Overview in General

This file contains a general overview of the data in the graph including node labels and relationships types.

References

- jqassistant
- Neo4j Python Driver

Node Labels

Table 1a - Highest node count by label combination

Lists the 30 label combinations with the highest number of nodes. The labels with the lowest node count are listed in table 1b. The total list would sum up to the total number of labels (100%).

The whole table can be found in the CSV report `Node_label_combination_count`.

	nodeLabels	nodesWithThatLabels	nodesWithThatLabelsPercent
0	[Type, TS, Primitive]	811	13.503164
1	[Json, Key]	668	11.122211
2	[Json, Value, Scalar]	603	10.039960
3	[Type, TS, Declared]	598	9.956710
4	[TS, ExternalDeclaration]	444	7.392607
5	[NPM, Dependency]	338	5.627706
6	[Type, TS, ObjectMember]	318	5.294705
7	[Type, TS, Literal]	274	4.562105
8	[Type, TS, Union]	246	4.095904
9	[TS, Property]	137	2.281052
10	[Json, Value, Object]	133	2.214452
11	[Value, TS, Literal]	124	2.064602
12	[Type, Object, TS]	109	1.814852
13	[TS, Function]	109	1.814852
14	[NPM, Script]	91	1.515152
15	[Value, TS, ObjectMember]	88	1.465201
16	[Type, TS, FunctionParameter]	80	1.332001
17	[TS, Parameter]	76	1.265401
18	[Type, TS, Function]	70	1.165501
19	[File, Directory, Local]	64	1.065601
20	[TS, Variable]	59	0.982351
21	[File, TS, Local, Module]	46	0.765901
22	[TS, Interface]	37	0.616051
23	[File, Directory]	34	0.566101
24	[Type, TS, Intersection]	34	0.566101
25	[Project, TS]	33	0.549451
26	[TS, ExternalModule]	33	0.549451
27	[TS, TypeAlias]	32	0.532801
28	[Value, TS, Declared]	30	0.499500
29	[Package, File, Json, NPM]	29	0.482850

Chart 1a - Highest node count by label combination

Values under 0.5% will be grouped into "others" to get a cleaner plot. The group "others" is then broken down in Chart 1b.

```
<Figure size 640x480 with 0 Axes>
```

