

---

# denoにおける外部モジュールimportの 組み合わせについての実証分析

— 実証的ソフトウェア工学研究室 —  
19T2083J 志良堂 泰成

---

# 背景・目的



## Deno

- ・ JavaScript / TypeScriptのサーバーサイドでの実行環境
- ・ 実装したい機能に合った外部モジュールをimportして、システム構築を行う
- ・ Node.jsでの様々な問題点を解決するために実装された

# 背景・目的

## Node.jsとの違い

### Node.js

ネットワークやファイル  
パーミッションルへのアクセス権を制  
限することができない



### Deno

プロセスごとにパーミッ  
ションを制限することができる。

モジュールの  
インポート

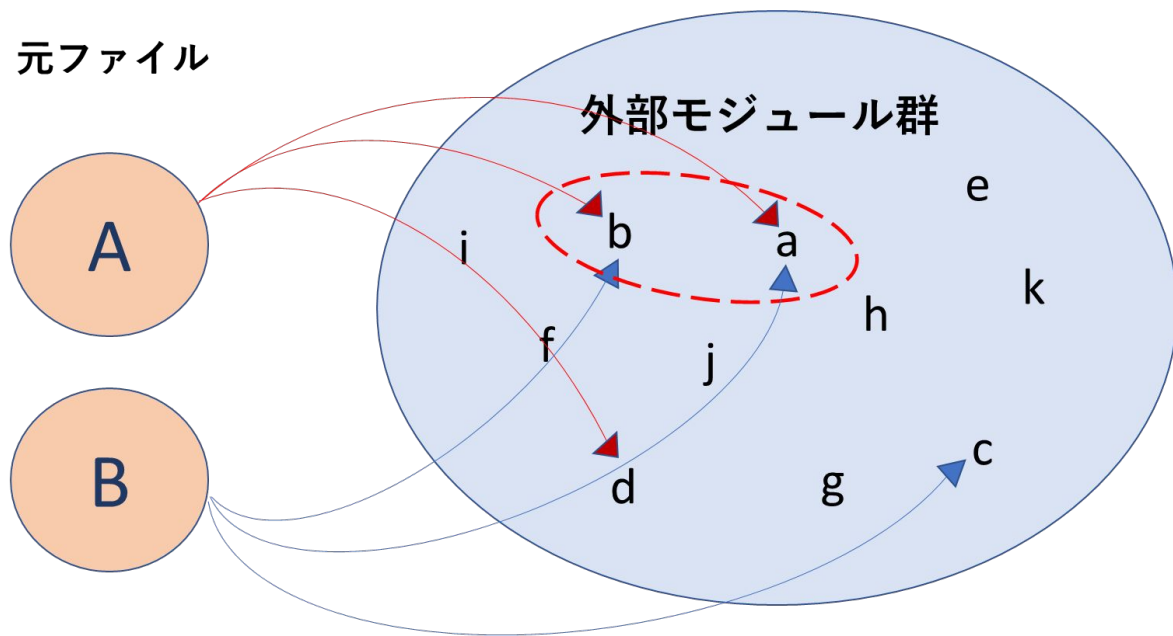
npm(Node Package Manager)  
Installでモジュールを  
取得しなければならない



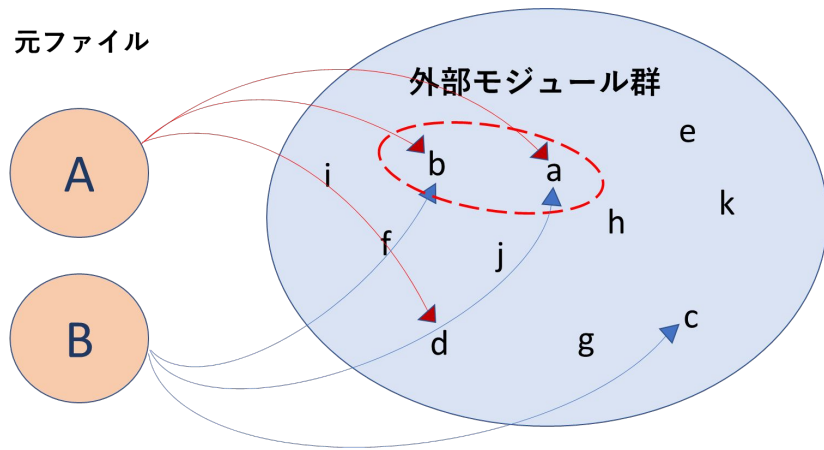
Import文でURLを渡すことで、  
直接外部モジュールをimport  
することができる。

# 背景・目的

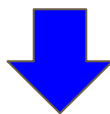
Denoは、複数の外部モジュールをimportして組み合わせることによって開発が行われることが多い



