From Indentation Shapes to Code Structures

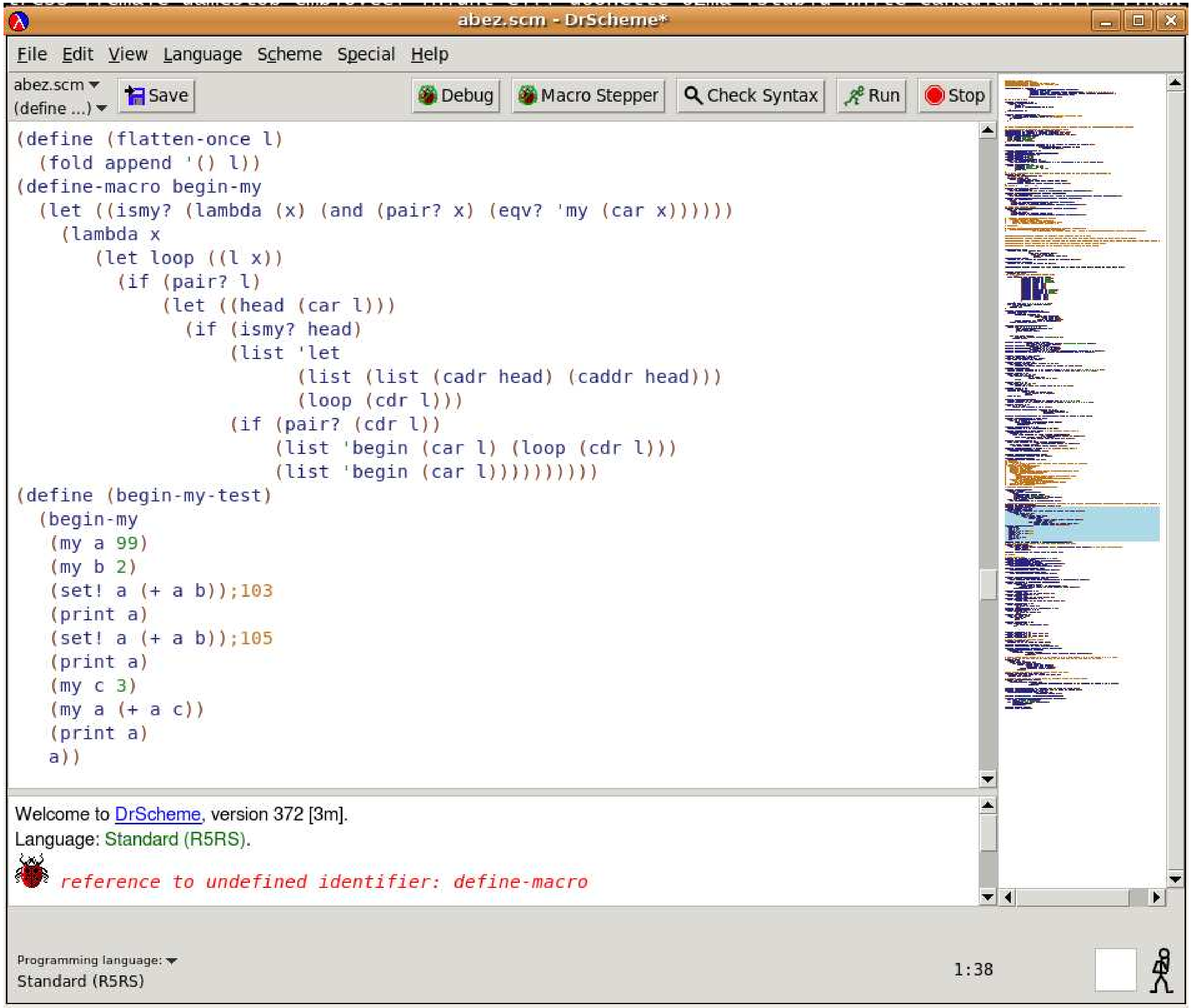
Abram Hindle, Michael W. Godfrey, Richard C. Holt

Software Architecture Group
David R. Cheriton School of Computer Science
University of Waterloo
Canada
http://swag.uwaterloo.ca/

{ahindle,migod,holt}@cs.uwaterloo.ca

Introduction

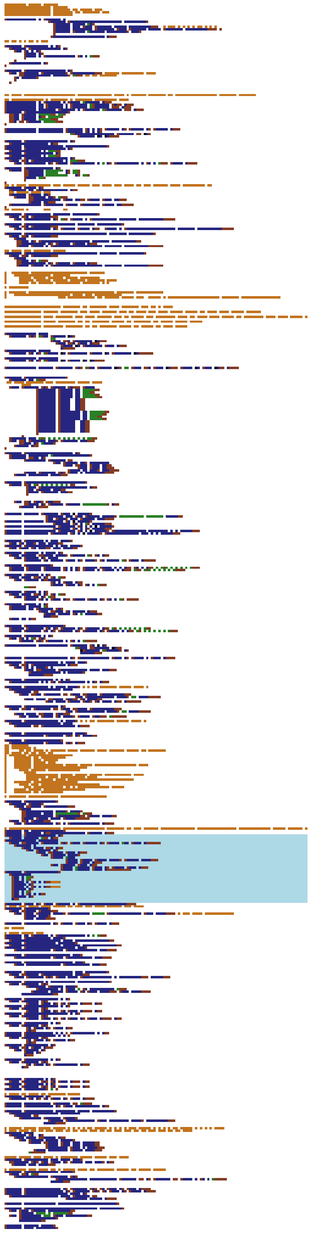
- Previously we showed at ICPC 2008 that variance and sum of indentation rank correlates with complexity.
- Quick and cheap methods for determining if revisions are worthwhile to investigate
- But we noticed something

