University of Glasgow

Functional Baby Talk:

Analysis of Code Fragments from Novice Haskell Programmers

Jeremy.Singer@glasgow.ac.uk

@jsinger_compsci @haskellmooc

Part I : Glasgow Haskell MOOC

from knowing *nothing* to understanding **`Nothing`** in 6 weeks

Glasgow Haskell MOOC

What is Haskell?

Haskell at Glasgow

Glasgow *was* the epicentre of Haskell development


the Haskell movement,
I suppose, was born.

Glasgow, Haskell and the GHC compiler

60 YEARS OF COMPUTING AT GLASGOW

The Haskell Programming Language

Haskell is a non-strict functional programming language widely used for research, and now increasingly adopted by industry. It is named after the logician Haskell Curry.

In 1987 the functional programming community formed a committee to define a standard non-strict language. The committee of around 25 people included several Glaswegians (Kevin Hammond, John Hughes, Simon Peyton Jones, John Launchbury, and Philip Wadler), four of whom also served as Editor for a major release of the Haskell Report. The committee sometimes met in Glasgow.

Philip Wadler

John Hughes

Despite being designed over 25 years ago, Haskell remains in a ferment of active development, with increasingly wide take-up in industry.

More info: www.haskell.org

Haskell

Simon Peyton Jones

Haskell Compilers and GHC

There were initially three Haskell compilers from Yale, Chalmers and Glasgow. Over the years there have been other Haskell implementations, but the Glasgow Haskell Compiler (GHC) remains the most widely used.

Simon Peyton Jones led the development of GHC originally at Glasgow, and latterly at Microsoft Research. Cambridge.

Glasgow contributors to GHC include the committee members, Cordy Hall, Simon Marlow, Hans-Wolfgang Loid Jim Mattson, Will Partain, Phil Trinder, Kevin Hammond and many clever postgrads.

More info: www.haskell.org/ghc/

To this day the School continues to define, implement and use novel programming language technologies.

Illustrations by Lovisa Sundin

Up to 2014

we used to teach Functional Programming as an optional final year course

25-50% of the cohort selected the course

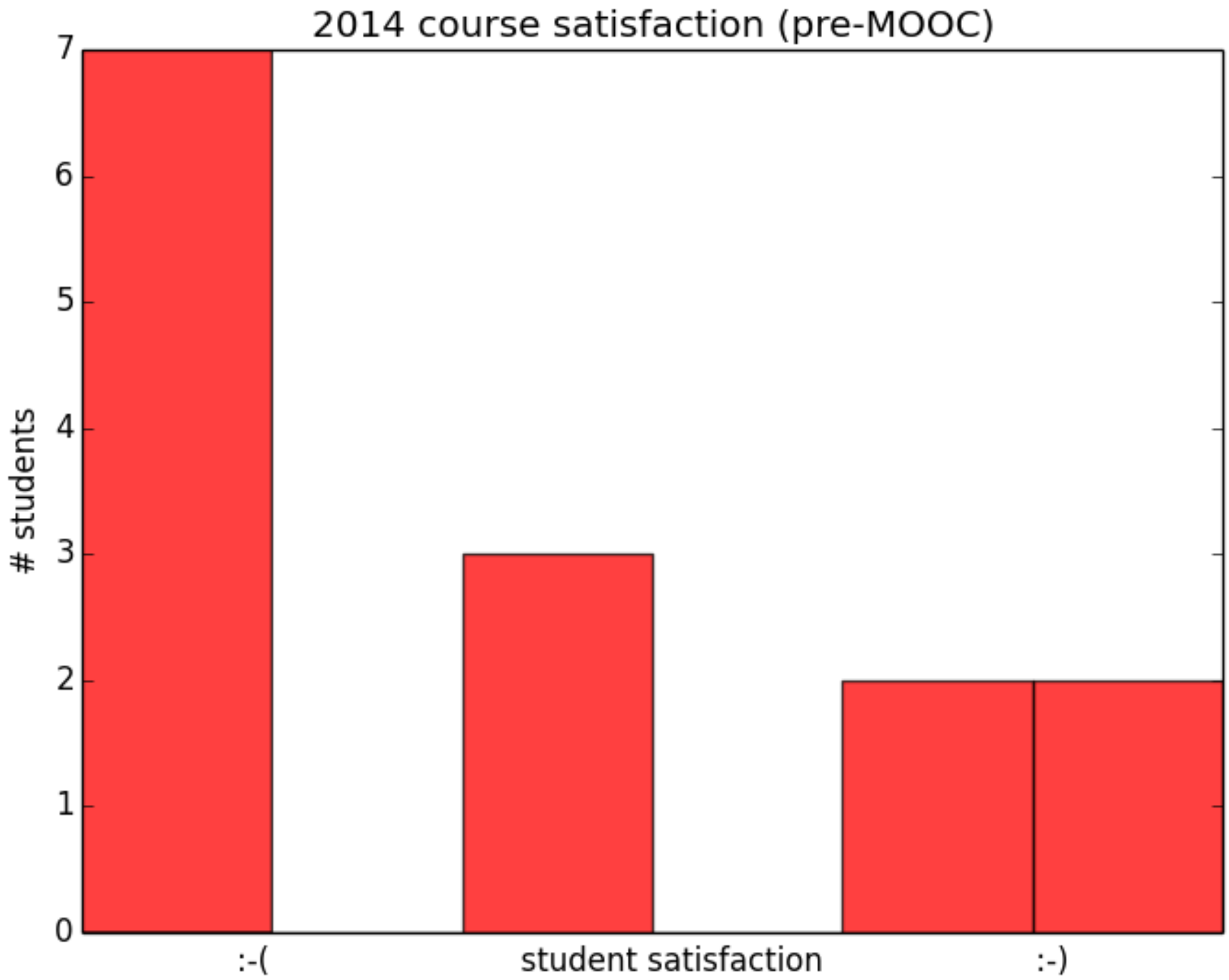
20 hours of face-to-face lectures, one big bang assignment (20%), 80% written exam

Course Catalogue

View Specification Document | Reading List

Functional Programming (H) COMPSCI4021

- **Academic Session:** 2016-17
- **School:** School of Computing Science
- **Credits:** 10
- **Level:** Level 4 (SCQF level 10)
- **Typically Offered:** Semester 1
- **Available to Visiting Students:** Yes
- **Available to Erasmus Students:** Yes


2014 course satisfaction (pre-MOOC)
students
0
1
2
3
4
5
6
7
:-(
student satisfaction
:-)

we need to fix this!

