CWI
Centrum Wiskunde & Informatica

bedatadriven

Latest developments around Renjin

Maarten-Jan Kallen & Hannes Mühleisen

R Summit & Workshop 2015, 2015-06-27

What is renjin

- R on the JVM
- Compatibility is paramount, not just academic exercise (e.g. automatic Fortran/C translations)
- R anywhere on any data format (e.g. Cloud environments)
- Increased performance through lazy evaluation, parallel execution, …
- Easy to plug any Java code into R analysis, easy to plug Renjin into java projects

Renjin Timeline

- 2013: Public launch of Renjin at useR!
- 2014: CRAN Package build system
  - Building, testing, regressions
  - Public repository, automagic package installation
- 2014: Start of “R as a query language” project
- 2015 (Q4): 3-year EU-funded project to implement S4 for Bioconductor compatibility

Renjin

- `anyNA(x)` introduced in R 3.0, as `any(is.na(x))` is inefficient for large vectors.
  - Solution: Mash into one function, farm implementation out to C
- But: Does not solve many similar cases

```
y <- is.na(x); any(y)
any(is.na(x) | is.na(y))
all(!is.na(x))
```

Abstraction in Renjin

```
> a <- 1:10^9
> a[1000000] <- NA #harr harr
```

```
> system.time(print(anyNA(a)))[[3]]
[1] TRUE
[1] 0.001
> system.time(print(any(is.na(a))))[[3]]
[1] TRUE
[1] 2.23
```

GNU R

```
> system.time(print(any(is.na(a))))[[3]]
[1] TRUE
[1] 0.05
```

Renjin

“R as a query language”

`dplyr::, data.table::`

`[` `subset()`

- Observation 1: Lots of data wrangling happening in R scripts

`merge()`

`$` `aggregate()`

“R as a query language”

- Observation 2: Things get slow quickly as vectors get longer
- Lots of optimisation opportunities, but how?
  - State of the art: Tactical optimisation/Band aids

“R as a query language”

- Proposal: Treat R scripts as declaration of intent (not as a procedural contract written in blood)
  - Then we can optimise strategically!

Rule-based query optimisation

$$\pi_{name,title}(\sigma_{dept='Music'}(instructor \bowtie course))$$

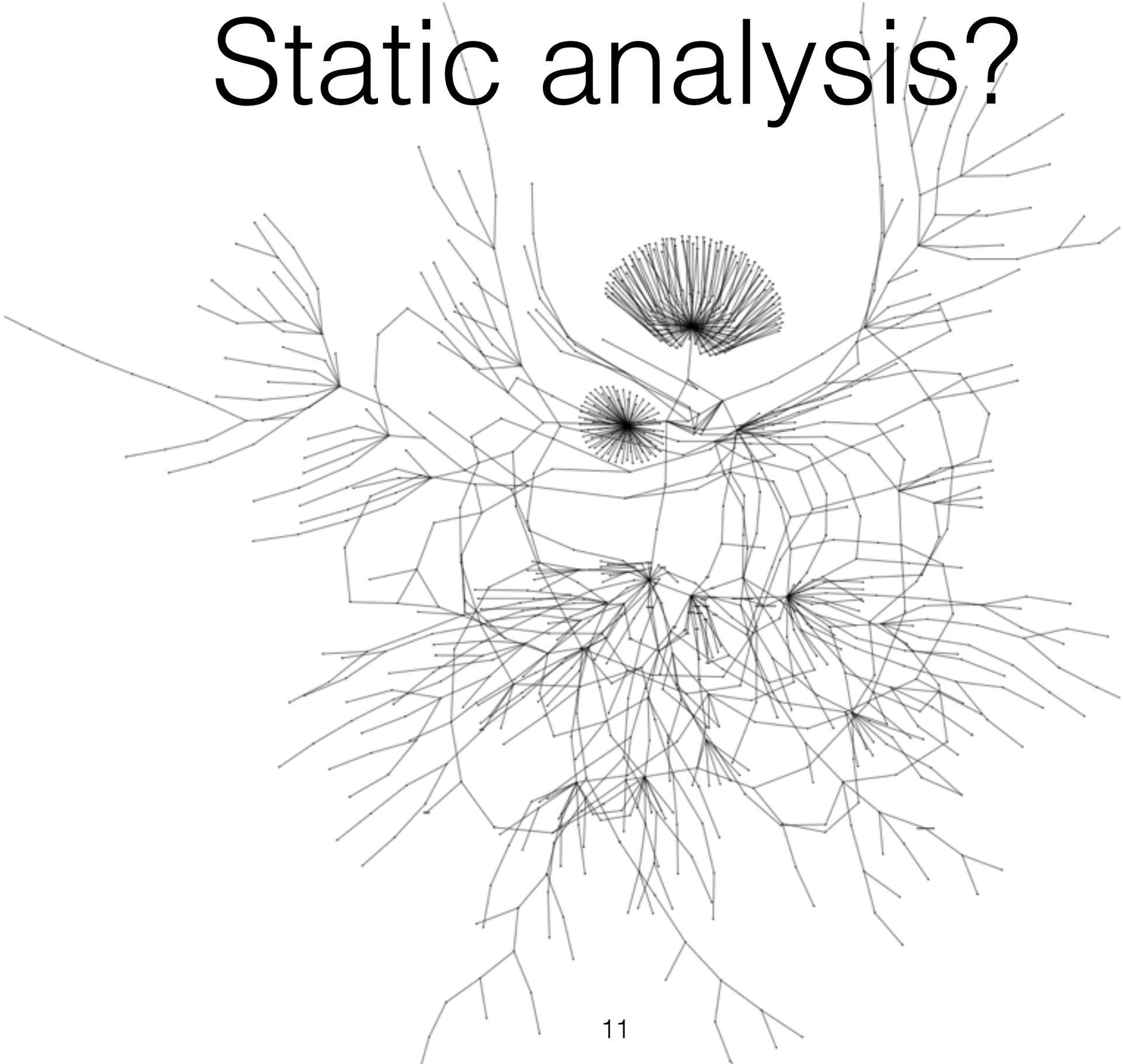
$$\sigma_{\theta 0}(E_1 \bowtie_\theta E_2) \equiv (\sigma_{\theta 0}(E_1)) \bowtie_\theta E_2$$

