How to write programs that are right

- lessons from science for software engineering

Greg Detre
@gregdetre

28th September, 2013
BarCamp Tampa

if you want to chat through some of these ideas, I'm new to Tampa and looking to be part of the community, so drop me a line

WHO IS THIS FOR?

better to fail than be invisibly wrong

this is about writing programs where you really care that the answer is right. for example, if you're analysing data, and you're going to make a big decision or publicise the results, you really care that the analysis is right, or at least, that you understand it and it's doing what you think it's doing

you'd rather it crashes than give you the wrong answer

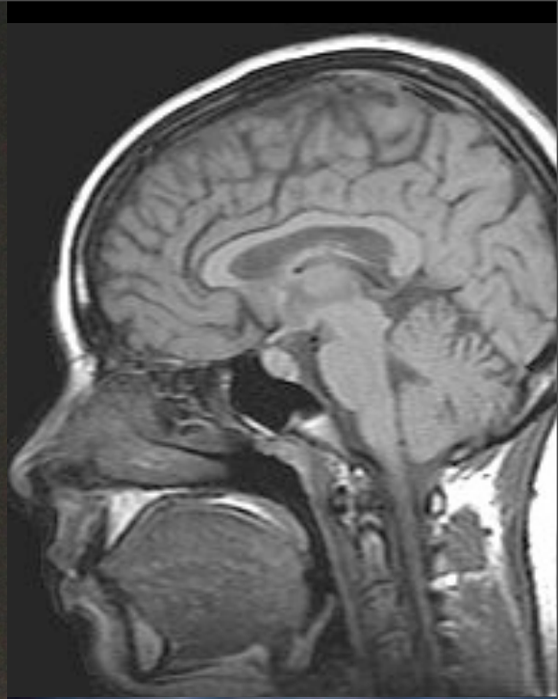
this is not about scalability either

you know what you want it to do

you don’t mind if it takes longer. though if 90% of your time is debugging, slowly & surely may even be faster in the long run