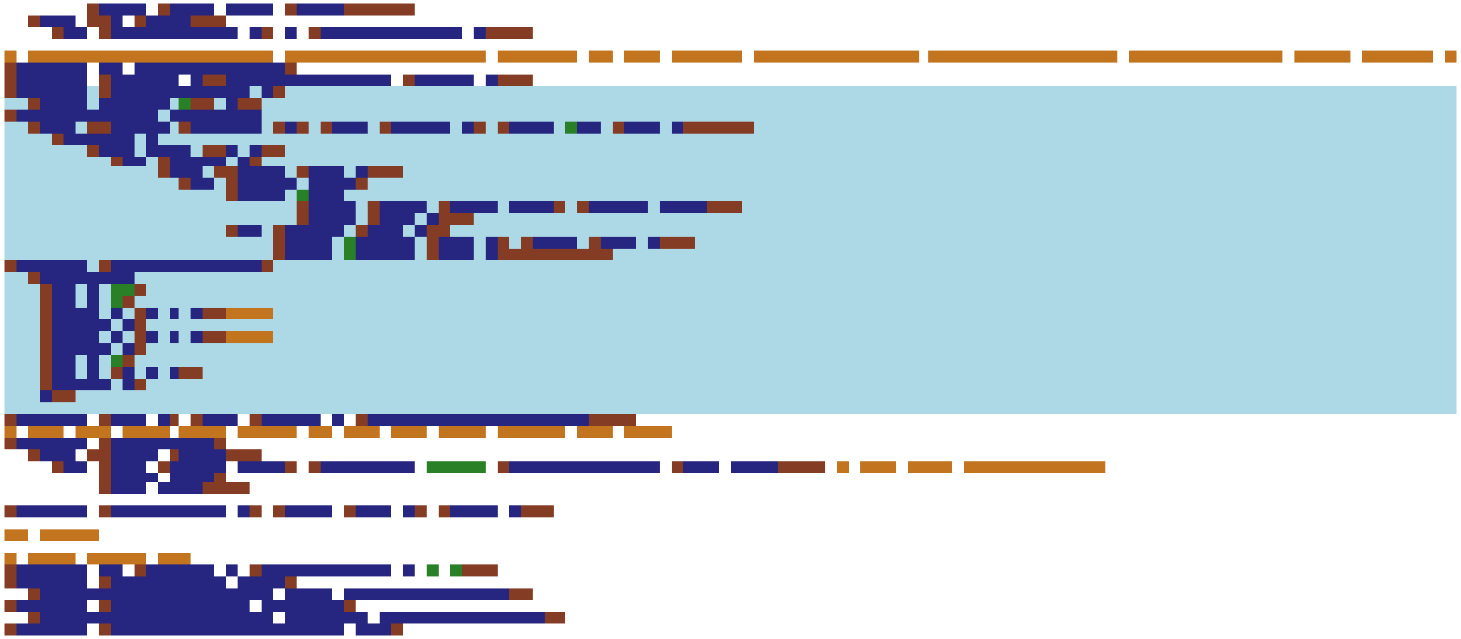


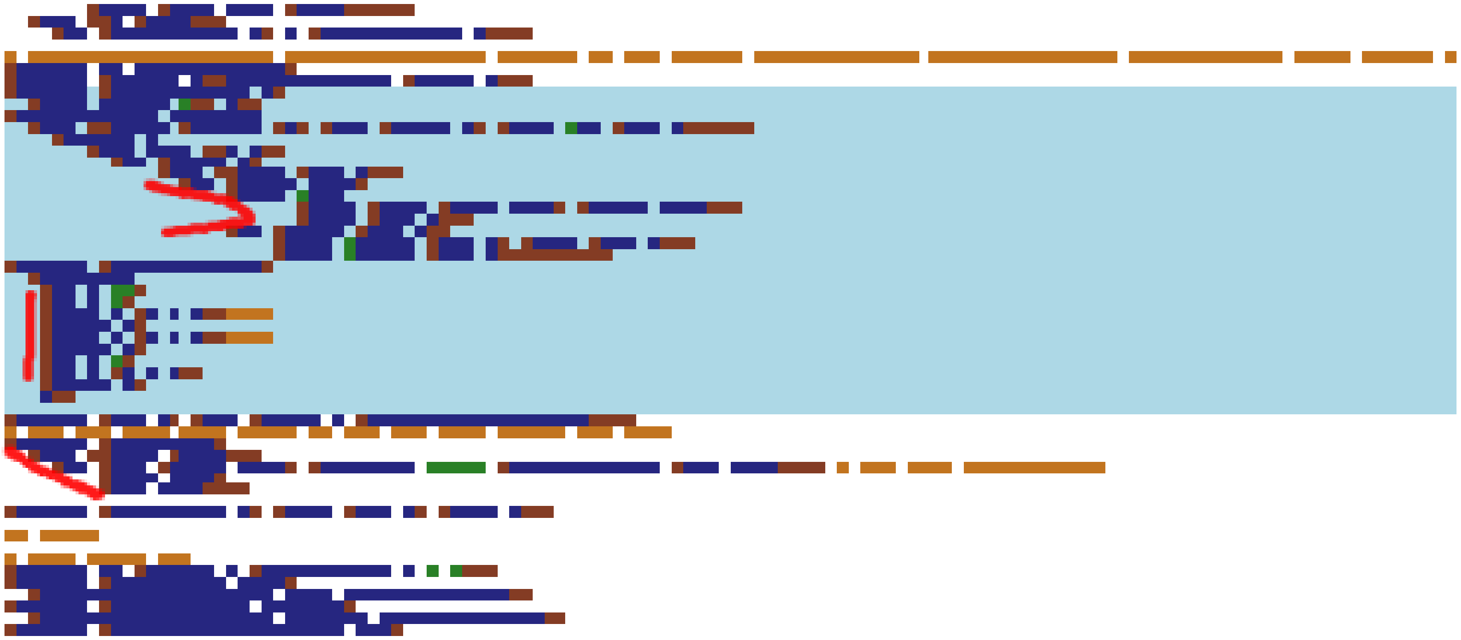
```
(define (flatten-once l)
  (fold append '() l))
(define-macro begin-my
  (let ((ismy? (lambda (x) (and (pair? x) (eqv? 'my (car x))))))
    (lambda x
       (let loop ((l x))
         (if (pair? l)
             (let ((head (car l)))
               (if (ismy? head)
                   (list 'let
                         (list (list (cadr head) (caddr head)))
                         (loop (cdr l)))
                   (if (pair? (cdr l))
                       (list 'begin (car l) (loop (cdr l)))
                       (list 'begin (car l))))))))))
(define (begin-my-test)
  (begin-my
   (my a 99)
   (my b 2)
   (set! a (+ a b));103
   (print a)
   (set! a (+ a b));105
   (print a)
   (my c 3)
   (my a (+ a c))
   (print a)
   a))
```

Welcome to DrScheme, version 372 [3m].
Language: Standard (R5RS).
reference to undefined identifier: define-macro

Standard (R5RS)





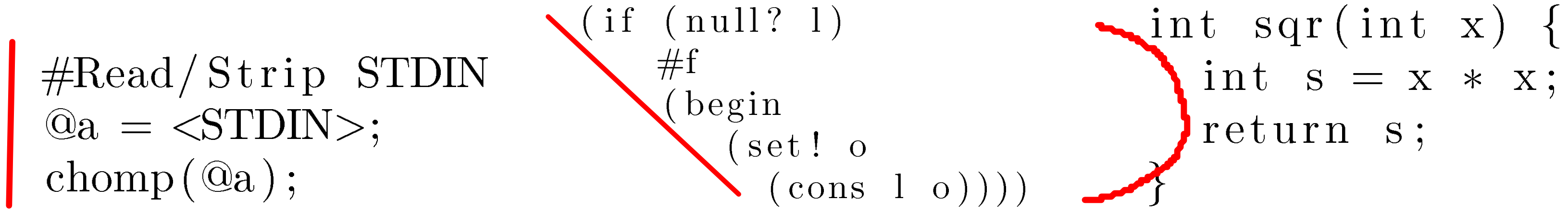


Introduction

- This code had a profile, it had shape
- Large code has a complex shape
  - What about changes to code, aren't they small?
- Does a revision's indentation shape tell you something about the underlying code?

Indentation Shapes

- Indentation shapes are meant to be detectable by man and machine
- Shape of changes
- Formalized 3 shapes we expected to see (and did occur)
  - flat
  - bubble
  - slash



```
#Read/Strip STDIN
@a = <STDIN>;
chomp(@a);
```

(a) Flat Indentation

```
(if (null? l)
     #f
     (begin
          (set! o
               (cons l o))))
```

(b) Slash Indentation

```
int sqr(int x) {
   int s = x * x;
   return s;
}
```

(c) Bubble Indentation

Figure 1: Examples of Flat, Slash and Bubble indentation shapes

Methodology (1/2)

- Mirrored CVS Repositories of most active and most downloaded SF projects
- Sampled a control set of revisions and source files
- Analyzed Indentation for both sets
- Selected Revisions matched the indentation shapes
- Annotated selected revisions
- Analyzed results

Methodology (2/2)

- Started with 51GB of CVS Repos
  - (control set sampled from here)
- Shape set was a subsample of 479 source files
- 84 C, 65 C++, 138 .h, 118 Java, 51 PHP, 10 Perl, 13 Python
- Matched 5660 revisions with indentation shapes
- Control set was 1001 revisions

Get the Diff

```
> void square( int * arr, int n ) {
> □□□□int i = 0;
> □□□□for ( i = 0 ; i < n ; i++ ) {
> □□□□□□□□arr[ i ] *= arr[ i ];
> □□□□}
> }
```

Measure the Indentation

Raw Indentation	0	4	4	8	4	0
Logical Indentation	0	1	1	2	1	0

Produce Summary Statistics

Metric	Raw	Logical
LOC	6.000	6.000
AVG	3.330	0.833
MED	4.000	1.000
STD	2.750	0.687
VAR	9.070	0.567
SUM	20.000	5.000
MCC	2.000	2.000
HVOL	152.000	152.000
HDIFF	15.000	15.000
HEFFORT	2127.000	2127.000

Annotations

- Comments
- Type Declarations
- Assignments
- Conditionals
- Function Calls
- Data
- Function Definition

- Macro
- Loop
- Conditional Macro
- Anomaly
- Exception
- Return
- Concurrency
- Expression

Questions:

- What kind of indentation correlates with function definition?
- What kinds of code correlate with zero variance indentation?
- What kinds of code correlate with non-zero variance indentation?

Flat Revisions

- No change in indentation
- Most common shape of our 3 shapes
- 3319 Flat Revisions
- Most likely: comments, assignments, type definitions
- Least Likely: conditionals, loops, etc.
- More formally, a revision of N lines ($N \geq 2$) is said to be flat if $\forall i : 1..N \bullet I_i = k$ for some constant $k \geq 0$ where I_i is the indentation of line i.