Learning & Teaching

Blended and online learning

- Moodle at the University of Glasgow

Online learning has typically replaced the term 'e-learning', and refers to students

Course Design

Learning type: Acquisition

Learning through acquisition is what learners are doing when they are listening to a lecture or podcast, reading from books or websites, and watching demos or videos

Learning type: Collaboration

Learning through collaboration embraces mainly discussion, practice, and production. Building on investigations and acquisition it is about taking part in the process of knowledge building itself

Learning type: Discussion

Learning through discussion requires the learner to articulate their ideas and questions, and to challenge and respond to the ideas and questions from the teacher, and/or from their peers

Learning type: Investigation

Learning through investigation guides the learner to explore, compare and critique the texts, documents and resources that reflect the concepts and ideas being taught

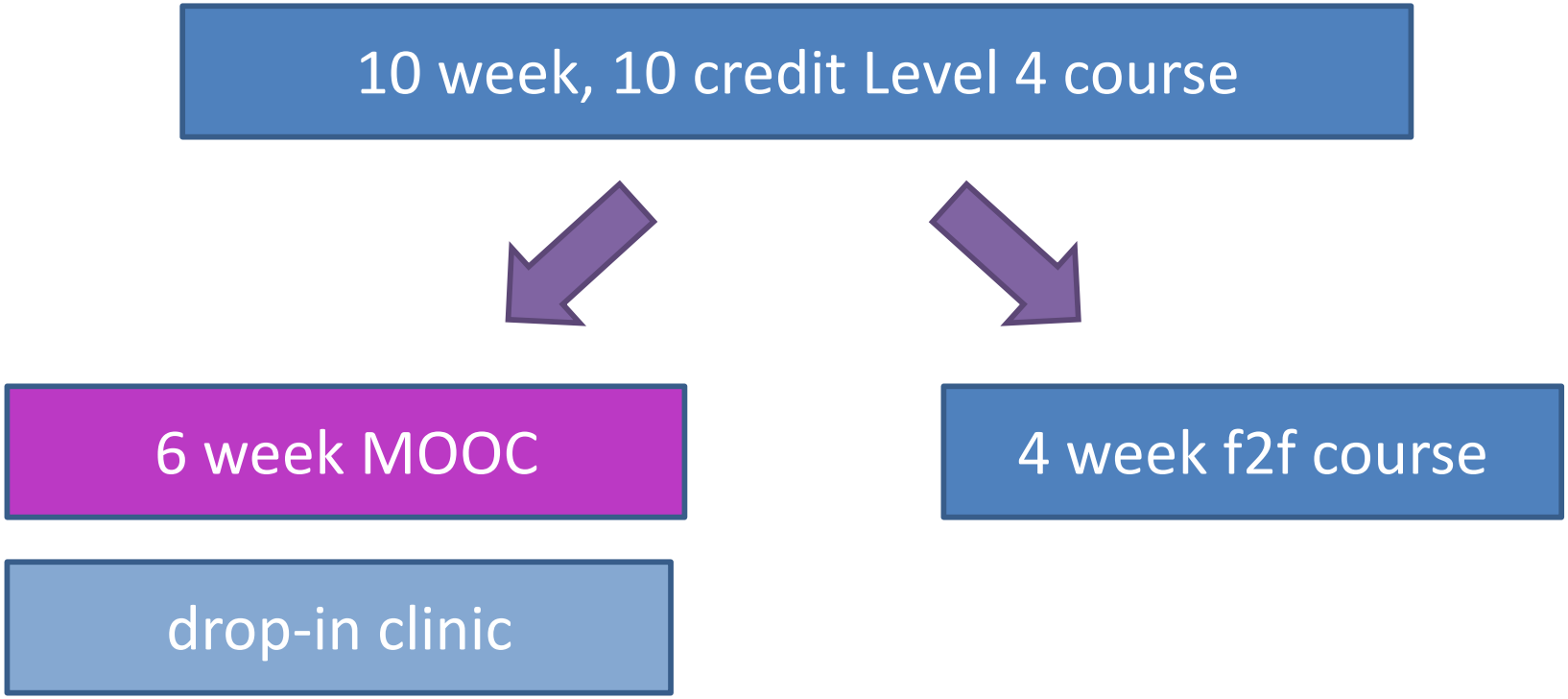
Learning type: Practice

Learning through practice enables the learner to adapt their actions to the task goal, and use the feedback to improve their next action. Feedback may come from self-reflection, from peers, from the teacher, or from the activity itself, if it shows them how to improve the result of their action in relation to the goal

Learning type: Production

Learning through production is the way the teacher motivates the learner to consolidate what they have learned by articulating their current conceptual understanding and how they used it in practice

What we did


10 week, 10 credit Level 4 course
6 week MOOC
4 week f2f course
drop-in clinic

ONLINE COURSE

Functional Programming in Haskell: Supercharge Your Coding

Get an introduction to Haskell, the increasingly popular functional programming language, with this University of Glasgow course.

Register


λ
Σ
Δ
Σ
Δ

the competition

coursera

Catalog Search catalog Institutions Log In Sign Up

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Functional Programming Principles in Scala

Learn about functional programming, and how it can be effectively combined with object-oriented programming. Gain practice in writing clean functional code, using the Scala programming language.

Introduction to Functional Programming

The aim of this course is to teach the foundations of functional programming and how to apply them in the real world.

TU Delft

Archived

Future Dates To Be Announced

Enroll Now

☑ I would like to receive email from

FUN Université Paris Diderot Introduct Search courses Sign up Login

Home • All courses • Introduction to Functional Programming in OCaml

Introduction to Functional Programming in OCaml

hematics Informatique Programmation

https://www.fun-mooc.fr/courses/parisdiderot/56002/session01/about#

PARIS DIDEROT

Enrollment is clos

End of Registration

Registration closed

ONLINE COURSE

Functional Programming in Erlang

University of Kent

Learn the theory and practice of functional programming in Erlang, through practical exercises and suggested projects.

Join free

Upgrade - £59

What's the difference?