$$\pi_{name,title}(\sigma_{dept='Music'}(instructor) \bowtie course)$$

Optimisations

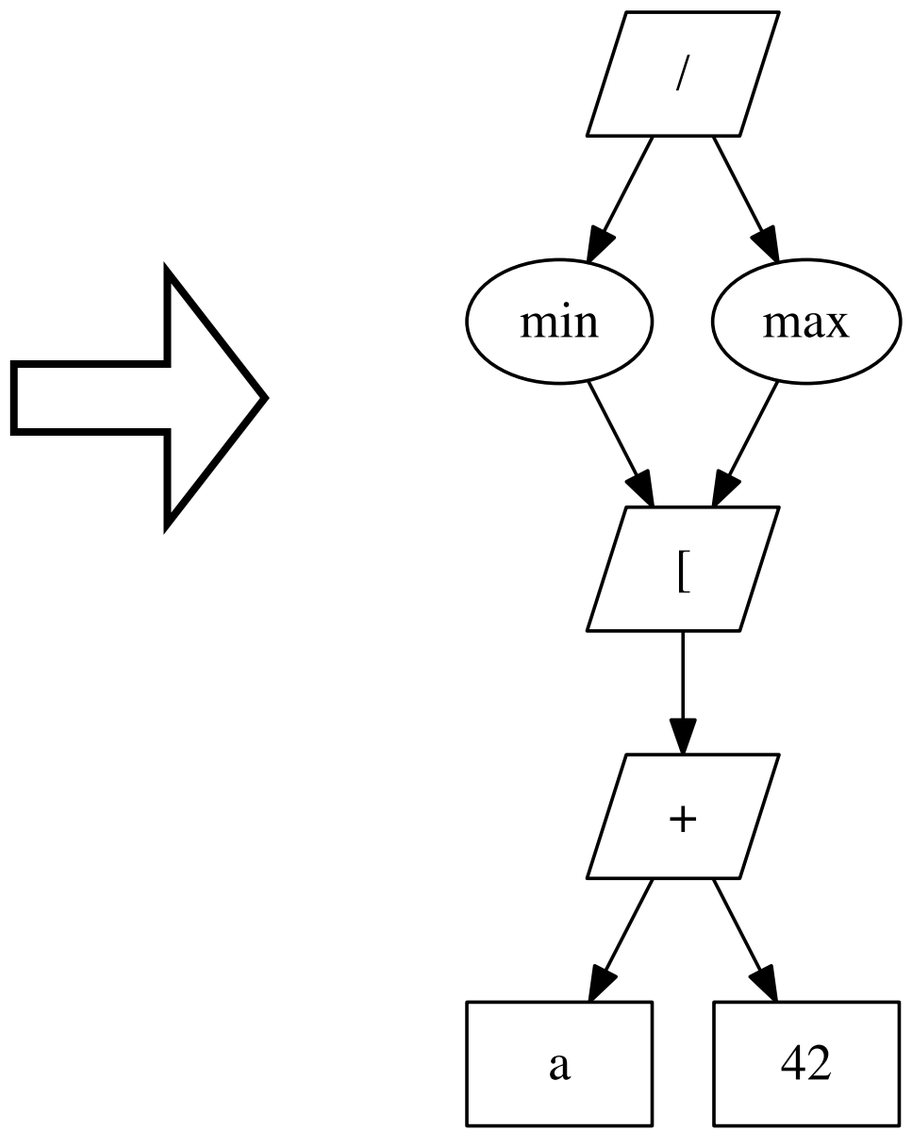
- Selection Pushdown
- Data-parallel scheduling
- Function specialisation/vectorisation
- Common expression elimination/caching
- Redundant computation elimination

Static analysis?