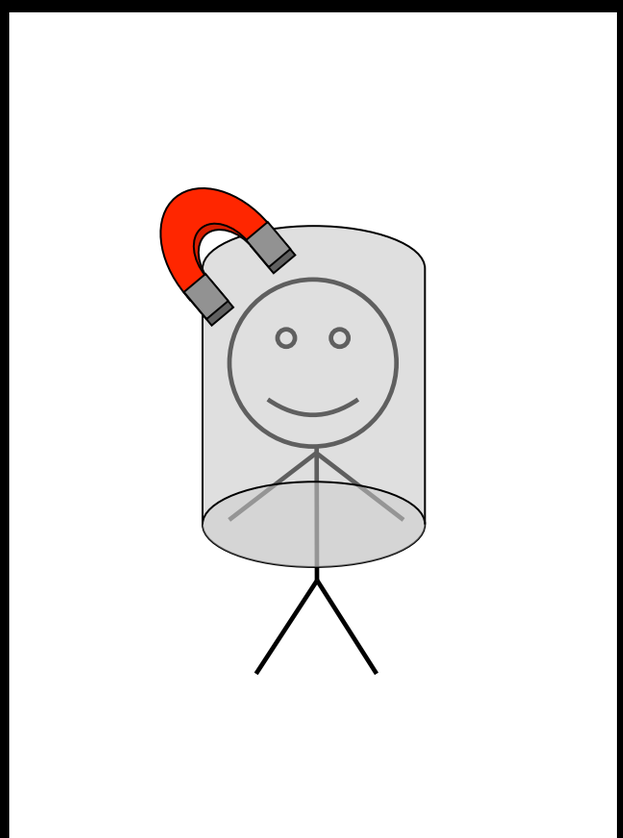
ME ME ME

Dr Greg Detre



I'm Greg Detre

I have a PhD in the neuroscience of human memory and forgetting from Princeton

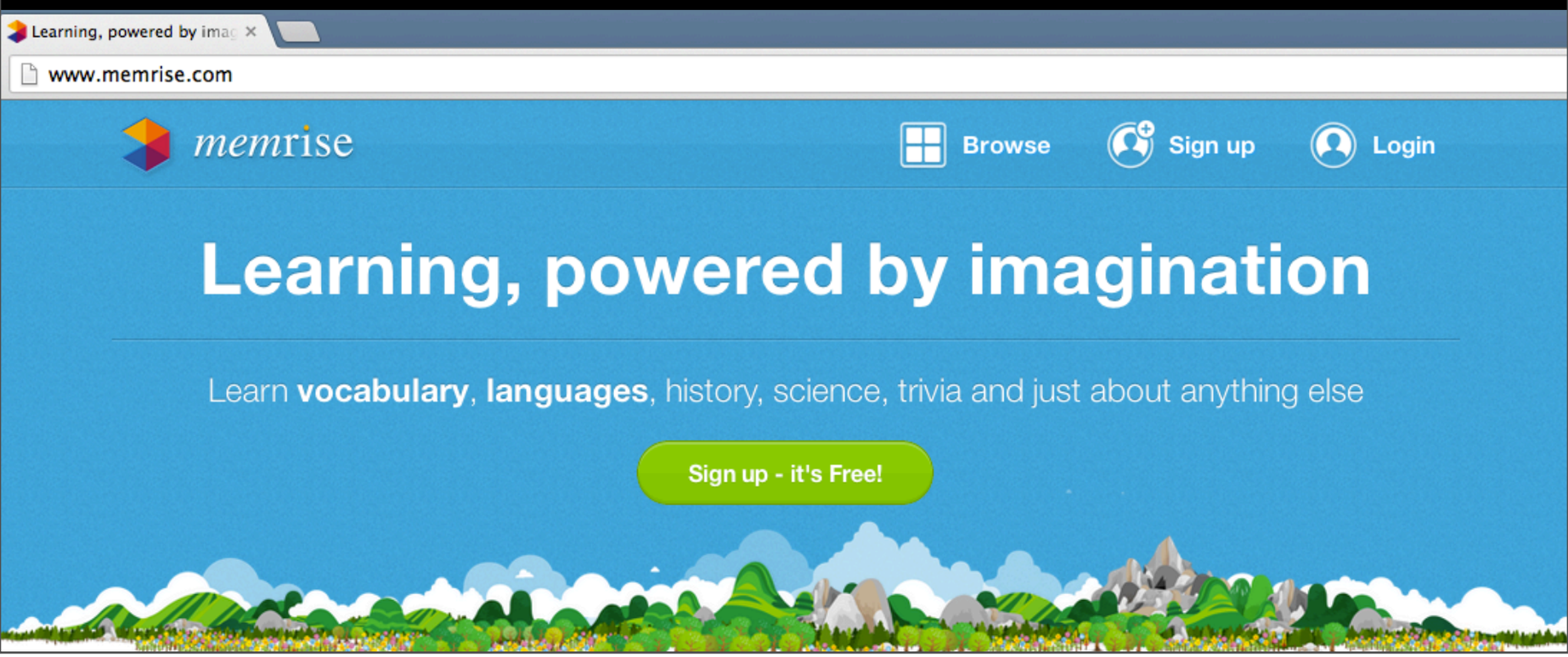


i spent my days scanning people's brains

including my own

it turned out to be smaller than I'd hoped

memrise


Learning, powered by imag
www.memrise.com
memrise
Browse
Sign up
Login
Learning, powered by imagination
Learn vocabulary, languages, history, science, trivia and just about anything else
Sign up - it's Free!

How to write programs that are right

- lessons from science for software engineering

@gregdetre, blog.gregdetre.co.uk

by the way, if you have a question, just make a noise like a wounded wildebeest and we can talk about them together

TOOLS

Version control

Git

@gregdetre, blog.gregdetre.co.uk

If you program but don't use version control, you're like a Michelin chef trying to cook over a bonfire

you absolutely should be

WRITE FOR A STRANGER

Imagine the person reading your code is hungry, tired, has a violent history, and knows where you live.

The person reading my code is usually ME (in which case, all 4 are true)

In a year's time, you will be a stranger to your present self.

Good comments

High-level goal: what is it trying to achieve?

What kinds of inputs does it expect? Examples

What kinds of outputs does it return? Examples

I tried another way, but ended up doing it this way because...

Explain unusual/complex bits

Comment before you write the code

Examples of bad comments:

Bad comments

```
% I'm so sorry about this next bit of code.
...

% Loop over 100 times
For x:1:100
```

Good coding practices

Break functions into bite-sized chunks

- each one a separate concept
- encapsulation

Don't repeat yourself

Variable naming

Etc

http://www.python.org/dev/peps/pep-0020/

https://github.com/thomasdavis/best-practices#programming-best-practices-tidbits

TESTING

Unit tests

If I call this function with input X, I expect to get output Y back

Helps you structure your code

And the tests serve as a kind of how-to guide

You're probably doing this anyway as you go

structuring
  if it's easy to test, it'll be easy to understand and refactor

probably doing this anyway as you go
  tests just reify that

Guard against new bugs in old code

Run your unit tests every time you run your analysis

Otherwise you might break something that used to work, and not realize it

Defensive coding

asserts and sanity checks

fail immediately if things are wrong

sanity checks
  e.g. confirm the dimensions, range of values, type of values

fail immediately
  that way you'll notice early on in time and near to the cause of the problem
rather than 2 weeks later and in a downstream part of the analysis

Eyeball it

examples of what this might help you see?

