

Dart Build Experiment Results

Grant Nelson

<https://github.com/grantnelson-wf/dartCompTest>

Main Idea

- The Dart build time consists of a dependency graph, cached items, and changed items that need to be rebuilt.
- Mini-libraries (File_Dep) are when neither “part of” nor “library” directive are being used. Instead, each file has its own list of imports.
- Part-of libraries (Library_Dep) is when a library consists of one or more files which use “part of”. The single library file contains the list of imports for all of its parts.
- Mini-libraries should build faster because less code needs to be rebuilt than with part-of libraries.
- Part-of libraries should build faster because the dependency tree is far simpler and there are fewer larger caches.
- So, which is actually faster?

Null Hypothesis:

The Dart build time of code using part-of libraries is the **same** as using mini-libraries where each file has its own import list.

$$H_0: \mu_{part\text{-}of} = \mu_{mini}$$

Method

- A generator program creates two Dart projects which are nearly identical. One project uses part-of libraries, the other uses mini-libraries. These projects are the treatments of the experiment.
- Each treatment has “pub get” and an initial build run on it (burn-in). This will prepare all the tools such as web-builder once before the experiment runs.
- The experiment program runs a prepare command followed by a build command.
- The prepare command make any changes needed based on what is being tested to prepare the treatment project for the build. For example, it will delete the build folder or it will modify a value in one of the treatment’s files.
- The experiment program measures the time that the build command takes.
- The order of the treatments are randomized per replication of the experiment to help prevent the order from influencing the recorded times.

Analysis

- Two-way ANOVA: seconds ~ order + index + order:index
- The results will contain the index of the treatment and the order as an incrementing value to indicate the order the builds were run in.
- Using a 95% confidence interval. Any p -value < 0.05 means there is evidence to **reject** the Null Hypotheses (i.e., the treatments took different amounts of time)

	sum_sq	df	F	PR(>F)
order	0.006789	1	0.598926	0.482184
index	0.660721	1	58.289553	0.001581
order:index	0.010114	1	0.892302	0.398341
Residual	0.045341	4	–	–

Check that order hasn't influenced results

Check that the different treatments are the same, $H_0: p$ -value

Check if there is an interaction between order and treatment index

Residual will be empty if there are no unaccounted for variables

Mini-library Code

```
1 import 'package:TreeGen/group82.dart';
2
3 class group41_branch5{
4     group82_branch13? _item0 = null;
5     group82_branch14? _item1 = null;
6
7     group41_branch5() {
8         _item0 = group82_branch13();
9         _item1 = group82_branch14();
10    }
11
12     int get hash {
13         int hashCode = 1430287;
14         hashCode *= 7302013 ^ (_item0?.hash ?? 0);
15         hashCode *= 7302013 ^ (_item1?.hash ?? 0);
16         return hashCode;
17     }
18
19     int get sum =>
20         (_item0?.sum ?? 0) +
21         (_item1?.sum ?? 0);
22
23     int get count =>
24         (_item0?.count ?? 0) +
25         (_item1?.count ?? 0);
26 }
```

```
1 class group902_leaf13{
2     final int _value;
3
4     group902_leaf13():
5         _value = 25;
6
7     int get hash => _value;
8
9     int get sum => _value;
10
11    int get count => 1;
12 }
```

```
1 export 'src/group29/group29_branch0.dart';
2 export 'src/group29/group29_branch1.dart';
3 export 'src/group29/group29_branch2.dart';
4 export 'src/group29/group29_branch3.dart';
5 export 'src/group29/group29_branch4.dart';
6 export 'src/group29/group29_branch5.dart';
7 export 'src/group29/group29_branch6.dart';
8 export 'src/group29/group29_branch7.dart';
9 export 'src/group29/group29_branch8.dart';
10 export 'src/group29/group29_branch9.dart';
11 export 'src/group29/group29_branch10.dart';
12 export 'src/group29/group29_branch11.dart';
13 export 'src/group29/group29_branch12.dart';
14 export 'src/group29/group29_branch13.dart';
15 export 'src/group29/group29_branch14.dart';
```

Part-of Library Code

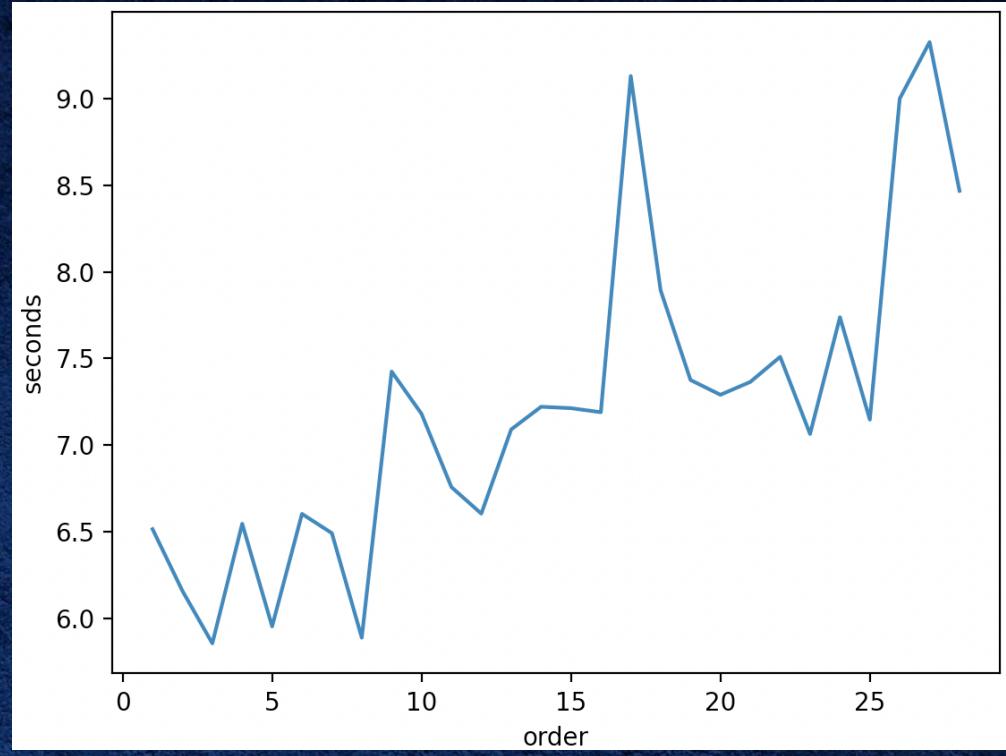
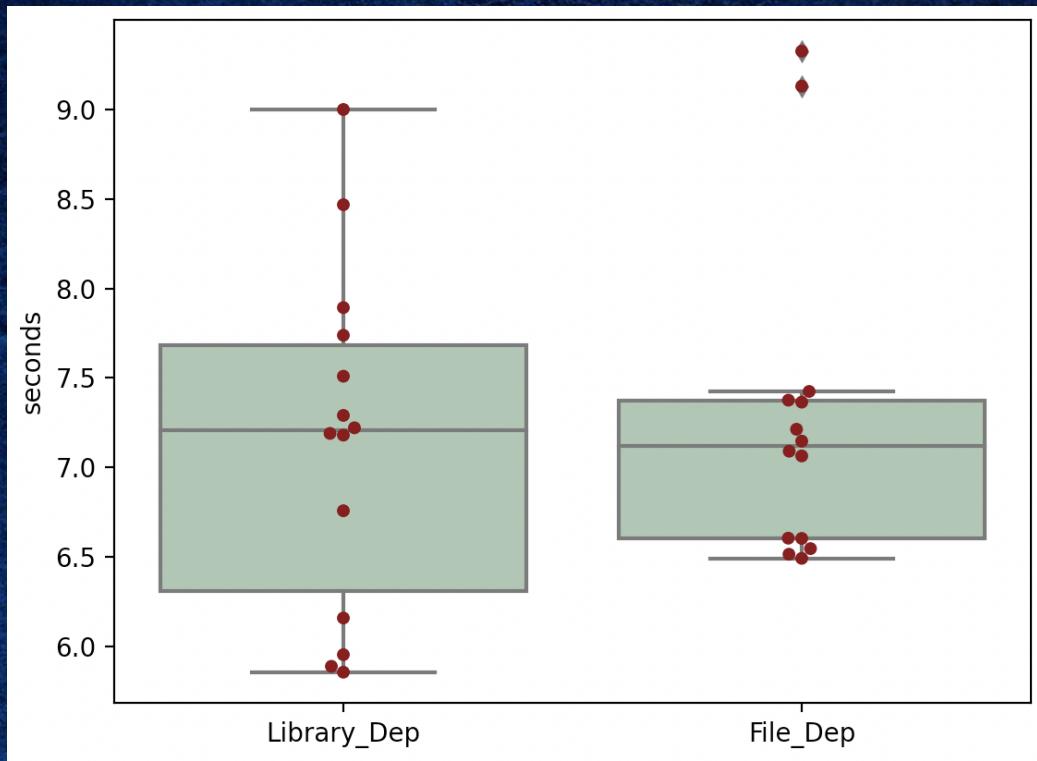
```
1 library group18;
2
3 import 'package:TreeGen/group36.dart';
4 import 'package:TreeGen/group37.dart';
5 import 'package:TreeGen/group38.dart';
6
7 part 'group18_branch0.dart';
8 part 'group18_branch1.dart';
9 part 'group18_branch2.dart';
10 part 'group18_branch3.dart';
11 part 'group18_branch4.dart';
12 part 'group18_branch5.dart';
13 part 'group18_branch6.dart';
14 part 'group18_branch7.dart';
15 part 'group18_branch8.dart';
16 part 'group18_branch9.dart';
17 part 'group18_branch10.dart';
18 part 'group18_branch11.dart';
19 part 'group18_branch12.dart';
20 part 'group18_branch13.dart';
21 part 'group18_branch14.dart';
```

```
1 part of group41;
2
3 class group41_branch5{
4     group82_branch13? _item0 = null;
5     group82_branch14? _item1 = null;
6
7     group41_branch5() {
8         _item0 = group82_branch13();
9         _item1 = group82_branch14();
10    }
11
12     int get hash {
13         int hashCode = 1430287;
14         hashCode *= 7302013 ^ (_item0?.hash ?? 0);
15         hashCode *= 7302013 ^ (_item1?.hash ?? 0);
16         return hashCode;
17    }
18
19     int get sum =>
20         (_item0?.sum ?? 0) +
21         (_item1?.sum ?? 0);
22
23     int get count =>
24         (_item0?.count ?? 0) +
25         (_item1?.count ?? 0);
26 }
```

```
1 part of group902;
2
3 class group902_leaf13{
4     final int _value;
5
6     group902_leaf13():
7         _value = 25;
8
9     int get hash => _value;
10
11    int get sum => _value;
12
13    int get count => 1;
14 }
```

```
1 library group29;
2
3 export 'src/group29/group29.dart';
```