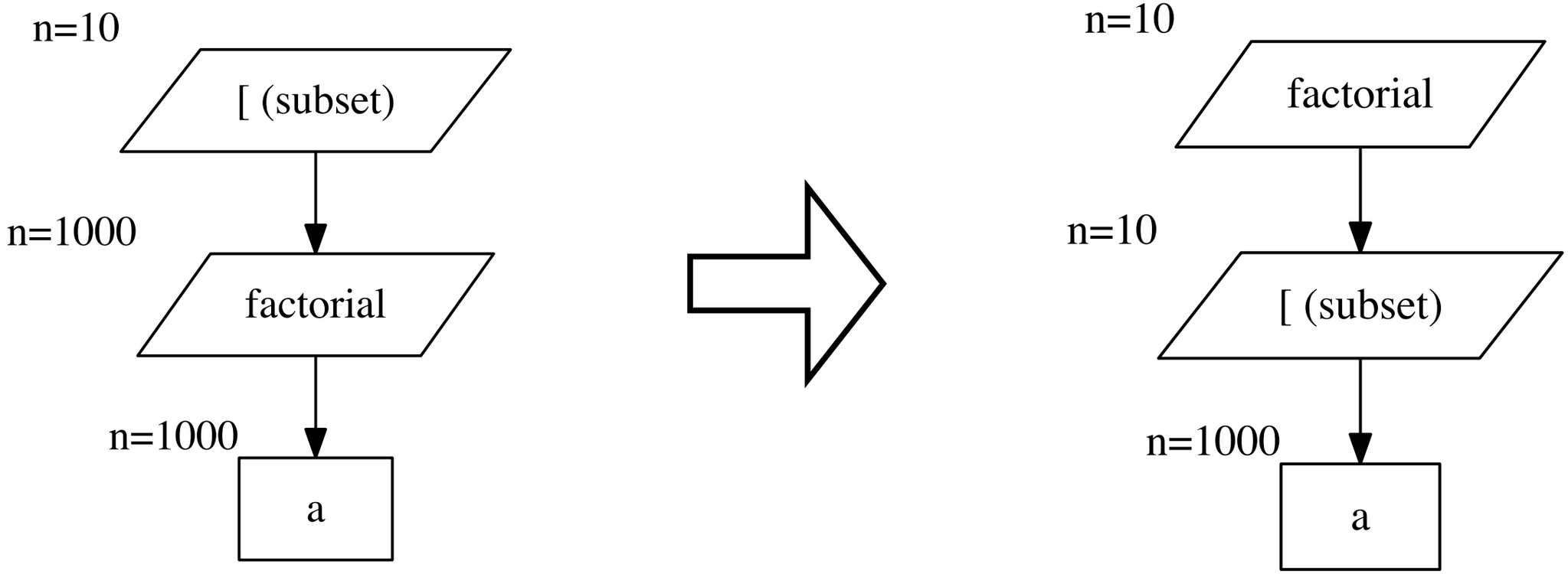
Deferred Evaluation

```
a <- 1:1000
b <- a + 42
c <- b[1:10]
d <- min(c) / max(c)
print(d)
```