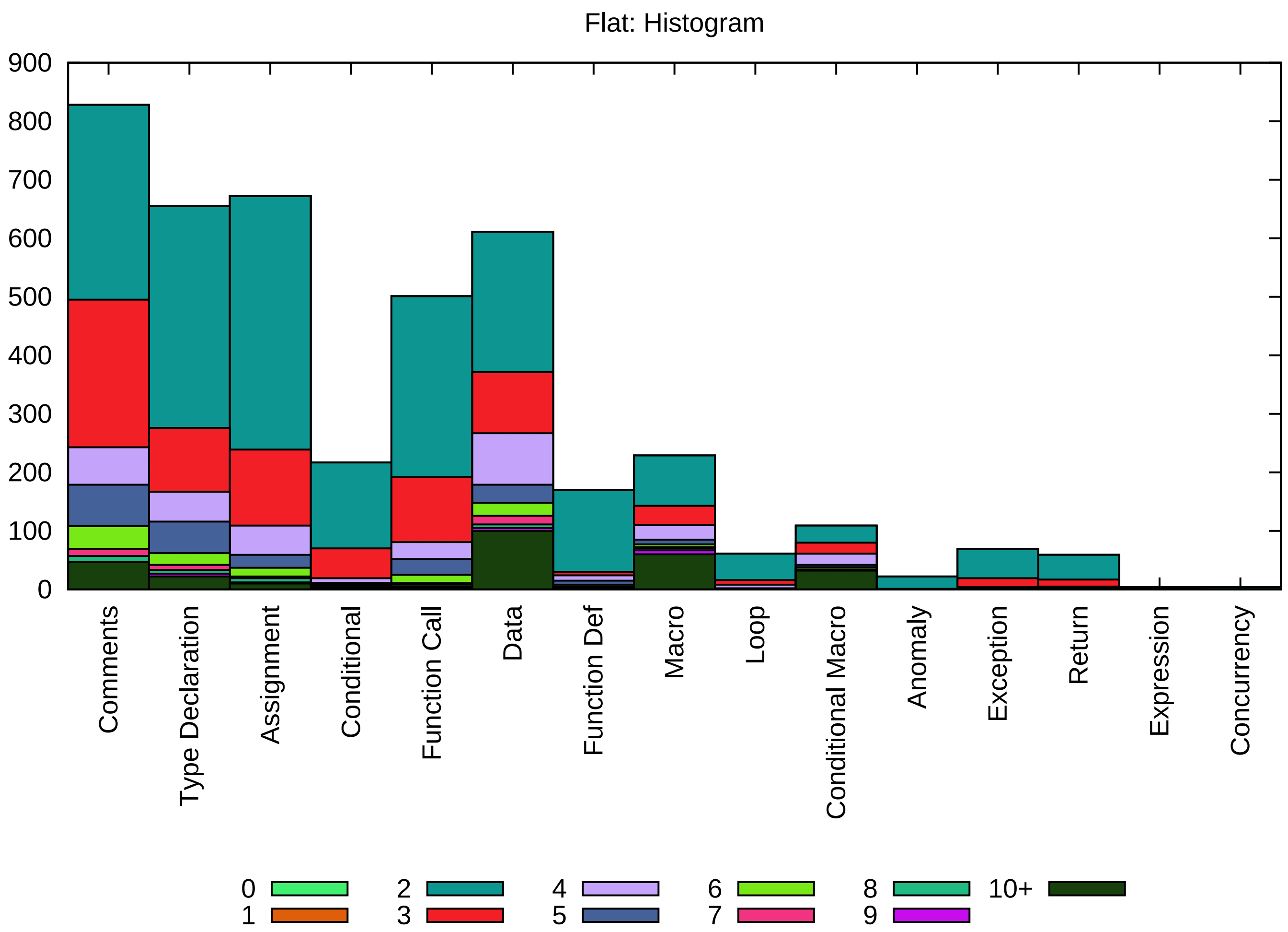


Figure 2: Distribution of revision length of flat shape revisions per annotation

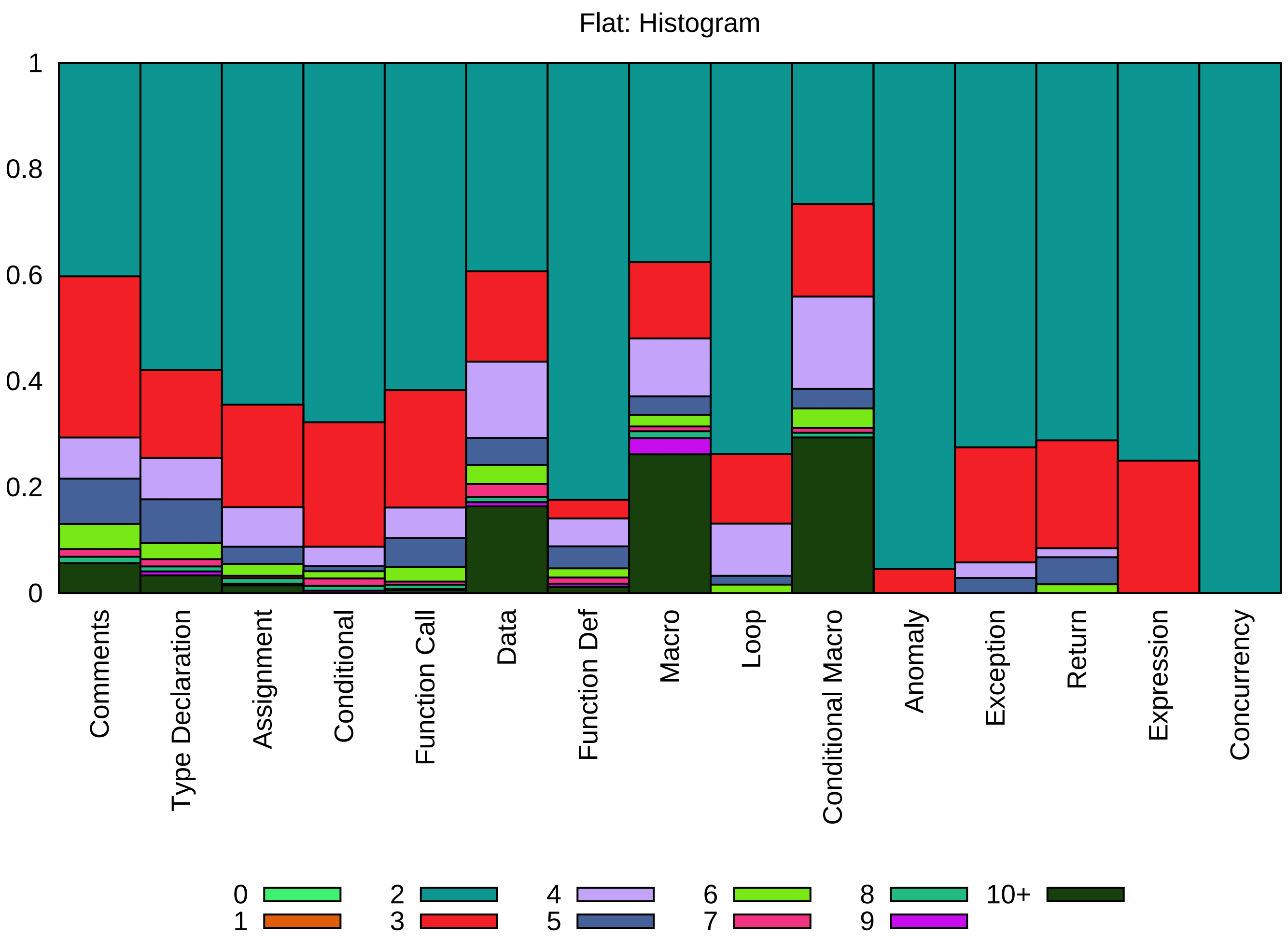


Figure 3: Proportional Distribution of revision length of flat shape revisions per annotation

Slash Revisions

- Increasing indentation depth
- 1552 Slash revisions
- Most likely: conditionals, type declarations
- Least likely: data, macros, conditional macros
- Slash revisions can be described as revisions of N lines (where $N >= 2$), where I_i represents the indentation of line i, $\forall i : 2..N \bullet I_i \geq I_{i-1}$, and $I_1 < I_N$.

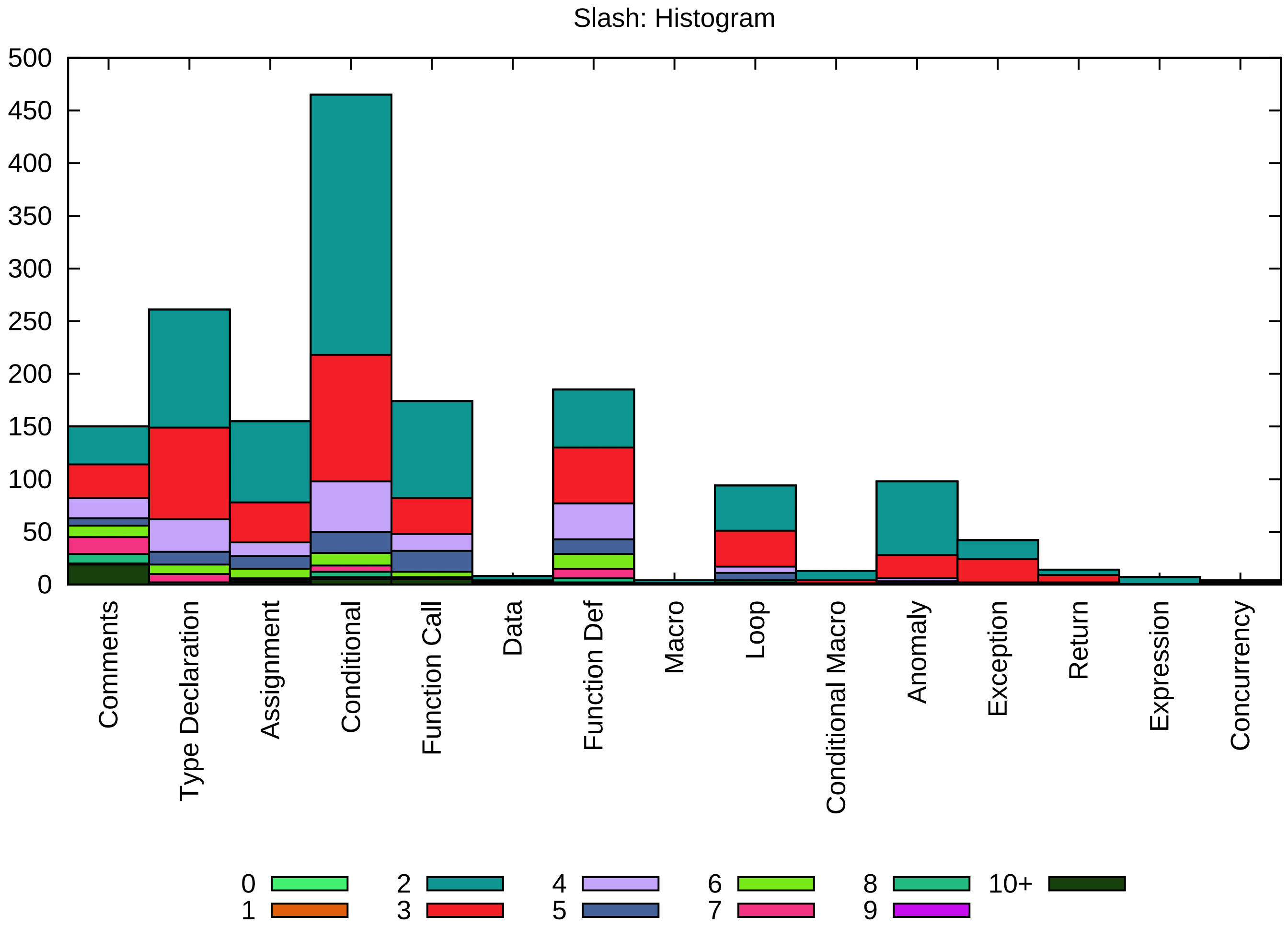


Figure 4: Distribution of revision length of slash shape revisions per annotation

