Towards the Extraction of Implicit Variability From R Research Scripts

FOSD Meeting 2026 | **Ruben Dunkel**, Florian Sihler, Matthias Tichy, Thomas Thüm | March 27, 2026

Software Engineering
Programming Languages

universität ulm

ISF

Technische Universität Braunschweig

Real-World R Code

[1] Drudze et al., *Apple phenology data set and R script, related to publication "Full flowering phenology of apple tree (Malus domestica) in Pūre orchard, Latvia from 1959 to 2019"* (2021, Zenodo)

Real-World R Code

- Very long and complex
- Partial data availability (reproducibility problem)
- Take long to run
- Where are the interesting parts?

[1] Drudze et al., *Apple phenology data set and R script, related to publication "Full flowering phenology of apple tree (Malus domestica) in Pūre orchard, Latvia from 1959 to 2019"* (2021, Zenodo)

Real-World R Code

- Very long and complex
- Partial data availability (reproducibility problem)
- Take long to run
- Where are the interesting parts?

Slicing

[1] Drudze et al., *Apple phenology data set and R script, related to publication "Full flowering phenology of apple tree (Malus domestica) in Pūre orchard, Latvia from 1959 to 2019"* (2021, Zenodo)

Real-World R Code

- Very long and complex
- Partial data availability (reproducibility problem)
- Take long to run

} Slicing

- Where are the interesting parts?

```
x ← 1
y ← 2
print(x)
```

[1] Drudze et al., *Apple phenology data set and R script, related to publication "Full flowering phenology of apple tree (Malus domestica) in Pūre orchard, Latvia from 1959 to 2019"* (2021, Zenodo)

Real-World R Code

- Very long and complex
- Partial data availability (reproducibility problem)
- Take long to run

} Slicing

- Where are the interesting parts?

```
x ← 1
y ← 2
print(x)
```

[1] Drudze et al., *Apple phenology data set and R script, related to publication "Full flowering phenology of apple tree (Malus domestica) in Pūre orchard, Latvia from 1959 to 2019"* (2021, Zenodo)

Real-World R Code

- Very long and complex
- Partial data availability (reproducibility problem)
- Take long to run

} Slicing

- Where are the interesting parts?

```
x ← 1
y ← 2
print(x)
```

```
x ← 1
y ← 2
print(x)
```

[1] Drudze et al., *Apple phenology data set and R script, related to publication "Full flowering phenology of apple tree (Malus domestica) in Pūre orchard, Latvia from 1959 to 2019"* (2021, Zenodo)

Real-World R Code

1 Import

2 Model

3 Model

4 Visualize

5 Visualize

6 Visualize

- Very long and complex
- Partial data availability (reproducibility problem)
- Take long to run

Slicing

- Where are the interesting parts?

```
x ← 1
y ← 2
print(x)
```

```
x ← 1
y ← 2
print(x)
```

[1] Drudze et al., *Apple phenology data set and R script, related to publication "Full flowering phenology of apple tree (Malus domestica) in Pūre orchard, Latvia from 1959 to 2019"* (2021, Zenodo)

Real-World R Code

1 } Import

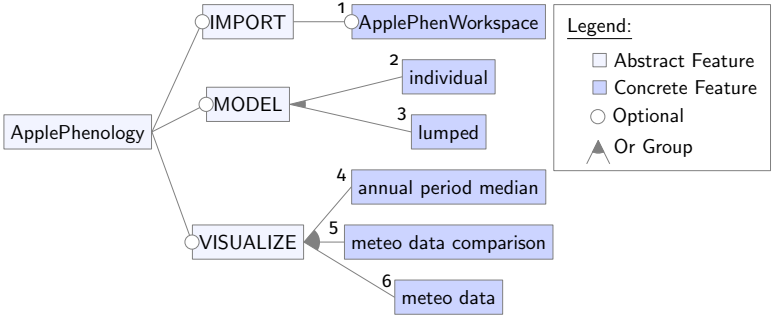
2 } Model

3 } Model

4 } Visualize

5 } Visualize

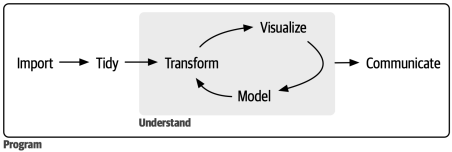
6 } Visualize



[1] Drudze et al., *Apple phenology data set and R script, related to publication "Full flowering phenology of apple tree (Malus domestica) in Pūre orchard, Latvia from 1959 to 2019"* (2021, Zenodo)

Research Questions

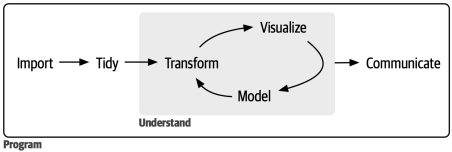
1. What are common PoIs in R scripts that are used for data analysis in research during the phases of the data science process described by Wickham [2]?