Course title	Platform	Lead Educators	Run (year)	Signups	Completions
Functional Programming Principles in Scala	Coursera	Odersky	1 (2012)	50K [14]	9.6K
Introduction to Functional Programming	edX	Meijer	1 (2014)	38K [12]	2.0K
Introduction to Functional Programming in OCaml	FUN	Di Cosmo / Regis-Gianas / Treinen	1 (2015)	3.7K	300
Functional Programming in Haskell	FutureLearn	Singer / Vanderbauwhede	1 (2016)	6.4K	900
Functional Programming in Erlang	FutureLearn	Thompson	1 (2017)	5.6K	400

more stats from our course

- MOOC signups: 6.4K
- active learners: 3.1K
- completions: 900
- Glasgow students: 120

Future Learn

IDENTITY VERIFIED

Issued 30th October 2016. futurelearn.com/certificates/abcdef

Certificate of Achievement

Susanna Rodriguez

has completed the following course:

FUNCTIONAL PROGRAMMING IN HASKELL: SUPERCHARGE YOUR CODING
UNIVERSITY OF GLASGOW

This online course explored the concepts of functional programming using the Haskell language. The course covered standard functional programming techniques, as well as some advanced concepts, which were applied to realistic programming problems.

6 weeks, 4 hours per week

Wim Vanderbauwhede
Senior Lecturer in Computing Science
University of Glasgow

Jeremy Singer
Lecturer in Computing Science
University of Glasgow

University of Glasgow

Part II : Analysis of Novice Code

"I have been trying almost all
tod

ous to do
nything in
nd it and
te Haskell
ssion."

"It's not my first attempt to learn Haskell. I always get stuck with monad part and do-notation."

Use data from MOOC to help novice learners

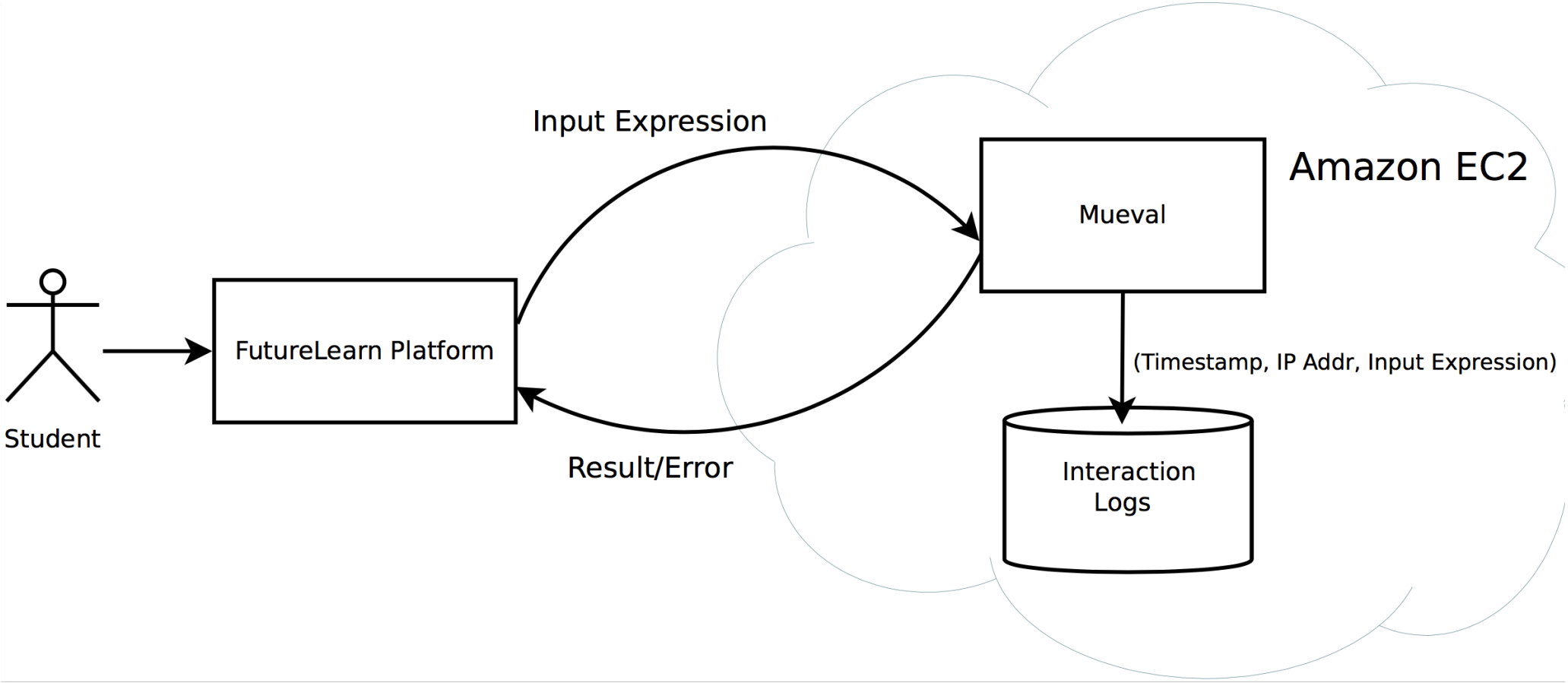
1. improved pedagogy
2. error-aware toolchain
3. modifications to language

We have lots of data

- forum comments and discussion contributions
- learner analytics from FutureLearn
- student feedback on evaluation forms
- coursework performance
- code submitted to REPL

We have lots of data

- forum comments and discussion contributions
- learner analytics from FutureLearn
- student feedback on evaluation forms
- coursework performance
- code submitted to REPL


Input Expression
Amazon EC2
Mueval
Student
FutureLearn Platform
(Timestamp, IP Addr, Input Expression)
Result/Error
Interaction
Logs

Haskell Tutorials -- Interactive tutorials in your browser - Mozilla Firefox

Fedora Project - Start Pa... | Haskell Tutorials -- Inter...

localhost:4001/tutorial11/#step3

University of Glasgow

Haskell Tutorials

Active users

```
Type Haskell expressions in here.
λ 41
41
:: Num a => a
λ
```

[step 3/25]