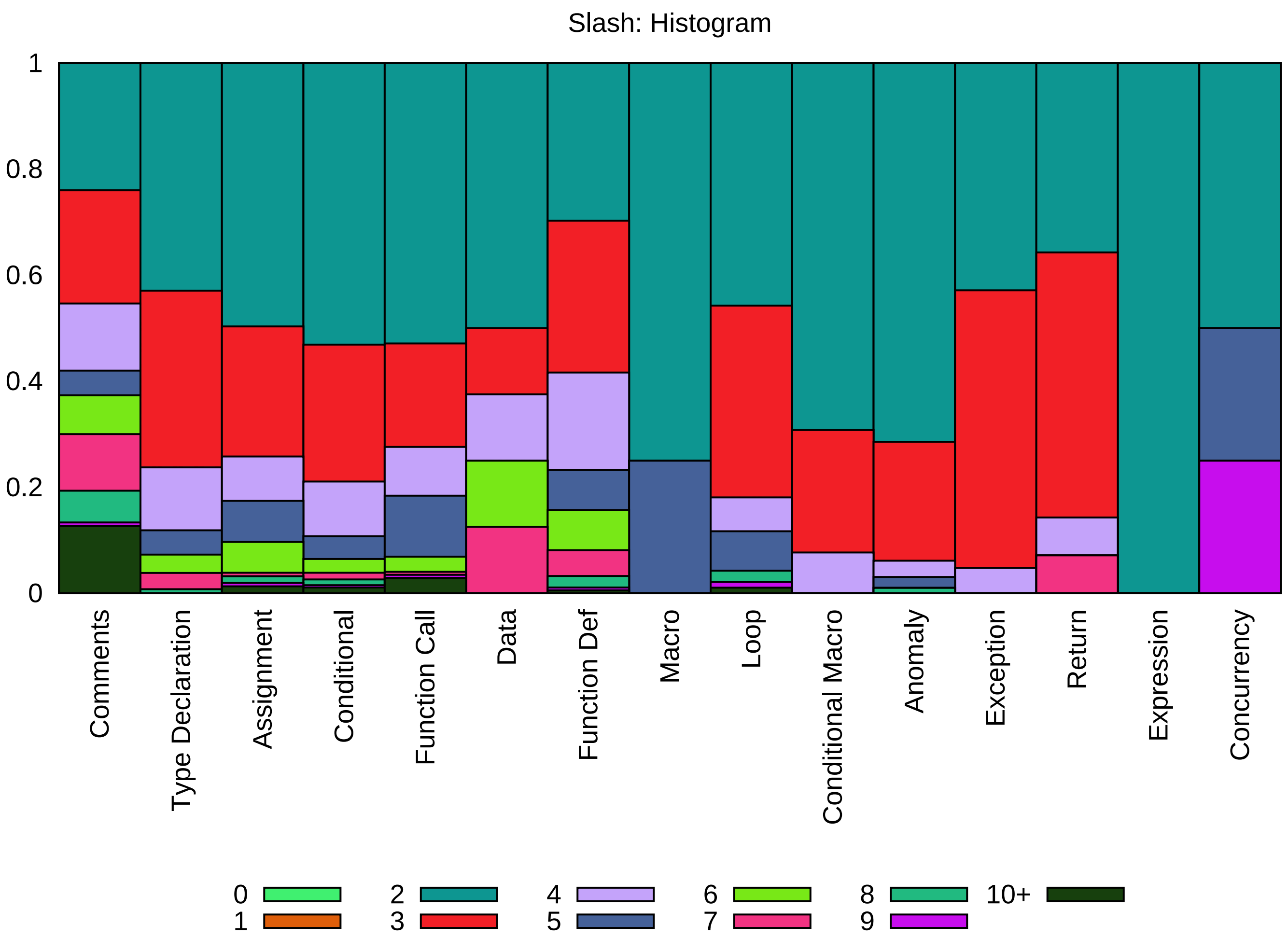


Figure 5: Proportional distribution of revision length of slash shape revisions per annotation

Bubble (1/2)

- Bubble revisions represent code which has a bubble-like shape
- 805 revisions
- Most likely: conditional, function implementations, assignments
- Least likely: data, macros

Bubble (2/2)

- Formally, a revision of N lines, where $N \geq 3$ and where I_i is the indentation of line i, is said to be a bubble revision if there exists a peak k where $\forall i : 2..k \; I_{i-1} \leq I_i \leq I_k$ and $\forall i : (k+1)..N \bullet I_k > I_{i-1} \geq I_i$ and $I_1 \leq I_n$. Thus indentation depth increases up till line k, then after, it decreases. The last line has the same or greater indentation than the first line.

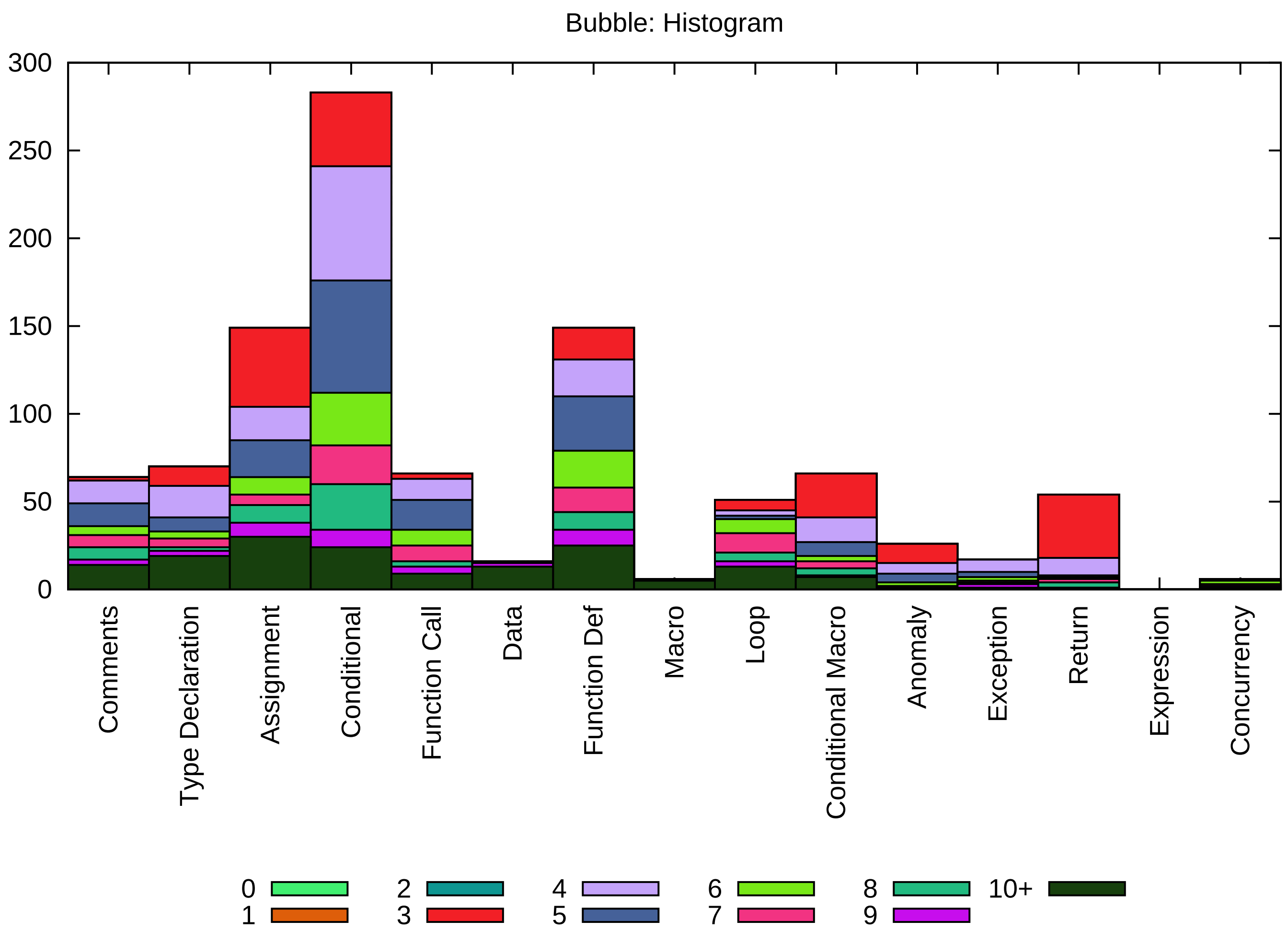


Figure 6: Distribution of revision length of bubble shape revisions per annotation