[2] Wickham et al., *R for Data Science* (2023, " O'Reilly Media, Inc.")

Research Questions

1. What are common PoIs in R scripts that are used for data analysis in research during the phases of the data science process described by Wickham [2]?

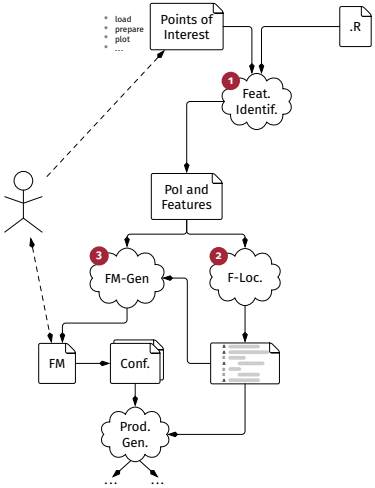


[2] Wickham et al., *R for Data Science* (2023, " O'Reilly Media, Inc.")

Research Questions

1. What are common PoIs in R scripts that are used for data analysis in research during the phases of the data science process described by Wickham [2]?

2. How can a prototype for converting R scripts into a SPL be realized with special focus on the identification of PoIs, the slicing and adding of variability annotation and the creation of the feature model?

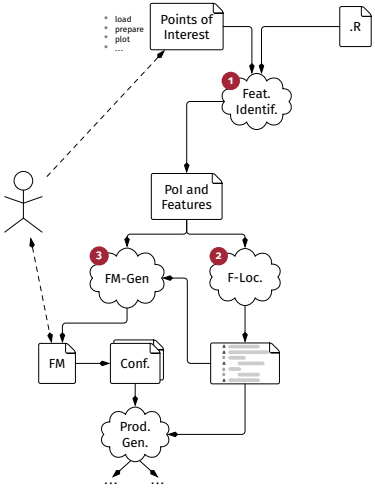


[2] Wickham et al., *R for Data Science* (2023, " O'Reilly Media, Inc.")

Research Questions

1. What are common PoIs in R scripts that are used for data analysis in research during the phases of the data science process described by Wickham [2]?

2. How can a prototype for converting R scripts into a SPL be realized with special focus on the identification of PoIs, the slicing and adding of variability annotation and the creation of the feature model?



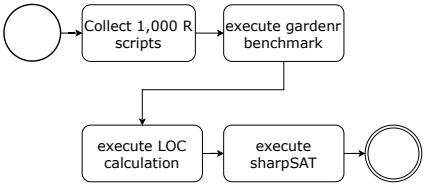
[2] Wickham et al., *R for Data Science* (2023, " O'Reilly Media, Inc.")

Research Questions

1. What are common PoIs in R scripts that are used for data analysis in research during the phases of the data science process described by Wickham [2]?

2. How can a prototype for converting R scripts into a SPL be realized with special focus on the identification of PoIs, the slicing and adding of variability annotation and the creation of the feature model?

3. How do the created variants compare to the original script regarding LoC, execution time of the tool, and correctness of the outputs, and how many variants are created?



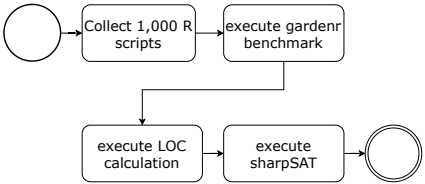
[2] Wickham et al., *R for Data Science* (2023, " O'Reilly Media, Inc.")

Research Questions

1. What are common PoIs in R scripts that are used for data analysis in research during the phases of the data science process described by Wickham [2]?

2. How can a prototype for converting R scripts into a SPL be realized with special focus on the identification of PoIs, the slicing and adding of variability annotation and the creation of the feature model?

