PYGOTHAM

RECAP

TALKS I LIKED

» "Playing with Python Bytecode" -- Scott Sanderson and Joe Jevnik

» "Python Performance Profiling: The Guts And The Glory" -- A. Jesse Jiryu Davis

» "A tale of two cellphones: Python on Android and iOS" -- Russell Keith-Magee

» "Vector space modeling on music data" -- Tim Schmeier

PLAYING WITH PYTHON BYTECODE

-- SCOTT SANDERSON & JOE JEVNIK

PLAYING WITH PYTHON BYTECODE

```
>>> def add(a, b):
...     return a + b
...
>>> add.__code__.co_code
'|\x00\x00|\x01\x00\x17S'
```

PLAYING WITH PYTHON BYTECODE

```
>>> import dis
>>> dis.dis(add)
  2           0 LOAD_FAST                0 (a)
              3 LOAD_FAST                1 (b)
              6 BINARY_ADD
              7 RETURN_VALUE
```

PLAYING WITH PYTHON BYTECODE

```
>>> from types import CodeType, FunctionType
>>> add = FunctionType(
...     CodeType(
...         2, 0, 2, 2, 0x0043,
...         b'|\x00\x00|\x01\x00\x17S',
...         (1,), (), ('a', 'b'),
...         'ayy lmao', 'add', 0, b'', (), ()
...     ), {})
>>> add(1, 2)
3
```

PLAYING WITH PYTHON BYTECODE

Code Transformer:
https://github.com/llllllllll/codetransformer

PYTHON PERFORMANCE PROFILING: THE GUTS AND THE GLORY

-- A. JESSE JIRYU DAVIS

PYTHON PERFORMANCE PROFILING: THE GUTS AND THE GLORY

» A profiler does not tell you where there are inefficiencies in your code;

» A profiler helps you make a hypothesis about where there are inefficiencies in your code;

» Before trying to optimize, you must test the hypothesis with an experiment!

A TALE OF TWO CELLPHONES: PYTHON ON ANDROID AND IOS

-- RUSSELL KEITH-MAGEE

A TALE OF TWO CELLPHONES: PYTHON ON ANDROID AND IOS

- iOS: clang - same C compiler used to compile CPython on macOS
  - Can compile CPython for any iOS device (with a patch to Python)
  - iOS native libraries are Objective C -- easy to call
  - Uses Python3's type annotations

A TALE OF TWO CELLPHONES: PYTHON ON ANDROID AND IOS

- Android (Java-ish):
  - You can use the JNI to call embedded C code from Java
  - You can compile and include Jython
  - Java bytecode looks like Python bytecode if you squint hard enough