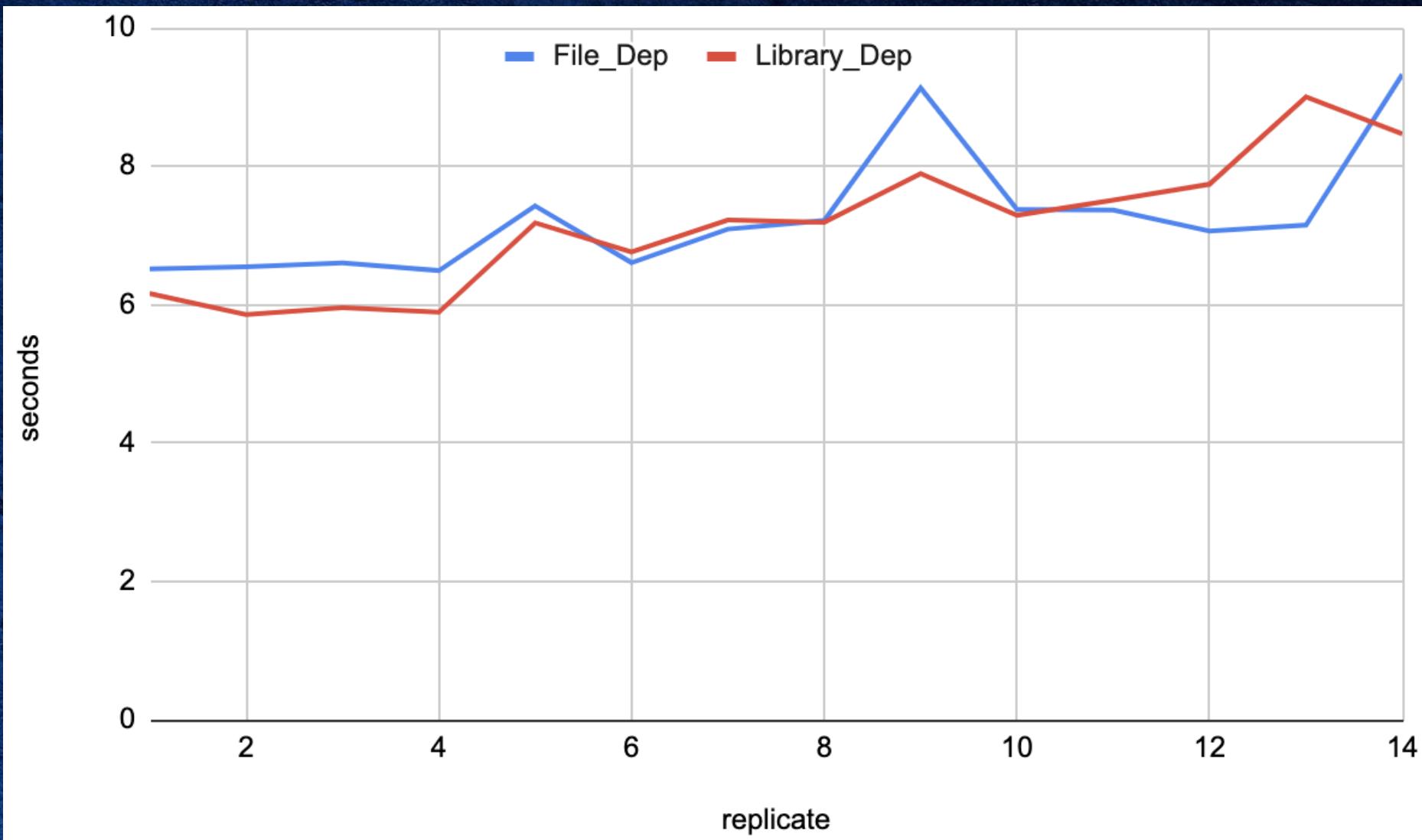
Trial 1 - Rebuild, Depth 10



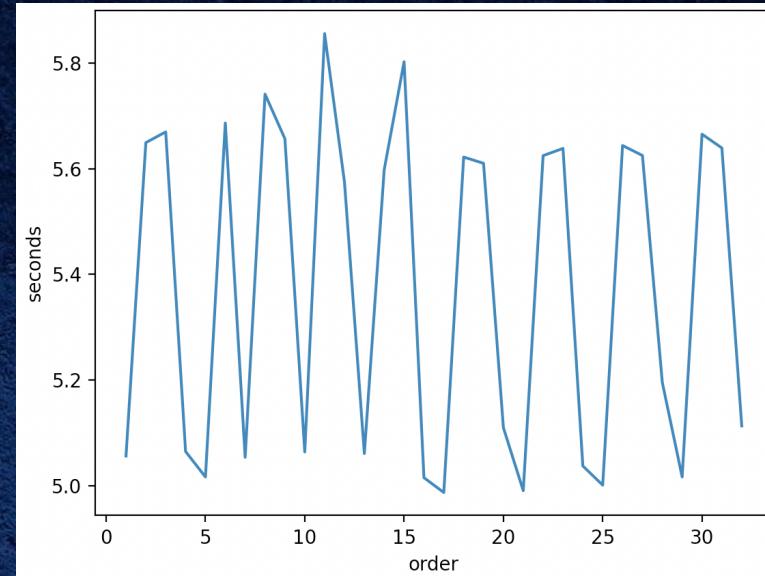
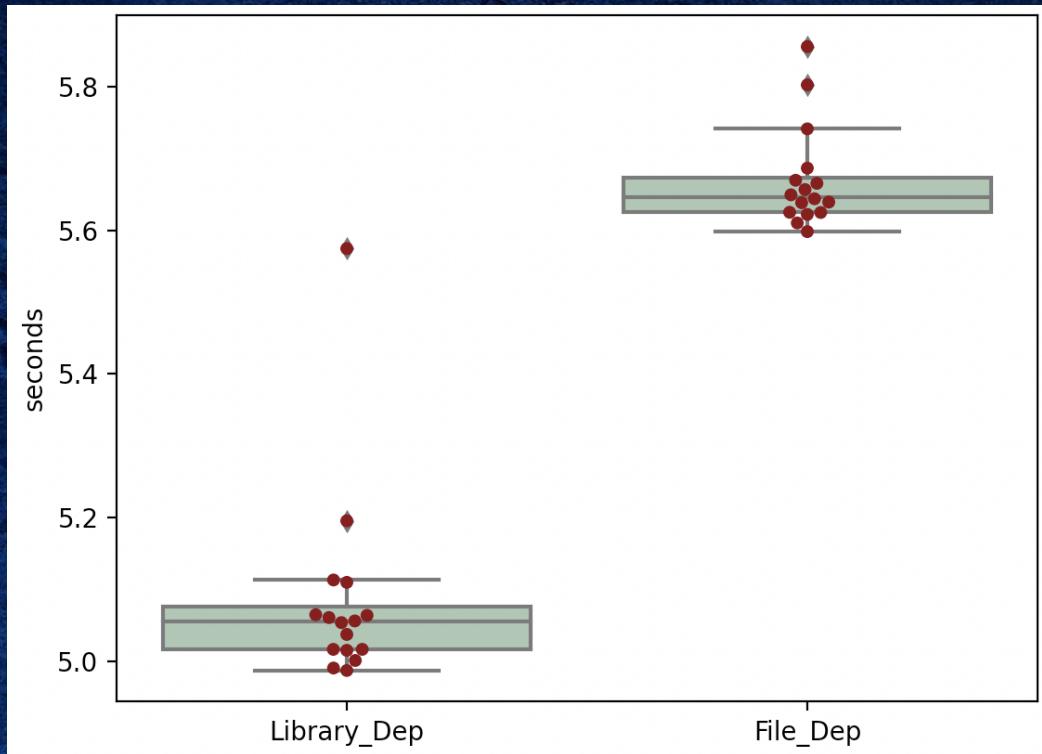
Depth: 10
 Total nodes: 4093
 Branch nodes: 2045
 Leaf nodes: 2048
 Groups: 273
 Dart version: 2.7.4
 Repetitions: 14 < build kept locking up during trial

	sum_sq	df	F	PR(>F)
order	13.610886	1	39.462483	0.000002
index	0.219831	1	0.637362	0.432497
order:index	0.565525	1	1.639645	0.212612
Residual	8.277767	24	—	—