3. How do the created variants compare to the original script regarding LoC, execution time of the tool, and correctness of the outputs, and how many variants are created?

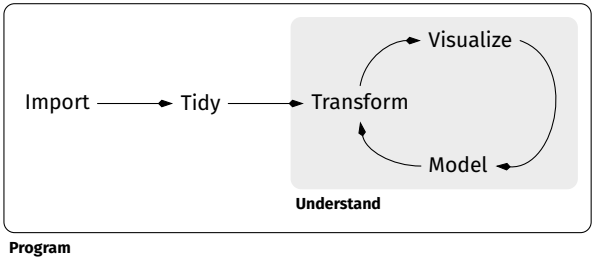


[2] Wickham et al., *R for Data Science* (2023, " O'Reilly Media, Inc.")

Labeling

RQ1

Using functions as PoI [3]



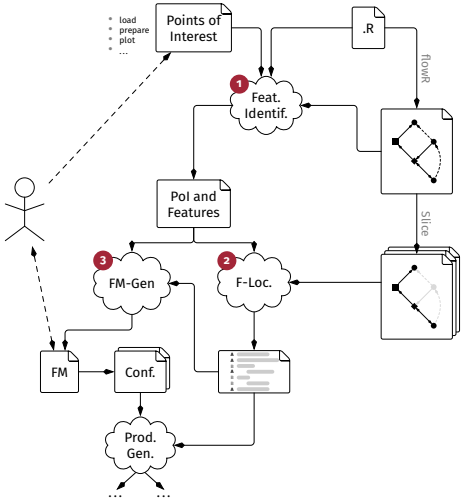
- PoI = function call
- labeled functions of most used R packages
  - 17 548 functions found
  - labeled by 5 researchers
  - 12 004 assigned to a step in process

[3] Walkinshaw et al., “Feature Location and Extraction using Landmarks and Barriers” (2007, IEEE)

Architecture

RQ2

Using functions as PoI [3]

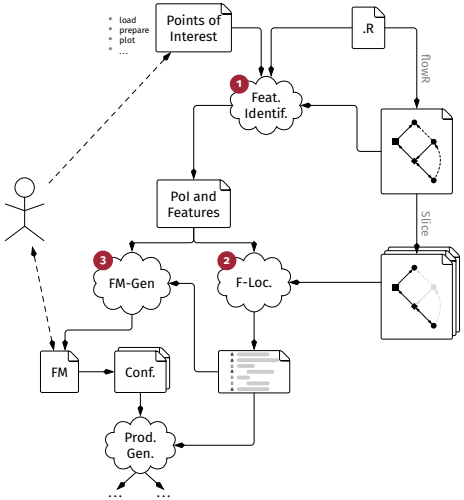


[3] Walkinshaw et al., "Feature Location and Extraction using Landmarks and Barriers" (2007, IEEE)

Feature Identification

- PoI -> feature
  - inspired by landmarks [3]
- find PoIs in script using a tool

Architecture



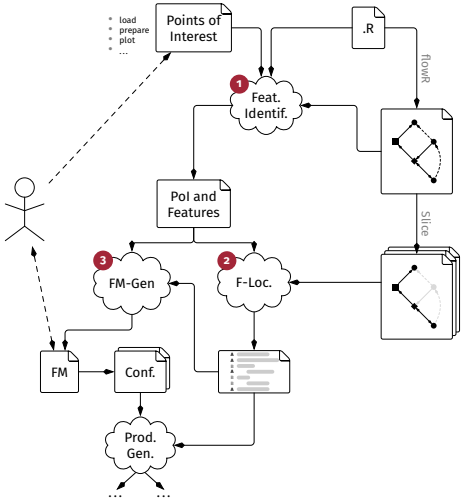
Feature Identification

- PoI -> feature
  - inspired by landmarks [3]
- find PoIs in script using a tool
- language independent
  - PoIs replaceable
  - slicer independent

Architecture

RQ2

Applying inter-feature dependencies [4]