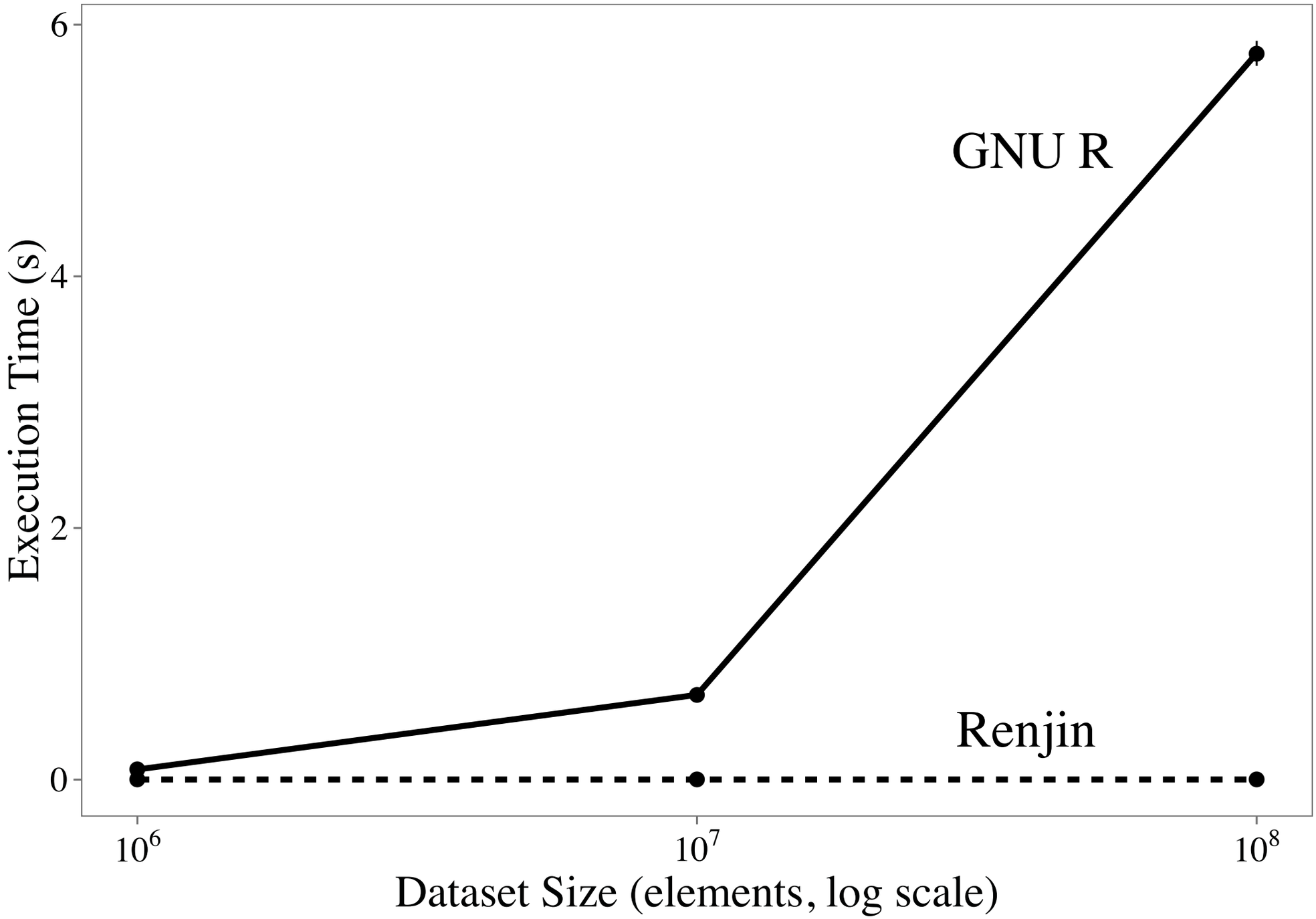


Pushdown

```
b <- factorial(a)
c <- b[1:10]
print(c)
```

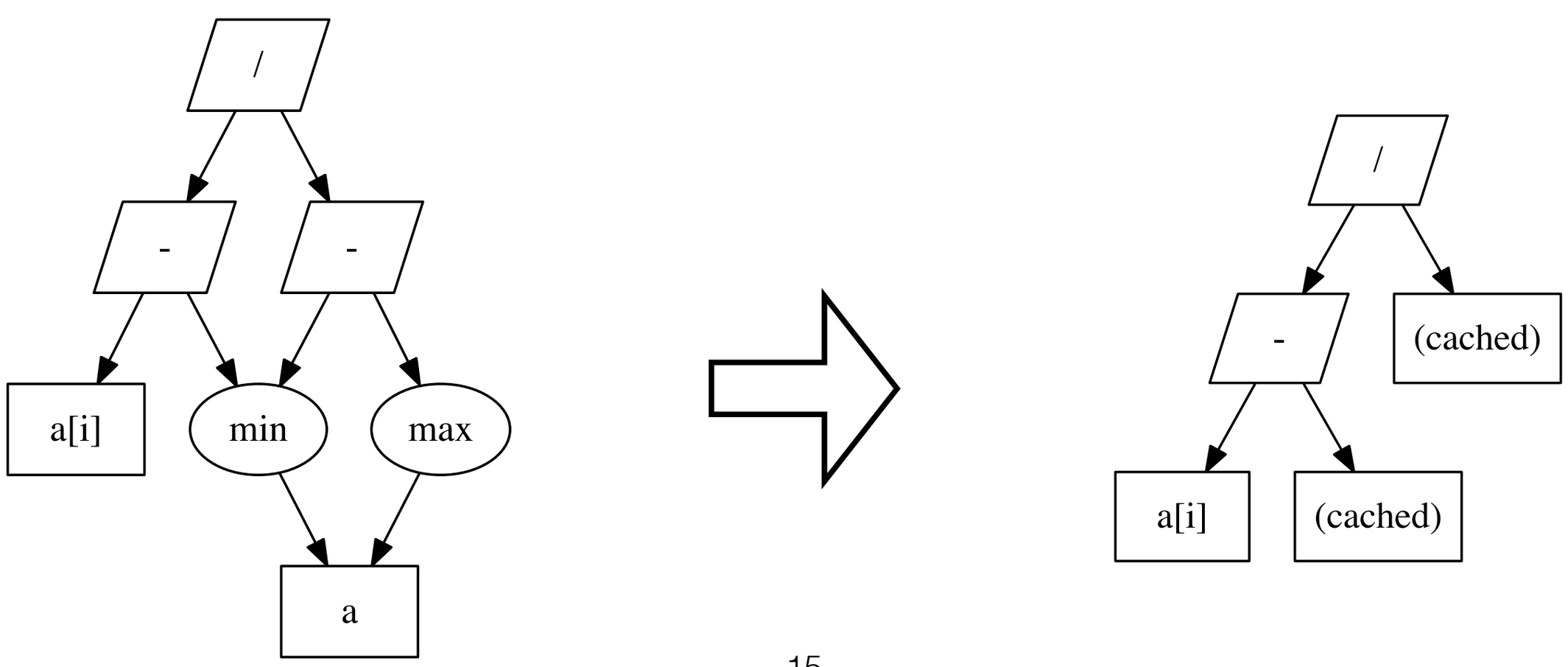


Pushdown