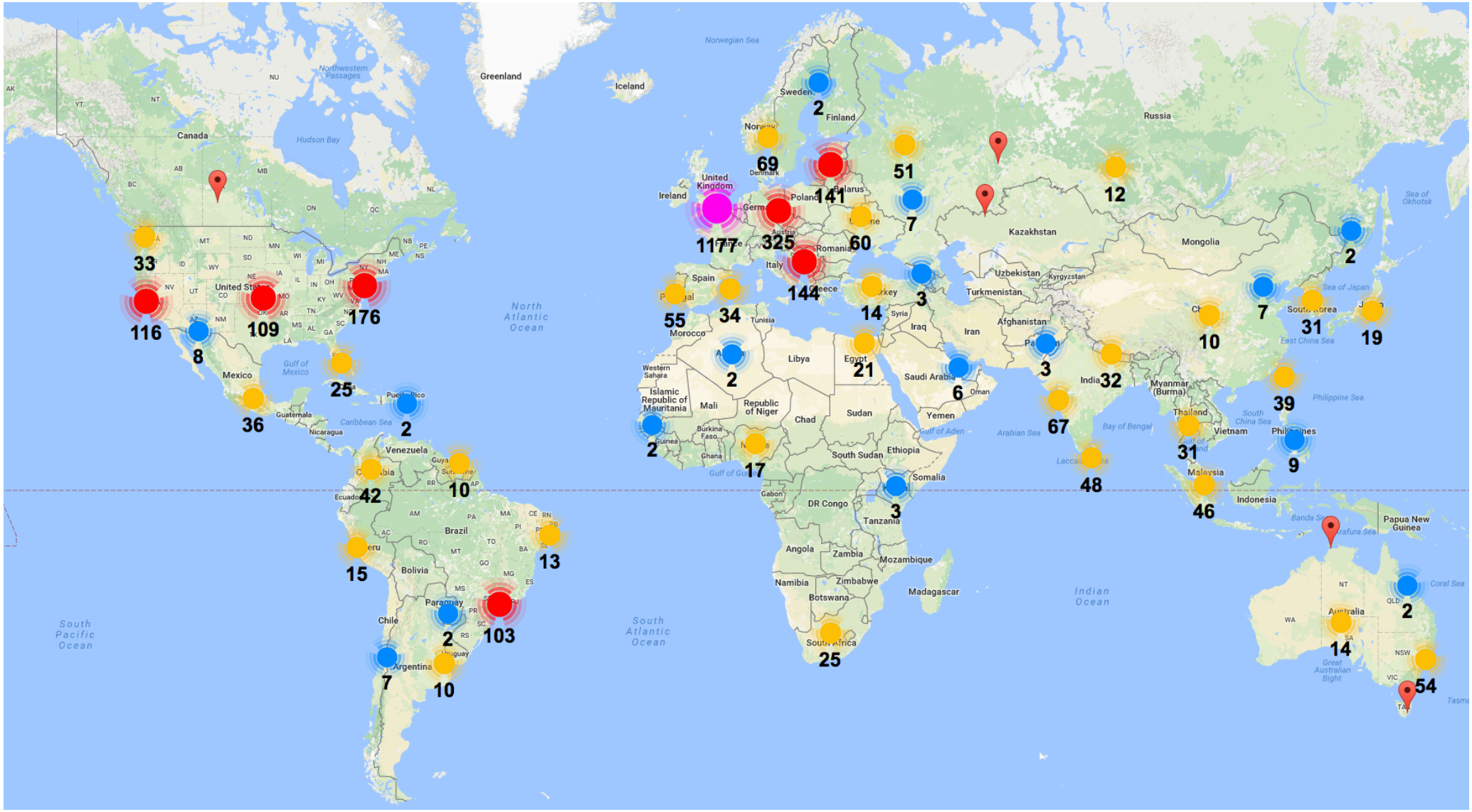
OK, no surprises so far, you got back the number 41 as expected.

Now type a simple integer arithmetic operation, e.g. `6*7`, and observe that it evaluates to the expected result.

Analysis of REPL logs

- 21 September to 3 November 2016
- 161K lines of interactions
- around 17MB of data
- 3.3K unique IP addresses

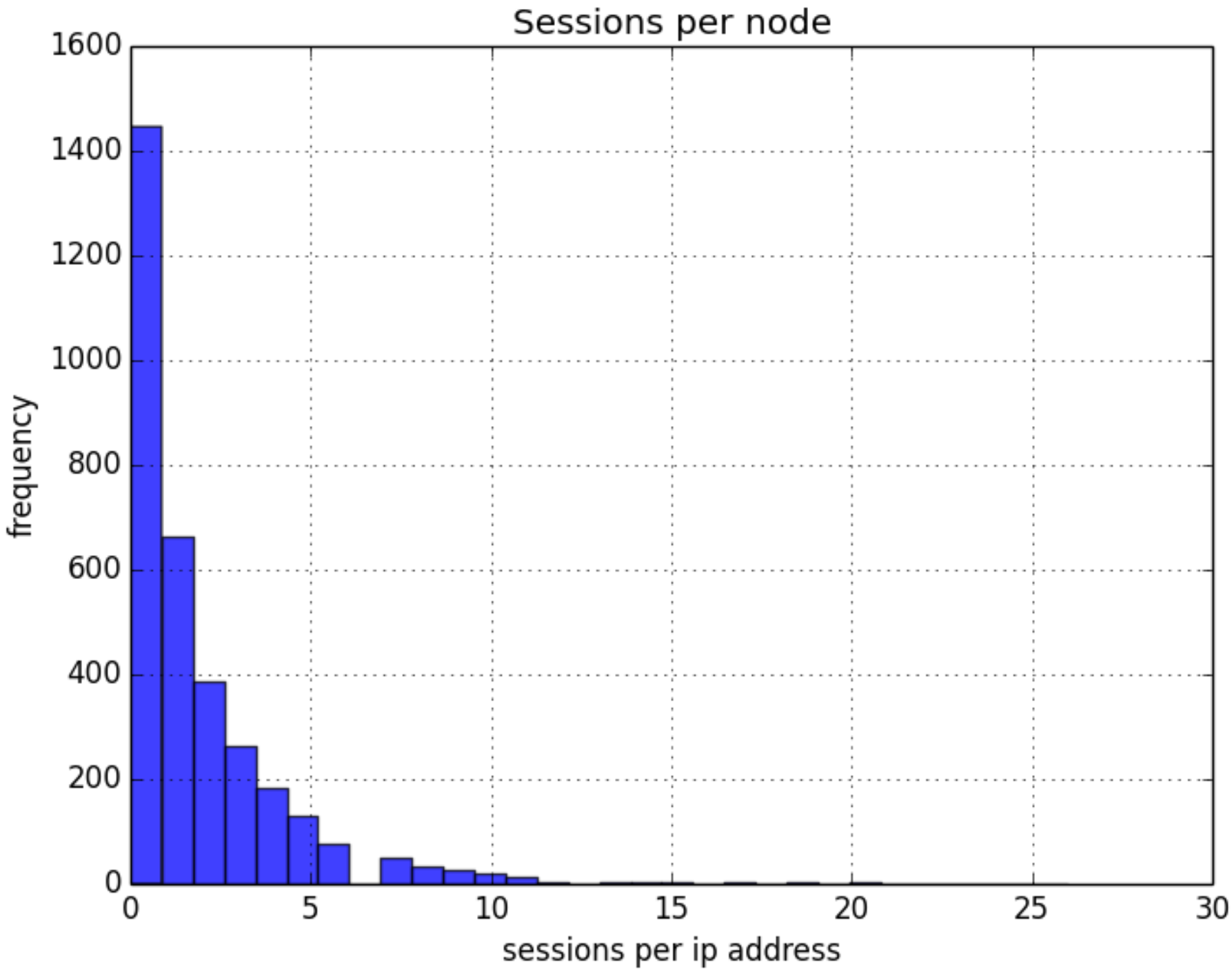
```
2016-09-21 20:34:01.078514 UTC
98.226.108.109> let { } in foldr
(\elt acc -> acc:elt) "" "Reversing
a string"
```