Trial 1 - Replicate Comparison



Trial 2 - Rebuild, Depth 8

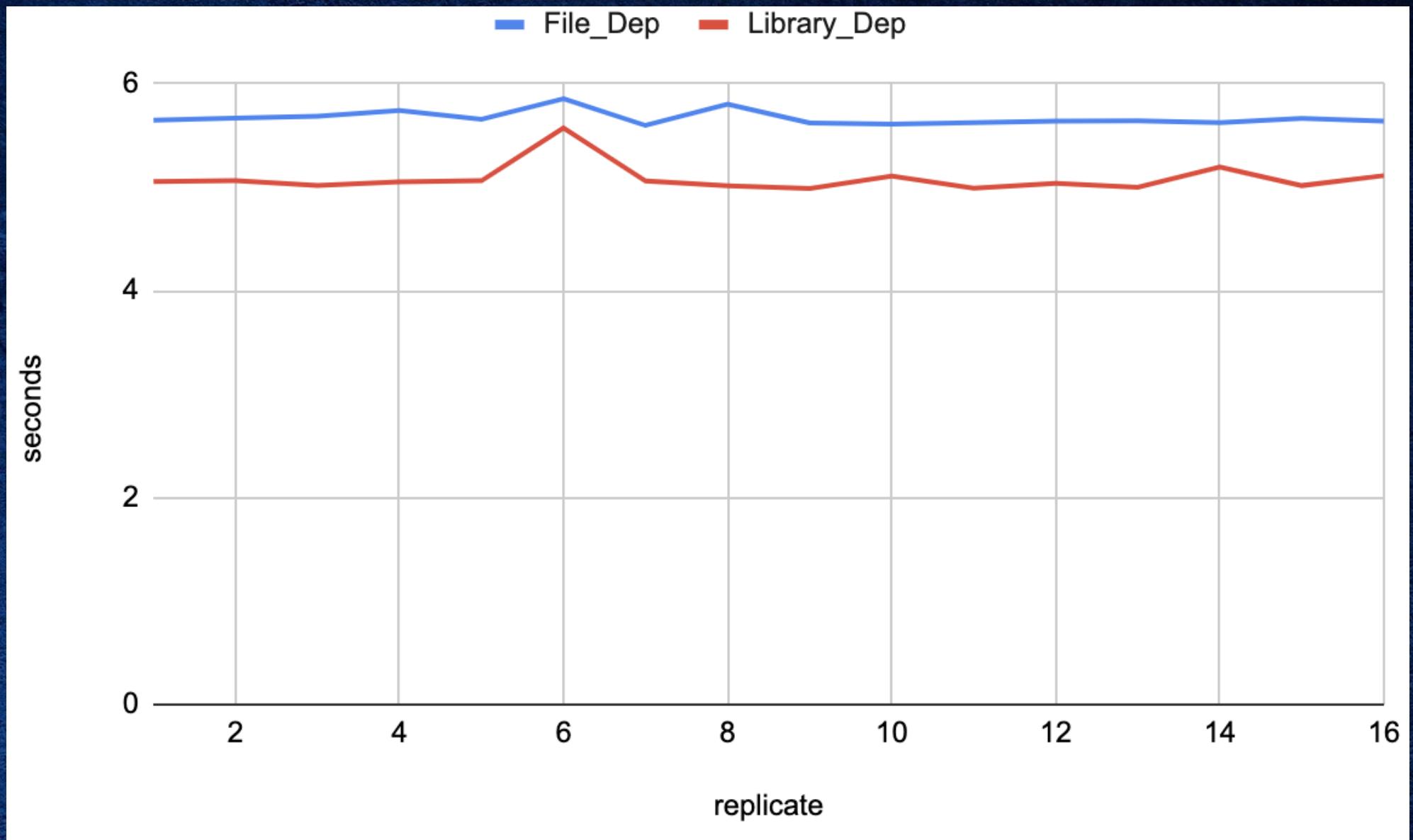


avg. File_Dep: 5.670822500
 avg. Library_Dep: 5.084755625 } = 0.586

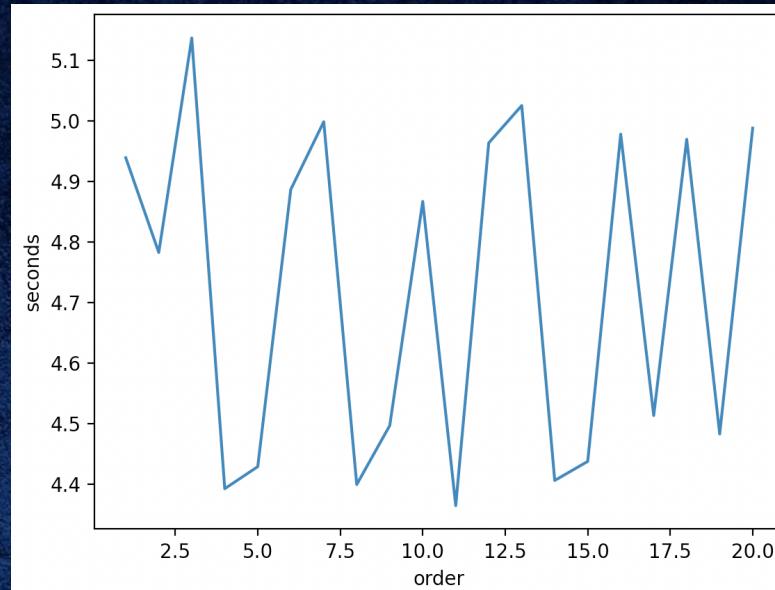
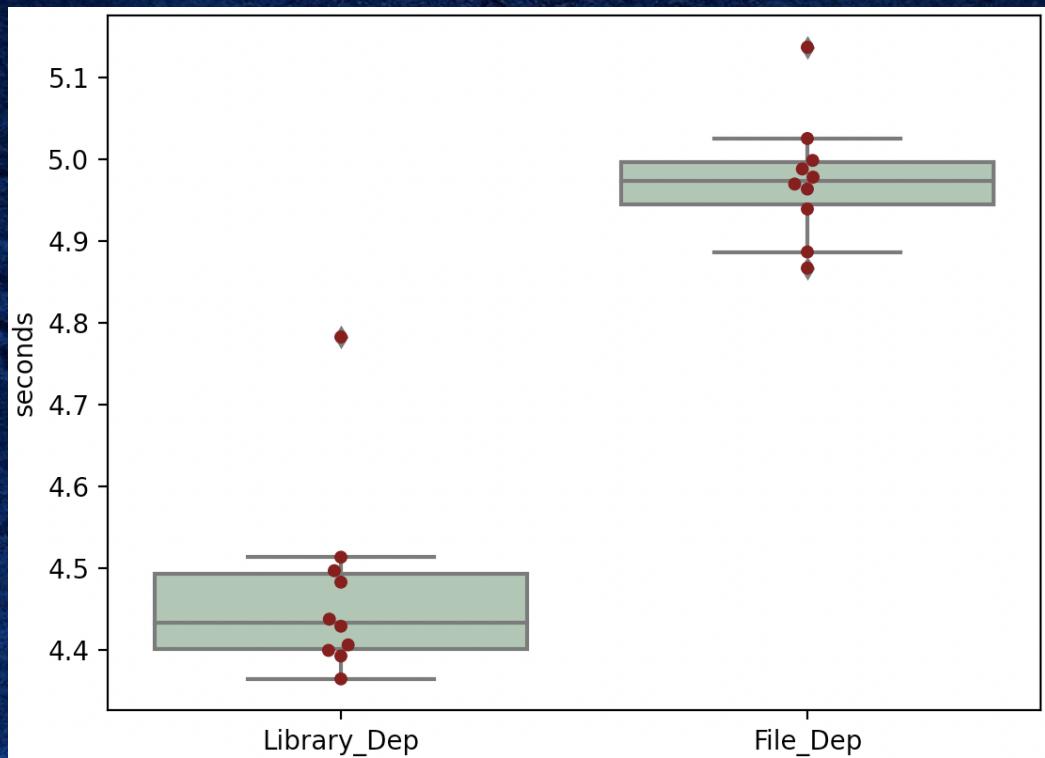
Depth: 8
 Total nodes: 1021
 Branch nodes: 509
 Leaf nodes: 512
 Groups: 69
 Dart version: 2.7.4
 Repetitions: 16

	sum_sq	df	F	PR(>F)
order	0.006132	1	0.469122	0.4990227
index	2.747795	1	210.206034	1.522e-14
order:index	0.002507	1	0.191803	0.6647802
Residual	0.366014	28	-	-

Trial 2 - Replicate Comparison



Trial 3 - 10% Leaves Changed

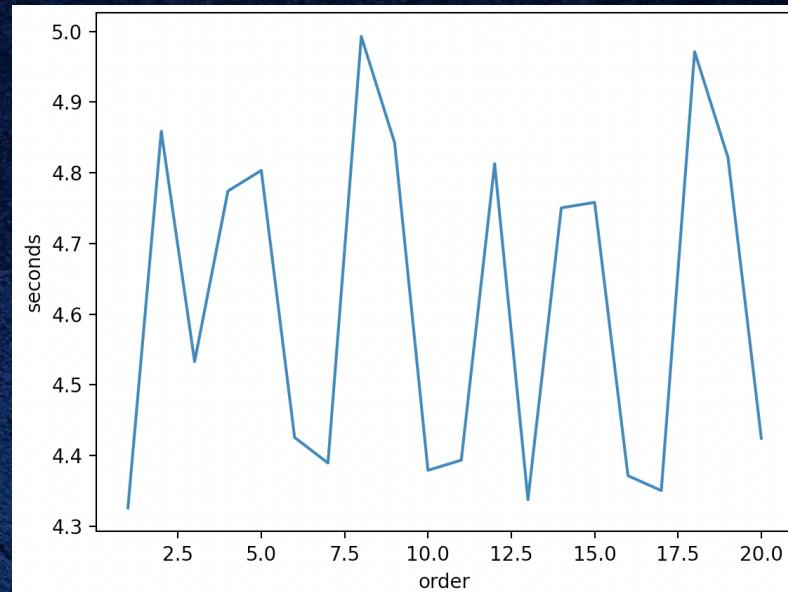
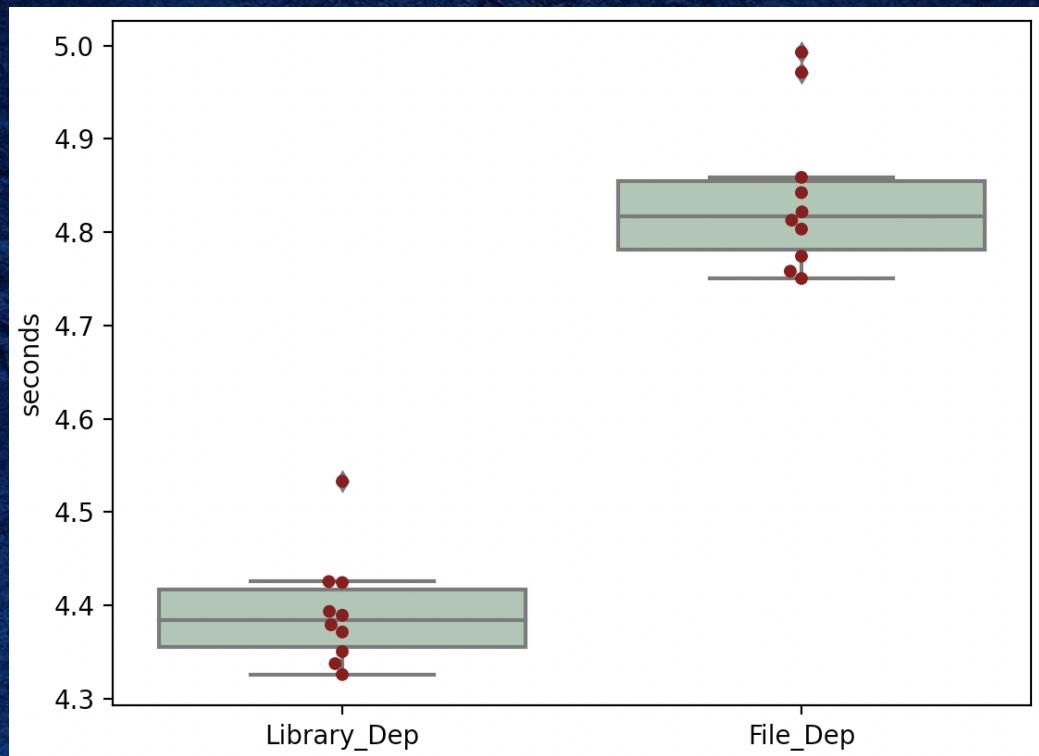


avg. File_Dep: 4.975775
 avg. Library_Dep: 4.470568 } = 0.505

Depth: 8
 Total nodes: 1021
 Branch nodes: 509
 Leaf nodes: 512
 Groups: 69
 Dart version: 2.7.4
 Repetitions: 10

	sum_sq	df	F	PR(>F)
order	0.006754	1	0.640073	0.4354028
index	1.279008	1	121.209072	7.092262e-09
order:index	0.004559	1	0.432031	0.5203383
Residual	0.168833	16	-	-