# 背景・目的



複雑な外部モジュールの依存関係が  
無数に存在している

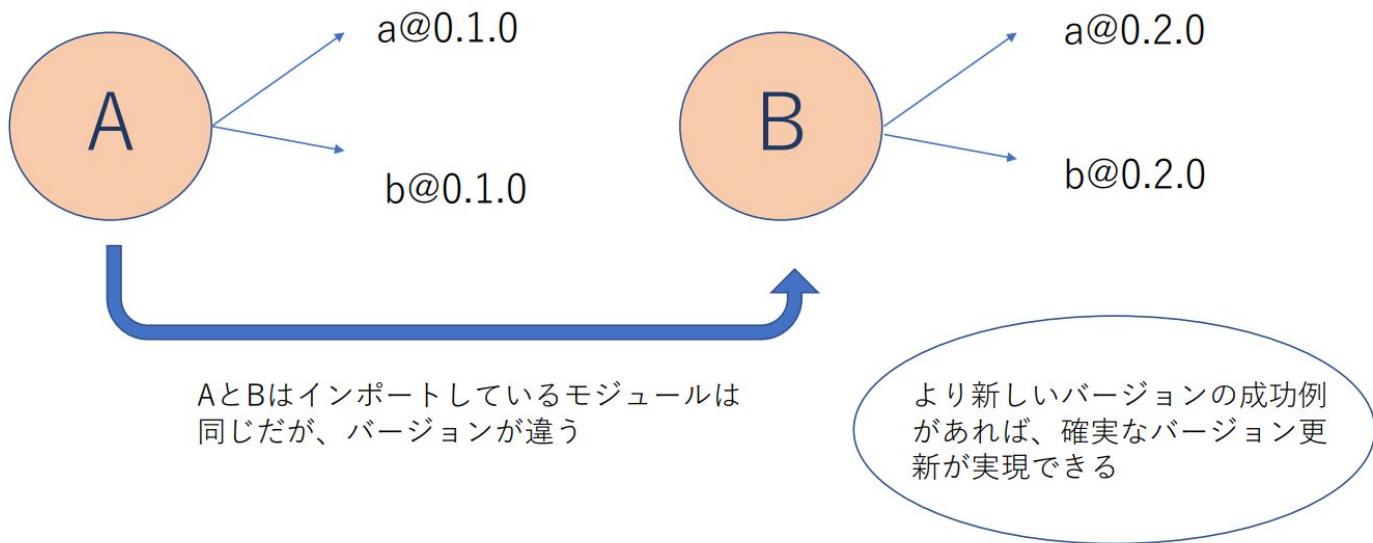


deno.land(Denoのモジュールレジストリ)に存在している、  
計4,811個(2022.07.08時点)のモジュールの  
依存関係の組み合わせについて調査

# 背景・目的

## 組み合わせを調べる意義

前提: Denoは基本的に、新しいバージョンのモジュールを使用することを推奨している



# 分析方法

deno.land上にある呼び出し元ファイル全てについてimport文を抽出し、importされているモジュールの情報を取得

```
Import {foo} from "https://hoge hoge.ts"
```

部を取得する



呼び出し元と外部モジュールの  
依存関係をリストアップ  
(データ数:11,363個)

呼び出し元	外部モジュール
A	a,b,d
B	a,b,c
C	d,e,f,g

module名、version等の情報が含まれている



# Research Questions

**RQ1...**importされている外部モジュールの組み合わせの全体像はどのような特徴があるか。

**RQ2...**外部モジュールの組み合わせごとのバージョン分布はどのようなになっているのか。

**RQ3...**同じモジュールの組み合わせがimportされている中で、上位のバージョンが指定されているものが存在するか。



# RQ1...importされている組み合わせの全体像はどのような特徴があるか

- ・どのくらいの種類・大きさの組み合わせが存在しているか

組み合わせの大きさ	種類	Importされている頻度
2	941	24,722
3	632	8,128
4	293	2,090
5	122	480
6	38	105
7	11	23
8	5	10
9	5	12
11	3	9
12	2	8
13	3	6
15	1	2
18	1	2
74	1	2

# RQ1...importされている組み合わせの全体像はどのような特徴があるか

- ・どのくらいの種類・大きさの組み合わせが存在しているか

組み合わせの大きさ	種類	Importされている頻度
2	941	24,722
3	632	8,128
4	293	2,090
5	122	480
6	38	105
7	11	23
8	5	10
9	5	12
11	3	9
12	2	8
13	3	6
15	1	2
18	1	2
74	1	2

・組み合わせの大きさが小さいものから順に、多くのファイルにimportされている傾向にある

・組み合わせが大きいものの中にも、より小さい組み合わせを包含しているものが多数存在している。

# RQ1...importされている組み合わせの全体像はどのような特徴があるか

・どのような外部モジュールの組み合わせで、多くimportされているのか  
(上位5つ)