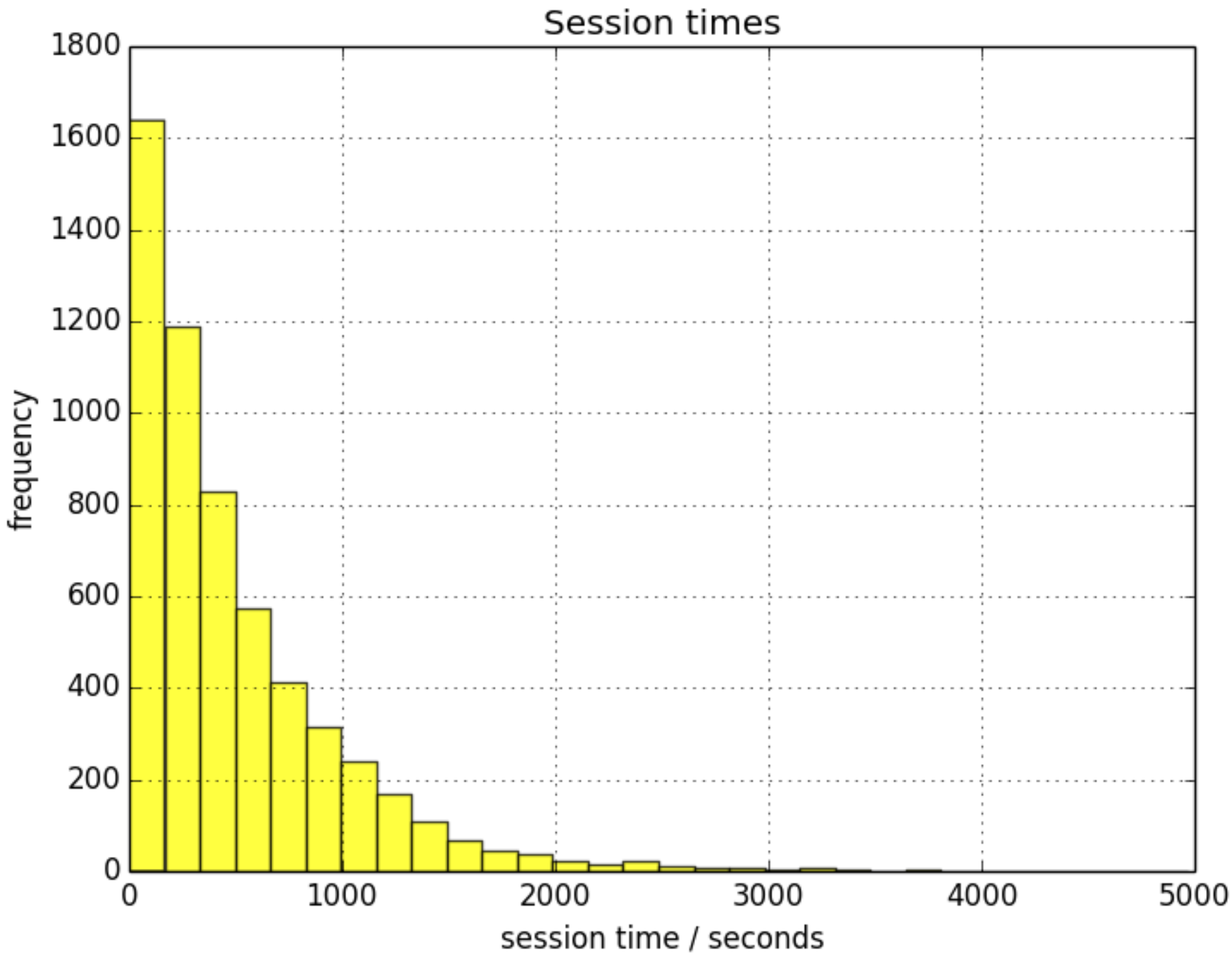

Canada
Greenland
Iceland
Norwegian Sea
Hudson Bay
Northwestern Passages
United States
Mexico
Gulf of Mexico
Caribbean Sea
Guatemala
Nicaragua
Venezuela
Brazil
Bolivia
Paraguay
Chile
Argentina
Ecuador
Peru
Uruguay
Ireland
United Kingdom
Spain
Portugal
Morocco
Algeria
Tunisia
Libya
Egypt
Italy
Greece
Romania
Poland
Belarus
Finland
Sweden
Norway
Denmark
Russia
Kazakhstan
Mongolia
Uzbekistan
Turkmenistan
Kyrgyzstan
Iran
Iraq
Syria
Turkey
Afghanistan
Saudi Arabia
Yemen
Oman
India
Myanmar (Burma)
Thailand
Vietnam
Malaysia
Indonesia
Philippines
Papua New Guinea
Australia
Mali
Republic of Niger
Chad
Sudan
South Sudan
Ethiopia
Somalia
DR Congo
Tanzania
Angola
Zambia
Mozambique
Namibia
Botswana
Zimbabwe
Madagascar
South Africa
Western Sahara
Islamic Republic of Mauritania
Guinea
Burkina Faso
Ghana
Gabon
North Atlantic Ocean
South Atlantic Ocean
South Pacific Ocean
Indian Ocean
Arabian Sea
Bay of Bengal
Gulf of Aden
Philippine Sea
South China Sea
East China Sea
Sea of Japan
Sea of Okhotsk
Coral Sea
Great Australian Bight
116
33
8
109
176
25
36
2
42
10
15
13
2
103
7
10
55
34
2
2
17
1177
325
144
69
141
51
7
60
14
21
3
6
3
25
12
2
7
10
3
32
67
48
46
39
9
31
31
19
2
14
54