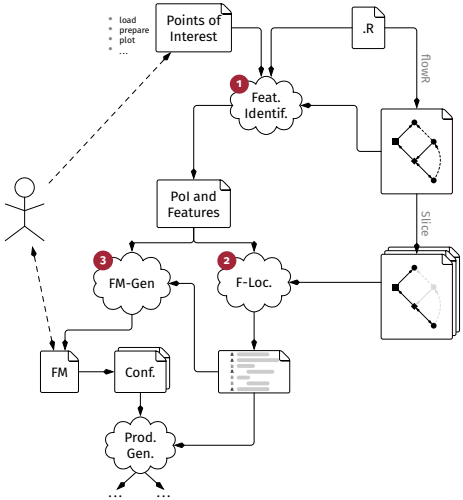


Feature Location

- slice for every single detected PoI
- compare slices to detect dependencies
  - create dependency graph
- dependency = implication constraint

[4] Li et al., “Using Feature-Oriented Analysis to Recover Legacy Software Design for Software Evolution.” (2005, Citeseer)

Architecture

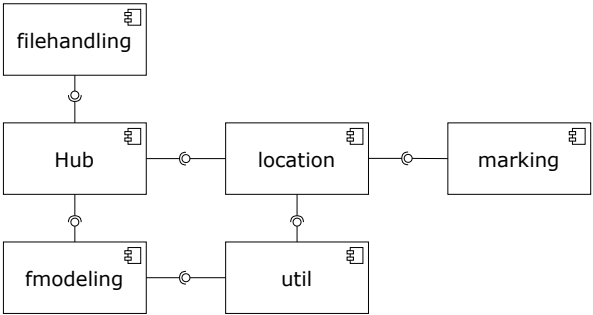


Feature Model Generation

- create feature from each PoI
- traverse dependency graph to create cross-tree constraints

Implementation

RQ2



Relevant Aspects

- Input:
  - PoI list from RQ1
  - R research scripts
- Slicer: flowr
- Output:
  - JSON file containing features and constraints
  - UVL file
  - modified script with *preprocessor directives*

Example

```
sample ← read.csv("sample.csv", sep = ";")

plot(sample$var1 ~ sample$var2, pch = 20)

abline(lm(sample$var1 ~ sample$var2))
```

Detected PoIs:

- read.csv
- plot
- abline
- lm

Example

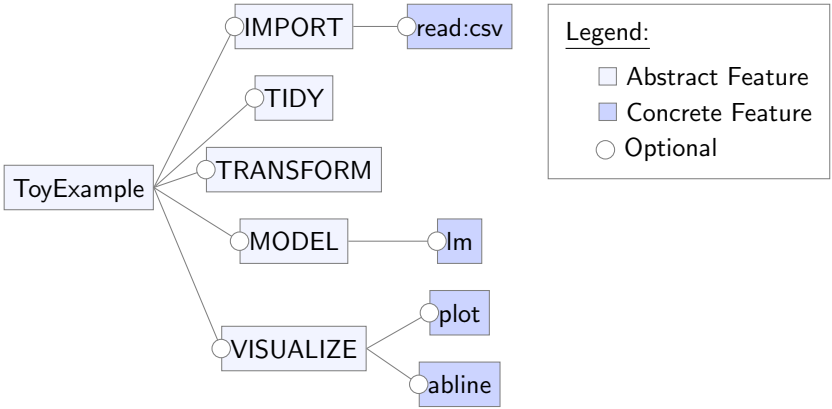
```
#if read.csv || plot
sample ← read.csv("sample.csv", sep = ";")
#if plot
plot(sample$var1 ~ sample$var2, pch = 20)
#if ‘abline‘ || ‘lm‘
abline(lm(sample$var1 ~ sample$var2))
```

Example

```
#if read.csv || plot
sample ← read.csv("sample.csv", sep = ";")
#if plot
plot(sample$var1 ~ sample$var2, pch = 20)
#if ‘abline‘ || ‘lm‘
abline(lm(sample$var1 ~ sample$var2))
```

```
namespace ToyExample
features
  ToyExample {abstract true}
    optional
      IMPORT {abstract true}
        optional
          "read:csv"
      MODEL {abstract true}
        optional
          "lm"
      VISUALIZE {abstract true}
        optional
          "plot"
          "abline"
constraints
  "lm" => "abline"
  "abline" => "plot"
  "plot" => "read:csv"
```