Run imagemat at large scale. You'll easily spot
- outliers
- stripes, e.g.
  if the scanner wasn't collecting for a while
  one row is all-zeros
  baseline differences before/after
  gradients/drift over time

HOSTILE WITNESS

@gregdetre, blog.gregdetre.co.uk

[cf the Cartesian demon]

i.e. your program/data are out to get you. ask leading questions and challenge it

If the examining attorney who called the witness finds that their testimony is antagonistic or contrary to the legal position of their client, the attorney may request that the judge declare the witness hostile

If the request is granted, the attorney may proceed to ask the witness leading questions. Leading questions either suggest the answer ("You saw my client sign the contract, correct?") or challenge (impeach) the witness' testimony.

(e.g. MovieLens/Netflix-style dataset)

movies

users

	About a boy	*Babel*	*Caddyshack*	...
anna	4		3	
bill		2	5	
charlie	1	2	1	
...				

3 teams

1. Analysis writers
2. White-box testers
3. Hostile witnesses

your data is a hostile witness

get a friend to be the hostile witness. ask them to try and create data that would trick the analysis

Write the analysis

Most popular movies?
Which movies are most similar to one another?
Which are the hardest movies to predict?
What subsets of movies tend to get rated together? Genres?
Recommendations
Who's the most accurate rater? Are some raters fake/spammers?

Write the analysis

Most popular movies?

Which movies are most similar to one another?

Which are the hardest movies to predict?

What subsets of movies tend to get rated together? Genres?

Recommendations

Who's the most accurate rater? Are some raters fake/spammers?

Most popular movies?
Which movies are most similar to one another?
Which are the hardest movies to predict?
What subsets of movies tend to get rated together? Genres?
Recommendations
Who's the most accurate rater? Are some raters fake/spammers?

Creating hostile datasets

try baseline increasing one movie by a big margin
try zeroing out an entire genre
try making all the movies belong to the same genre
try something subtle that won't be obvious visually, e.g. add a little randomness to each of the values (they're supposed to be ints/bools)
steganography

Creating hostile datasets

try baseline increasing one movie by a big margin

try zeroing out an entire genre

try making all the movies belong to the same genre

try something subtle that won't be obvious visually, e.g. add a little randomness to each of the values (they're supposed to be ints/bools)

steganography

try baseline increasing one movie by a big margin
try zeroing out an entire genre
try making all the movies belong to the same genre
try something subtle that won't be obvious visually, e.g. add a little randomness to each of the values (they're supposed to be ints/bools)
steganography

LOTS OF BABY STEPS

How do you eat an elephant?

Validate on small data, iterate quickly, scale up

- Define your metric
- Run it on small data - subsample (carefully)
- Show that you get better as you add more data

how do you eat an elephant? one bite at a time. start small, with a tiny subset of your data. that way, the algorithm runs quickly while you're prototyping

CANARIES IN THE DATACOALMINE

@gregdetre, blog.gregdetre.co.uk

Fake data

Generate data that looks exactly the way you expect

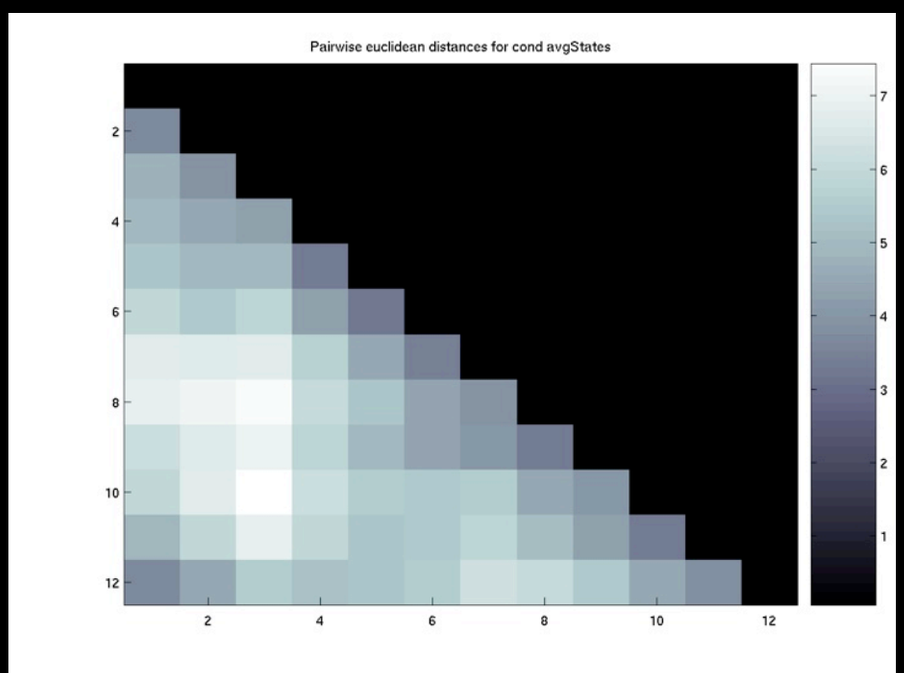
Can be hard to do, but often helps you think things through