Trial 4 - 1% Leaves Changed

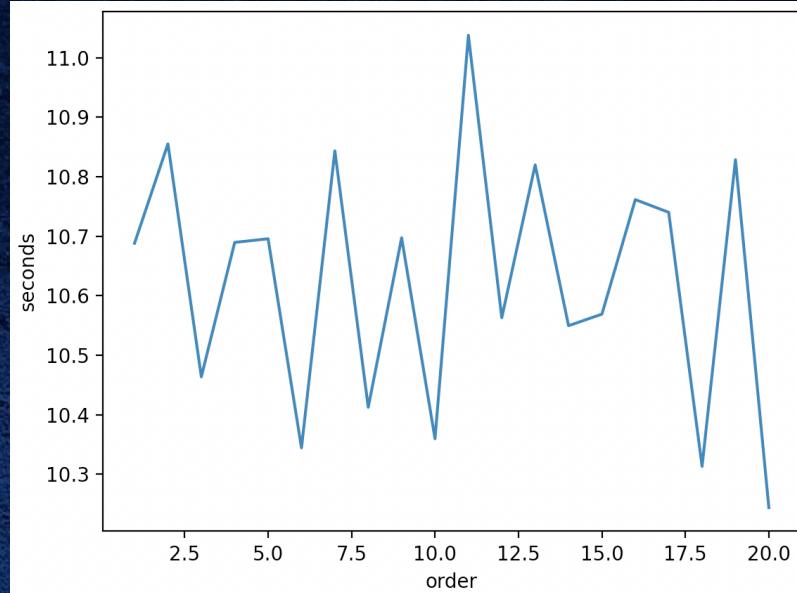
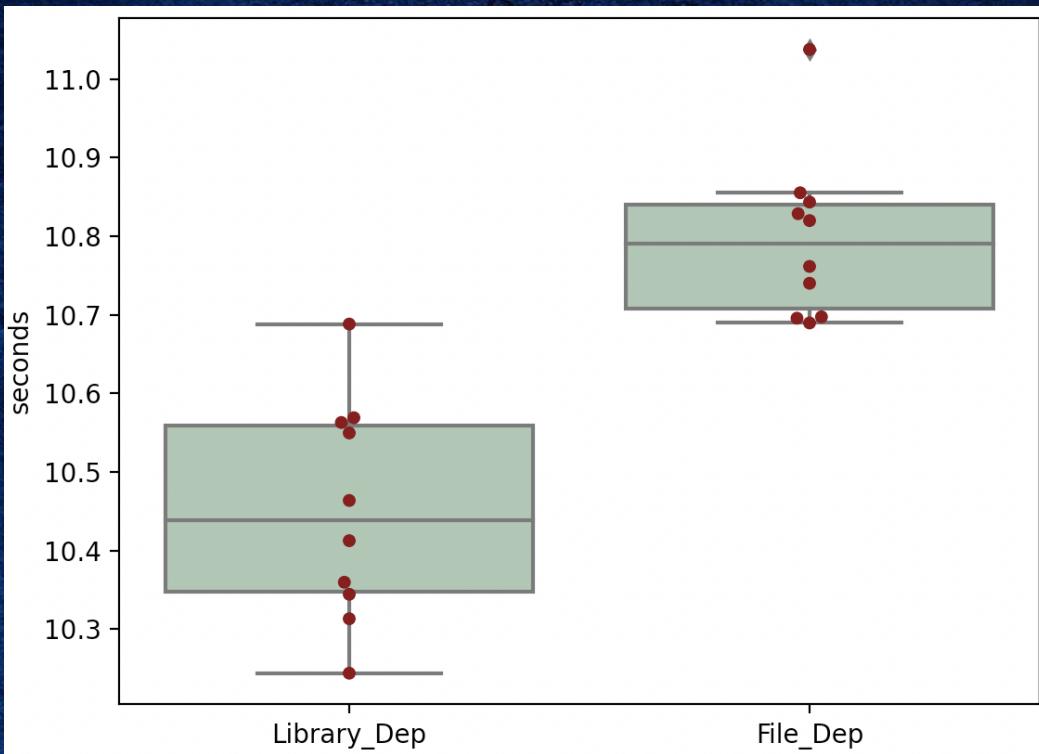


$$\text{avg. File_Dep: } 4.838602 \quad \text{avg. Library_Dep: } 4.393067 \quad \} = 0.4455$$

Depth: 8
 Total nodes: 1021
 Branch nodes: 509
 Leaf nodes: 512
 Groups: 69
 Dart version: 2.7.4
 Repetitions: 10

	sum_sq	df	F	PR(>F)
order	0.000508	1	0.088114	0.770407
index	0.992988	1	172.181131	5.578636e-10
order:index	0.001323	1	0.229367	0.6384735
Residual	0.092274	16	-	-

Trial 5 - Rebuild, Depth 12

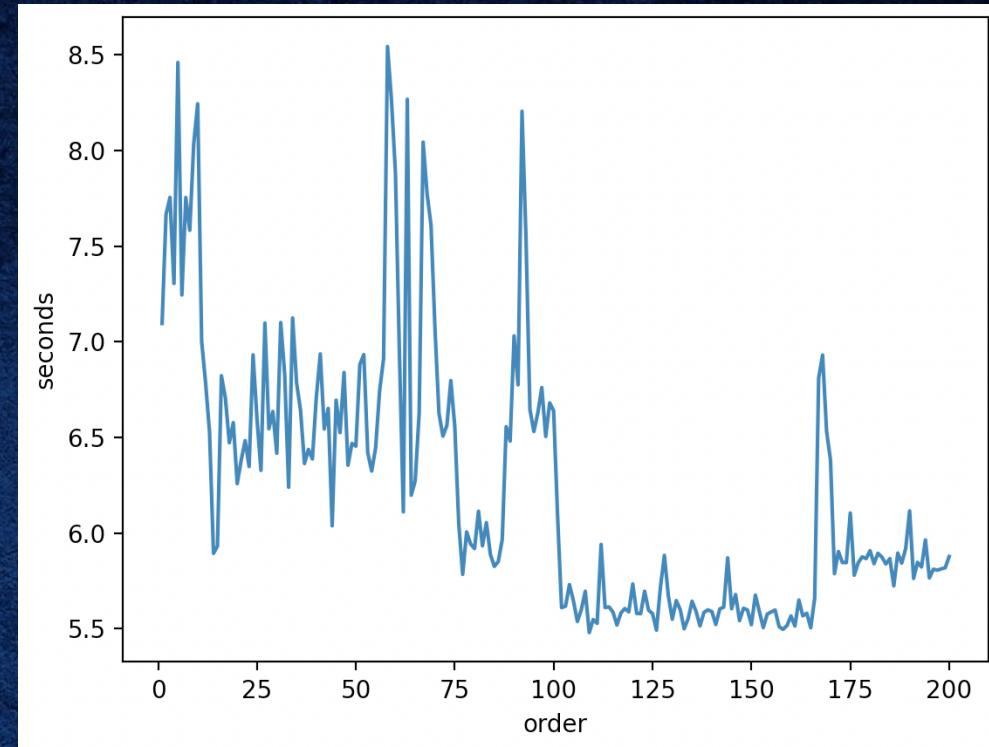
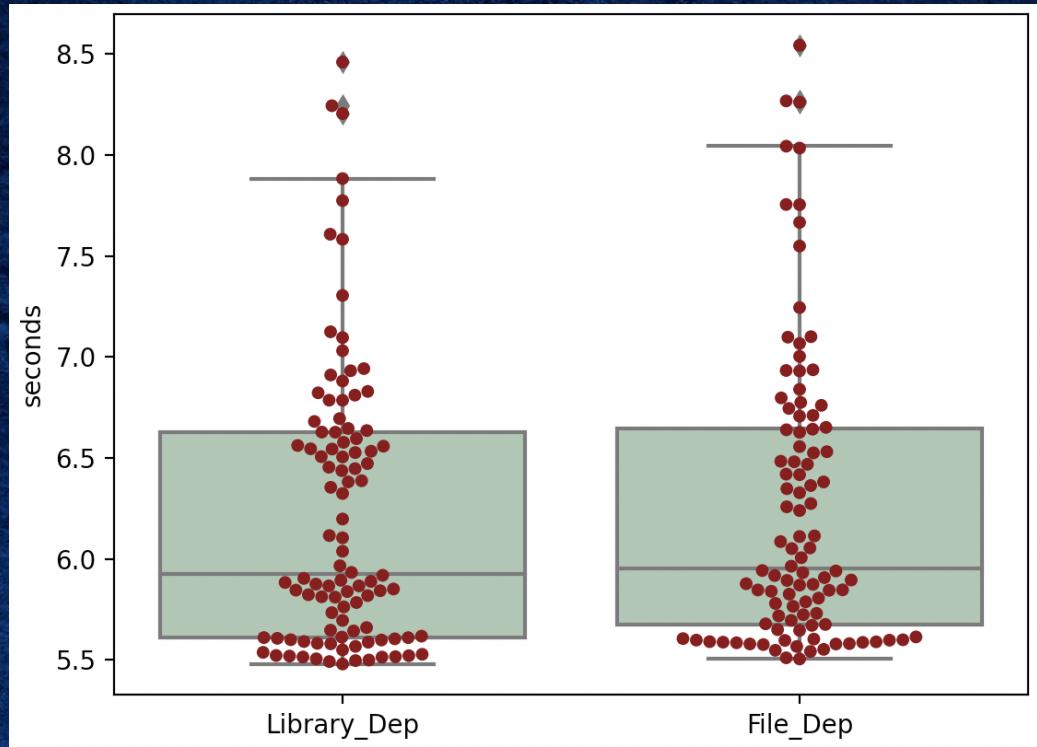


avg. File_Dep: 10.796764 } = 0.346
 avg. Library_Dep: 10.450601

Depth: 12
 Total nodes: 16381
 Branch nodes: 8189
 Leaf nodes: 8192
 Groups: 1093
 Dart version: 2.7.4
 Repetitions: 10

	sum_sq	df	F	PR(>F)
order	0.014045	1	0.947131	0.344935
index	0.592081	1	39.926724	0.000010
order:index	0.025535	1	1.721929	0.207959
Residual	0.237267	16	-	-