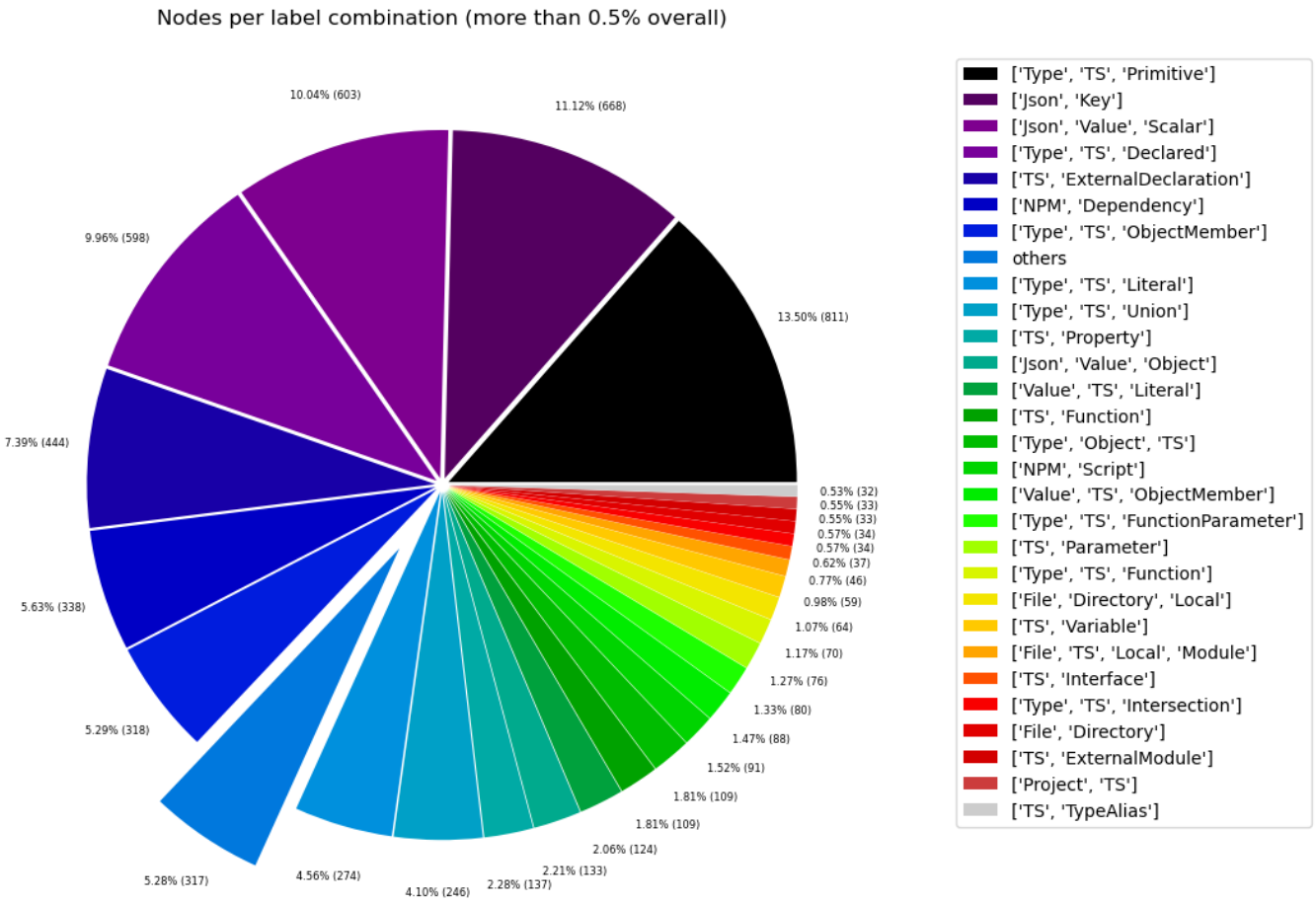


Table 1b - Lowest node count by label combination

Lists the 30 label combinations with the lowest number of nodes until they reach 0.5% of the total node count, which are shown above.

	nodeLabels	nodesWithThatLabels	nodesWithThatLabelsPercent
0	[Analyze, Task, jQAssistant]	1	0.01665
1	[Value, TS, Null]	1	0.01665
2	[TS, Class]	2	0.03330
3	[TS, Constructor]	2	0.03330
4	[TS, Enum]	4	0.06660
5	[TS, Method]	4	0.06660
6	[Value, Array, TS]	5	0.08325
7	[Type, TS, Tuple]	6	0.09990
8	[NPM, Engine]	6	0.09990
9	[TS, TypeParameter]	8	0.13320
10	[Value, TS, Complex]	11	0.18315
11	[Type, TS, TypeParameterReference]	12	0.19980
12	[Json, Value, Array]	12	0.19980
13	[Value, TS, Function]	13	0.21645
14	[Value, TS, Call]	14	0.23310
15	[Value, TS, Member]	14	0.23310
16	[TS, EnumMember]	16	0.26640
17	[jQAssistant, Rule, Concept]	19	0.31635
18	[Type, TS, NotIdentified]	23	0.38295
19	[Value, Object, TS]	28	0.46620
20	[File, Local]	28	0.46620
21	[File, TS, Scan]	29	0.48285
22	[Package, File, Json, NPM]	29	0.48285
23	[Value, TS, Declared]	30	0.49950

Chart 1b - Lowest node count by label combination

Shows the lowest (less than 0.5% overall) node count label combinations. Therefore, this plot breaks down the "others" slice of the pie chart above. Values under 0.01% will be grouped into "others" to get a cleaner plot.

```
<Figure size 640x480 with 0 Axes>
```

