

# Directed Random

*Abdullah Alsharif*

*3 January 2017*

## Analyse Wilcox Rank

Some Text Description (TODO: Table it write)

p.value	dbms	vs
0.93791	Postgres	AVM vs Directed Random Coverage
1.00000	SQLite	AVM vs Directed Random Coverage
1.00000	HyperSQL	AVM vs Directed Random Coverage
0.00000	Postgres	AVM vs Directed Random Mutation Score
0.00000	SQLite	AVM vs Directed Random Mutation Score
0.00000	HyperSQL	AVM vs Directed Random Mutation Score
0.00000	Postgres	AVM vs Directed Random Number Of evaluations
0.00000	SQLite	AVM vs Directed Random Number Of evaluations
0.00000	HyperSQL	AVM vs Directed Random Number Of evaluations
0.00000	Postgres	AVM vs Directed Random Test Generation Time
0.00000	SQLite	AVM vs Directed Random Test Generation Time
0.00000	HyperSQL	AVM vs Directed Random Test Generation Time

## Effect Size for AVM and Directed Random

Directed Random technique is greater in regard of time in test generation. As the following table shows that size is larger meaning that the effect size of the first technique is much superior when compared to AVM. Thresh-holding meant it investigate the effect size with different levels, to establish sensitivity. Meaning that it is precipitable to humans.

Some Text Description (TODO: Table it write)

value	size	rank.sum	dbms	vs	threshold
0	large	465	Postgres	Directed Random vs AVM Test Generation Time	0
0	large	465	SQLite	Directed Random vs AVM Test Generation Time	0
0	large	465	HyperSQL	Directed Random vs AVM Test Generation Time	0
0	large	465	Postgres	Directed Random vs AVM Test Generation Time	100
0	large	465	SQLite	Directed Random vs AVM Test Generation Time	100
0	large	465	HyperSQL	Directed Random vs AVM Test Generation Time	100
0	large	465	Postgres	Directed Random vs AVM Test Generation Time	200
0	large	465	SQLite	Directed Random vs AVM Test Generation Time	200
0	large	465	HyperSQL	Directed Random vs AVM Test Generation Time	200
0	large	465	Postgres	Directed Random vs AVM Test Generation Time	300
0	large	465	SQLite	Directed Random vs AVM Test Generation Time	300
0	large	465	HyperSQL	Directed Random vs AVM Test Generation Time	300
0	large	465	Postgres	Directed Random vs AVM Test Generation Time	400
0	large	465	SQLite	Directed Random vs AVM Test Generation Time	400
0	large	465	HyperSQL	Directed Random vs AVM Test Generation Time	400
0	large	465	Postgres	Directed Random vs AVM Test Generation Time	500
0	large	465	SQLite	Directed Random vs AVM Test Generation Time	500

value	size	rank.sum	dbms	vs	threshold
0	large	465	HyperSQL	Directed Random vs AVM Test Generation Time	500
0	large	465	Postgres	Directed Random vs AVM Test Generation Time	600
0	large	465	SQLite	Directed Random vs AVM Test Generation Time	600
0	large	465	HyperSQL	Directed Random vs AVM Test Generation Time	600
0	large	465	Postgres	Directed Random vs AVM Test Generation Time	700
0	large	465	SQLite	Directed Random vs AVM Test Generation Time	700
0	large	465	HyperSQL	Directed Random vs AVM Test Generation Time	700
0	large	465	Postgres	Directed Random vs AVM Test Generation Time	800
0	large	465	SQLite	Directed Random vs AVM Test Generation Time	800
0	large	465	HyperSQL	Directed Random vs AVM Test Generation Time	800
0	large	465	Postgres	Directed Random vs AVM Test Generation Time	900
0	large	465	SQLite	Directed Random vs AVM Test Generation Time	900
0	large	465	HyperSQL	Directed Random vs AVM Test Generation Time	900
0	large	465	Postgres	Directed Random vs AVM Test Generation Time	1000
0	large	465	SQLite	Directed Random vs AVM Test Generation Time	1000
0	large	465	HyperSQL	Directed Random vs AVM Test Generation Time	1000

## HeatMap Plot of mutant analysis per technique for each Operator

Some Text Description for the heat map

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