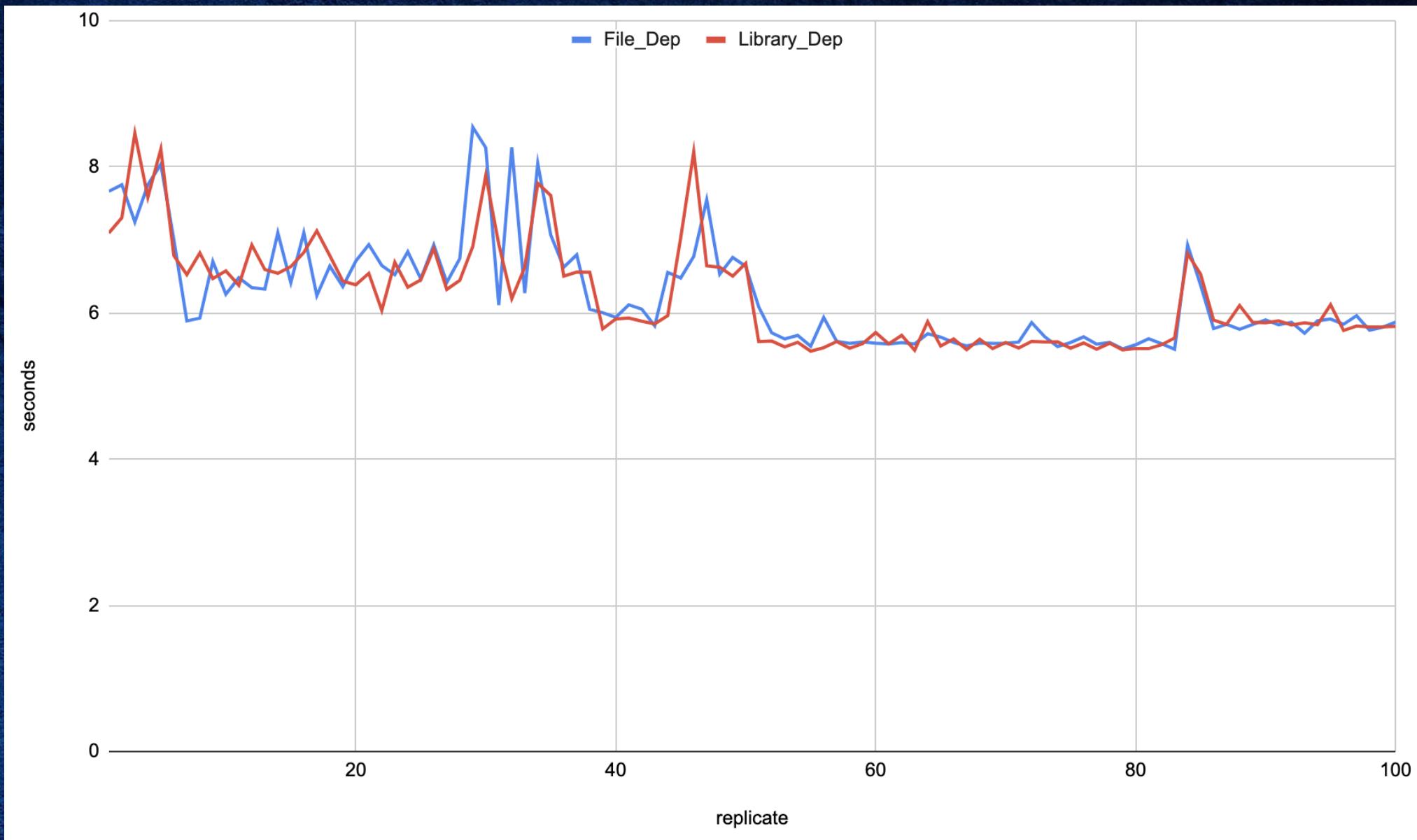
Trial 6 - Rebuild, Depth 10



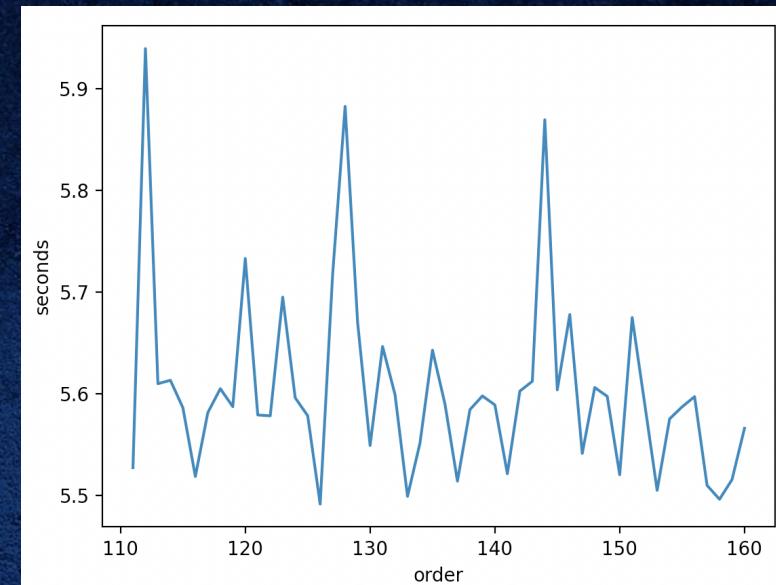
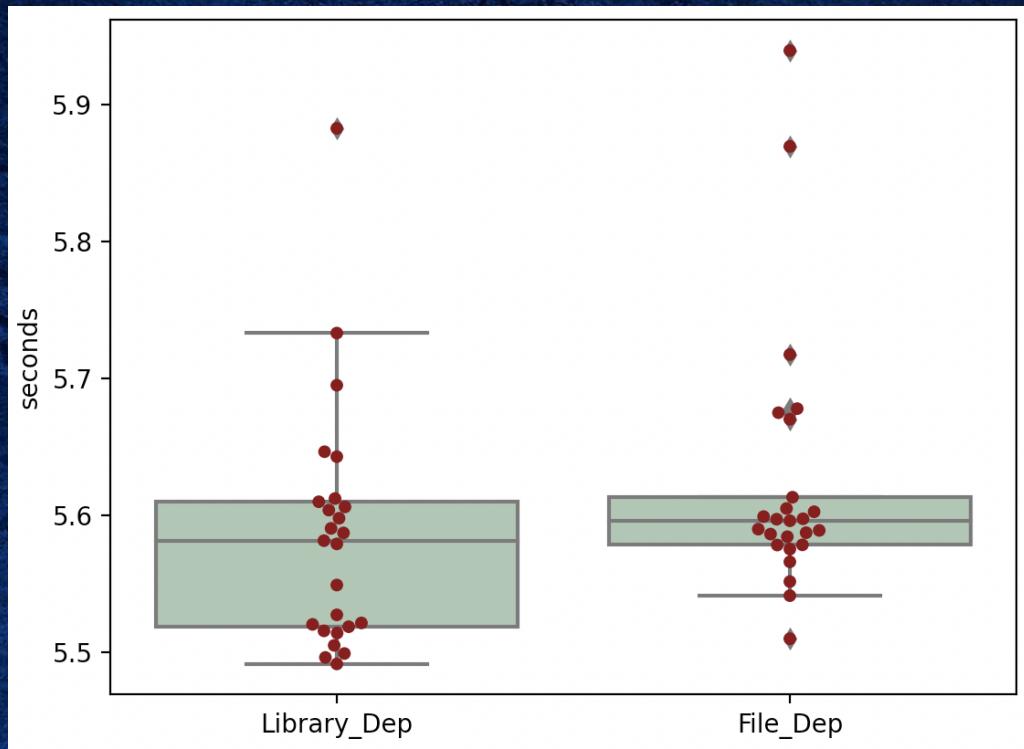
Depth: 10
 Total nodes: 4093
 Branch nodes: 2045
 Leaf nodes: 2048
 Groups: 273
 Dart version: **2.13.4**
 Repetitions: 100

	sum_sq	df	F	PR(>F)
order	42.133695	1	142.913831	4.286981e-25
index	0.050572	1	0.171538	0.6792019
order:index	0.000208	1	0.000704	0.9788601
Residual	57.7845	196	-	-

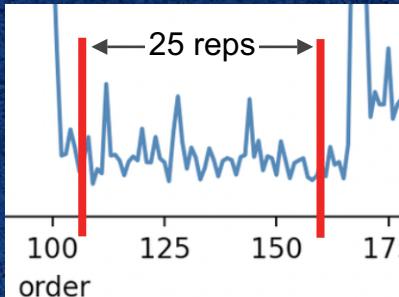
Trial 6 - Replicate Comparison



Trial 6 - Subset

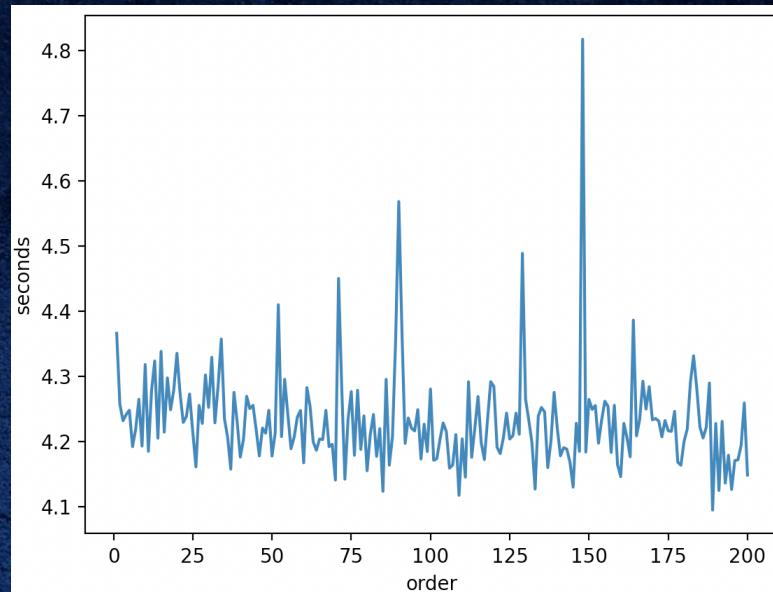
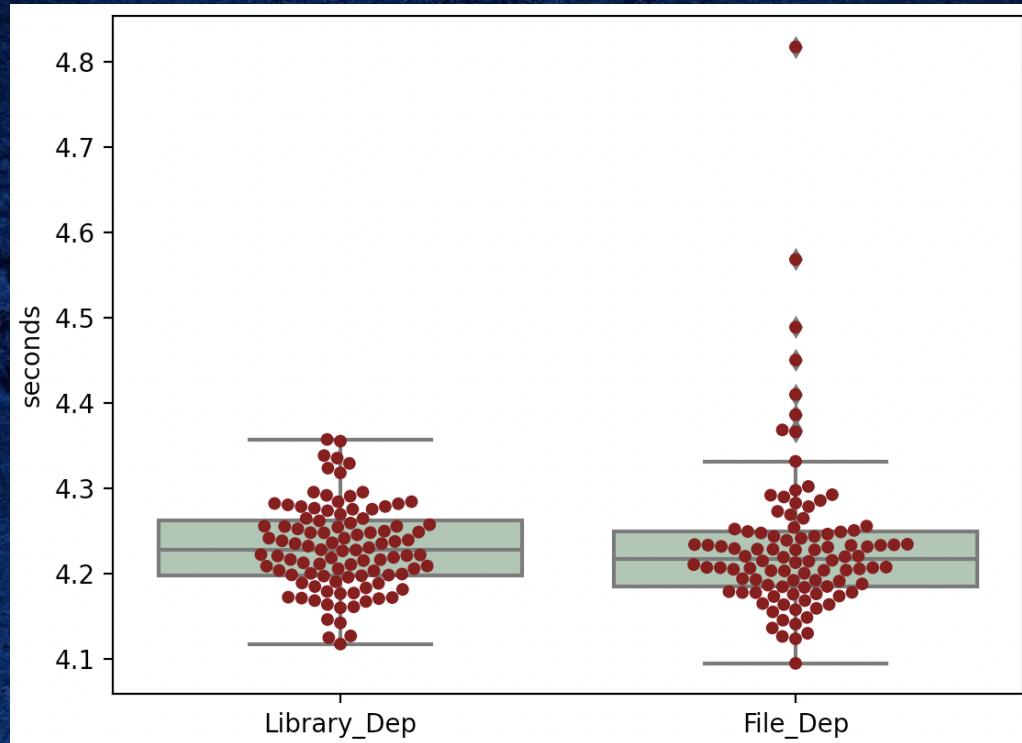