Example



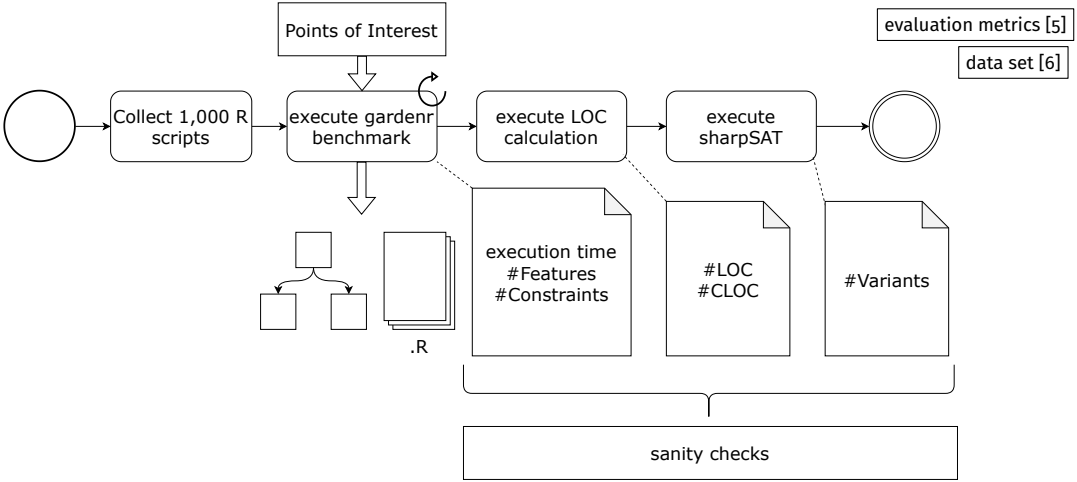
$\text{lm} \Rightarrow \text{abline}$

$\text{abline} \Rightarrow \text{plot}$

$\text{plot} \Rightarrow \text{"}read : csv\text{"}$

Evaluation

RQ3

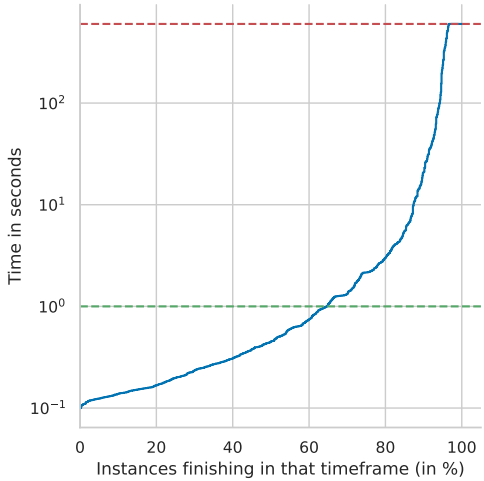


[5] Linsbauer et al., "Variability extraction and modeling for product variants" (2017, Springer)
[6] Sihler et al., "Statically Analyzing the Dataflow of R Programs" (2025, Association for Computing Machinery)

Evaluation

RQ3

responsiveness metric [7]



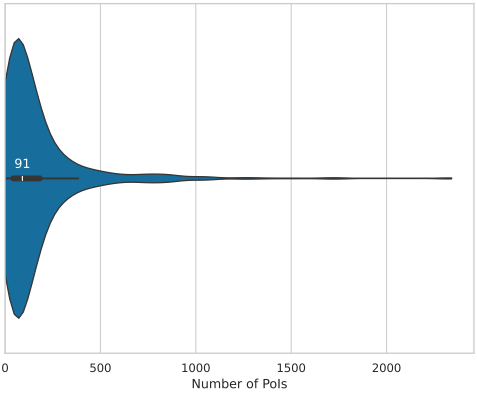
- logarithmic y scale
- 10 min, timeout
  - >95 %
- 1 s, keep train of thought
  - >60 %

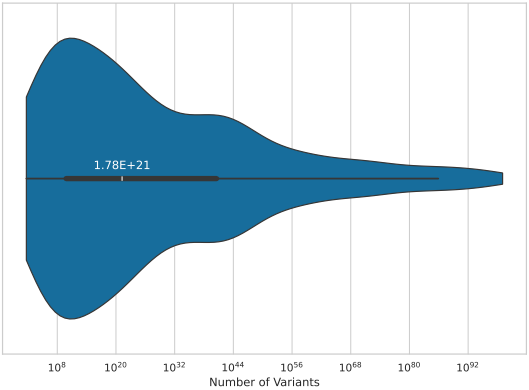
[7] Nielsen, “Chapter 5 - Usability Heuristics” (1993, Morgan Kaufmann)