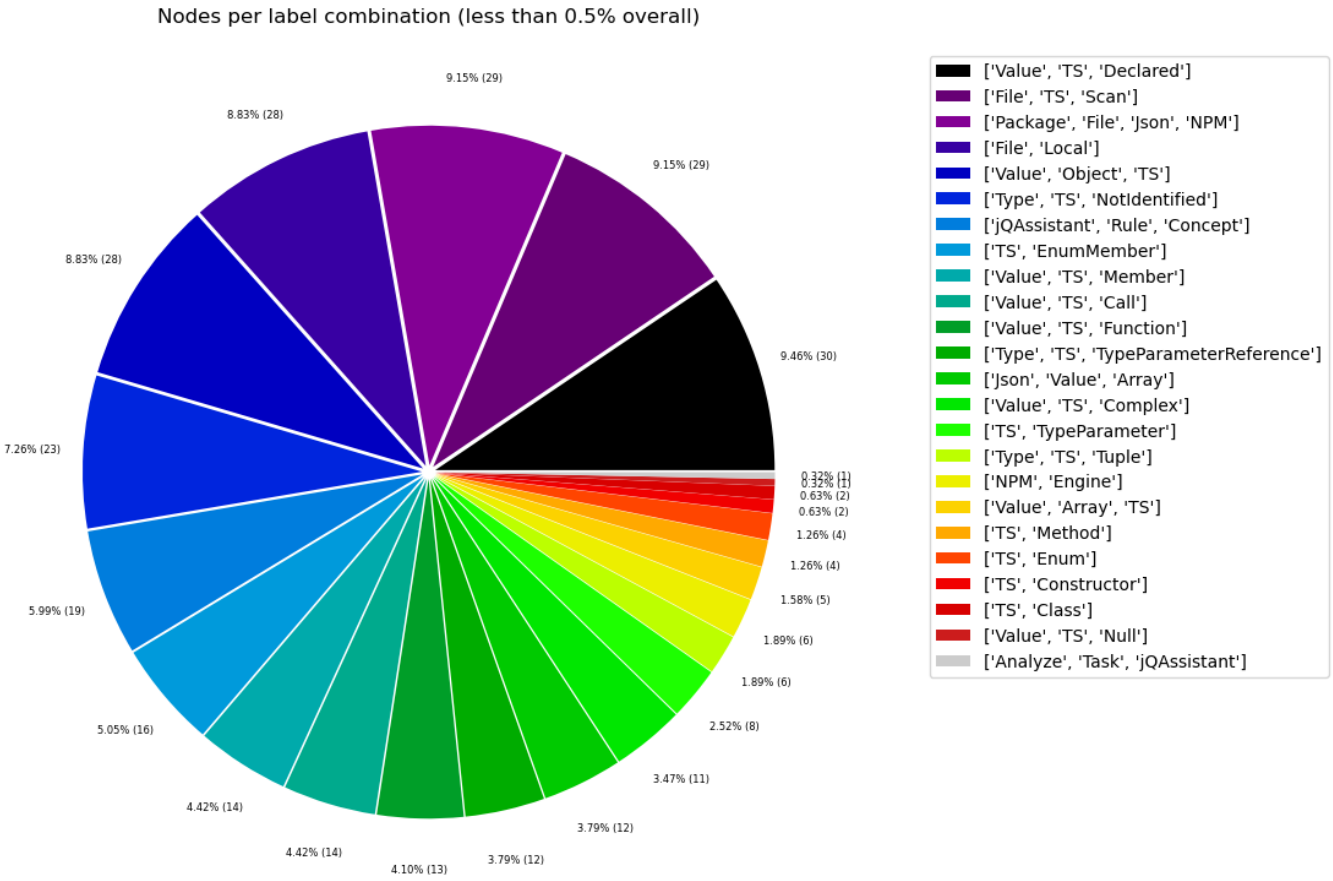


Table 1c - Highest node count by single label

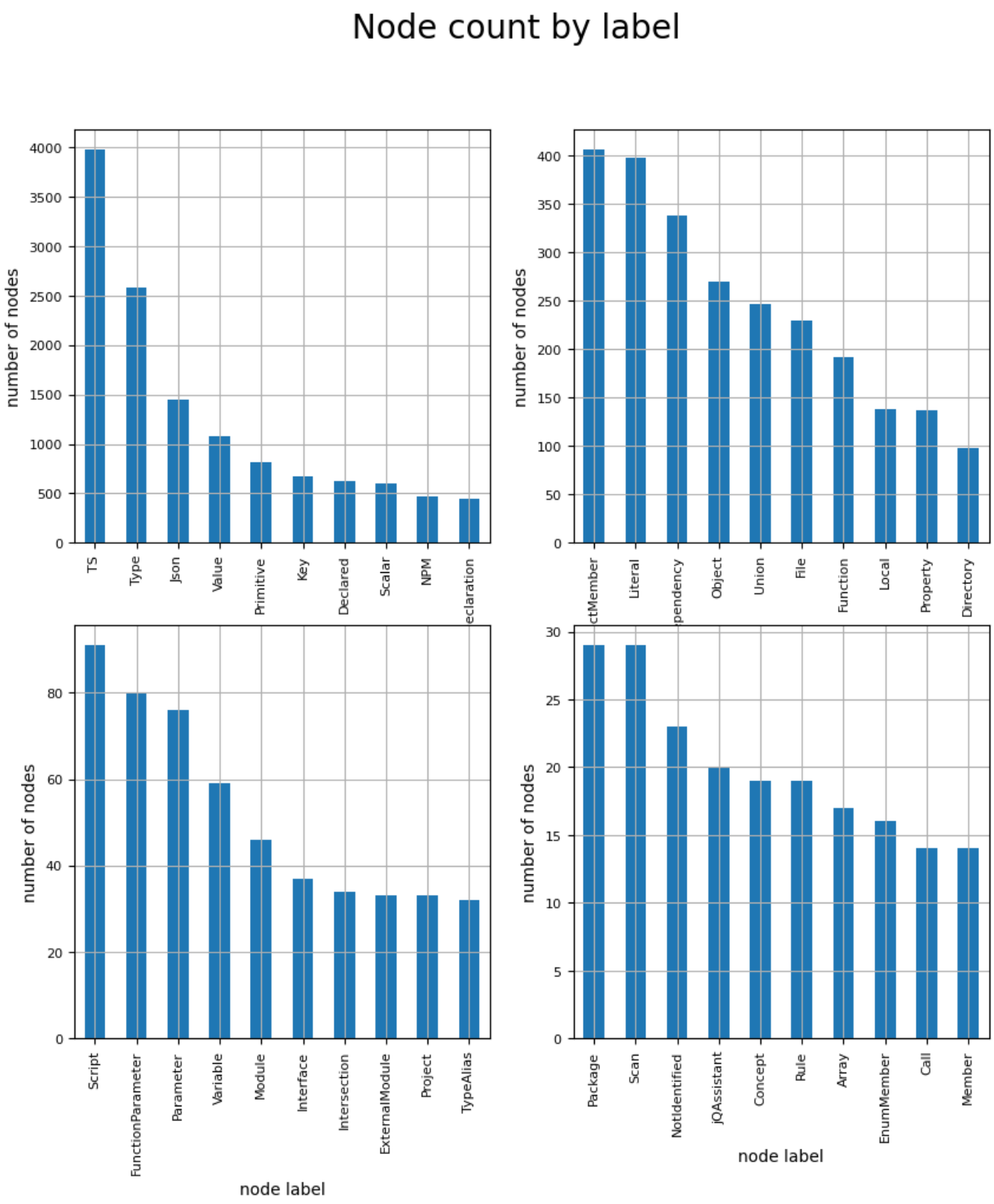
Lists the 40 labels with the highest number of nodes. Doesn't sum up to the total number of nodes or 100% because one node can have multiple labels. Helps to identify commonly used labels.