Confirm that the output looks as it should

Useful for orienting audience in presentations

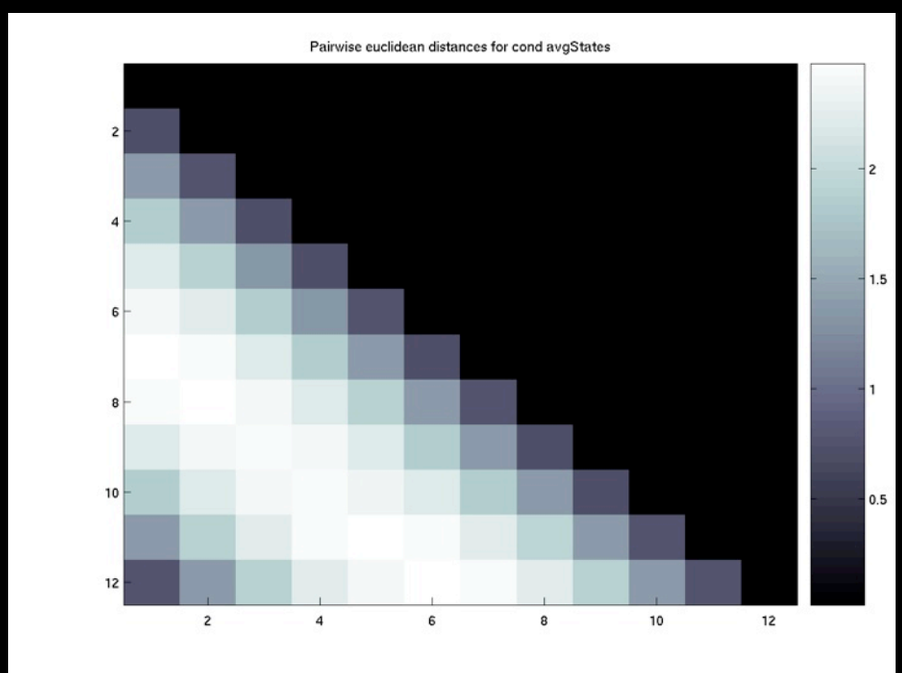
Set expectations with fake data

?


Pairwise euclidean distances for cond avgStates
2
4
6
8
10
12
2
4
6
8
10
12
7
6
5
4
3
2
1