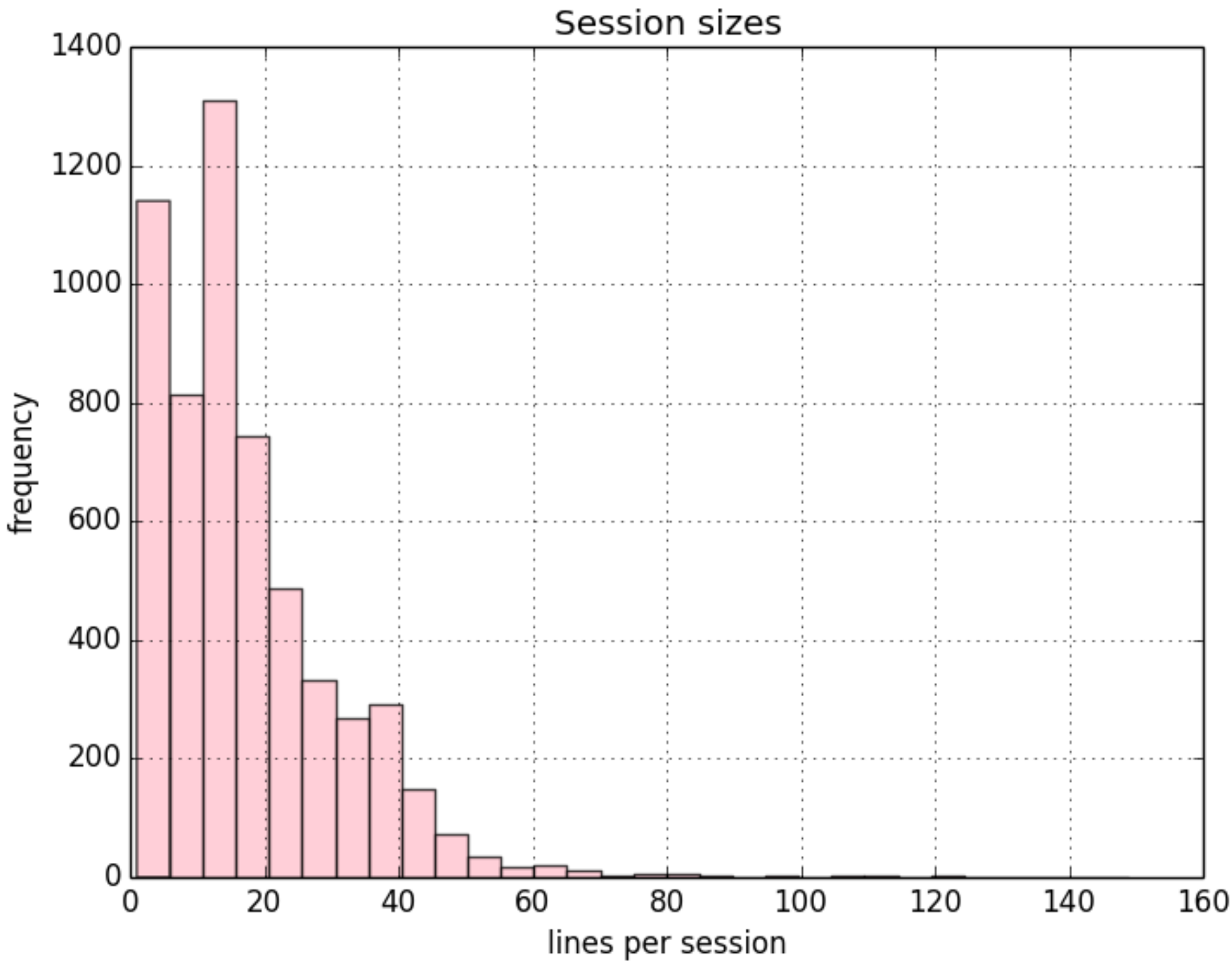
Group individual lines into *sessions*

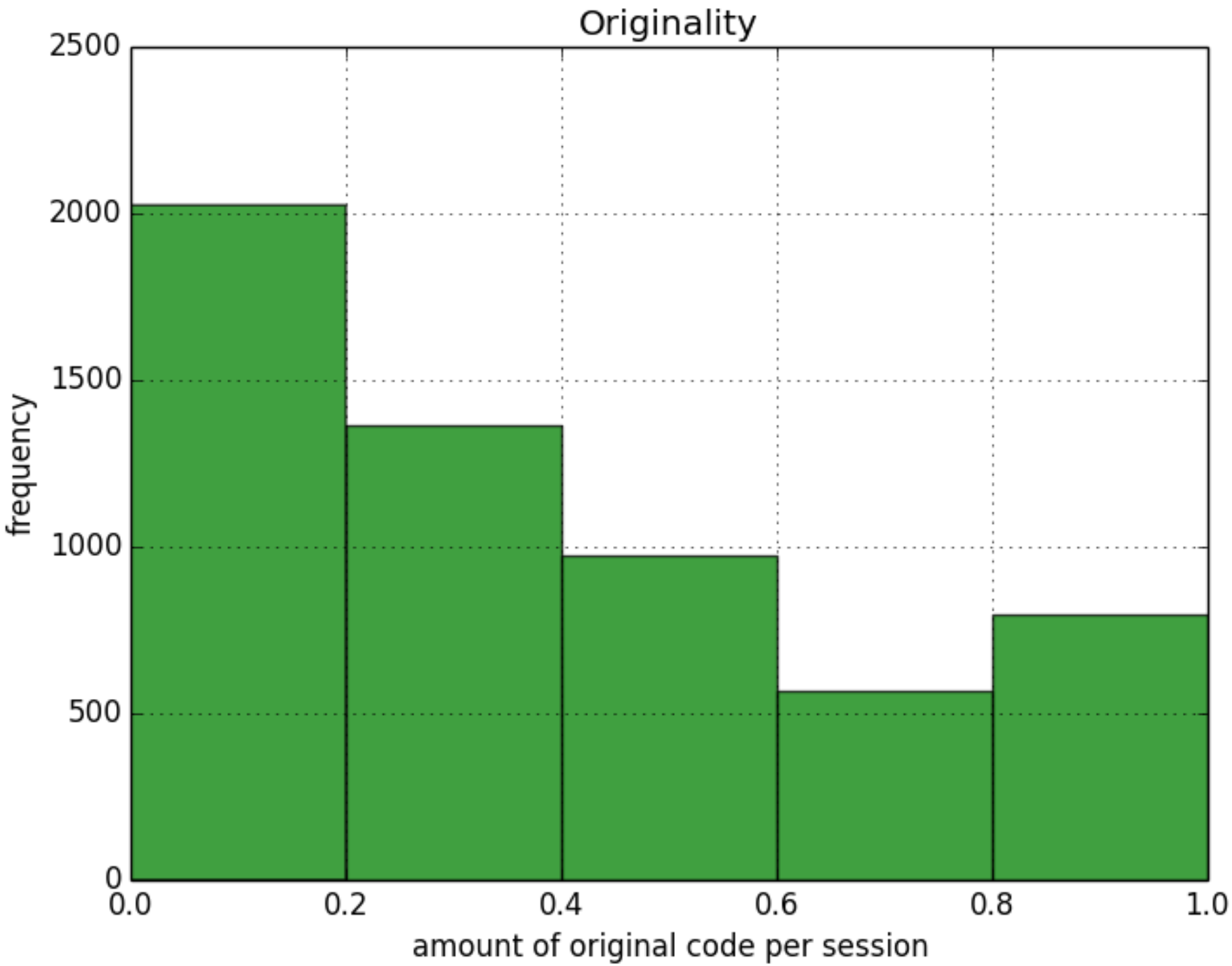
- Associate log entries that originate from the same IP address, with an interval of less than 10 minutes between successive interactions.

- 5.7K sessions, 1.7 sessions/IP address


Sessions per node
1600
1400
1200
1000
800
600
400
200
0
frequency
0
5
10
15
20
25
30
sessions per ip address


Session times
1800
1600
1400
1200
1000
800
600
400
200
0
frequency
0
1000
2000
3000
4000
5000
session time / seconds


Session sizes
1400
1200
1000
800
600
400
200
0
frequency
0
20
40
60
80
100
120
140
160
lines per session


Originality
2500
2000
1500
1000
500
0
frequency
0.0
0.2
0.4
0.6
0.8
1.0
amount of original code per session

Top Five Errors

based on parsing raw Haskell lines from log files and finding syntax errors.

(1) Parenthesis mismatch:

Closing parenthesis characters are frequently missing

```
min((max 3 4) 5))
```

— Heeren et al. [10] also state that “illegal nesting of parentheses, braces, and brackets is a very common lexical error”.

(2) *Bad scoping*:

Problems with **`let`** and **`where`** constructs are apparent. The mueval expression parser does not support where clauses properly. Some users had confusion with the syntax of let.

```
let (a,b) (10, 12) in a * 2
```

(3) *Misunderstanding do blocks*:

Many people tried to bind values to names with <- outside a do block, or bind names in a do block as the final action

```
do putStrLn "nnnnn "; z <- getLine;
```

(4) *Complex constructs*:

The mueval interpreter is particularly restrictive. It does not support data or type definitions, or definitions of multi-line functions. Several users attempted to enter such code, which did not parse correctly.

(5) Incorrect syntax for **`enumFromThenTo`** *syntactic sugar*:

People misunderstood the .. notation, generating incorrect list expressions like:
`[0,1,3..10]` or `[0, 2, ..]` or
`[1.1, 1.2, .. 2.0]`

Threats to Validity

- Single course, single run
- TryHaskell is a language subset
- Browser incompatibility

Conclusions

- need to correlate a broad range of analytics
- TryHaskell is not an ideal platform
- User engagement shows standard MOOC drop-off
- MOOC runs again in September – please sign up at https://www.futurelearn.com/courses/functional-programming-haskell