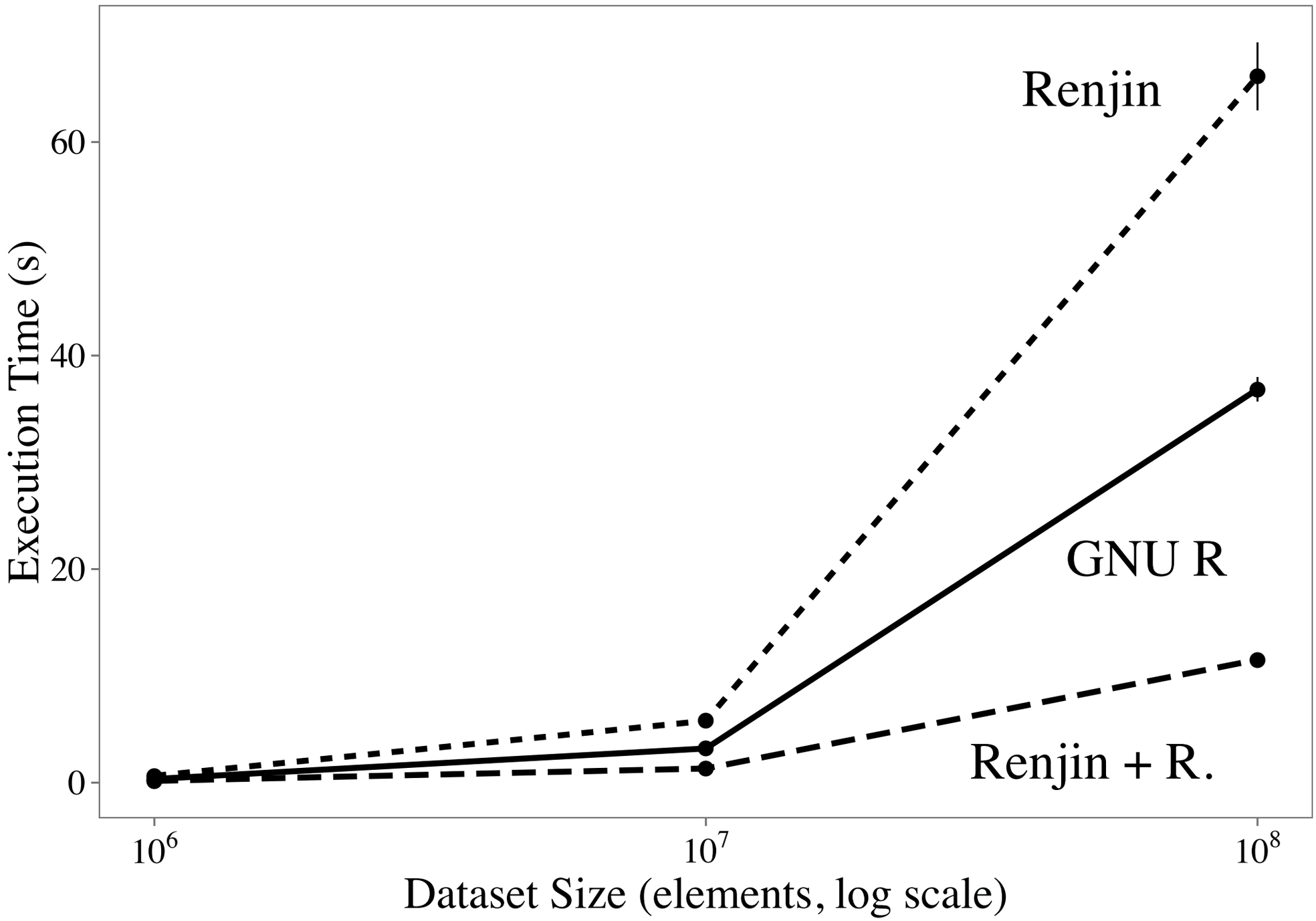

6
4
2
0
Execution Time (s)
GNU R
Renjin
10^6
10^7
10^8
Dataset Size (elements, log scale)

Recycling

```
for (i in 1:100)
  print((a[i] - min(a))/(max(a)-min(a)))