avg. File_Dep: 5.623734
 avg. Library_Dep: 5.584836 } = 0.039



	sum_sq	df	F	PR(>F)
order	0.027053	1	3.231171	0.078815
index	0.019353	1	2.311441	0.135269
order:index	0.000013	1	0.001606	0.968206
Residual	0.38514	46	-	-

Trial 7 - Rebuild, Depth 8

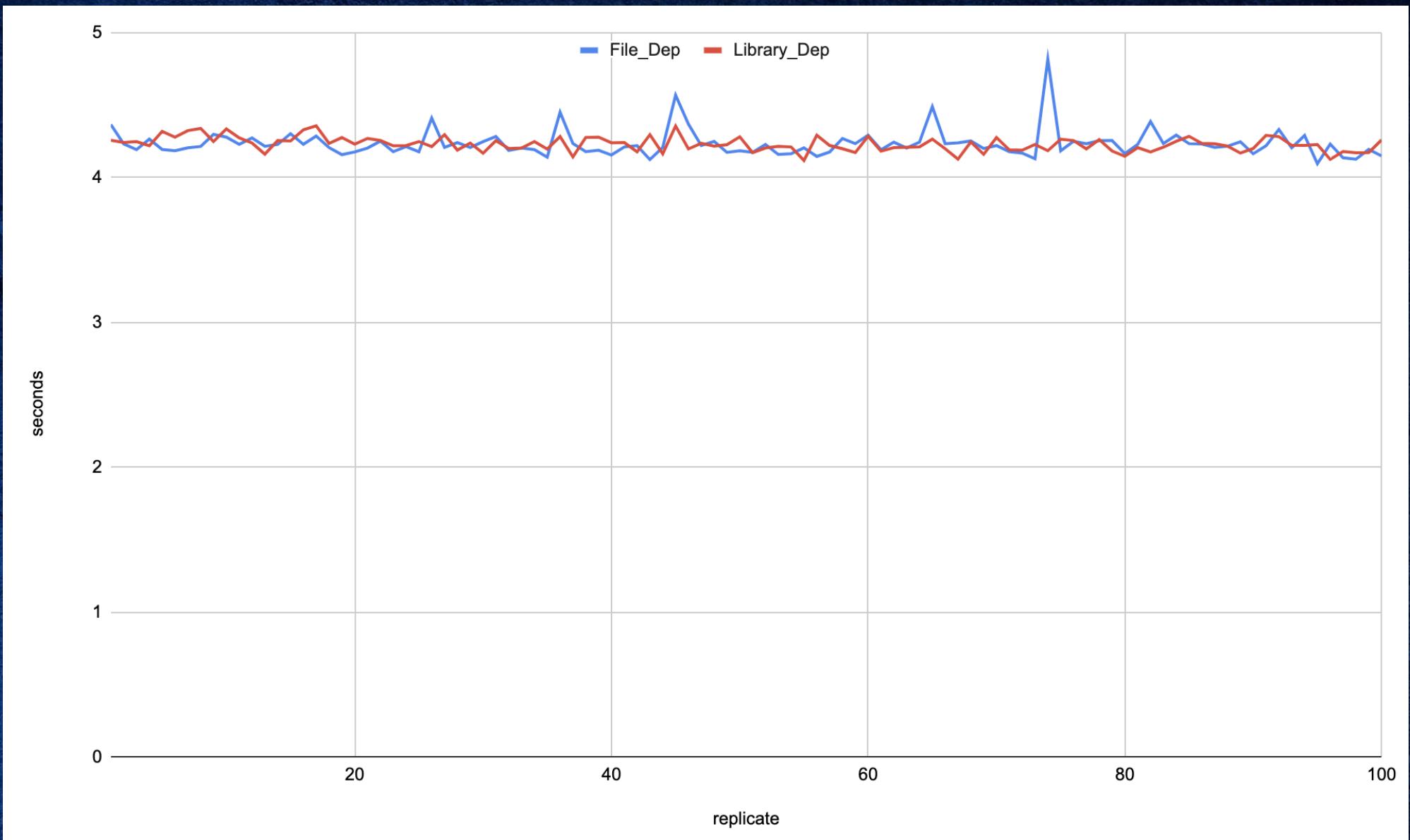


avg. File_Dep: 4.23432
 avg. Library_Dep: 4.23050 } = 0.038

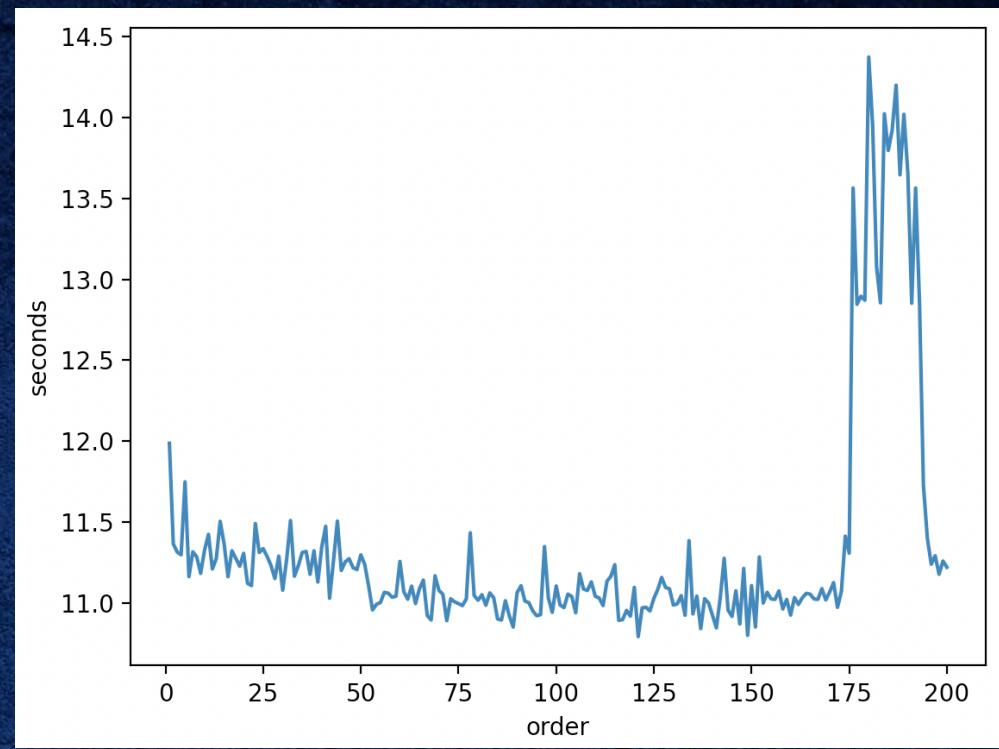
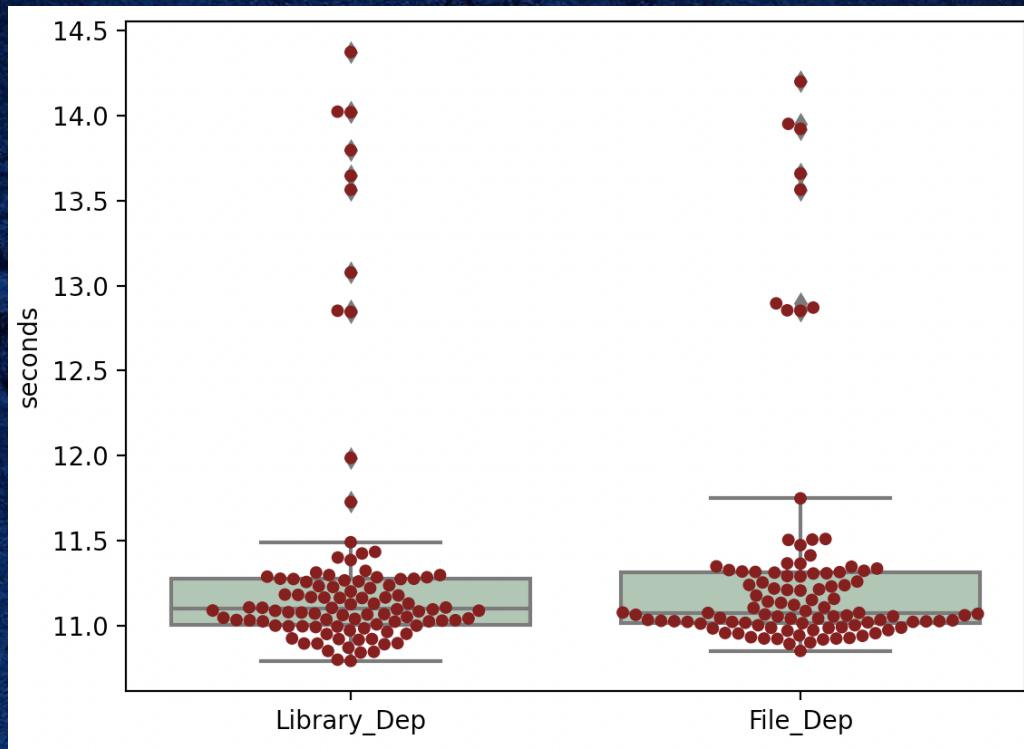
Depth: 8
 Total nodes: 1021
 Branch nodes: 509
 Leaf nodes: 512
 Groups: 69
 Dart version: **2.13.4**
 Repetitions: 100

	sum_sq	df	F	PR(>F)
order	0.023597	1	4.217357	0.04134
index	0.000727	1	0.129966	0.718855
order:index	0.012643	1	2.259554	0.134402
Residual	1.096681	196	—	—

