



flowR – A Dataflow Analysis Framework and Program Slicer for R

Florian Sihler, Oliver Gerstl, Julian Schubert, Matthias Tichy | useR! 2026 • Lightning Talk



Software Engineering
Programming Languages



universität
uulm



We do **static analysis** for R.

Analyzing the code without executing it.



Non Executable — research artifacts might be broken



Security — no potential side-effects



IDE & Development — linting, auto-completion & more

I use **LLMs**?

Program Slicing

focus on a single visualization



```
1 sum ← 0
2 prod ← 1
3 n ← 10
4 for (i in 1:(n-1)) {
5   sum ← sum + i
6   prod ← prod * i
7 }
8 cat("Sum:", sum, "\n")
9 cat("Product:", prod, "\n")
```

slice(8@sum) →

```
1 sum ← 0
3 n ← 10
4 for (i in 1:(n-1)) {
5   sum ← sum + i
7 }
8 cat("Sum:", sum, "\n")
```

On real scripts, slices are 80 % to 90 % smaller.^[1]

[1] Sihler et al., "Statically Analyzing the Dataflow of R Programs" (2025, OOPSLA)

Dependency View



Libraries 8 items

lme4 by "library" in (L. 15)

ggplot2 by "library" in (L. 16)

emmeans by "library" in (L. 17)

Go to Dependency

Show Influence of Dependency (Forward Slice)

Show Relevant Code for Dependency (Backward Slice)

scales by "::" in (L. 713)

Imported Data 5 items

Sourced Scripts 0 items

Outputs Disabled

Visualizations 3 items

function "plot" 9 items

function "plot" in (L. 146)

function "abline" in (L. 147, linked to 1 id)

Data-Frame Shapes

Data-Frame Shape:

Data-Frame Shapes
hover tooltips in the editor
Cols: [1, 1]

patients.csv

```
id,age,weight,height  
1,34,70,1.75  
2,51,82,1.80  
3,29,60,1.65  
4,42,95,1.90
```

```
1 df <- read.csv("patients.csv") |>  
2   mutate(bmi = weight / height^2) |>  
3   select(-c(weight, height))
```

data frame shape:
Rows: 4
Cols: 3
Columns: id, age, bmi

Linting

unused-definitions

cube is never called

```
1 applyTwice ← function(f, x) f(f(x))
2 square     ← function(x) x * x
3 cube       ← function(x) x * x * x
4
5 p ← readline()
6 pdf(p)
7 print(applyTwice(square, 3))
```

tainted-input

untrusted p reaches pdf()

What flowR Offers



Dataflow Graphs
& control-flow, in ~570 ms



Program Slicing
focus on what matters



Lint
reproducibility & security



Dependency View
libraries, sources & I/O

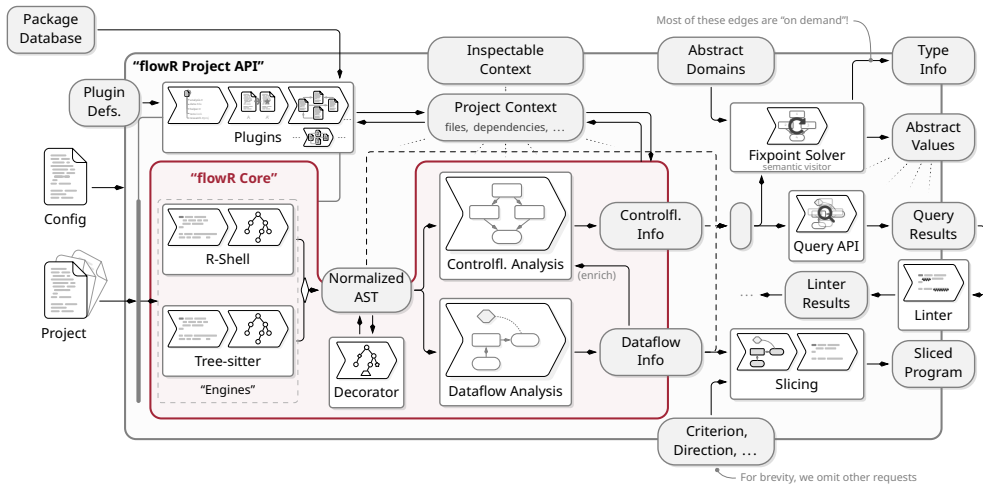


IDE Integrations
VS Code & Positron, RStudio

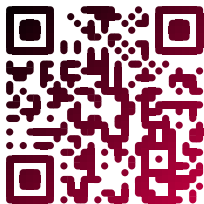


Data-Frame Shapes
hover tooltips in the editor
Cols: [1, 1]

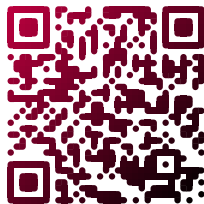
flowR's Architecture



Try flowR and tell us what you want!



github.com/flowr-analysis/flowr



VS Code & **Positron** extension

Florian Sihler • florian.sihler@uni-ulm.de