```



Recycling

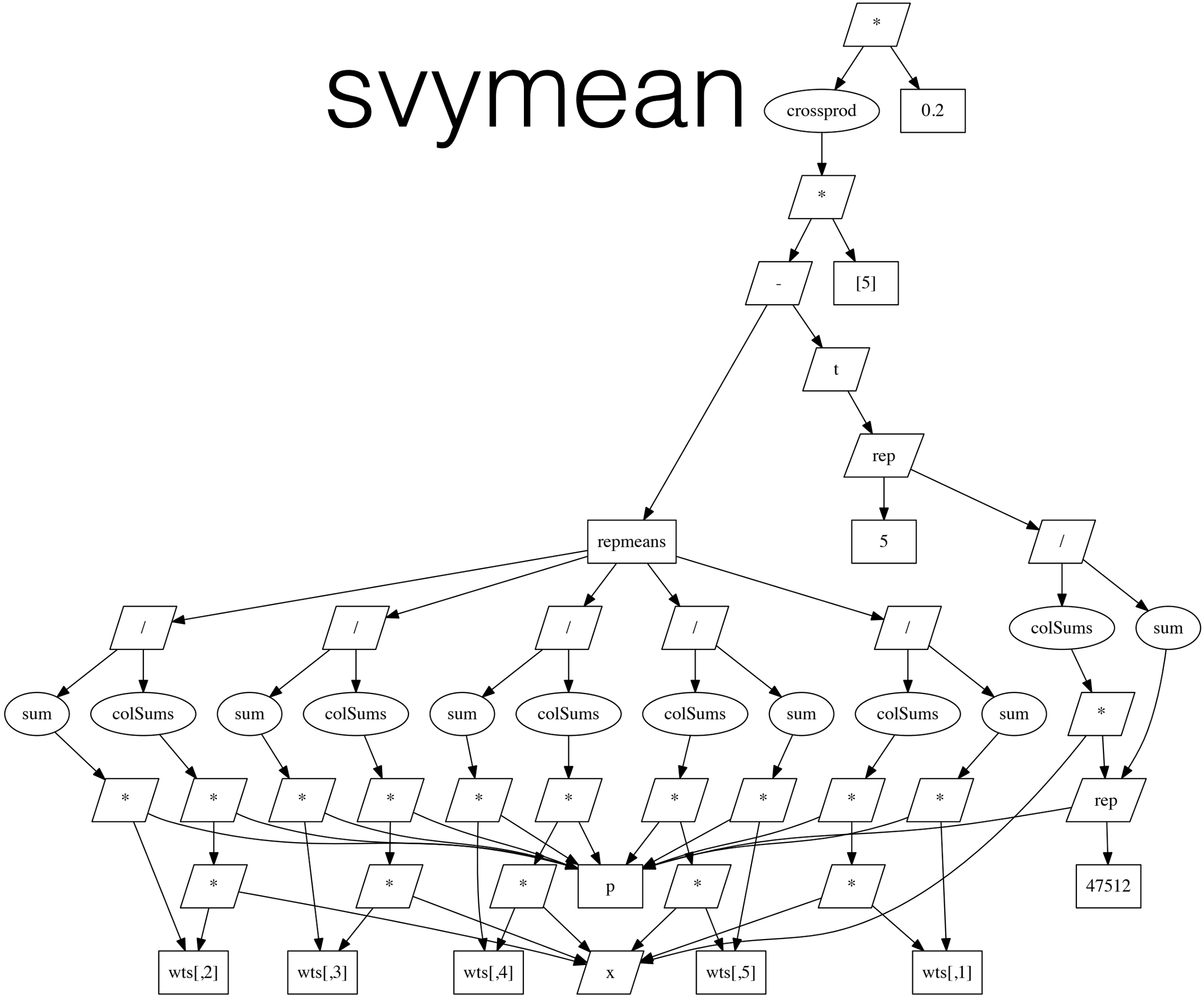

Renjin
GNU R
Renjin + R.
Execution Time (s)
0
20
40
60
10^6
10^7
10^8
Dataset Size (elements, log scale)