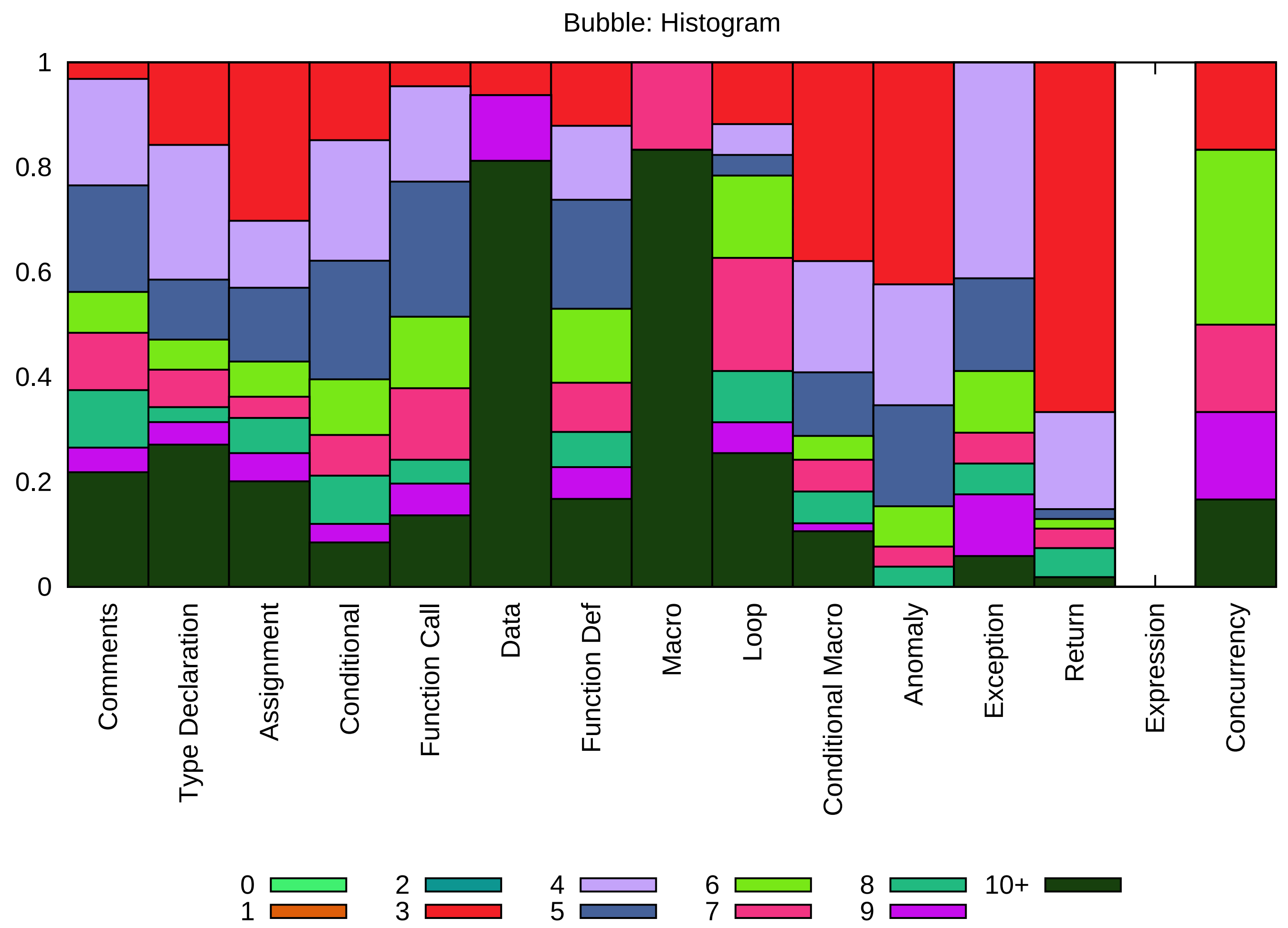


Figure 7: Proportional Distribution of revision length of bubble shape revisions per annotation

Control

- 1001 Randomly Sampled Revisions
- Length of revisions followed a power law/exponential like distribution
- Most likely: comments, type declarations, assignments, function definitions
- Least likely: exceptions, returns, concurrency, conditionals

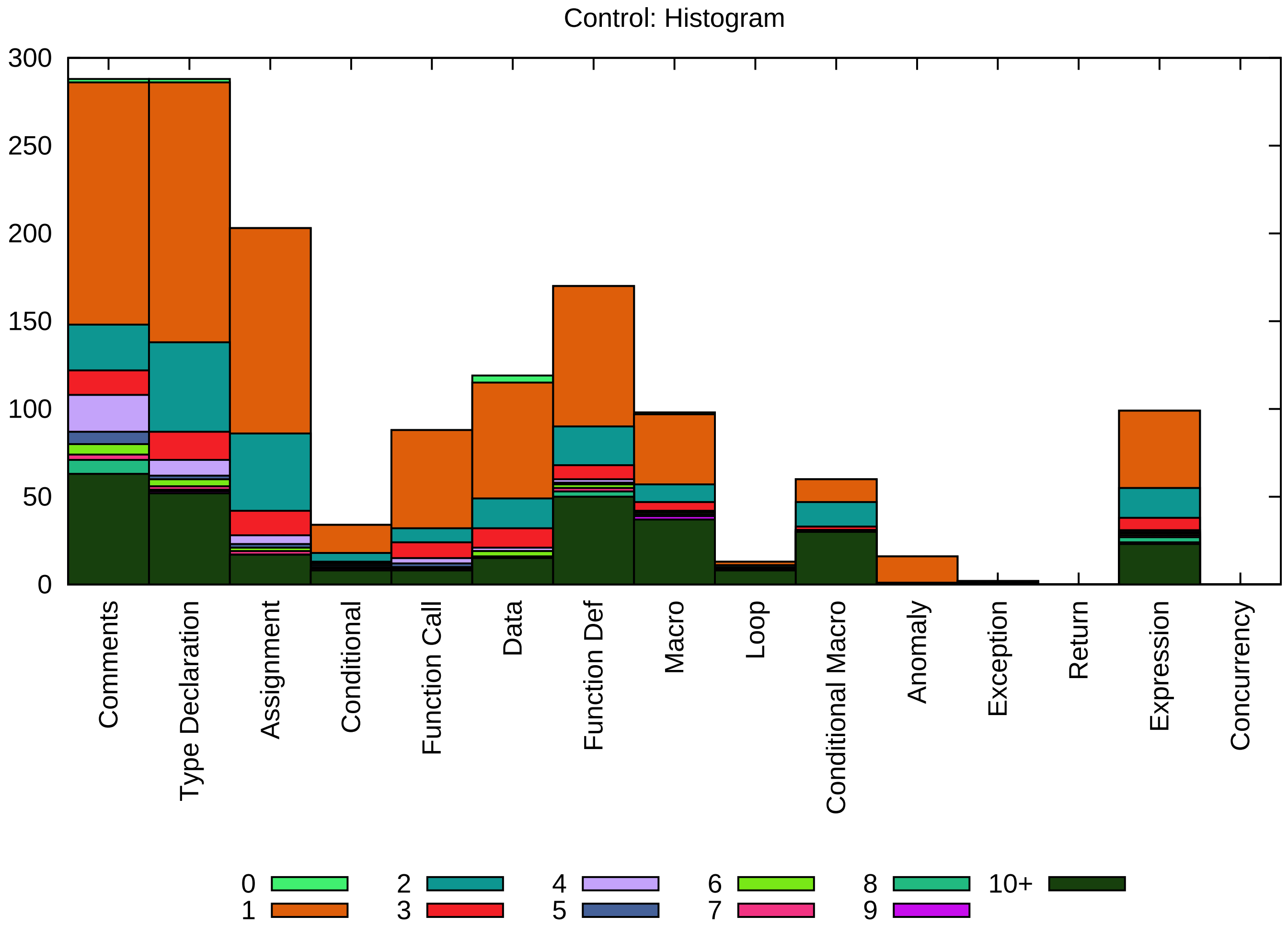


Figure 8: Distribution of revision length of control sampled revisions per annotation