組み合わせ	Importされている頻度
<ul style="list-style-type: none"><li>• <a href="https://deno.land/std/path/mod.ts">https://deno.land/std/path/mod.ts</a></li><li>• <a href="https://deno.land/std/fs/mod.ts">https://deno.land/std/fs/mod.ts</a></li></ul>	1,274
<ul style="list-style-type: none"><li>• <a href="https://deno.land/std/path/mod.ts">https://deno.land/std/path/mod.ts</a></li><li>• <a href="https://deno.land/std/fmt/colors.ts">https://deno.land/std/fmt/colors.ts</a></li></ul>	988
<ul style="list-style-type: none"><li>• <a href="https://deno.land/std/testing/asserts.ts">https://deno.land/std/testing/asserts.ts</a></li><li>• <a href="https://deno.land/std/path/mod.ts">https://deno.land/std/path/mod.ts</a></li></ul>	675
<ul style="list-style-type: none"><li>• <a href="https://deno.land/std/fs/mod.ts">https://deno.land/std/fs/mod.ts</a></li><li>• <a href="https://deno.land/std/fmt/colors.ts">https://deno.land/std/fmt/colors.ts</a></li></ul>	673
<ul style="list-style-type: none"><li>• <a href="https://deno.land/std/path/mod.ts">https://deno.land/std/path/mod.ts</a></li><li>• <a href="https://deno.land/std/flags/mod.ts">https://deno.land/std/flags/mod.ts</a></li></ul>	548

# RQ1...importされている組み合わせの全体像はどのような特徴があるか

・どのような外部モジュールの組み合わせで、多くimportされているのか  
(上位5つ)

組み合わせ	Importされている頻度
・ <a href="https://deno.land/std/path/mod.ts">https://deno.land/std/path/mod.ts</a> ・ <a href="https://deno.land/std/fs/mod.ts">https://deno.land/std/fs/mod.ts</a>	1,274
・ <a href="https://deno.land/std/path/mod.ts">https://deno.land/std/path/mod.ts</a> ・ <a href="https://deno.land/std/fmt/colors.ts">https://deno.land/std/fmt/colors.ts</a>	988
・ <a href="https://deno.land/std/testing/asserts.ts">https://deno.land/std/testing/asserts.ts</a> ・ <a href="https://deno.land/std/path/mod.ts">https://deno.land/std/path/mod.ts</a>	675
・ <a href="https://deno.land/std/fs/mod.ts">https://deno.land/std/fs/mod.ts</a> ・ <a href="https://deno.land/std/fmt/colors.ts">https://deno.land/std/fmt/colors.ts</a>	673
・ <a href="https://deno.land/std/path/mod.ts">https://deno.land/std/path/mod.ts</a> ・ <a href="https://deno.land/std/flags/mod.ts">https://deno.land/std/flags/mod.ts</a>	548

・全てのモジュールが std(標準ライブラリ)内のモジュールである。

→汎用性のあるモジュールが多く組み合わせられて使われている。

・1番多く使われている組み合わせの頻度は **1,274** であるが、これは 標本数全体 (11,363個)の **11.3%**にあたる

## RQ1...importされている組み合わせの全体像はどのような特徴があるか

- ・どのくらいの種類・大きさの組み合わせが存在しているか

全てstdモジュールの組み合わせ → **976**種類  
一部stdを含んでいるモジュールの組み合わせ → **745**種類  
一つもstdを含んでいないモジュールの組み合わせ → **338**種類 (全2,059種類)

stdを一つも含んでいない組み合わせの中で、一番importされているものは [deno.land/x/ramda/mod.ts](#) と [deno.land/x/functional/library/utilities.js](#) で、importされている頻度は**140**である。

## RQ2...組み合わせごとのバージョン分布はどのようなになっているのか。

存在しているimportモジュールの組み合わせについて、  
「全てバージョン定義されている」か「一部バージョン定義されていない」かを分析

### 例

[deno.land/x/ModuleA/mod.ts, deno.land/x/ModuleB/mod.ts]  
の組み合わせが3回importされていると仮定。



[0.140.0, 0.10.0], [0.141.0, 0.10.0], [0.150.0, 0.12.0]

全てバージョン定義されている

[0.140.0, 0.10.0], [0.141.0, 0.10.0], [0.150.0, nan]