A TALE OF TWO CELLPHONES: PYTHON ON ANDROID AND IOS

» https://github.com/pybee/Python-iOS-template

» https://github.com/pybee/Python-Android-template

» https://github.com/pybee/Python-macOS-template

A TALE OF TWO CELLPHONES: PYTHON ON ANDROID AND IOS

» http://pybee.org/project/projects/bridges/rubicon/

» http://pybee.org/project/projects/bridges/voc/

» http://pybee.org/project/projects/libraries/toga/

VECTOR SPACE MODELING ON MUSIC DATA

-- TIM SCHMEIER

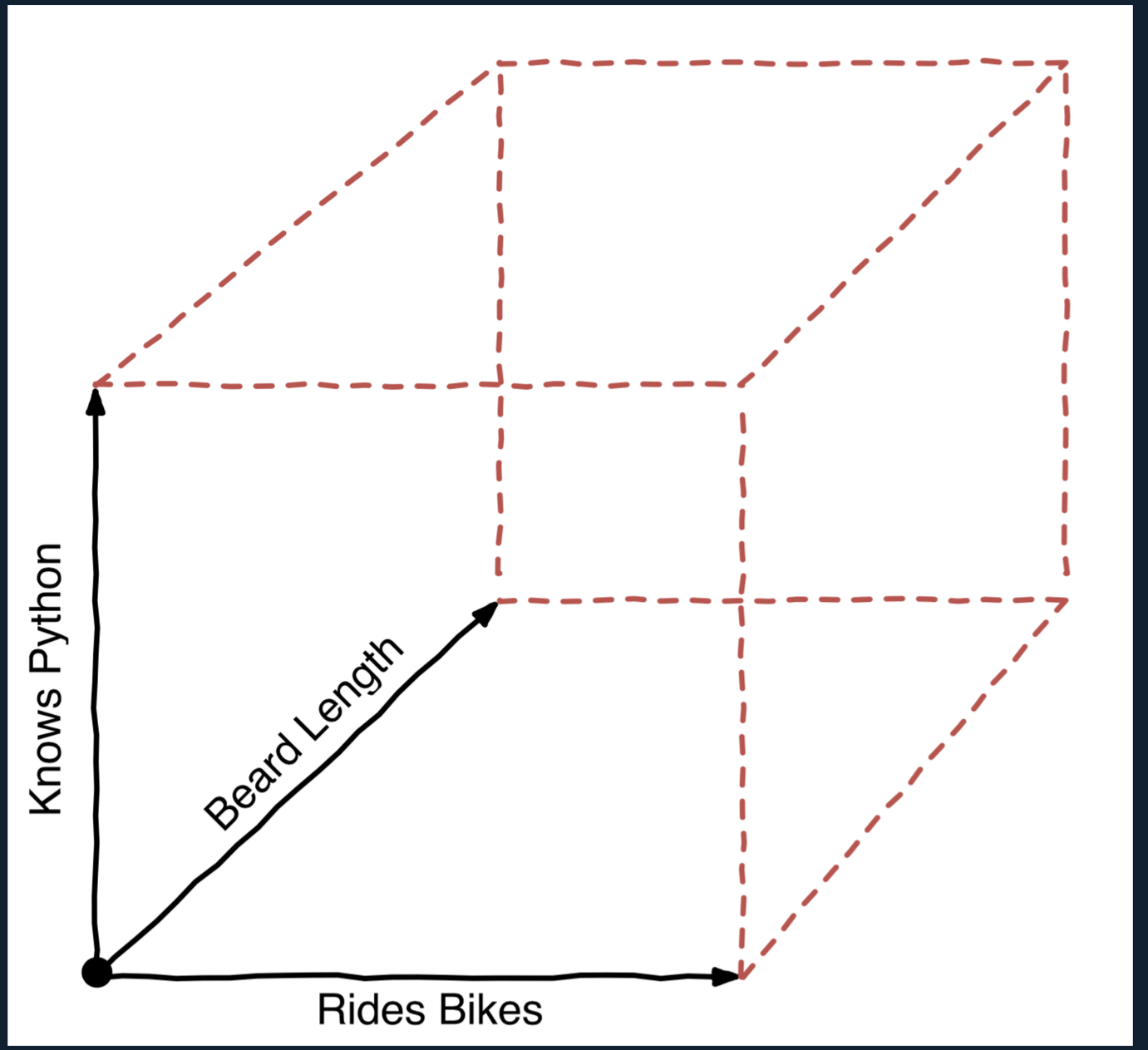
VECTOR SPACE MODELING ON MUSIC DATA

Developer Identifiers:

» rides bikes;