	nodeLabel	nodesWithThatLabel	nodesWithThatLabelPercent
0	TS	3980	66.267066
1	Type	2581	42.973693
2	Json	1445	24.059274
3	Value	1076	17.915418
4	Primitive	811	13.503164
5	Key	668	11.122211
6	Declared	628	10.456210
7	Scalar	603	10.039960
8	NPM	464	7.725608
9	ExternalDeclaration	444	7.392607
10	ObjectMember	406	6.759907
11	Literal	398	6.626707
12	Dependency	338	5.627706
13	Object	270	4.495504
14	Union	246	4.095904
15	File	230	3.829504
16	Function	192	3.196803
17	Local	138	2.297702
18	Property	137	2.281052
19	Directory	98	1.631702
20	Script	91	1.515152
21	FunctionParameter	80	1.332001
22	Parameter	76	1.265401
23	Variable	59	0.982351
24	Module	46	0.765901
25	Interface	37	0.616051
26	Intersection	34	0.566101
27	ExternalModule	33	0.549451
28	Project	33	0.549451
29	TypeAlias	32	0.532801
30	Package	29	0.482850
31	Scan	29	0.482850
32	NotIdentified	23	0.382950
33	jQAssistant	20	0.333000
34	Concept	19	0.316350
35	Rule	19	0.316350
36	Array	17	0.283050
37	EnumMember	16	0.266400
38	Call	14	0.233100
39	Member	14	0.233100

Chart 1c - Highest node count by label

Shows the 40 labels with the highest number of nodes.

```
<Figure size 640x480 with 0 Axes>
```



Relationship Types

Table 2a - Highest relationship count by type

Lists the 30 relationship types with the highest number of occurrences. The whole table can be found in the CSV report `Relationship_type_count`.

```
Total number of relationships: 8796
```

	relationshipType	nodesWithThatRelationshipType	nodesWithThatRelationshipTypePercent
0	DEPENDS_ON	1823	20.725330
1	CONTAINS	1195	13.585721
2	OF_TYPE	1030	11.709868
3	HAS_KEY	668	7.594361
4	HAS_VALUE	668	7.594361
5	EXPORTS	651	7.401091
6	REFERENCES	489	5.559345
7	DECLARES	410	4.661210
8	HAS_MEMBER	406	4.615734
9	HAS_TYPE_ARGUMENT	202	2.296498
10	RETURNS	183	2.080491
11	DECLARES_DEV_DEPENDENCY	169	1.921328
12	DECLARES_DEPENDENCY	161	1.830377
13	HAS_PARAMETER	155	1.762165
14	DECLARES_SCRIPT	91	1.034561
15	INITIALIZED_WITH	75	0.852660
16	CONTAINS_VALUE	51	0.579809
17	CONTAINS_PROJECT	33	0.375171
18	HAS_CONFIG	33	0.375171
19	HAS_NPM_PACKAGE	33	0.375171
20	HAS_ROOT	33	0.375171
21	IS_DESCRIBED_IN_NPM_PACKAGE	33	0.375171
22	USES	33	0.375171
23	REQUIRES_CONCEPT	28	0.318327
24	PROVIDED_BY_NPM_DEPENDENCY	20	0.227376
25	INCLUDES_CONCEPT	19	0.216007
26	CALLS	14	0.159163
27	HAS_ARGUMENT	14	0.159163
28	MEMBER	14	0.159163
29	PARENT	14	0.159163

Chart 2a - Highest relationship count by type

Values under 0.5% will be grouped into "others" to get a cleaner plot. The group "others" is then broken down in the second chart.

```
<Figure size 640x480 with 0 Axes>
```