一部バージョン定義されていない

## RQ2...組み合わせごとのバージョン分布はどのようになっているのか。

存在しているimportモジュールの組み合わせについて、  
「全てバージョン定義されている」か「一部バージョン定義されていない」かを分析

全2,059種類について

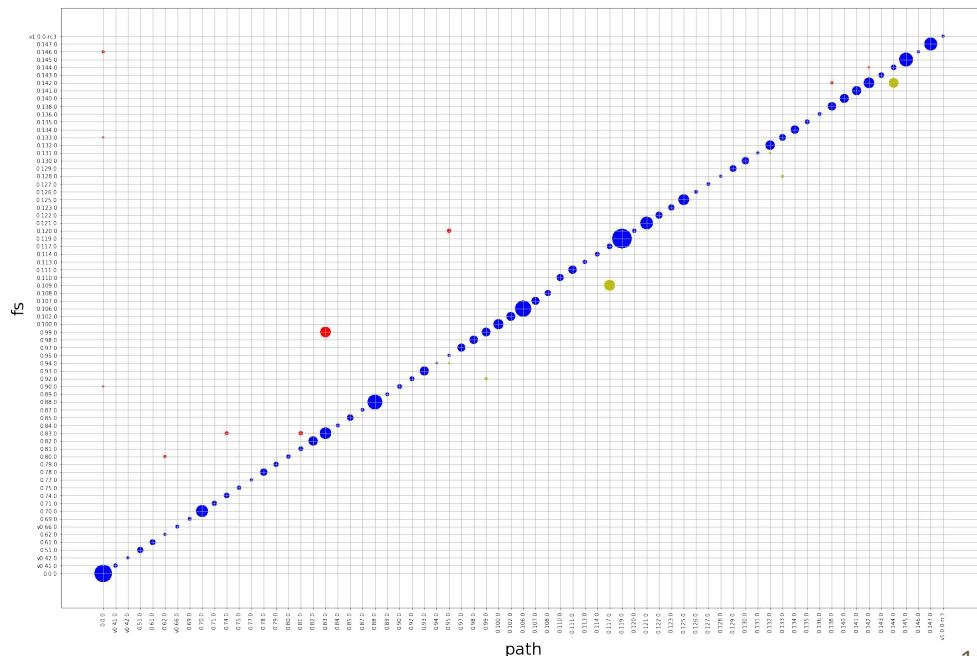
全てバージョン定義されている → 1,356種類  
一部バージョン定義されていない → 703種類