svymean

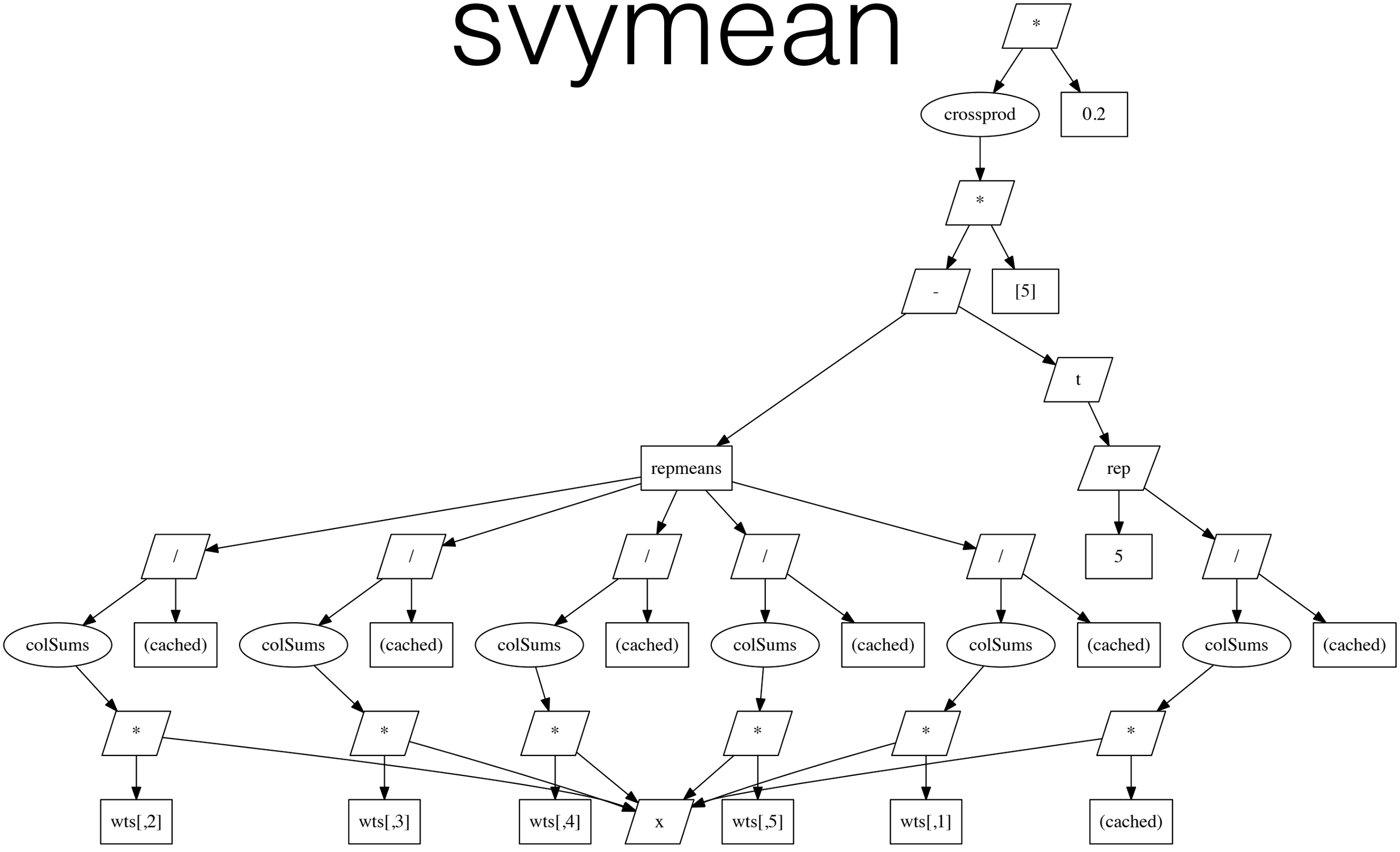
```
agep <- svymean(~agep, svydsgn, se=TRUE)


for(i in 1:ncol(wts)) {
  repmeans[i,]<-t(colSums(wts[,i]*x*pw)/
      sum(pw*wts[,i]))
}
[…]
v<-crossprod(sweep(thetas,2,
meantheta,"-")*sqrt(rscales))*scale
```