Trial 7 - Replicate Comparison



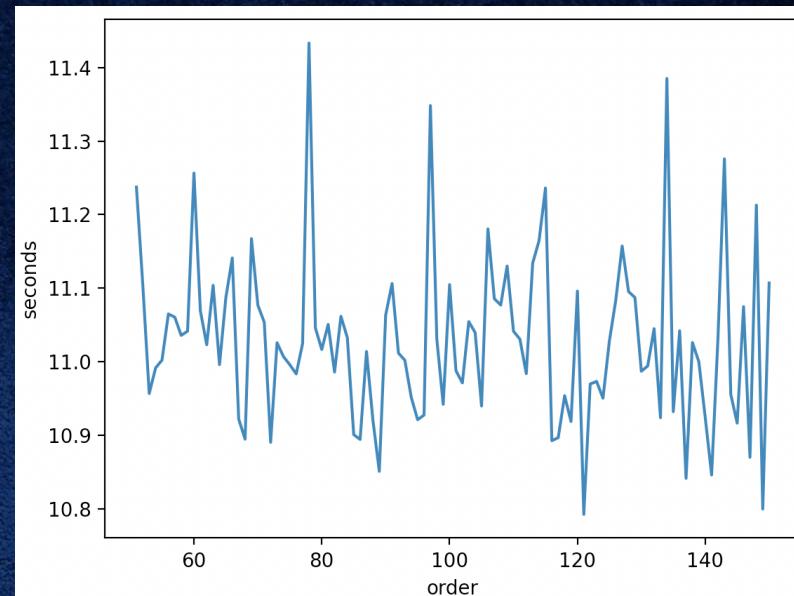
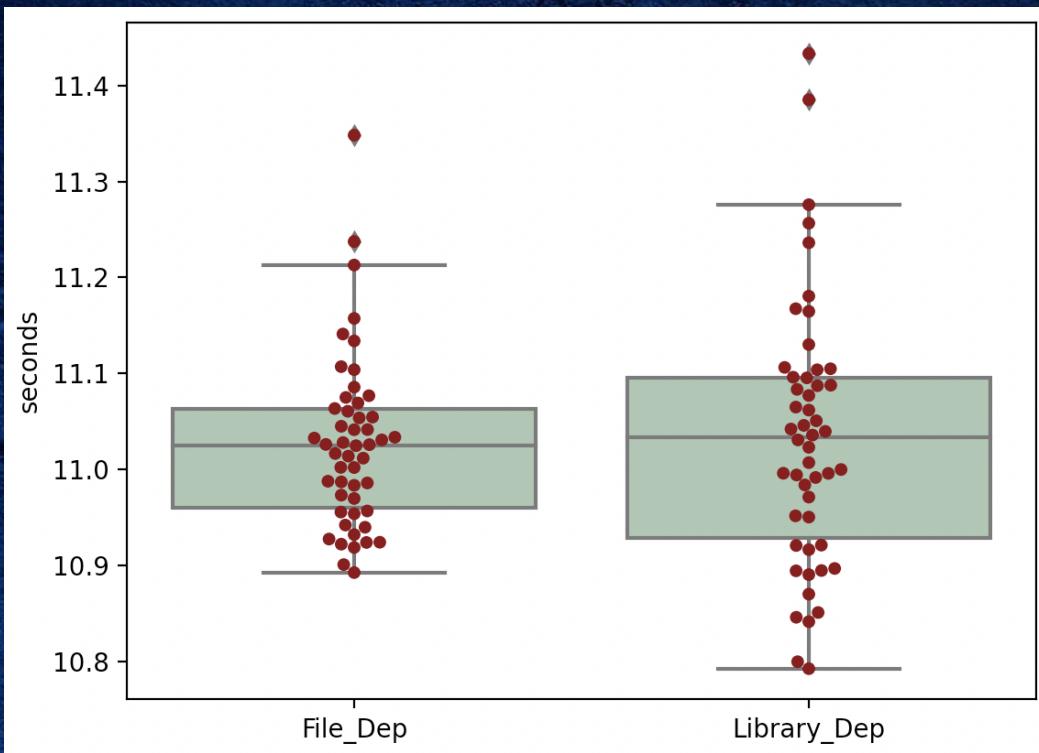
Trial 8 - Rebuild, Depth 12



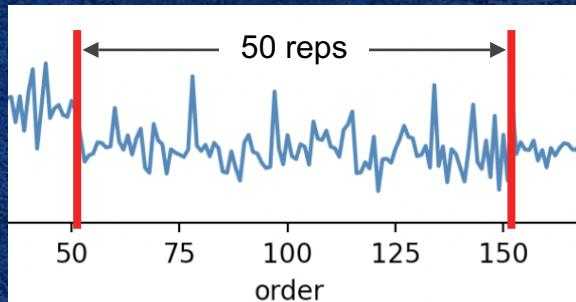
Depth: 12
Total nodes: 16381
Branch nodes: 8189
Leaf nodes: 8192
Groups: 1093
Dart version: **2.13.4**
Repetitions: 100

	sum_sq	df	F	PR(>F)
order	12.700661	1	27.412943	4.215337e-07
index	0.002548	1	0.0055	0.9409546
order:index	0.061905	1	0.133614	0.7151082
Residual	90.808548	196	-	-

Trial 8 - Subset

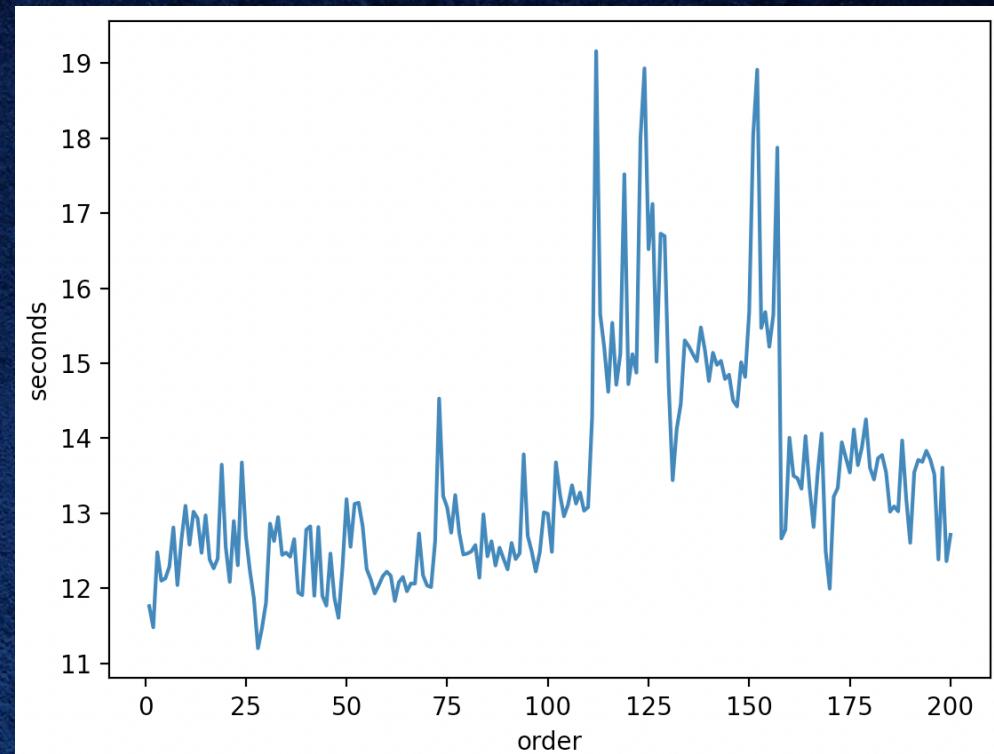
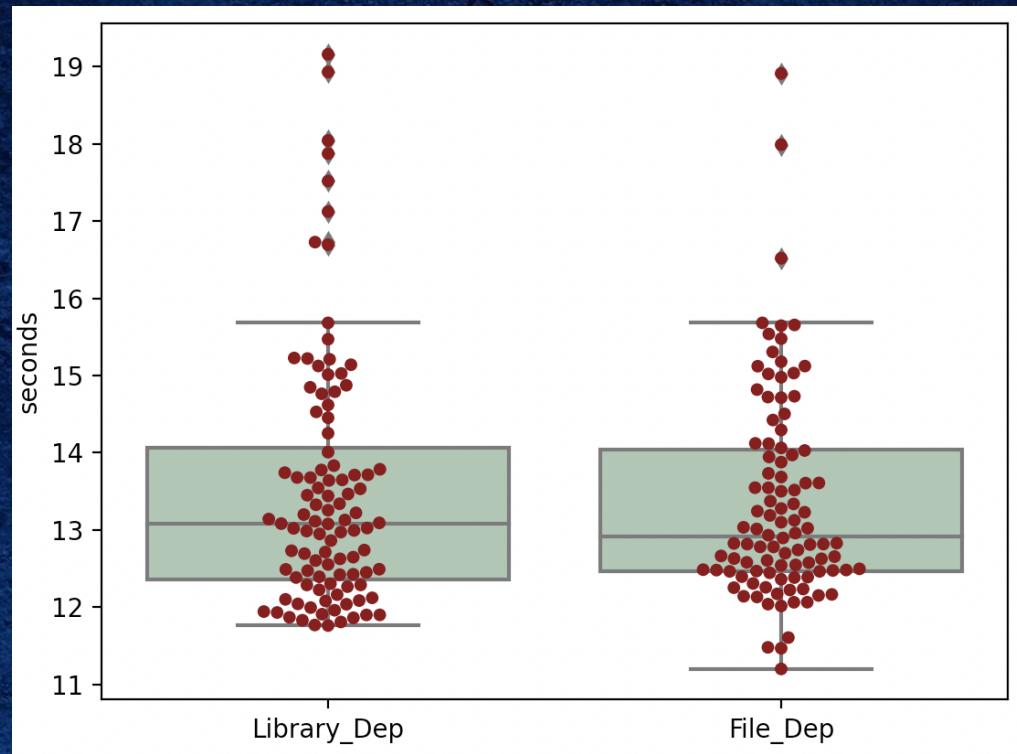


avg. File_Dep: 11.026511 } = 0.006
 avg. Library_Dep: 11.032707



	sum_sq	df	F	PR(>F)
order	0.019458	1	1.439978	0.233096
index	0.000978	1	0.072363	0.788505
order:index	0.002122	1	0.157064	0.692753
Residual	1.297207	46	—	—