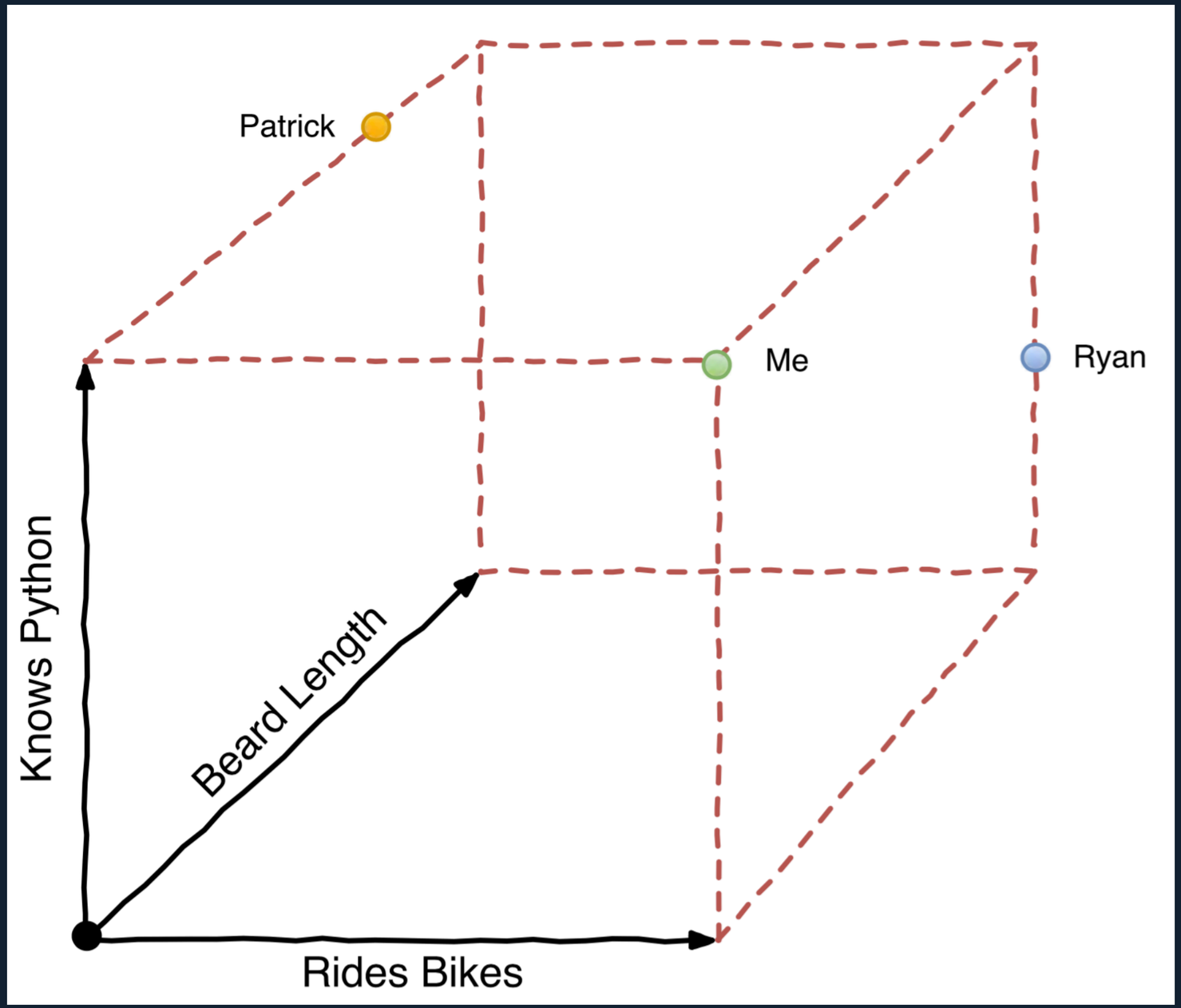
» knows Python;

» length of beard.


Knows Python
Beard Length
Rides Bikes

VECTOR SPACE MODELING ON MUSIC DATA

```
me = [1.0, 1.0, 0.0]
ryan = [1.0, 0.4, 1.0]
patrick = [0.0, 1.0, 0.8]
```


Patrick
Me
Ryan
Knows Python
Beard Length
Rides Bikes

VECTOR SPACE MODELING ON MUSIC DATA

```
>>> def combine(*vectors):
...     return [
...         sum(i)/len(i)
...         for i in zip(*vectors)
...     ]
...
>>> combine(me, ryan)
[1.0, 0.7, 0.5]
```

