A tutorial introduction to Make for reproducible research

Iain Davies and Daniel Nuest and Stephen J Eglen

August 24, 2020

1 Introduction

2 Introduction

Why make is useful. Originally designed into 1970s for the efficient compilation of programs. Gradually adopted to other situations.

Note the workflow here

1. Generate 2. Analyse 3. Summarise

If Generating the data takes 10 hours, and analysing takes only 2 minutes, you don't want to rerun whole pipeline to regenerate data.

3 Example problem

3.1 Estimating pi

how to estimate pi with a dartboard. **Iain: can you write this section, explain how we solve the problem in the three steps above.**

3.2 Traditional approach - write a batch file

```
#!/bin/sh
./throw-darts inputs.dat > darts.txt
./count-inside darts.txt > inside.out
./estimate inside.out > pi.est
./draw-figure darts.txt inside.out pi.est output.pdf
```

3.3 Version 1 (R makefile)

```
.PHONY: clean all

all: darts.pdf

darts.xy: inputs.dat throw-darts
	./throw-darts inputs.dat > darts.xy

inside.out: darts.xy count-inside
	./count-inside darts.xy > inside.out

pi.est: inside.out estimate
	./estimate inside.out > pi.est



darts.pdf: darts.xy inside.out pi.est draw-figure
	./draw-figure darts.xy inside.out pi.est darts.pdf


clean:
	rm -f darts.xy inside.out pi.est darts.pdf



## This depends on https://github.com/lindenb/makefile2graph

graph.pdf: Makefile
	make -Bnd | makefile2graph | dot -Tpdf > graph.pdf
```

TODO Anatomy of a rule.

what is a rule? target? dependency? how it does it decide whether to re-execute the rule (timestamps of files).

3.4 Running it (for real)

```
cd v1-R
make

## make[1]: Entering directory '/home/stephen/papers/2020/make-tutorial/v1-R'
## ./throw-darts inputs.dat > darts.xy
## ./count-inside darts.xy > inside.out
## ./estimate inside.out > pi.est
## ./draw-figure darts.xy inside.out pi.est darts.pdf
## make[1]: Leaving directory '/home/stephen/papers/2020/make-tutorial/v1-R'
```

This should generate a new pdf, 'darts.pdf'. If we then delete darts.pdf by accident, when we run 'make' again, it will not need to run all steps of the analysis, as the intermediate files are still present.

```
cd v1-R
rm darts.pdf
make

## make[1]: Entering directory '/home/stephen/papers/2020/make-tutorial/v1-R'
## ./draw-figure darts.xy inside.out pi.est darts.pdf
## make[1]: Leaving directory '/home/stephen/papers/2020/make-tutorial/v1-R'
```

3.5 DAG

One nice thing about using a Makefile is usually you can visualize the dependency graph. e.g. Figure 2. Input files are clearly shown in green, and everything else in red can be regenerated.

(This figure is generated by analysing the structure of the output from the make program, thanks to a program from `makefile2graph`)

3.6 Explain what PHONY targets are

Two common PHONY targets are the ”all” and ”clean” targets. ”make all” by convention is a request to regenerate all the files in the directory, rather than to generate a file called ”all”. Likewise, ”clean” is a convention to remove all the files that can be deleted.

