HW01 — STAT/CS 287  
NAME: Michael Arnold   
DATE: 09/14/18

## P1.1.a

## Sets differ from dictionaries in that sets only have unique, immutable elements, whereas dictionaries have unique immutable elements (called keys), which are paired with some value. Obviously, they are also different types.

## P1.1.b

Sets are similar to dictionaries in that sets and dictionaries are primarily store unique, immutable objects, and both use a hash table, so they are unordered. Superficially, they both use curly braces.

## P1.1.c

You could just use a dictionary to replace a set, and just give all the keys a dummy value 0 or something. The dictionary would still enforce the immutable, unique restriction on the keys, that a set would impose on its elements.

## P1.2

## P1.3

set 0 is {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

set 1 is {0, 2, 4, 6, 8, 10, 12, 14, 16, 18}

set 2 is {1, 4, 7, 10, 13, 16, 19, 22, 25, 28}

The similarity of set1 and set2 is 0.333

The similarity of set2 and set3 is 0.176

The similarity of set1 and set3 is 0.176

## P2

# runs 0.2 0.4 0.6 0.8

length

------------------------------

1 133 143 97 35

2 25 59 61 24

3 6 25 34 22

4 2 12 23 21

5 0 0 13 16

6 0 1 6 16

7 0 0 2 4

8 0 1 1 5

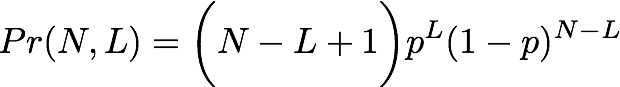
9 0 0 2 3

10 0 0 1 7

## ave 2.09 3.98 5.75 7.93

## P2 Bonus

The probability that we start flipping coins and get L heads in a row is just

We can ask a more complicated question, of given a number of rolls N, what is the probability we get a run of heads of length L ≤ N, which is given by the binomial distribution

since we need to multiply the probability of getting L heads with the probability of getting N-L tails, times the number of ways we can pick the position of the first heads in the run, while still having enough space left to complete it, (N-L+1) ways.

## P3.1

Caching is important so we don’t have to download the file every time we run the script. If we cache we can just open the file right away.

## P3.2

word count

--------------------

the 8186

and 4993

of 4124

to 3543

a 2974

in 2638

it 2013

his 2005

i 1917

that 1904

he 1833

was 1764

you 1455

with 1351

had 1297

as 1148

at 1045

her 1038

for 972

him 965

on 932

not 860

is 842

be 780

have 742

said 660

were 658

but 654

my 653

mr 620

this 588

so 582

by 578

all 571

there 567

they 564

no 548

from 529

me 522

if 471

she 459

out 446

one 438

been 435

or 434

when 434

which 409

them 393

who 375

what 371

an 349

your 345

would 341

lorry 336

are 333

into 319

their 318

do 315

up 309

will 295

upon 291

could 282

defarge 280

man 279

little 265

its 261

more 261

any 261

time 260

now 256

then 253

hand 247

before 232

down 232

miss 232

know 230

some 229

again 227

am 225

himself 219

very 217

than 216

two 211

good 209

like 198

see 198

other 193

looked 193

long 192

never 192

doctor 192

madame 191

face 187

these 187

old 186

made 185

here 184

night 182

much 181

way 180

## 

## P3 Bonus

We can use the tool Counter from the library collections, to quickly count the number of each word. Helpful people on the internet also suggested using an itertools tool chain, to map the words through, but unless there was multiprocessing this is more a memory saver, than a way to make the code quicker, and doesn’t really make it simpler to implement.

## P4.1

It is important to be able to reuse code that you wrote to save time and energy, but also to build up more complex behavior from functions that accomplish tasks simple enough to understand.

## P4.2

Yes, if we let q1 equal q2, the probability of the second coin giving heads or tails will not be affected by the outcome of the first coin.

## P4.3

-------------------------------

for p = 0.5 q1 = 0.75 q2 = 0.25

-------------------------------

event count

-------------------------------

HH 368

HT 121

TH 113

TT 398

-------------------------------

for p = 0.2 q1 = 0.75 q2 = 0.25

-------------------------------

event count

-------------------------------

HH 143

HT 59

TH 207

TT 591

## P4.4

Given a list of events for a random variable the easiest way to estimate the probability of an event is to add up the number of times an event occurs and divide by the total number of measurements. Can this method miss events? If by miss, we mean is it possible to not observe an event with happens with small probability, yes it is possible to not observe any instance of an event. But we will not miss any of the events which were observed using this method.

## P4.5

-------------------------------

for p = 0.5 q1 = 0.75 q2 = 0.25

-------------------------------

event joint prob

-------------------------------

P(H,H) 0.356

P(H,T) 0.111

P(T,H) 0.114

P(T,T) 0.419

-------------------------------

for p = 0.2 q1 = 0.75 q2 = 0.25

-------------------------------

event joint prob

-------------------------------

P(H,H) 0.131

P(H,T) 0.059

P(T,H) 0.191

P(T,T) 0.619

## P4 Bonus 1

If two random variable are independent, then their joint probability is factorable, mathematically written as

So if we can calculate P(X) and P(Y) independently by marginalizing over the others, then we can compare to see if P(X,Y) is really separable or not. I did this by taking the absolute value of the difference between the joint probability and the product of the marginalized probabilities for the first and second coin, and comparing in each case. If the difference was smaller than some tolerate for all pairs of joint probabilities, then the function prints independent, but if the difference is too large, they are declared dependent.

## P4 Bonus 2

I’m reasonably confident at around n = 50 if the coin is very dependent. To investigate this, I computed an estimate of joint probability and the product of the marginalized probabilities corresponding to that joint probability at each coin flip. If this pair for each unique event, which is colored the same approaches its partner, then the coins are independent, and if they remain distinct, the coins are dependent since