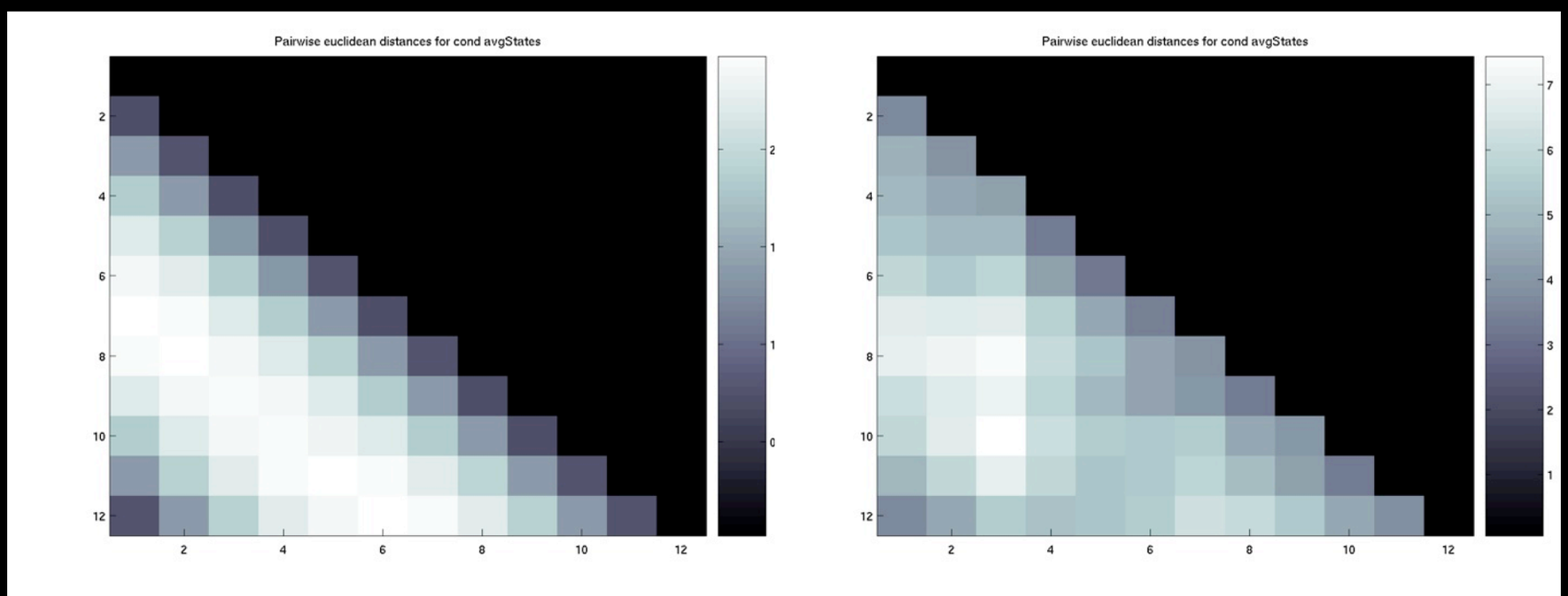
real data

it's supposed to look like this



synthetic data

... now it makes sense



synthetic

real

Nonsense/scrambled data

Set a trap. Feed your algorithm nonsense data. It had better tell you the results aren't significant!

Easy: shuffle regressors/labels or feed in random numbers as data

This. Will Save. Your. Bacon.

e.g. guard against peeking

Peeking in machine learning

REPRODUCIBILITY

Scripts

Version control everything non-data

- including config files

Commit often

Data

- Keep old versions of your files
  - Structured naming scheme
- Idempotent pipeline scripts
  - so you can effortlessly delete and regenerate intermediate steps

On idempotent:
- i.e. they don't mind (give the same results) if you run them multiple times in a row - automatically fill in the blanks as they go, so you can delete intermediate generated data - there are pipeline frameworks that I think are designed handle these kinds of dependencies for you (e.g. based on the old-school 'Make')

Results

101	100118d	allr_vr_dspk_norm_sm4 + mask: inters_groupana_091012a_t2.880	...sm4_z
102	100118e	allr_vr_dspk_norm_sm4 + mask: inters_groupana_091012a_t2.880	...sm4_z
103	100118f	allr_vr_dspk_norm_sm4 + mask: groupana_091012a_t2.880	...sm4_z
104	100118g	allr_vr_dspk_norm_sm4 + mask: groupana_091012a_t2.880	...sm4_z
105	100118h	allr_vr_dspk_norm_sm4 + mask: groupana_091012a_t2.880	...sm4_z
106	100118i	allr_vr_dspk_norm_sm4 + mask: groupana_091012a_t2.880	...sm4_z
107	100118j	allr_vr_dspk_norm_sm4 + mask: groupana_091012a_t2.880	...sm4_z
108	100118k	allr_vr_dspk_norm_sm4 + mask: groupana_091012a_t2.880	...sm4_z
109	100119a	allr_vr_dspk_norm_sm4 + mask: groupana_091012a_t2.101	...sm4_z
110	100119b	allr_vr_dspk_norm_sm4 + mask: groupana_091012a_t2.101	...sm4_z
111	100121a	allr_vr_dspk_norm_sm4 + mask: groupana_091012a_t2.880	...sm4_z

Structured file names will only get you so far

Spreadsheets are a step up, but hard to manipulate with programs

Use a database!

```
Result.objects \
    .filter(experiment__name='Shiny expt') \
    .filter(classifier__type='ridge',
            classifier__lambda=.2) \
    .filter(mask='PPA') \
    .values('pct_correct', 'running_time')
```

Open sourcing your code

It's good science

Ties you to the mast – standardize data formats, preserve backwards compatibility

Gets you into good habits

Write your code for a reader

Documentation

Package up requirements

Easier to collaborate

Gifts from smart strangers shower down from the sky

Glory!

THE END

@gregdetre, blog.gregdetre.co.uk

How to write programs that are right

- lessons from science for software engineering

Greg Detre
@gregdetre

28th September, 2013
BarCamp Tampa