6割近くのモジュールの組み合わせについて、  
バージョン管理が徹底されている

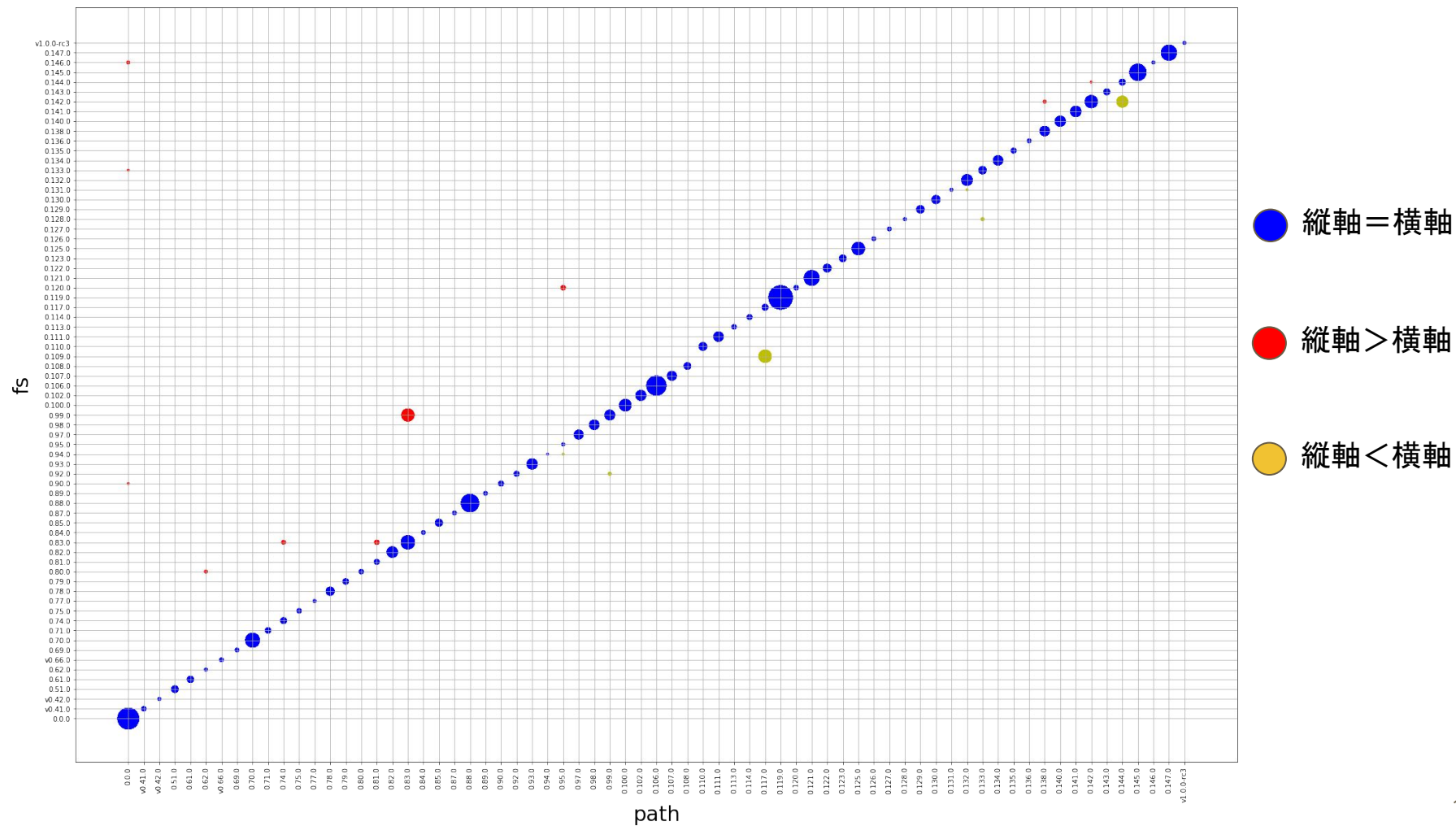
# RQ2...組み合わせごとのバージョン分布はどのようなになっているのか。

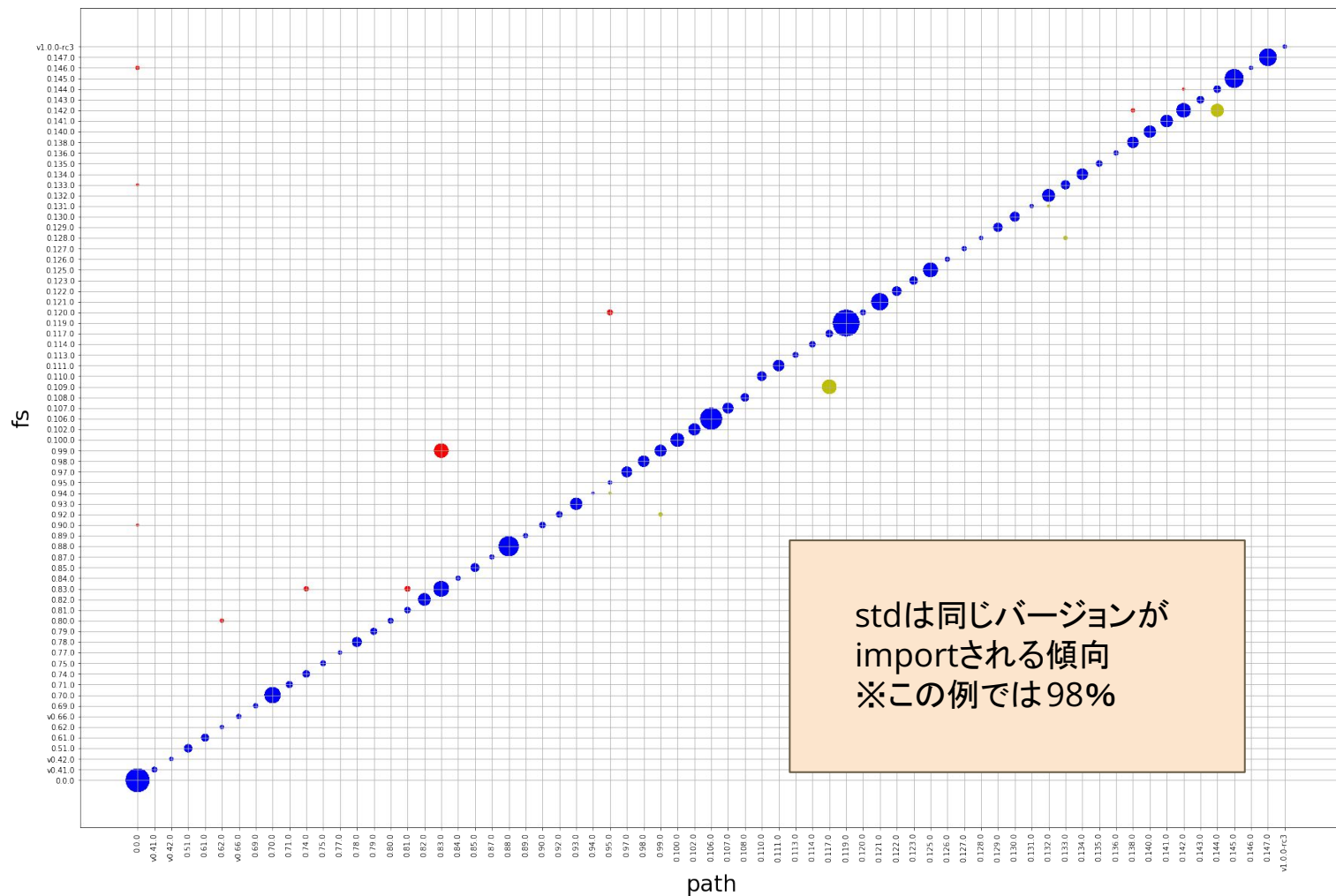
## stdの組み合わせの特徴

一番import回数の多い組み合わせ  
[std/path/mod.ts,std/fs/mod.ts]  
についてのバージョン分布 →









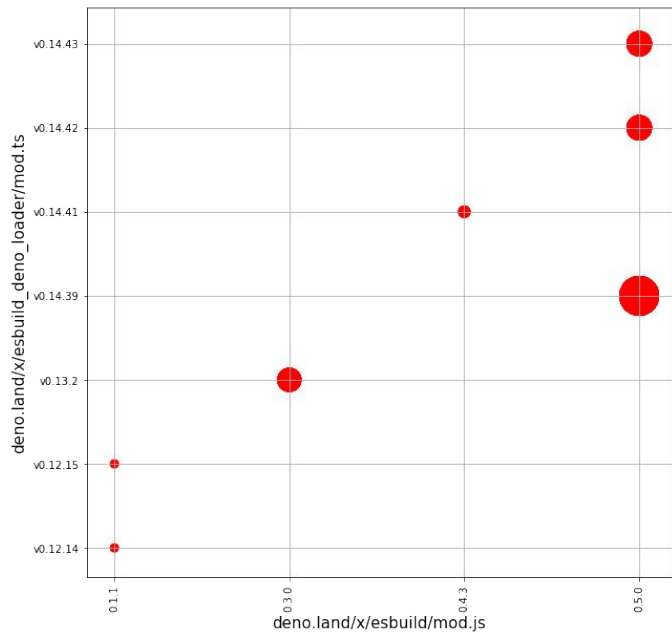