VECTOR SPACE MODELING ON MUSIC DATA

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2}.$$

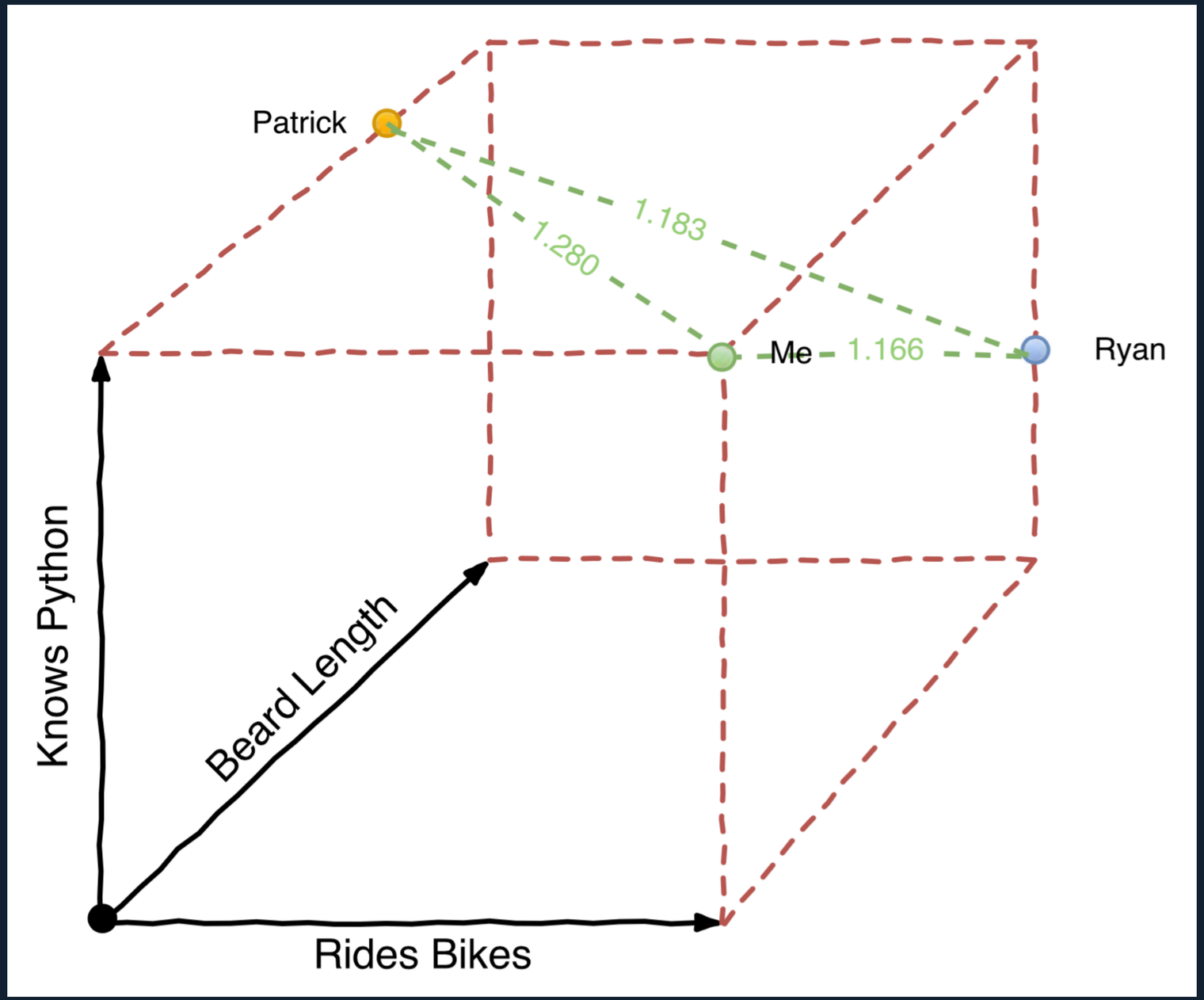
$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_i - q_i)^2 + \cdots + (p_n - q_n)^2}.$$

VECTOR SPACE MODELING ON MUSIC DATA

```
>>> def distance(a, b):
...     return sum(
...         (ai - bi)**2
...         for ai, bi in zip(a, b)
...     )**0.5
```