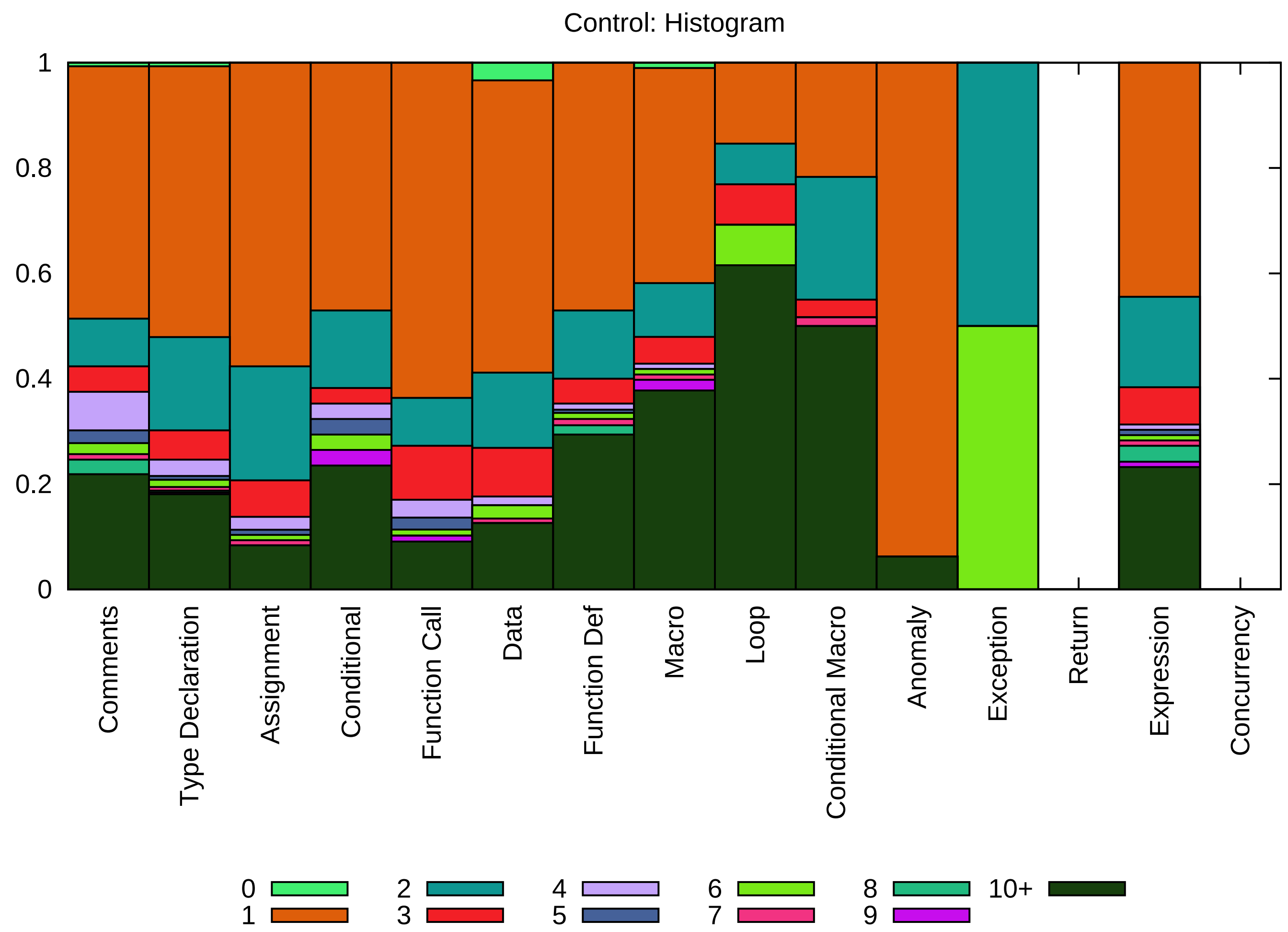


Figure 9: Proportional Distribution of revision length of control sampled revisions per annotation

Indentation Variance

- Evaluated the Variance of revisions
- We broke down the variance by quartiles
- Note: sometimes there are a lot of 0 variance revisions

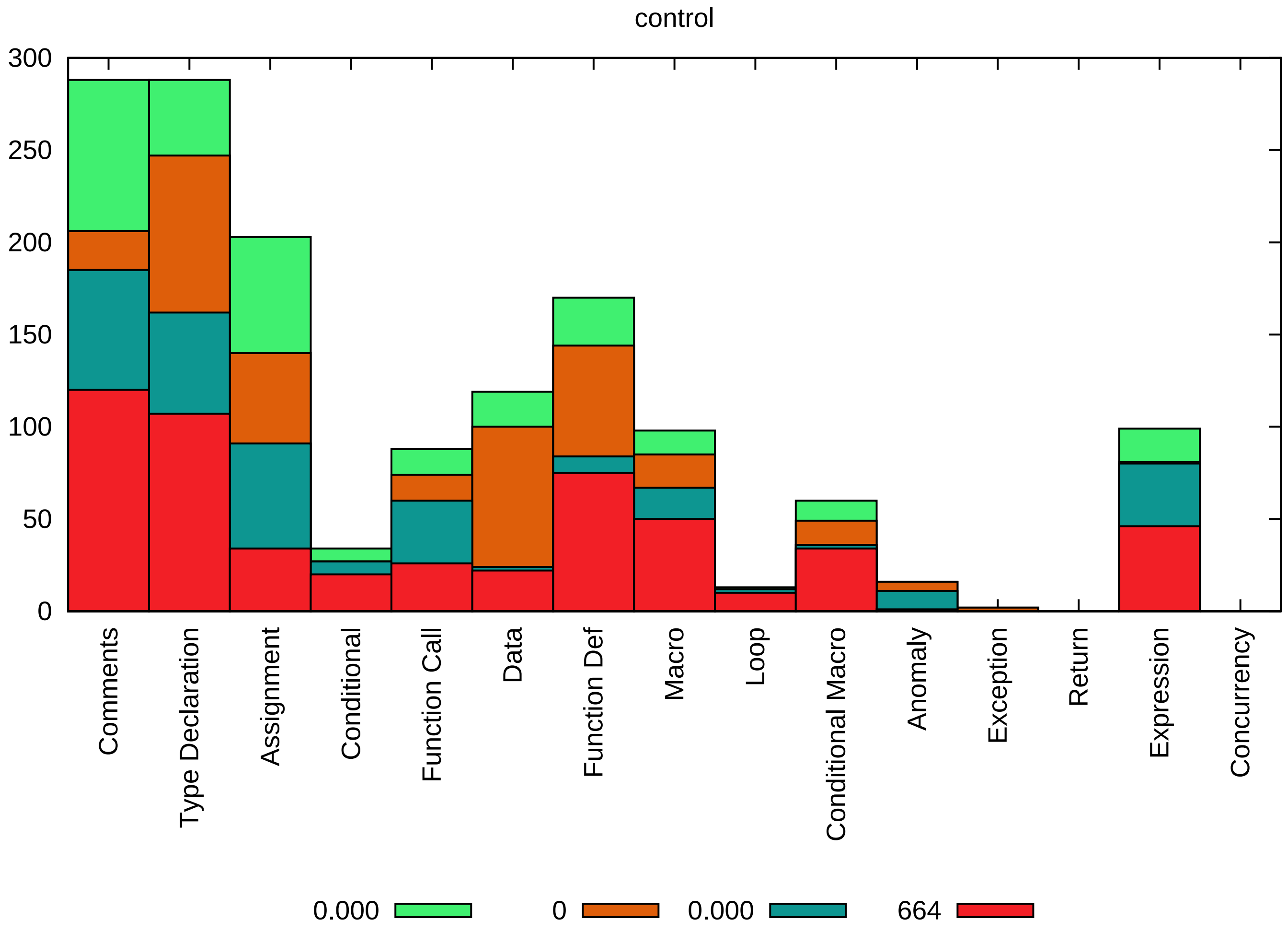


Figure 10: Distribution of control revisions per quartile of variance of indentation.