● 縦軸＝横軸

● 縦軸＞横軸

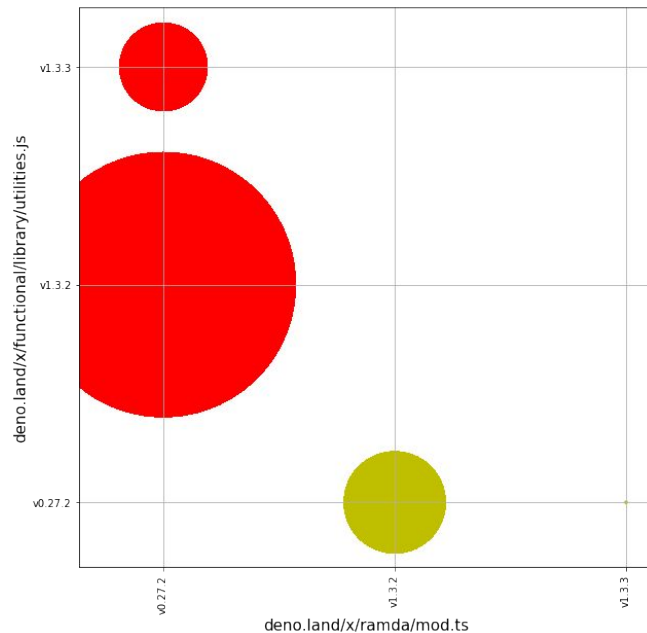
● 縦軸＜横軸

## RQ2...組み合わせごとのバージョン分布はどのようになっているのか。

std以外での組み合わせのバージョン分布の例



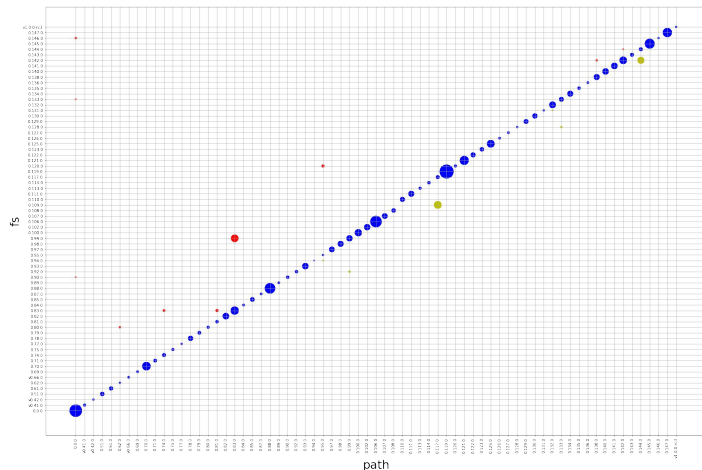
分布にばらつきがある例



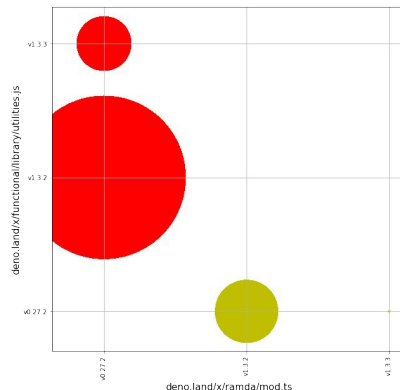
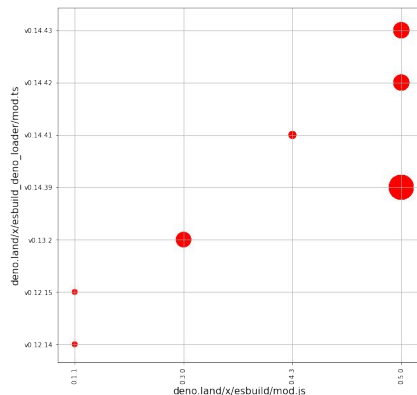
特定のバージョンに集中している例