Evaluation

RQ3

Evaluation Metrics [5]



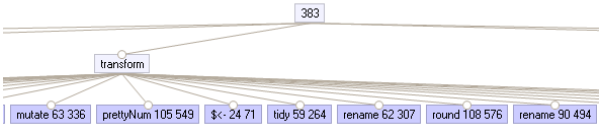


[5] Linsbauer et al., “Variability extraction and modeling for product variants” (2017, Springer)

Evaluation

RQ3

62 Features



"rename 58 252" ⇒ "factor 67 389"
"tidy 59 264" ⇒ "factor 67 389"
"nobs 107 557" ⇒ "c 121 653"
"$<- 67 354" ⇒ "xtable 96 520"
"train 47 200" ⇒ "c 121 653"
"round 108 576" ⇒ "c 121 653"
"library 4 15" ⇒ "library 1 3"

Future Work

1. Evaluation

Future Work

1. Evaluation

- correctness evaluation
- reduction of…
  - code to comprehend
  - execution time
- partial execution

Future Work

1. Evaluation
2. Setting it free

Future Work

1. Evaluation
2. Setting it free

- combat overtaxing the user
- practitioner survey
- hub for annotated research scripts

Future Work

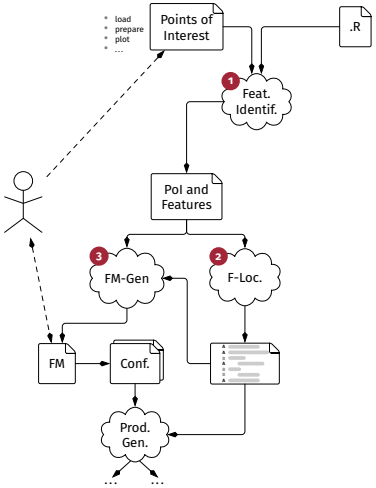
1. Evaluation
2. Setting it free
3. Expanding the scope

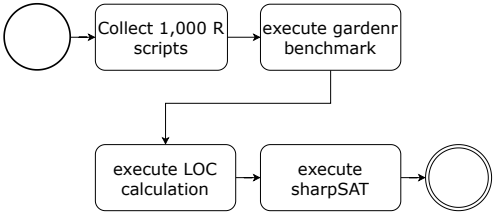
Future Work

1. Evaluation
2. Setting it free
3. Expanding the scope

- other programming languages
- …?

Summary


load
prepare
plot
...
Points of Interest
.R
1
Feat. Identif.
PoI and Features
3
FM-Gen
2
F-Loc.
FM
Conf.
Prod. Gen.


Collect 1,000 R scripts
execute gardenr benchmark
execute LOC calculation
execute sharpSAT

Towards the Extraction of Implicit Variability From R Research Scripts

FOSD Meeting 2026 | **Ruben Dunkel**, Florian Sihler, Matthias Tichy, Thomas Thüm | March 27, 2026

Software Engineering
Programming Languages

universität ulm

ISF

Technische Universität Braunschweig

Towards the Extraction of Implicit Variability From R Research Scripts

1. **Motivation**
2. **Research Questions**
3. **Labeling**
4. **Architecture**
   - Concept
   - Implementation
   - Example
5. **Evaluation**
6. **Future Work**
7. **Summary**

References

[1] Inese Drudze et al. *Apple phenology data set and R script, related to publication "Full flowering phenology of apple tree (Malus domestica) in Pūre orchard, Latvia from 1959 to 2019".* June 2021

[2] Hadley Wickham, Mine Çetinkaya-Rundel, and Garrett Grolemund. *R for Data Science.* 2023

[3] Neil Walkinshaw, Marc Roper, and Murray Wood. "Feature Location and Extraction using Landmarks and Barriers". Oct. 2007

[4] Shaoyun Li et al. "Using Feature-Oriented Analysis to Recover Legacy Software Design for Software Evolution.". 2005

[5] Lukas Linsbauer, Roberto Erick Lopez-Herrejon, and Alexander Egyed. "Variability extraction and modeling for product variants". Oct. 2017

[6] Florian Sihler and Matthias Tichy. "Statically Analyzing the Dataflow of R Programs". Oct. 2025

[7] Jakob Nielsen. "Chapter 5 - Usability Heuristics". 1993