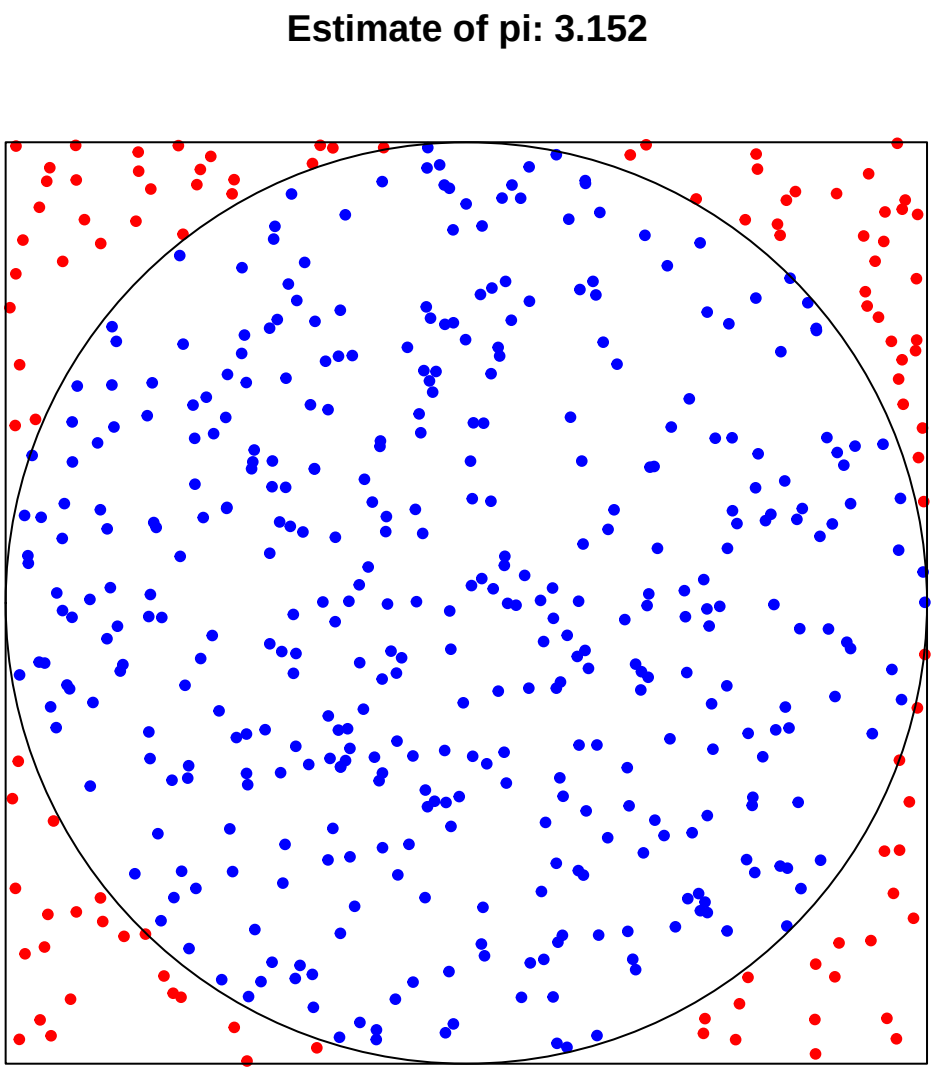

Figure 1: Example output file, darts.pdf, created by "make" command. Blue (or red) points are those that were determined to be inside (or outside) the circle. The estimate of pi, given in the title, was then given as 4*d/n, where d is the number of darts inside the circle and n is the total number of darts thrown.

