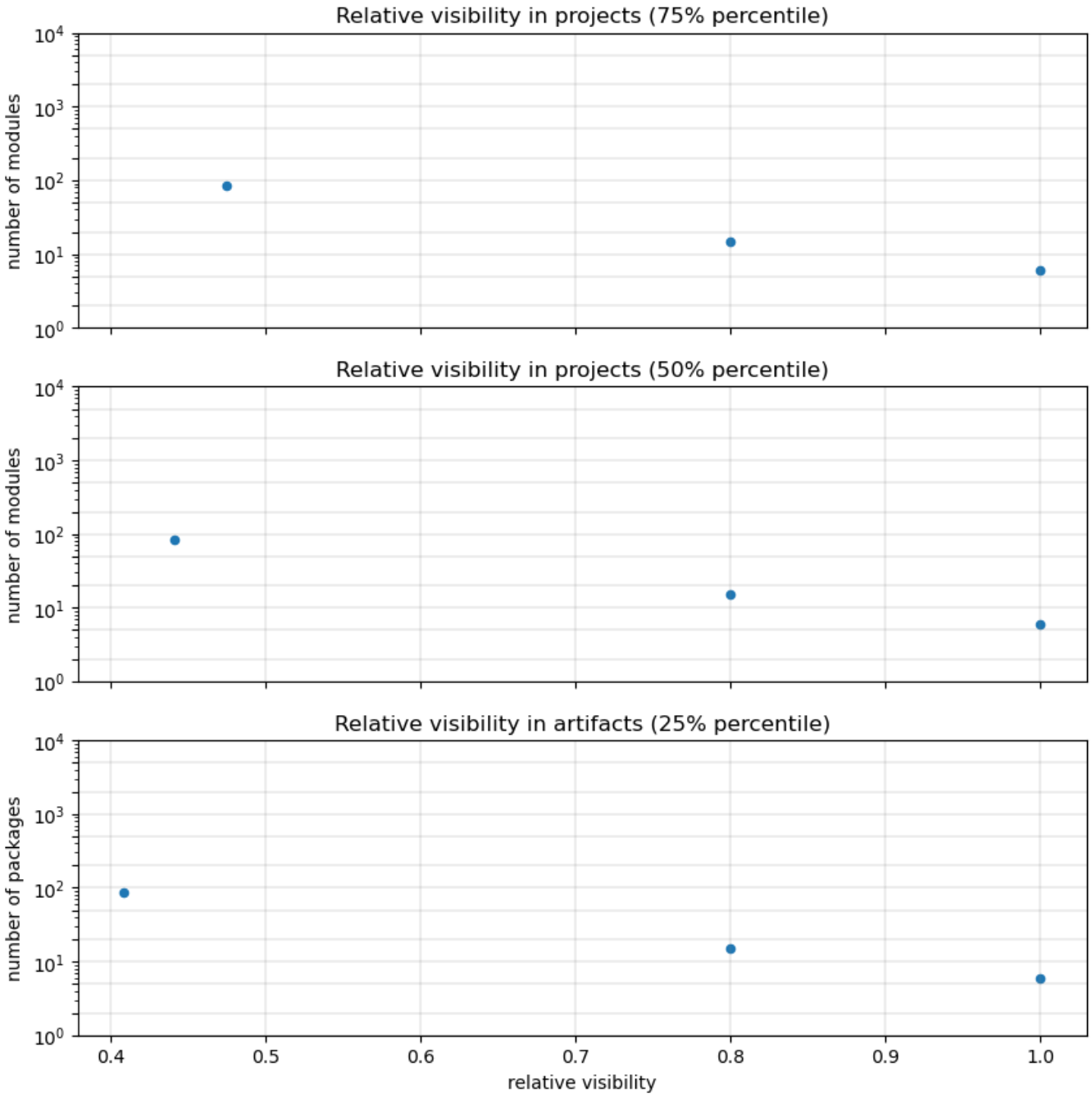

Table 2a - Top 40 modules with the highest visibility and lowest encapsulation

This table shows the relative visibility statistics per module and project and focusses on modules with many elements (variables, functions, classes, ...), hardly any non-exported ones and therefore the highest relative visibility (lowest encapsulation). Module directories with an index file and intentional exporting helps to improve encapsulation.

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

	projectPath	modulePath	moduleName	exportedElements	allElements	relativeVisibility
0	/home/runner/work/code-graph-analysis-pipeline...	index.ts	react-router	6	6	1.000000
1	/home/runner/work/code-graph-analysis-pipeline...	index.tsx	react-router-native	13	15	0.866667
2	/home/runner/work/code-graph-analysis-pipeline...	index.tsx	react-router-dom	35	69	0.507246
3	/home/runner/work/code-graph-analysis-pipeline...	server.tsx	server	6	16	0.375000

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

This table shows the relative visibility statistics per modules and project and focusses on modules with many elements (variables, functions, classes, ...), many non-exported ones and therefore the lowest relative visibility (highest encapsulation). Non-exported elements help to improve encapsulation. Zero percent visibility and therefore modules with no exported elements are suspicious to contain 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`

	projectPath	modulePath	moduleName	exportedElements	allElements	relativeVisibility
0	/home/runner/work/code-graph-analysis-pipeline...	server.tsx	server	6	16	0.375000
1	/home/runner/work/code-graph-analysis-pipeline...	index.tsx	react-router-dom	35	69	0.507246
2	/home/runner/work/code-graph-analysis-pipeline...	index.tsx	react-router-native	13	15	0.866667
3	/home/runner/work/code-graph-analysis-pipeline...	index.ts	react-router	6	6	1.000000

Table 2 Chart 1 - Relative visibility of modules

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