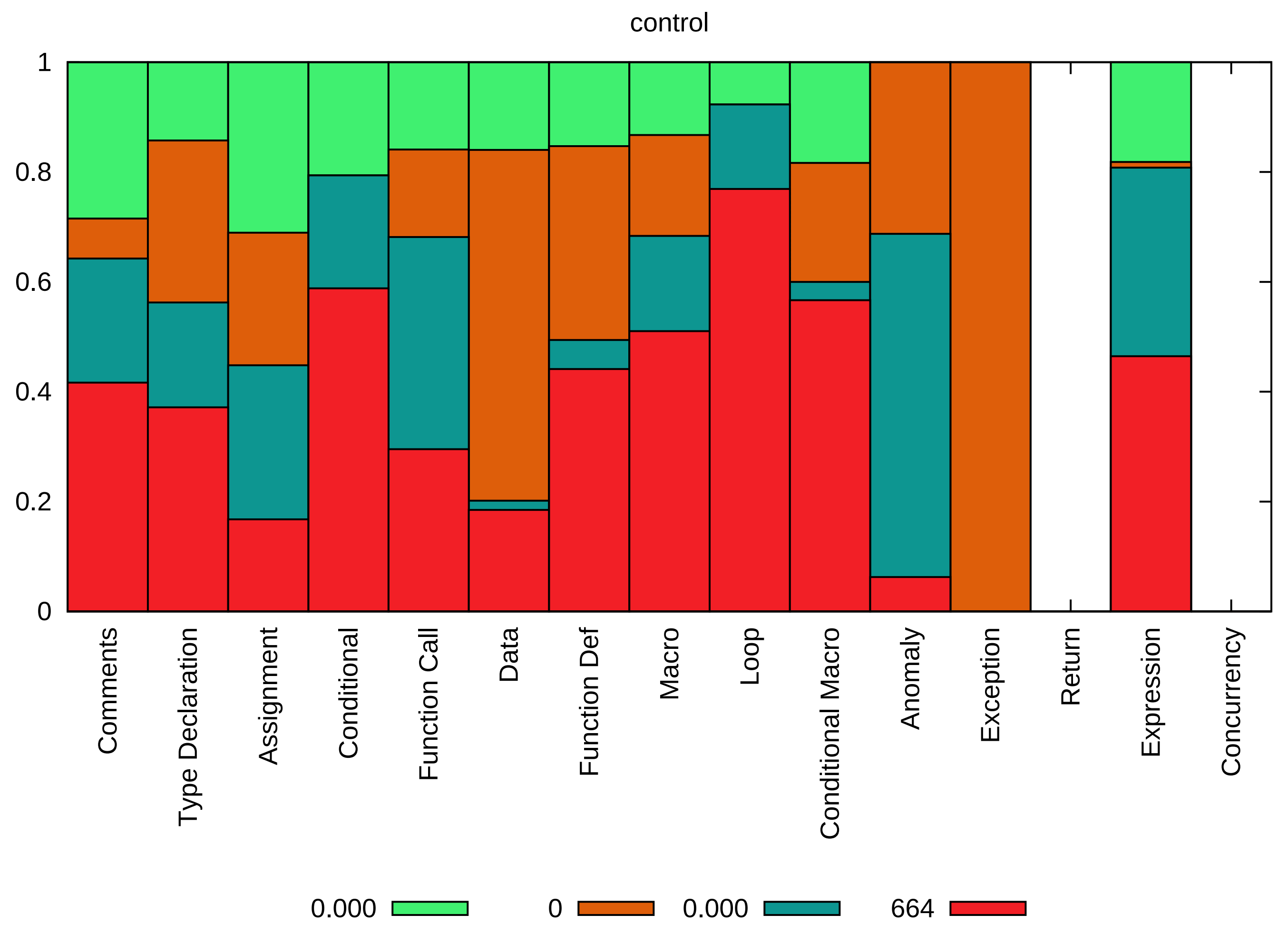


Figure 11: Proportional Distribution of control revisions per quartile of variance of indentation.

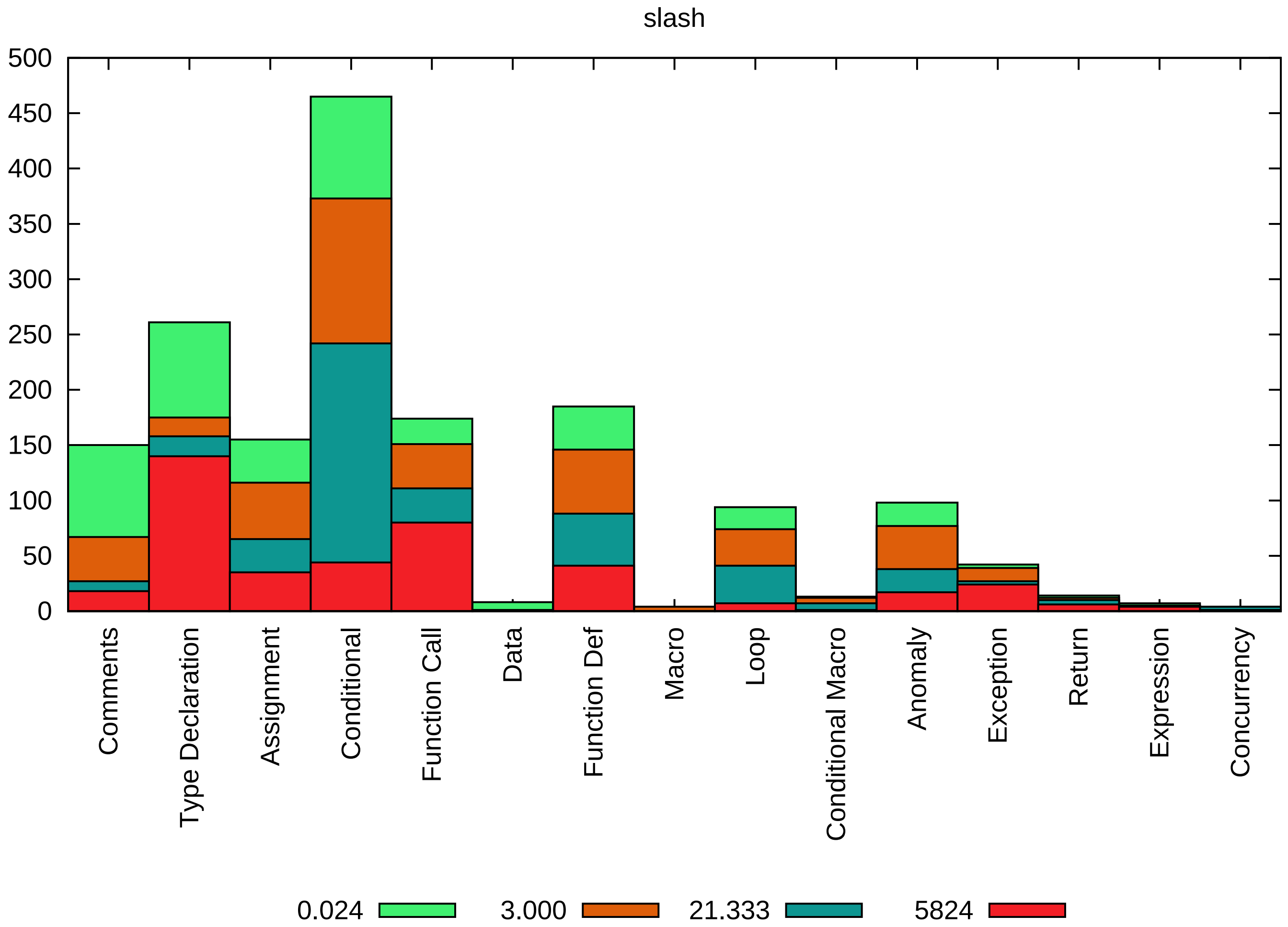


Figure 12: Distribution of slash revisions per quartile of variance of indentation.

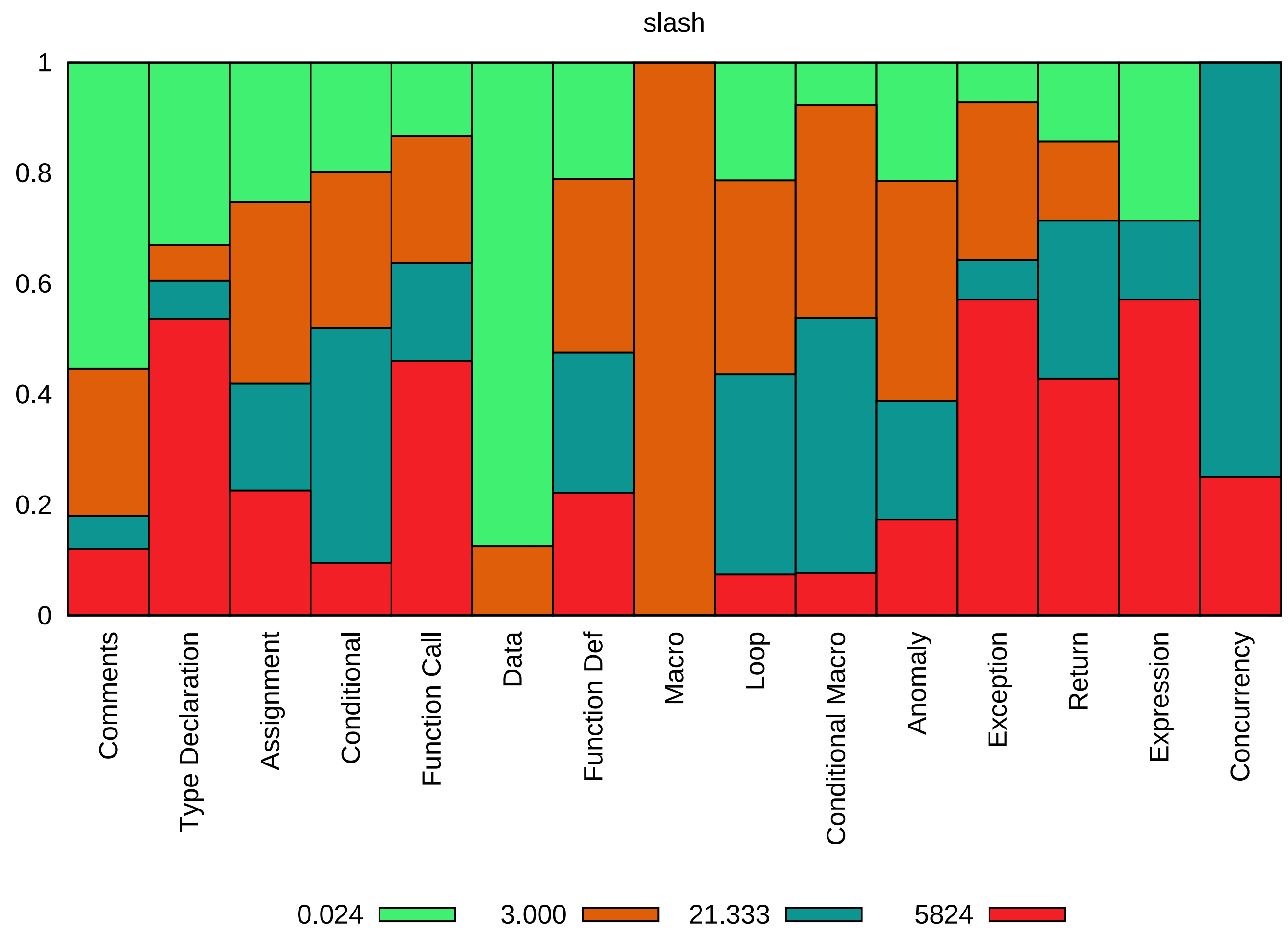


Figure 13: Proportional Distribution of slash revisions per quartile of variance of indentation.