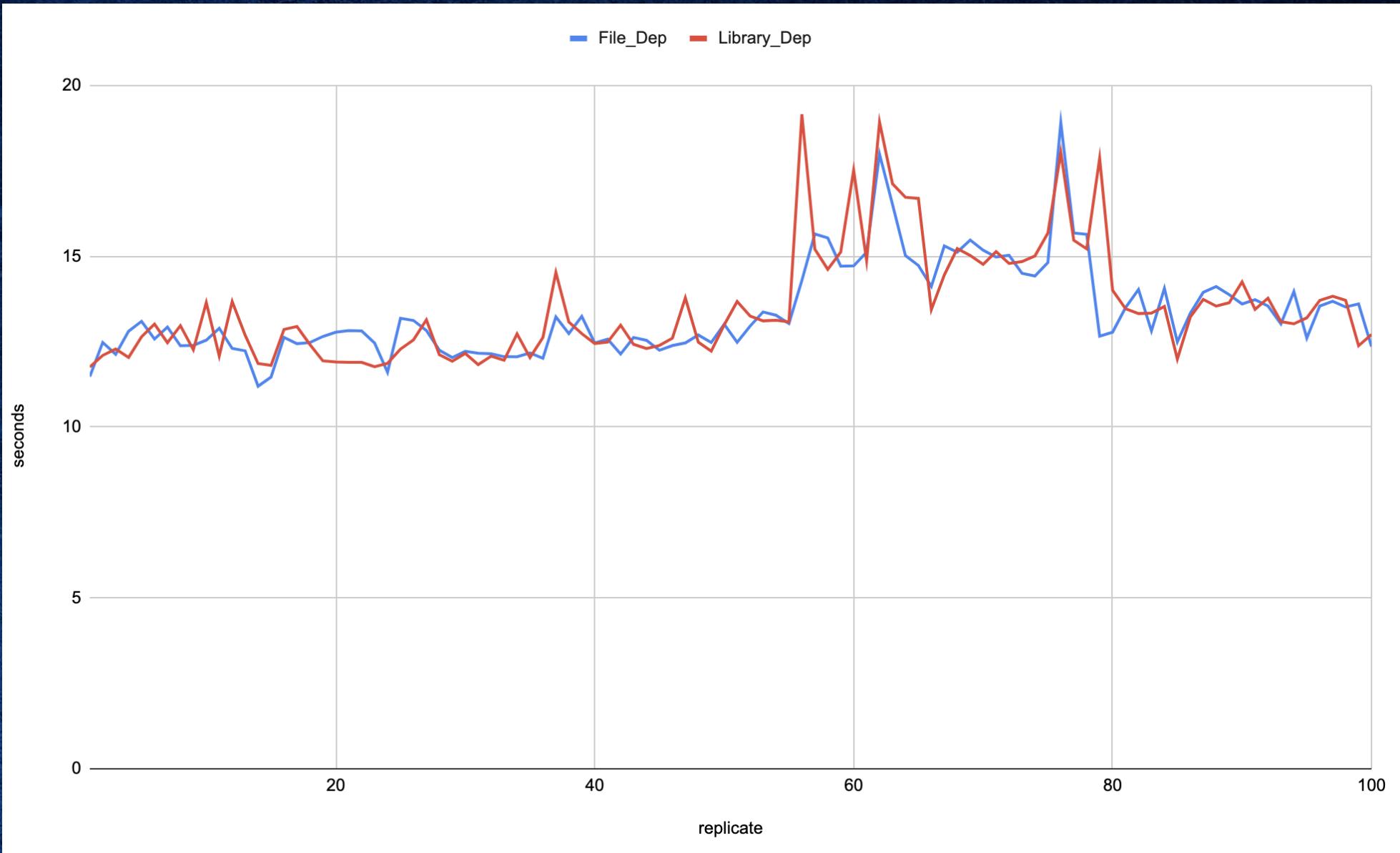
Trial 9 - 1% Leaves Changed



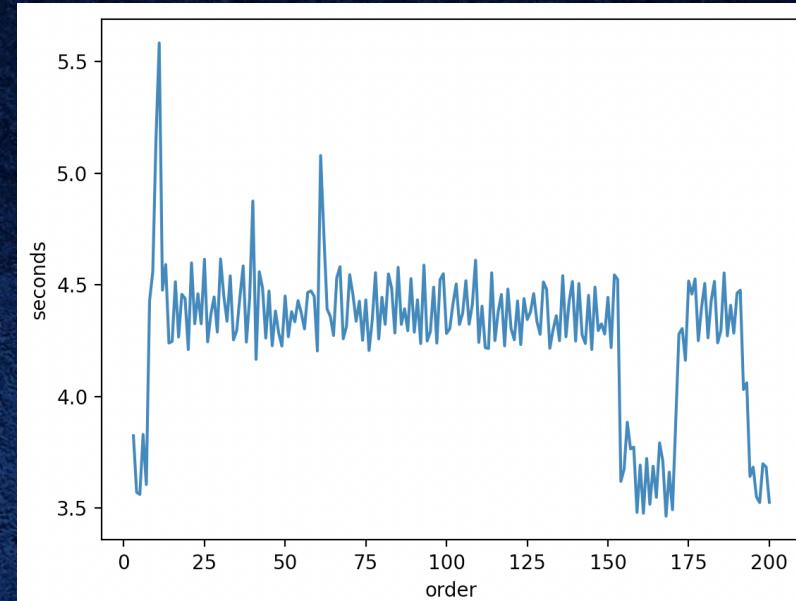
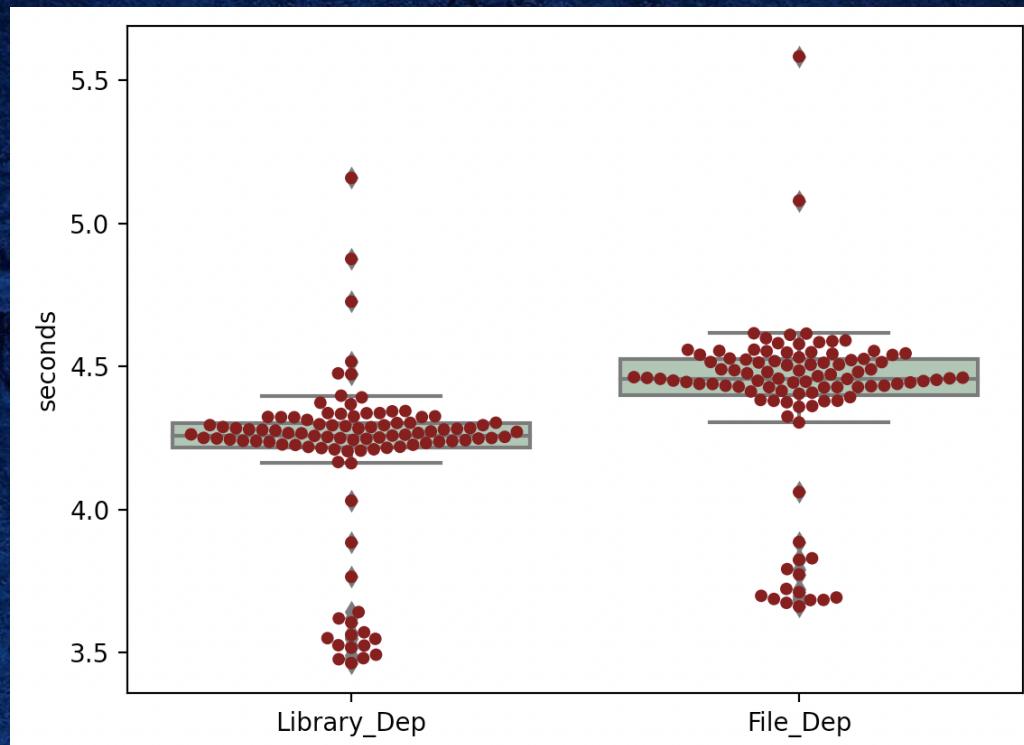
Depth: 12
 Total nodes: 16381
 Branch nodes: 8189
 Leaf nodes: 8192
 Groups: 1093
 Dart version: **2.13.4**
 Repetitions: 100

	sum_sq	df	F	PR(>F)
order	103.692577	1	60.050063	4.881253e-13
index	1.272163	1	0.73673	0.3917588
order:index	0.210486	1	0.121896	0.7273621
Residual	338.446692	196	-	-

Trial 9 - Replicate Comparison



Trial 10 - Block Chain Example



$$\begin{aligned} \text{avg. File_Dep: } & 4.3870 \\ \text{avg. Library_Dep: } & 4.1848 \end{aligned} \quad \} = 0.202$$

Block Chain Example
Dart version: **2.13.4**
Repetitions: 99

	sum_sq	df	F	PR(>F)
order	2.831682	1	36.323898	8.266298e-09
index	2.023438	1	25.956007	8.250686e-07
order:index	0.014726	1	0.188906	0.6643122
Residual	15.123551	194	-	-

Conclusion

- There is **strong evidence** in several of these trials to **reject** the null hypothesis.
- The majority of the v2.13 trials show **no evidence** against the null hypothesis.
- In the trials where the null hypothesis was rejected, the Part-of Libraries are faster.
- However, the trials also showed evidence in the order measurement, meaning that the variance between the different types of libraries is negligible compared to different computer pressures (likely the CPU time was being consumed by something else).
- Even though these trials show that Part-of Libraries are faster to build (assumption is that they create smaller dependency trees), the work required to make this change and the amount of improvement is diminishing returns.

Observations

- v2.13 is **way more stable**. The v2.7 trials would periodically lockup, making measurements difficult and making the strength of the analysis weaker due to the lower repetitions.
- Although no statistics were performed on the measurements for burn-in (the initial build which includes actions for prebuilding tools and creating the caches) whenever a burn-in was measured the Part-of Library appeared to be faster. (See detailed trial notes in the [repo](#) for the burn-ins which were measured.)
- No statistics directly comparing v2.13 and v2.7. However, v2.13 and v2.7 appeared to take approximately the same amount of time (Note that the generated code had to be slightly changed from v2.7 to v2.13, since v2.13 requires null awareness).
- The changes required to upgrade to v2.13 (null aware) will be a lot of work but with the stability and consistency of the builds it feels like it would be worth it.