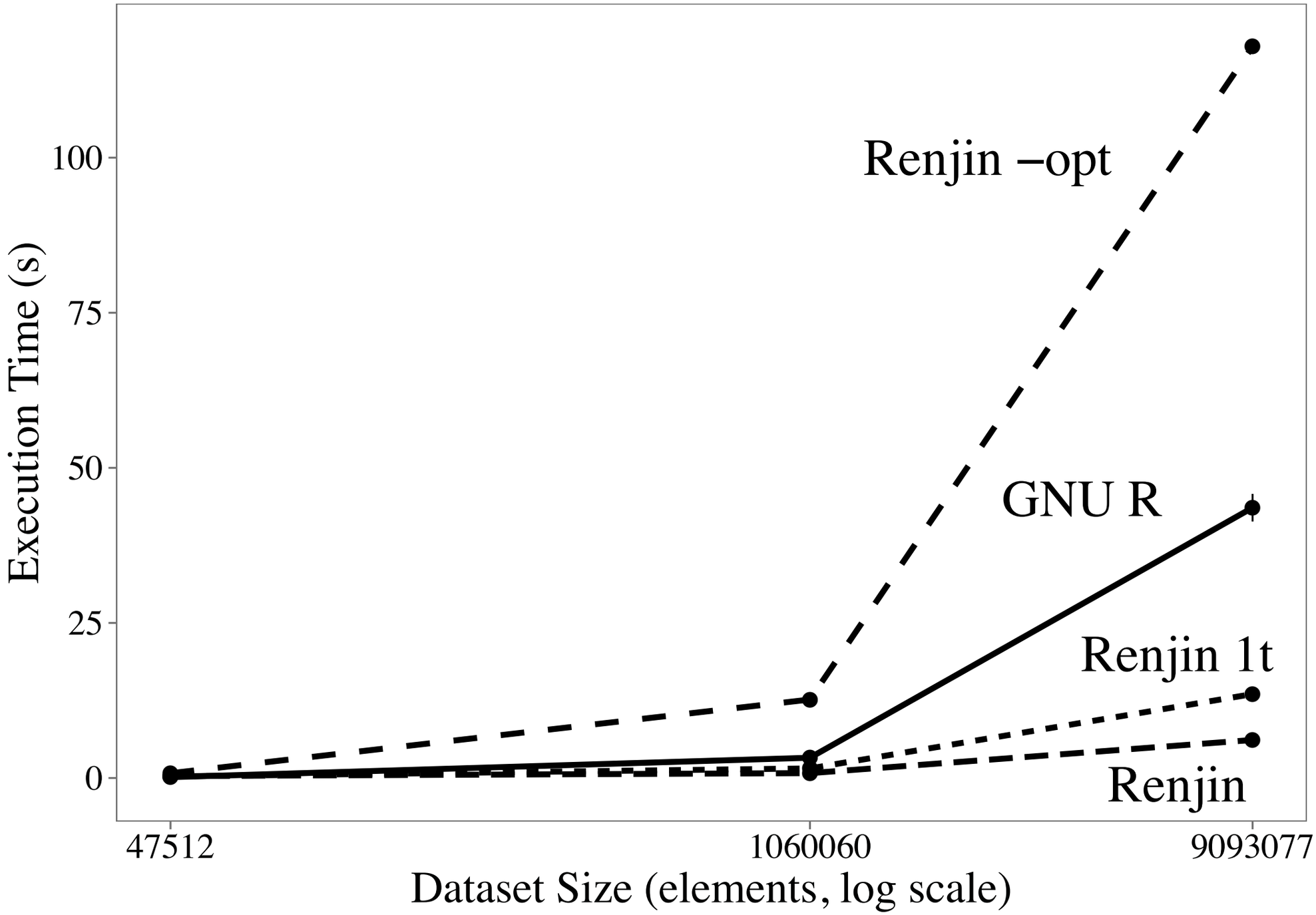
svymean


*
crossprod
0.2
*
-
[5]
t
rep
repmeans
5
/
/
/
/
/
/
colSums
sum
sum
colSums
sum
colSums
sum
colSums
colSums
sum
colSums
sum
*
*
*
*
*
*
*
*
*
*
*
*
rep
*
*
*
p
*
*
47512
wts[,2]
wts[,3]
wts[,4]
x
wts[,5]
wts[,1]

svymean


*
crossprod
0.2
*
-
[5]
t
repmeans
rep
/
/
/
/
/
5
/
colSums
(cached)
colSums
(cached)
colSums
(cached)
colSums
(cached)
colSums
(cached)
colSums
(cached)
*
*
*
*
*
*
wts[,2]
wts[,3]
wts[,4]
x
wts[,5]
wts[,1]
(cached)

svymean


Renjin −opt
GNU R
Renjin 1t
Renjin
100
75
50
25
0
Execution Time (s)
47512
1060060
9093077
Dataset Size (elements, log scale)

CWI
Centrum Wiskunde & Informatica

renjin

Thank You

Questions?

http://www.renjin.org

@mj_kallen
@hfmuehleisen