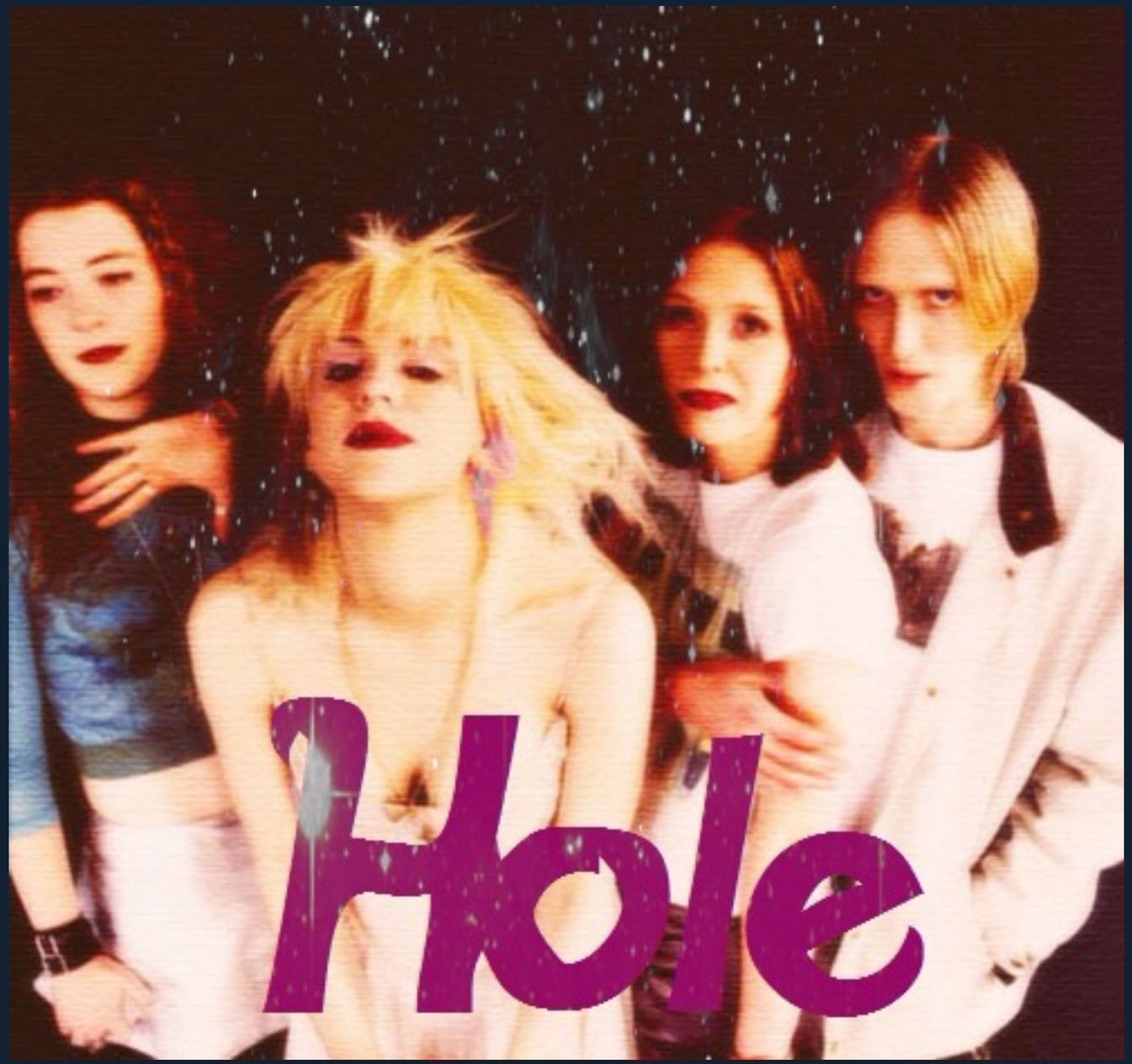
VECTOR SPACE MODELING ON MUSIC DATA

```
>>> distance(me, patrick)
1.28062484749
>>> distance(ryan, patrick)
1.18321595662
>>> distance(me, ryan)
1.16619037897
```


Patrick
1.183
1.280
Me
1.166
Ryan
Knows Python
Beard Length
Rides Bikes

VECTOR SPACE MODELING ON MUSIC DATA

```
nirvana - male_vocals + female_vocals
```


Hole

THANKS!