## RQ2...組み合わせごとのバージョン分布はどのようなになっているのか。

std(標準)モジュール  
同士の組み合わせ



stdではないモジュールの組  
み合わせ

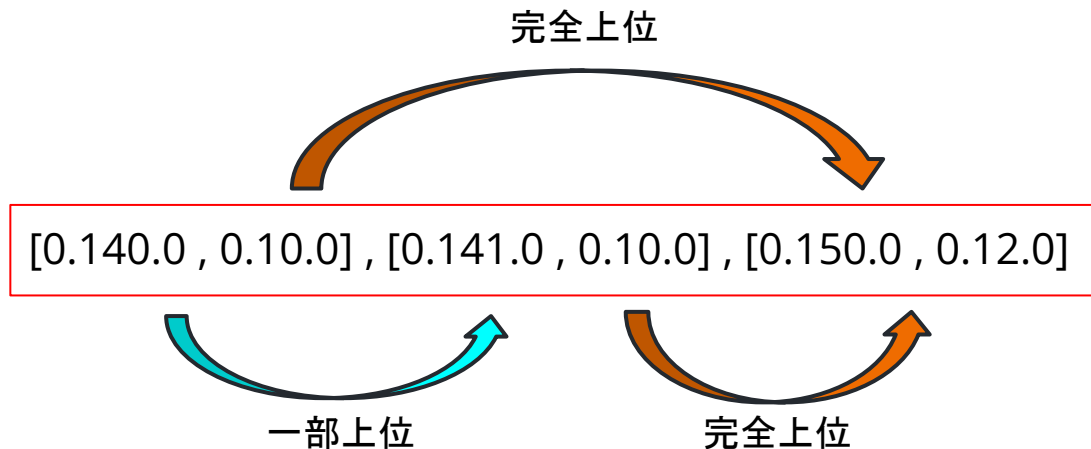


- std(標準)モジュールをimportする場合、同じバージョンを指定する場合が多く幅広い分布。
- stdモジュール以外だと、バージョンの組み合わせに規則性を確認できない。

# RQ3...同じモジュールの組み合わせがimportされている中で、上位のバージョンが指定されているものが存在するか。

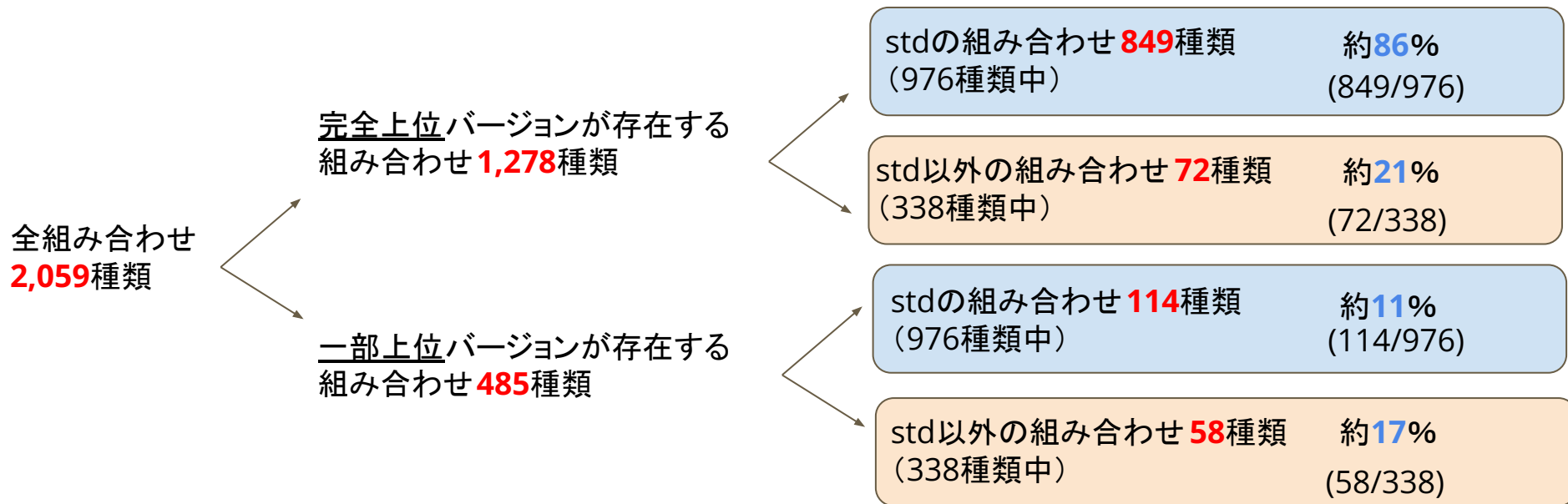
例

[deno.land/x/ModuleA/mod.ts , deno.land/x/ModuleB/mod.ts]  
の組み合わせが3回importされていると仮定。