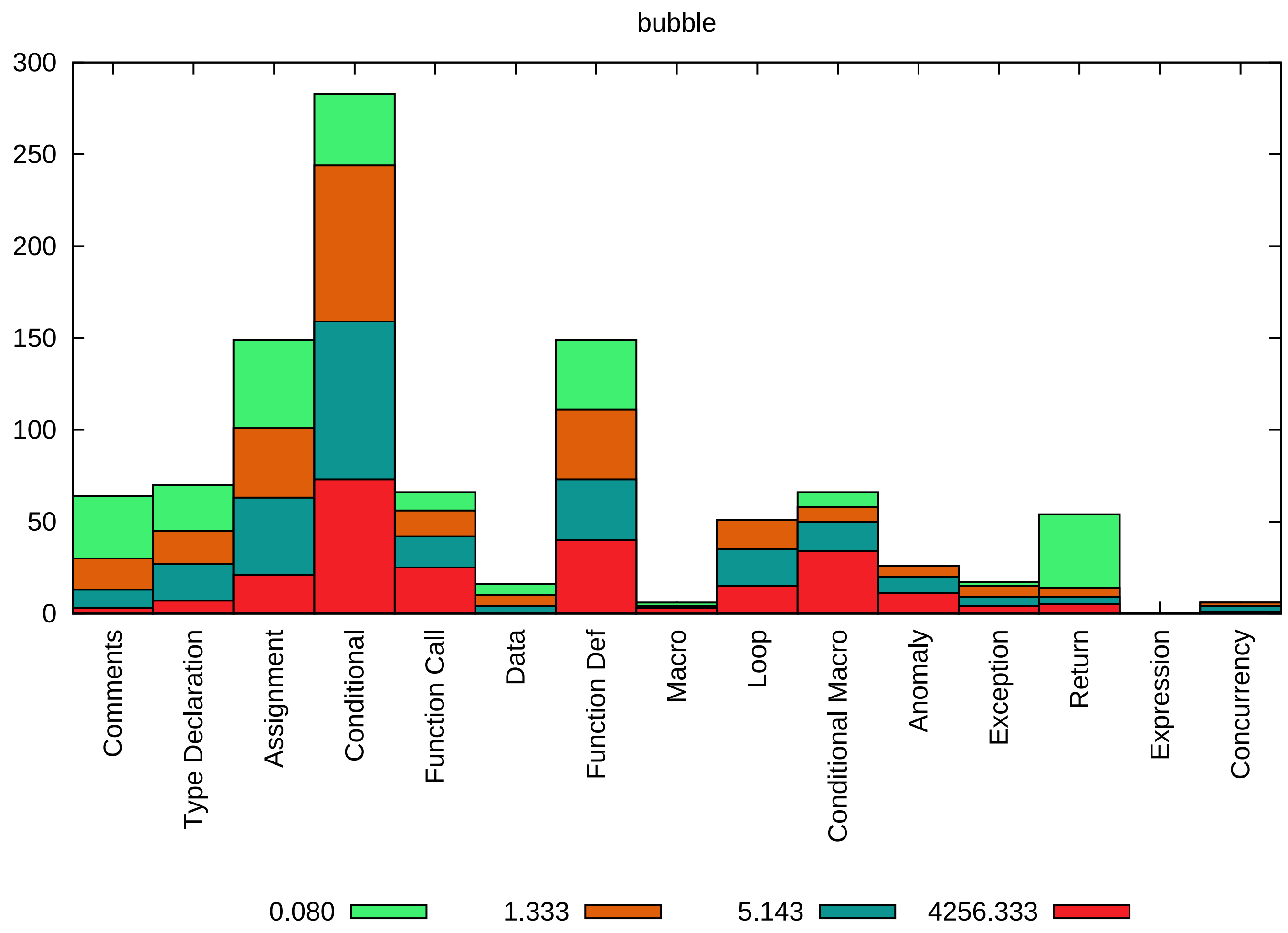


Figure 14: Distribution of bubble revisions per quartile of variance of indentation.

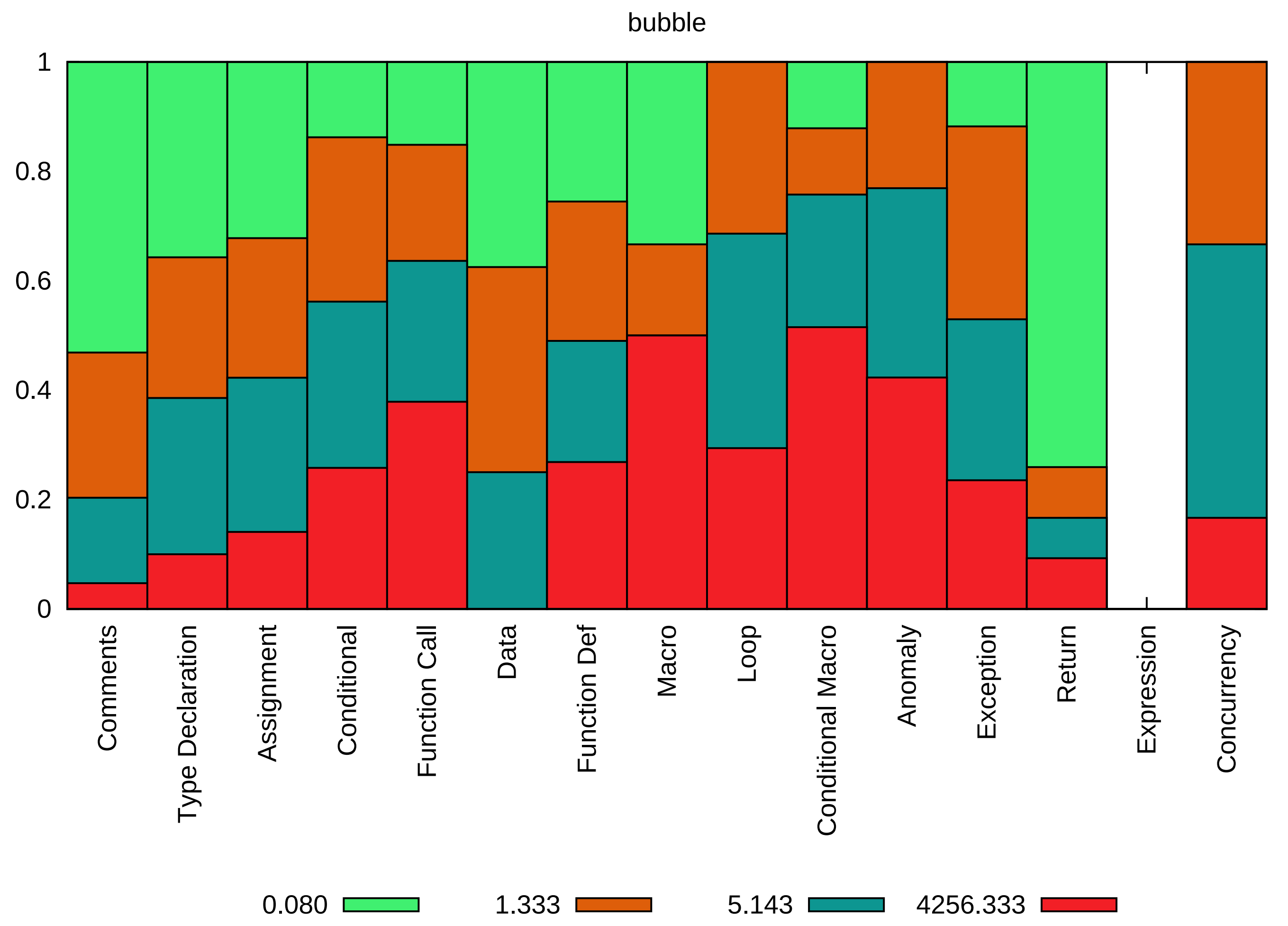


Figure 15: Proportional Distribution of bubble revisions per quartile of variance of indentation.

Questions (1/3)

- What kind of indentation correlates with function definitions?
  - Higher variance indentation
  - Bubble
  - Slash
  - Upper quartile of revision length

Questions (2/3)

- What kinds of code correlate with zero variance indentation?
  - comments
  - type declarations
  - assignments
  - data

Questions (3/3)

- What kinds of code correlate with non-zero variance indentation?
  - conditionals,
  - type declarations
  - function definitions,
  - comments
  - assignments.

Do these observations hold for other languages?

- Seems to hold for
  - Wirth-like syntax
  - C-style syntax
  - Even header files
- Best practices
  - Use consistent Indentation
  - IDEs use consistent Indentation

Issues

- Sampling
  - Projects
  - Revisions
- Biased Annotations

Conclusions

- Bubbles and Slashes relate to branching
- Low variance shapes are often comments, assignments, data and type declarations
- Filtering and sorting by shape can allow you to partition and grab revisions with specific code structures that you’re looking for.

What Kind of Indentation Do You Prefer?

- Tabs, 1 logical unit, 1 character
  - Tab-stops can be set in IDEs
- Personally I just rely on what emacs gives me