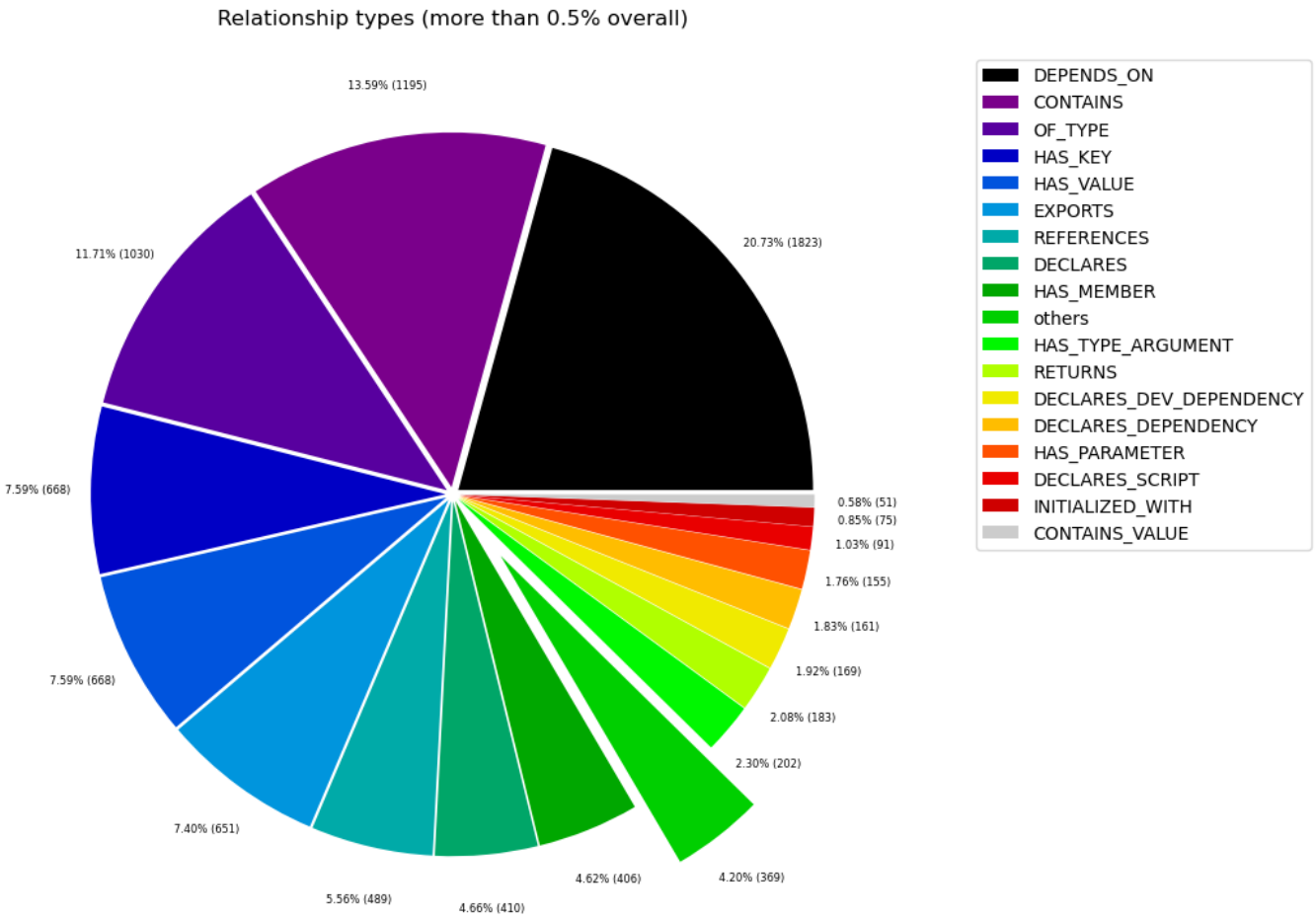


Table 2b - Lowest relationship count by type

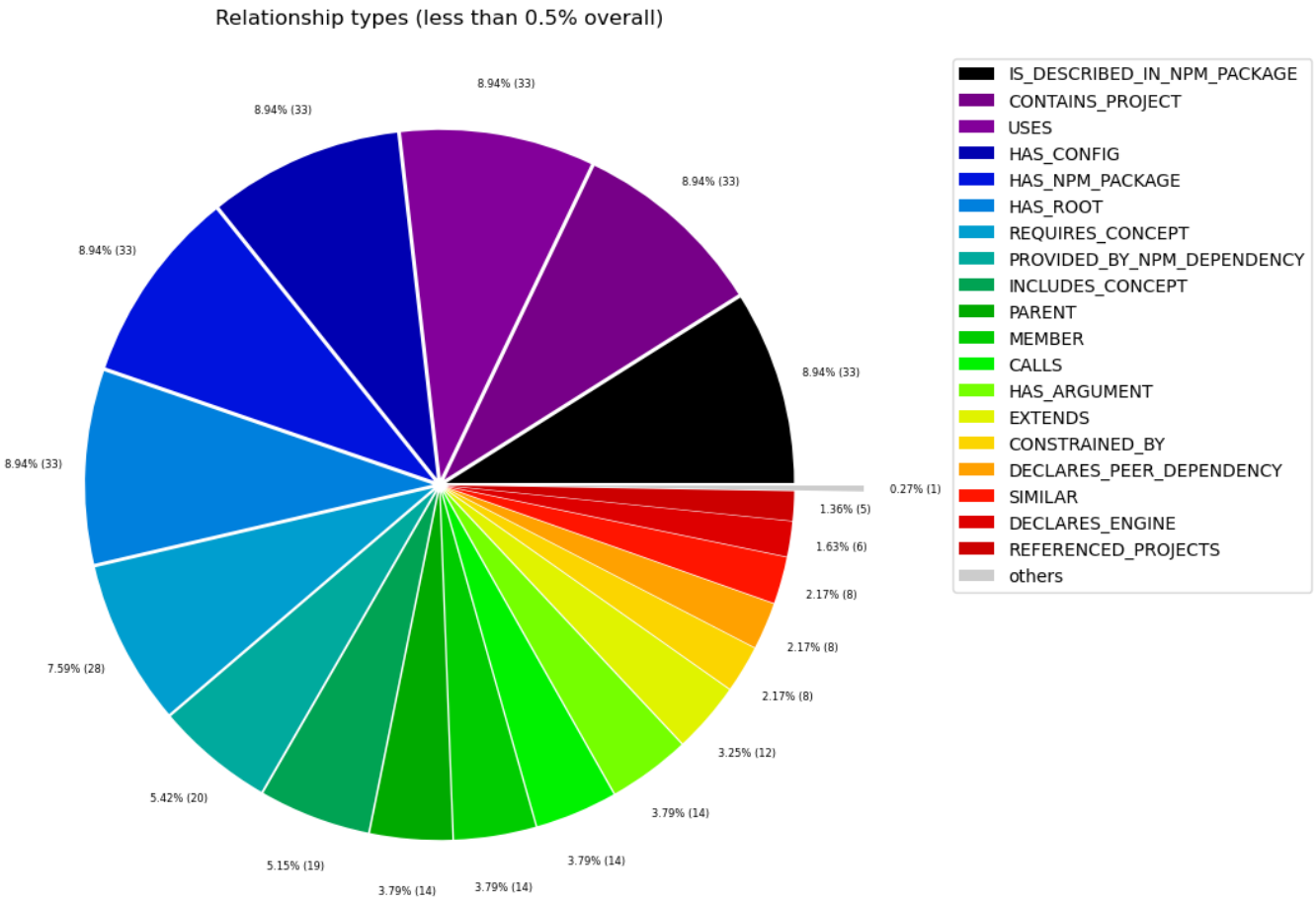
Lists the 30 relationships type with the lowest number of occurrences up to 0.5% of the total node count. This is essentially breaking down the "others" slice from the chart above.

	relationshipType	nodesWithThatRelationshipType	nodesWithThatRelationshipTypePercent
0	HAS	1	0.011369
1	REFERENCED_PROJECTS	5	0.056844
2	DECLARES_ENGINE	6	0.068213
3	SIMILAR	8	0.090950
4	DECLARES_PEER_DEPENDENCY	8	0.090950
5	CONSTRAINED_BY	8	0.090950
6	EXTENDS	12	0.136426
7	PARENT	14	0.159163
8	MEMBER	14	0.159163
9	CALLS	14	0.159163
10	HAS_ARGUMENT	14	0.159163
11	INCLUDES_CONCEPT	19	0.216007
12	PROVIDED_BY_NPM_DEPENDENCY	20	0.227376
13	REQUIRES_CONCEPT	28	0.318327
14	HAS_NPM_PACKAGE	33	0.375171
15	USES	33	0.375171
16	IS_DESCRIBED_IN_NPM_PACKAGE	33	0.375171
17	HAS_ROOT	33	0.375171
18	HAS_CONFIG	33	0.375171
19	CONTAINS_PROJECT	33	0.375171