haskell mooc

All Shopping News Videos Images More Search tools

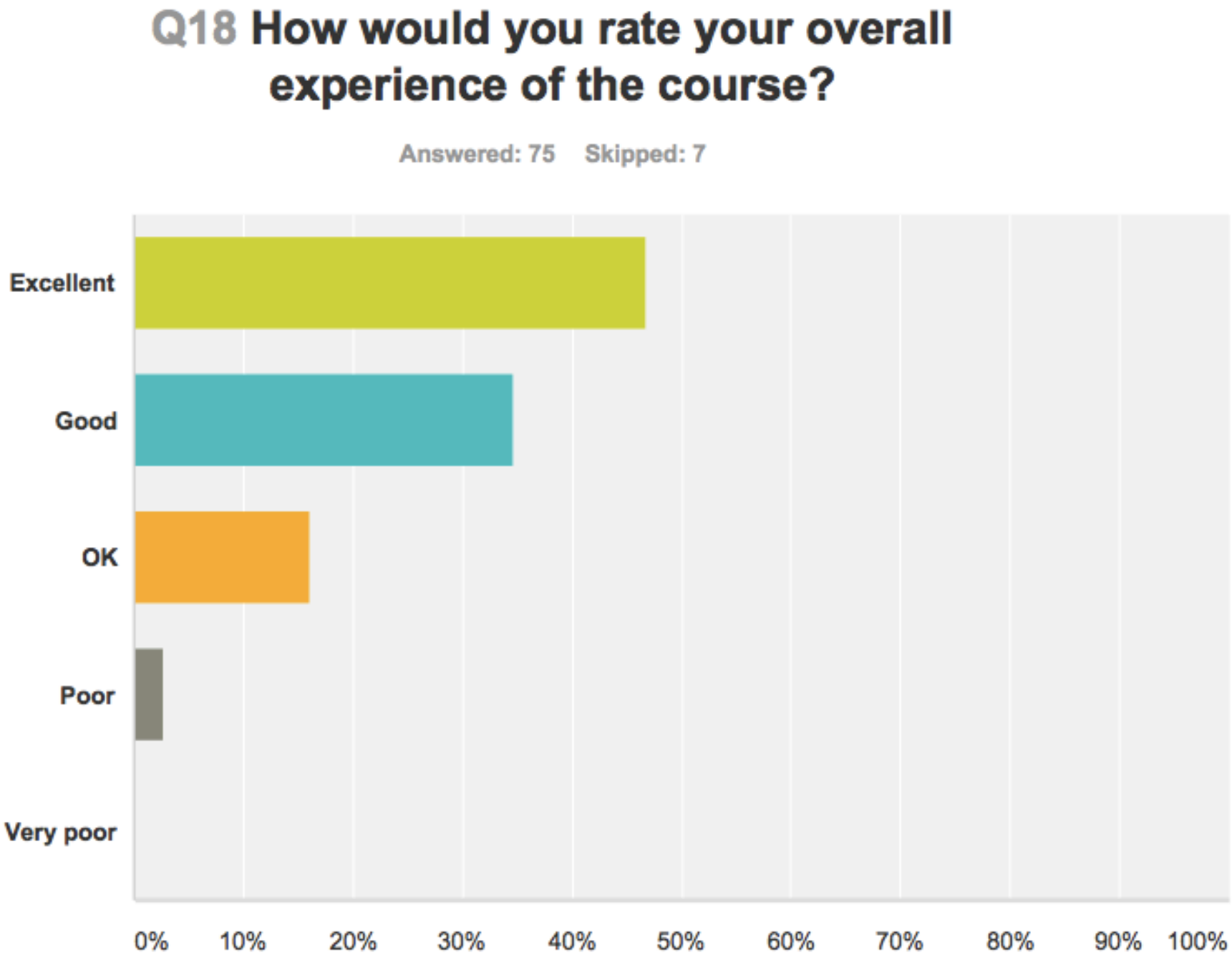
About 17,500 results (0.37 seconds)

Functional Programming in Haskell - Free online course - FutureLearn
https://www.futurelearn.com/courses/functional-programming-**haskell**
Get an introduction to **Haskell**, the increasingly popular functional programming language, with this University of Glasgow course. ... **Haskell** is a functional programming language, based on formal mathematical principles. ... On this introductory course, you will discover the power ...

Introduction to Functional Programming | edX
https://www.edx.org/course/introduction-functional-programming-delftx-fp101x-0
This course will use **Haskell** as the medium for understanding the basic ... theme, which is properly assessed in the homework, like a decent **MOOC** should do.

GitHub - wimvanderbauwhede/HaskellMOOC: "Introduction to ...
https://github.com/wimvanderbauwhede/**HaskellMOOC**
HaskellMOOC - "Introduction to Functional Programming in Haskell", a Glasgow University Computing Science MOOC to be hosted on

FutureLearn Summary Feedback




of Glasgow
An advantage of this is that
it's available free, online.
-1:23
1x
HD

Functional Programming in Haskell

supercharge your coding

University of Glasgow

Future Learn

- ~ 6k learners
- 6 week course
- interactive programming environment (fork of tryhaskell)
- basics of Haskell
- really ‘how to think functionally’

- google ‘haskell mooc’
- follow @HaskellMOOC
- or visit http://bit.ly/haskellmooc

Haskell MOOC

Google haskell mooc Anmelden

Alle Shopping Bilder News Videos Mehr Suchoptionen

Ungefähr 17 100 Ergebnisse (0,33 Sekunden)

Introduction to Functional Programming | edX
https://www.edx.org/.../introduction-functional-prog... Diese Seite übersetzen
This course will use **Haskell** as the medium for understanding the basic principles of functional programming. While the specific language isn't all that important, ...

Functional Programming in Haskell - University of Glasgow
https://www.futurelearn.com/.../functional-program... Diese Seite übersetzen
Get an introduction to **Haskell**, the increasingly popular functional programming language, with this University of Glasgow course.
Mo., 19. Sep. Functional Programming in ...

Haskell | MOOC List
https://www.**mooc**-list.com/tags/**haskell** Diese Seite übersetzen
Computer Science: Programming & Software Engineering, Engineering · Programming, Functional Programming, Functional Languages, **Haskell** · More Info ...

Functional Programming in Haskell - Course
https://onlinecourses.nptel.ac.in/noc15_cs13/ Diese Seite übersetzen
Haskell brings together the best features of functional programming and is

haskell mooc

All Shopping News Videos Images More Search tools

About 17,500 results (0.37 seconds)

Functional Programming in Haskell - Free online course - FutureLearn
https://www.futurelearn.com/courses/functional-programming-**haskell**
Get an introduction to **Haskell**, the increasingly popular functional programming language, with this University of Glasgow course. ... **Haskell** is a functional programming language, based on formal mathematical principles. ... On this introductory course, you will discover the power ...

Introduction to Functional Programming | edX
https://www.edx.org/course/introduction-functional-programming-delftx-fp101x-0
This course will use **Haskell** as the medium for understanding the basic ... theme, which is properly assessed in the homework, like a decent **MOOC** should do.

GitHub - wimvanderbauwhede/HaskellMOOC: "Introduction to ...
https://github.com/wimvanderbauwhede/**HaskellMOOC**
HaskellMOOC - "Introduction to Functional Programming in Haskell", a Glasgow University Computing Science MOOC to be hosted on

Feedback from Glasgow students

Beginning of the course was online with activi elp co ning.

Video lectures were great and gave students a chance to learn at own pace.

Particularly enjoyed the online platform and community

teachers

wim

jeremy