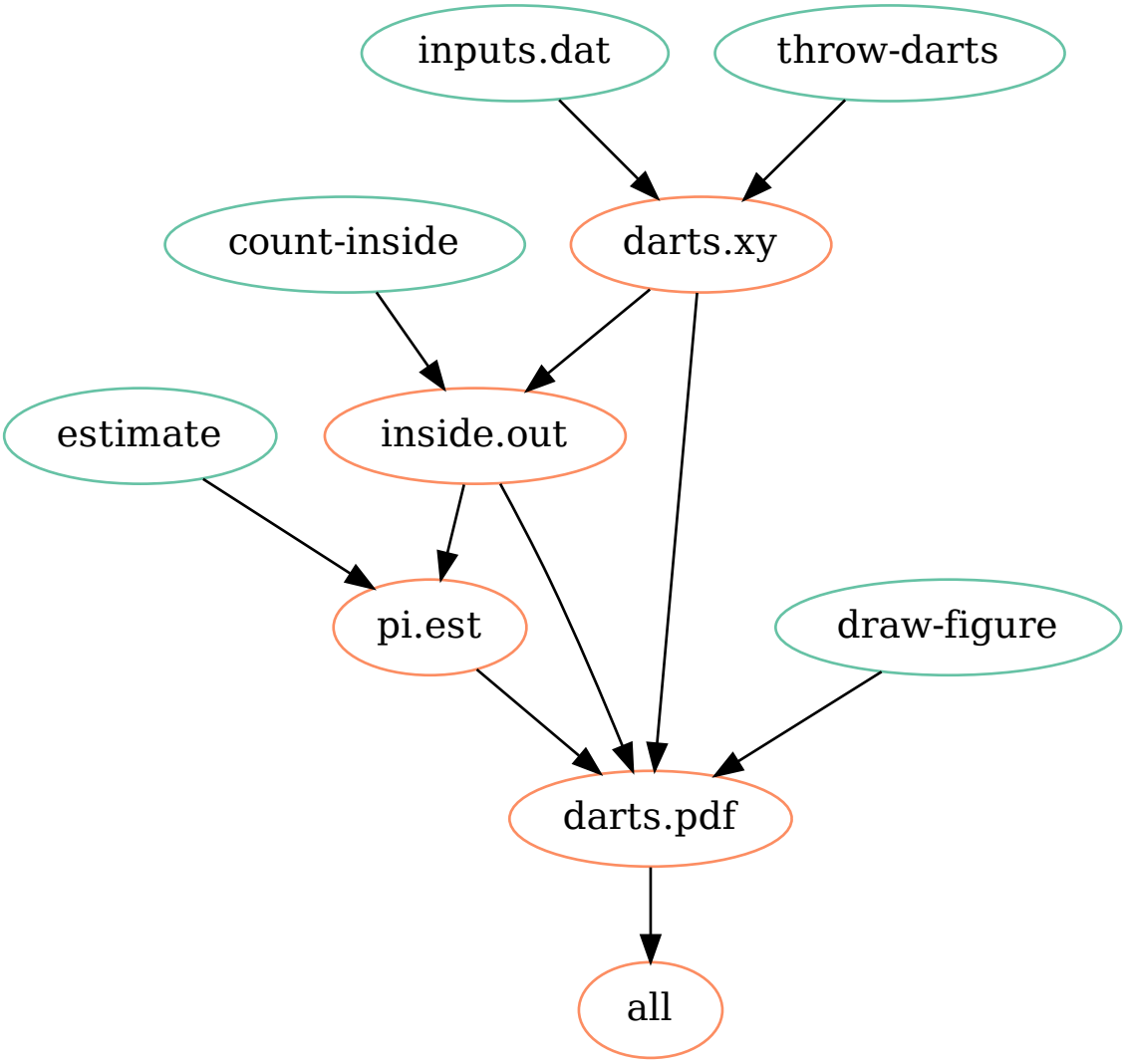

Figure 2: DAG for Makefile version 1. green ellipses correspond to those files that are up to date and do not need to be regenerated; red ellipsis are those files that need to be remade.

4 ersion 2 – generate lots of simulations

In version 2, we want to highlight how to make rules more flexible. Example application here would be generating lots of simulation runs, rather than just one.

```
# Tell Emacs this is a -*- Makefile -*-

.PHONY: all clean

REPEATS := $(shell seq 1 15)

TARGETS := $(patsubst %, pi-%.est, $(REPEATS))

all: $(TARGETS)

darts-%.xy: inputs.dat
	./throw-darts inputs.dat > $@

inside-%.out: darts-%.xy
	./count-inside $^ > $@


pi-%.est: inside-%.out
	./estimate $^ > $@


# https://blog.jgc.org/2015/04/the-one-line-you-should-add-to-every.html
print-%: ; @echo $*=$($*)


clean:
	rm -f $(TARGETS)
```

Try to generate B=15 samples, and then show a histogram of the distribution of the B estimates of pi.

```
cd v1-R
make -f Makefile2
cat pi-*est

## make[1]: Entering directory '/home/stephen/papers/2020/make-tutorial/v1-R'
## ./throw-darts inputs.dat > darts-1.xy
## ./count-inside darts-1.xy > inside-1.out
## ./estimate inside-1.out > pi-1.est
## ./throw-darts inputs.dat > darts-2.xy
## ./count-inside darts-2.xy > inside-2.out
## ./estimate inside-2.out > pi-2.est
## ./throw-darts inputs.dat > darts-3.xy
## ./count-inside darts-3.xy > inside-3.out
## ./estimate inside-3.out > pi-3.est
## ./throw-darts inputs.dat > darts-4.xy
## ./count-inside darts-4.xy > inside-4.out
## ./estimate inside-4.out > pi-4.est
## ./estimate inside-5.out > pi-5.est
```

```
## ./throw-darts inputs.dat > darts-6.xy
## ./count-inside darts-6.xy > inside-6.out
## ./estimate inside-6.out > pi-6.est
## ./throw-darts inputs.dat > darts-7.xy
## ./count-inside darts-7.xy > inside-7.out
## ./estimate inside-7.out > pi-7.est
## ./throw-darts inputs.dat > darts-8.xy
## ./count-inside darts-8.xy > inside-8.out
## ./estimate inside-8.out > pi-8.est
## ./throw-darts inputs.dat > darts-9.xy
## ./count-inside darts-9.xy > inside-9.out
## ./estimate inside-9.out > pi-9.est
## ./throw-darts inputs.dat > darts-10.xy
## ./count-inside darts-10.xy > inside-10.out
## ./estimate inside-10.out > pi-10.est
## ./throw-darts inputs.dat > darts-11.xy
## ./count-inside darts-11.xy > inside-11.out
## ./estimate inside-11.out > pi-11.est
## ./throw-darts inputs.dat > darts-12.xy
## ./count-inside darts-12.xy > inside-12.out
## ./estimate inside-12.out > pi-12.est
## ./throw-darts inputs.dat > darts-13.xy
## ./count-inside darts-13.xy > inside-13.out
## ./estimate inside-13.out > pi-13.est
## ./throw-darts inputs.dat > darts-14.xy
## ./count-inside darts-14.xy > inside-14.out
## ./estimate inside-14.out > pi-14.est
## ./throw-darts inputs.dat > darts-15.xy
## ./count-inside darts-15.xy > inside-15.out
## ./estimate inside-15.out > pi-15.est
## rm inside-7.out darts-9.xy inside-11.out darts-7.xy inside-10.out inside-6.out darts-6.
## make[1]: Leaving directory '/home/stephen/papers/2020/make-tutorial/v1-R'
## 3.08
## 3.184
## 3.2
## 3.104
## 3.112
## 3.2
## 3.152
## 3.192
## 3.12
## 3.184
## 3.152
## 3.264
## 3.176
## 3.264
## 3.144
```

Describe make -j8 to run this in parallel.
NOTE: intermediate files are not kept.
Perhaps show a 4x3 grid of 12 simulations?

5 What next

Further reading (books); other programs that build upon idea of make (snakemake, drake).
`https://www.frontiersin.org/articles/10.3389/fninf.2016.00002/full`
Portable make code: `https://github.com/markpiffer/gmtt`

5.1 Acknowledgements

Mozilla CODECHECK for funding