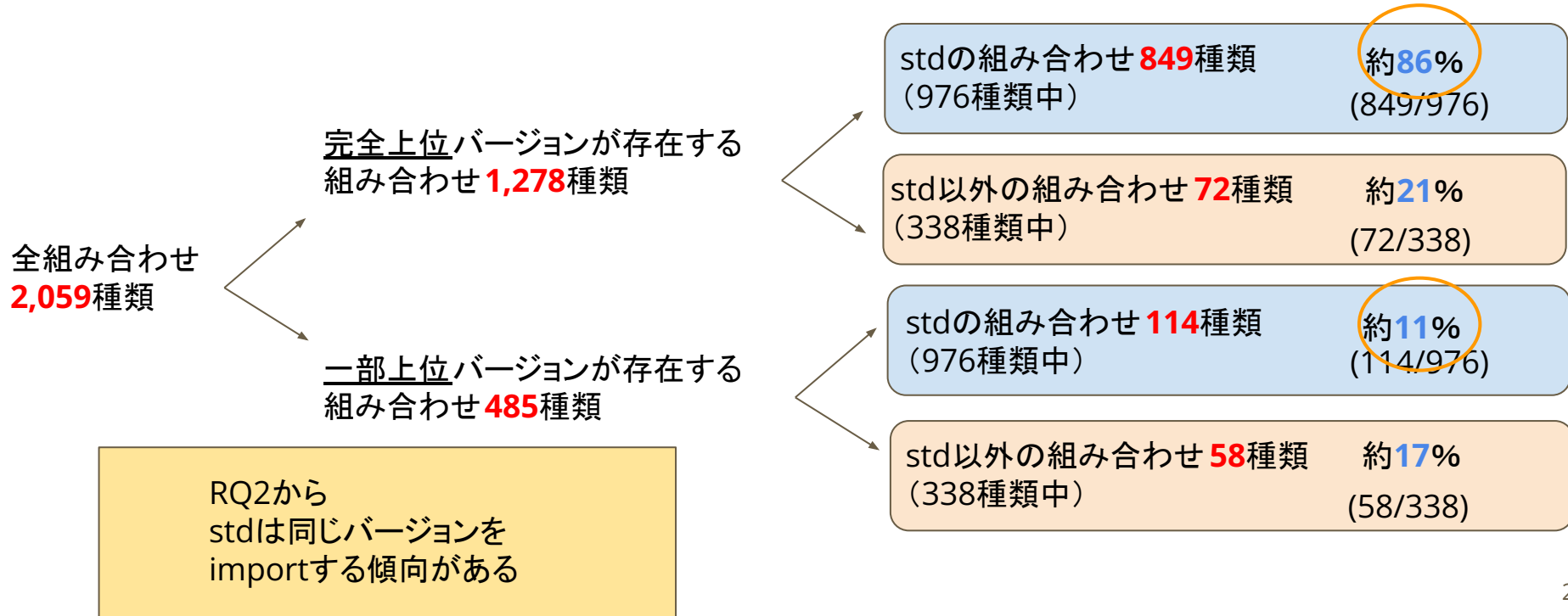


完全上位→2個  
一部上位→1個 としてカウント

# RQ3...同じモジュールの組み合わせがimportされている中で、上位のバージョンが指定されているものが存在するか。



# RQ3...同じモジュールの組み合わせがimportされている中で、上位のバージョンが指定されているものが存在するか。



# まとめ

## std

- ・幅広い開発者に複数同時に用いられ、同じバージョンがimportされる傾向
- ・複数のstdモジュールはほとんどの場合同じバージョンが指定され、そのバージョンは開発する時期に依存するため上位が多数存在する

## std以外

- ・サードパーティのモジュールであるため、組み合わせの種類・頻度が少ない傾向
- ・バージョン指定に規則性がなく、同じ開発者が特定のバージョンを複数回指定する場合が多い



# 今後の展望

stdモジュール同士の組み合わせは、  
多数の上位バージョンが存在している

しかし...



ほとんどの開発においてstdは利用されるため  
バージョン更新は容易であり、  
そこまで重要ではない。

std以外のモジュールの組み合わせに焦点を当てる事が、  
「確実に安全なバージョン更新」という目的において重要である