Chart 2b - Lowest relationship count by type

Shows the lowest (less than 0.5% overall) relationship types. This plot breaks down the "others" slice of the pie chart above. Values under 0.01% will be grouped into "others" to get a cleaner plot.

```
<Figure size 640x480 with 0 Axes>
```



Node labels with their relationships

Table 3a - Highest relationship count by node labels and relationship type

Lists the 30 node labels and their relationship types with the highest number of occurrences.

	sourceLabels	relationType	targetLabels	numberOfRelationships	numberOfNodesWithSameLabelsAsSource	numberOfNodes
0	[Json, Value, Object]	HAS_KEY	[Json, Key]	668	133	
1	[TS, Function]	DEPENDS_ON	[TS, ExternalDeclaration]	580	109	
2	[Json, Key]	HAS_VALUE	[Json, Value, Scalar]	552	668	
3	[TS, ExternalModule]	EXPORTS	[TS, ExternalDeclaration]	444	33	
4	[Type, Object, TS]	HAS_MEMBER	[Type, TS, ObjectMember]	318	109	
5	[File, TS, Local, Module, Mark4ModuleWeaklyCon...	DEPENDS_ON	[TS, ExternalDeclaration]	308	1	
6	[Type, TS, Union]	CONTAINS	[Type, TS, Primitive]	303	246	
7	[Type, TS, Declared]	REFERENCES	[TS, ExternalDeclaration]	288	598	
8	[Type, TS, Union]	CONTAINS	[Type, TS, Literal]	238	246	
9	[Type, TS, ObjectMember]	OF_TYPE	[Type, TS, Primitive]	173	318	
10	[Package, File, Json, NPM]	DECLARES_DEV_DEPENDENCY	[NPM, Dependency]	169	29	
11	[Package, File, Json, NPM]	DECLARES_DEPENDENCY	[NPM, Dependency]	161	29	
12	[TS, Function]	DEPENDS_ON	[TS, ExternalModule]	148	109	
13	[Type, TS, Union]	CONTAINS	[Type, TS, Declared]	145	246	
14	[TS, Interface]	DECLARES	[TS, Property]	129	37	
15	[File, TS, Local, Module, Mark4ModuleWeaklyCon...	DEPENDS_ON	[TS, ExternalDeclaration]	124	2	
16	[Value, TS, Literal]	OF_TYPE	[Type, TS, Primitive]	118	124	
17	[Json, Key]	HAS_VALUE	[Json, Value, Object]	104	668	
18	[TS, Property]	OF_TYPE	[Type, TS, Union]	93	137	
19	[Package, File, Json, NPM]	DECLARES_SCRIPT	[NPM, Script]	91	29	
20	[Value, Object, TS]	HAS_MEMBER	[Value, TS, ObjectMember]	88	28	
21	[TS, Variable]	DEPENDS_ON	[TS, ExternalDeclaration]	88	59	
22	[Type, TS, Declared]	HAS_TYPE_ARGUMENT	[Type, TS, Declared]	86	598	
23	[Value, TS, ObjectMember]	OF_TYPE	[Type, TS, Primitive]	84	88	
24	[Value, TS, ObjectMember]	REFERENCES	[Value, TS, Literal]	83	88	
25	[Type, TS, Function]	HAS_PARAMETER	[Type, TS, FunctionParameter]	80	70	
26	[TS, Function]	HAS_PARAMETER	[TS, Parameter]	75	109	
27	[Type, TS, ObjectMember]	OF_TYPE	[Type, TS, Union]	70	318	
28	[TS, Function]	DEPENDS_ON	[TS, Function]	67	109	
29	[File, Directory]	CONTAINS	[File, Directory]	65	34	

Graph Density

```
total_number_of_nodes (vertices): 6006
total_number_of_relationships (edges): 8796
-> total directed graph density: 0.00024388600575111816
-> total directed graph density in percent: 0.024388600575111816
```