Untitled

LuchaoQi

February 28, 2019

Q1

1a

Compute the edit distance:

```
Alpha:  EALERMFLSFPTTKTYFPHFDLSHGSAQVK
Beta:   EALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKVK
```

```
str1 = 'EALERMFLSFPTTKTYFPHFDLSHGSAQVK'
str2 = 'EALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKVK'
m = len(str1)
n = len(str2)
dp = [[0 for x in range(n+1)] for x in range(m+1)]
for i in range(m+1):
  for j in range(n+1):
    if i == 0:
      dp[i][j] = j
    elif j == 0:
      dp[i][j] = i
    elif str1[i-1] == str2[j-1]:
      dp[i][j] = dp[i-1][j-1]
    else:
      dp[i][j] = 1 + min(dp[i][j-1], dp[i-1][j],dp[i-1][j-1])
for i in range(len(dp)):
  print(dp[i])
```

Results:

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36]
[1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35]
[2, 1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34]
[3, 2, 1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33]
[4, 3, 2, 1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32]
[5, 4, 3, 2, 2, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32]
[6, 5, 4, 3, 3, 2, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 25, 26, 27, 28, 29, 30, 31]
[7, 6, 5, 4, 4, 3, 3, 3, 4, 5, 6, 7, 8, 9, 10, 11, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31]
[8, 7, 6, 5, 5, 4, 3, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 12, 13, 14, 15, 16, 17, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30]
[9, 8, 7, 6, 6, 5, 4, 4, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 13, 13, 14, 15, 16, 17, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29]
[10, 9, 8, 7, 7, 6, 5, 5, 5, 5, 6, 7, 8, 9, 10, 11, 11, 12, 13, 14, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29]
[11, 10, 9, 8, 8, 7, 6, 6, 6, 6, 6, 6, 7, 8, 9, 10, 11, 12, 13, 14, 14, 14, 15, 16, 17, 18, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28]
[12, 11, 10, 9, 9, 8, 7, 7, 7, 7, 7, 7, 7, 7, 8, 9, 10, 11, 12, 13, 14, 15, 15, 16, 17, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28]
[13, 12, 11, 10, 10, 9, 8, 8, 8, 8, 8, 8, 8, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 16, 17, 17, 18, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28]
[14, 13, 12, 11, 11, 10, 9, 9, 9, 9, 9, 9, 9, 8, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 17, 18, 18, 19, 20, 21, 22, 23, 24, 25, 25, 26, 27]
[15, 14, 13, 12, 12, 11, 10, 10, 10, 10, 10, 10, 10, 9, 9, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 26, 27]
[16, 15, 14, 13, 13, 12, 11, 11, 11, 11, 10, 11, 11, 10, 10, 10, 10, 11, 12, 13, 14, 15, 16, 17, 18, 18, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 27]
[17, 16, 15, 14, 14, 13, 12, 12, 12, 12, 11, 11, 12, 11, 11, 11, 10, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28]
[18, 17, 16, 15, 15, 14, 13, 13, 13, 13, 12, 11, 12, 12, 12, 12, 11, 11, 11, 12, 13, 14, 15, 16, 17, 18, 18, 19, 20, 21, 22, 23, 24, 24, 25, 26, 27]
[19, 18, 17, 16, 16, 15, 14, 14, 14, 14, 13, 12, 12, 13, 13, 13, 12, 12, 12, 12, 13, 14, 15, 16, 17, 18, 19, 19, 20, 21, 22, 23, 24, 25, 25, 26, 27]
[20, 19, 18, 17, 17, 16, 15, 15, 15, 15, 14, 13, 13, 13, 14, 14, 13, 12, 13, 13, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 26, 27]
[21, 20, 19, 18, 18, 17, 16, 16, 16, 16, 15, 14, 14, 14, 14, 15, 14, 13, 13, 14, 13, 13, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27]
[22, 21, 20, 19, 19, 18, 17, 16, 17, 17, 16, 15, 15, 15, 15, 15, 15, 14, 14, 14, 14, 14, 14, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26]
[23, 22, 21, 20, 20, 19, 18, 17, 17, 18, 17, 16, 16, 16, 16, 16, 16, 15, 15, 14, 15, 15, 15, 14, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25]
[24, 23, 22, 21, 21, 20, 19, 18, 18, 18, 18, 17, 17, 17, 17, 17, 17, 16, 16, 15, 15, 16, 16, 15, 14, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25]
[25, 24, 23, 22, 21, 21, 20, 19, 19, 19, 19, 18, 18, 18, 18, 18, 18, 17, 17, 16, 16, 15, 16, 16, 15, 15, 15, 16, 17, 18, 19, 19, 20, 21, 22, 23, 24]
[26, 25, 24, 23, 22, 22, 21, 20, 20, 20, 20, 19, 19, 19, 19, 19, 19, 18, 18, 17, 17, 16, 16, 17, 16, 16, 16, 16, 17, 18, 19, 20, 20, 21, 22, 23, 24]
[27, 26, 25, 24, 23, 23, 22, 21, 21, 21, 21, 20, 20, 20, 20, 20, 20, 19, 19, 18, 18, 17, 17, 17, 17, 17, 17, 17, 16, 17, 18, 19, 20, 21, 22, 23, 24]
[28, 27, 26, 25, 24, 24, 23, 22, 22, 22, 22, 21, 21, 21, 20, 21, 21, 20, 20, 19, 19, 18, 18, 18, 18, 18, 18, 18, 17, 17, 18, 19, 20, 21, 22, 23, 24]
[29, 28, 27, 26, 25, 25, 24, 23, 22, 22, 23, 22, 22, 22, 21, 21, 22, 21, 21, 20, 20, 19, 19, 19, 19, 19, 19, 19, 18, 17, 18, 19, 20, 21, 22, 22, 23]
[30, 29, 28, 27, 26, 26, 25, 24, 23, 23, 23, 23, 23, 23, 22, 22, 22, 22, 22, 21, 21, 20, 20, 20, 20, 20, 20, 20, 19, 18, 18, 19, 20, 21, 21, 22, 22]
```

So the edit distance is 22.
Sequences:

EALERMFLSFPTTKTYFPHF - DLS - - - - - HGSAQVK
EALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKVK

Q2

2a

Commands:

```
bowtie2-build chr22.fa chr22
bowtie2 -x chr22 -1 sample/pair.1.fq -2 sample/pair.2.fq > sample.sam
samtools flagstat sample.sam
```

Results:

```
leon@ubuntu:~/hw/applied_genomics/hw4$ samtools flagstat sample.sam
3367692 + 0 in total (QC-passed reads + QC-failed reads)
0 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
3365708 + 0 mapped (99.94% : N/A)
3367692 + 0 paired in sequencing
1683846 + 0 read1
1683846 + 0 read2
1698350 + 0 properly paired (50.43% : N/A)
3363862 + 0 with itself and mate mapped
1846 + 0 singletons (0.05% : N/A)
0 + 0 with mate mapped to a different chr
0 + 0 with mate mapped to a different chr (mapQ>=5)
```

So 3365708 reads align to the chr22 reference. 1984 reads did not align. 1846 reads had AKA singletons.

2b

Commands:

```
samtools view -c -f 16 sample.sam
```

We can get 1678219 reads mapped to the reverse strand.

2c

Commands:

```
samtools sort sample.sam > sample.bam
samtools index sample.bam
freebayes -f chr22.fa sample.bam > sample.vcf
bcftools filter -e "QUAL<20" sample.vcf > filtered.vcf
bcftools stats filtered.vcf > stats.txt
vim stats.txt
```

Results:

```
leon@ubuntu: ~/hw/applied_genomics/hw4
File  Edit  View  Search  Terminal  Help
# number of multiallelic sites     .. number of rows with multiple alternate alleles
# number of multiallelic SNP sites .. number of rows with multiple alternate alleles, all SNPs
#
# Note that rows containing multiple types will be counted multiple times, in each
# counter. For example, a row with a SNP and an indel increments both the SNP and
# the indel counter.
#
# SN    [2]id   [3]key  [4]value
SN      0       number of samples:      1
SN      0       number of records:      16536
SN      0       number of no-ALTs:      0
SN      0       number of SNPs: 14965
SN      0       number of MNPs: 152
SN      0       number of indels:       1403
SN      0       number of others:       18
SN      0       number of multiallelic sites:   6
SN      0       number of multiallelic SNP sites:       4
# TSTV, transitions/transversions:
# TSTV  [2]id   [3]ts   [4]tv   [5]ts/tv        [6]ts (1st ALT) [7]tv (1st ALT) [8]ts/tv (1st ALT)
TSTV    0       5342    9612    0.56    5339    9610    0.56
# SiS, Singleton stats:
# SiS   [2]id   [3]allele count [4]number of SNPs       [5]number of transitions       [6]number of transversions      [7]
  number of indels     [8]repeat-consistent    [9]repeat-inconsistent  [10]not applicable
SiS     0       1       12990   4472    8518    1231    0       0       1231
                                                                                                  25,1          1%
```

So there are 14965 single nucleotides with QUAL>20 and 1403 indels.

```
# IDD   [2]id   [3]length (deletions negative)  [4]count
IDD     0       -6      1
IDD     0       -5      4
IDD     0       -4      10
IDD     0       -3      60
IDD     0       -2      122
IDD     0       -1      510
IDD     0       1       483
IDD     0       2       144
IDD     0       3       49
IDD     0       4       12
IDD     0       5       6
IDD     0       9       2
```

Of the indels, there are 707 deletions, 696 insertions.

Q3

3a

Python:

```
import pysam
chr22 = pysam.FastaFile("chr22.fa")
seq = chr22.fetch("chr22",21000000,22000000)

sub = open("sub.fa","w")
sub.write(">chr22:21000000,22000000 \n")
sub.write(seq + '\n')
sub.close()

reads = open("reads.fa","w")
for i in range(len(seq)-35+1):
    number  = ">"+str(i+1)#pos in chr22
    subseq  = seq[i:i+35]
    reads.write(number + "\n")
    reads.write(subseq + "\n")
reads.close()
```

Commands:

```
python3 reads.py #get the ref and pseudo reads
bowtie2-build sub.fa sub #build up the index
bowtie2 -x sub -f -U reads.fa -S sub.sam
```

Results:

```
(base) leon@ubuntu:~/hw/applied_genomics/hw4$ bowtie2 -x sub -f -U reads.fa -S sub.fa
999966 reads; of these:
  999966 (100.00%) were unpaired; of these:
    0 (0.00%) aligned 0 times
    595713 (59.57%) aligned exactly 1 time
    404253 (40.43%) aligned >1 times
100.00% overall alignment rate
```

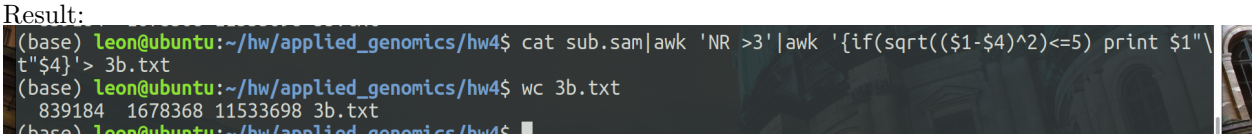
So 404253 aligned more than one times. The alignment rate is 100%. 0 read did not align.

3b

Commands:

```
samtools sub.sam -o sub.sam
cat sub.sam|awk 'NR >3'|awk '{if(sqrt(($1-$4)^2)<=5) print $1"\t"$4}'> 3b.txt
wc 3b.txt
```

Result:



So 839184 reads aligned correctly(within 5bp)

3c

Commands:

```
cat sub.sam|awk 'NR>3 {print $1"\t"$5}' >3c.txt
```

Python:

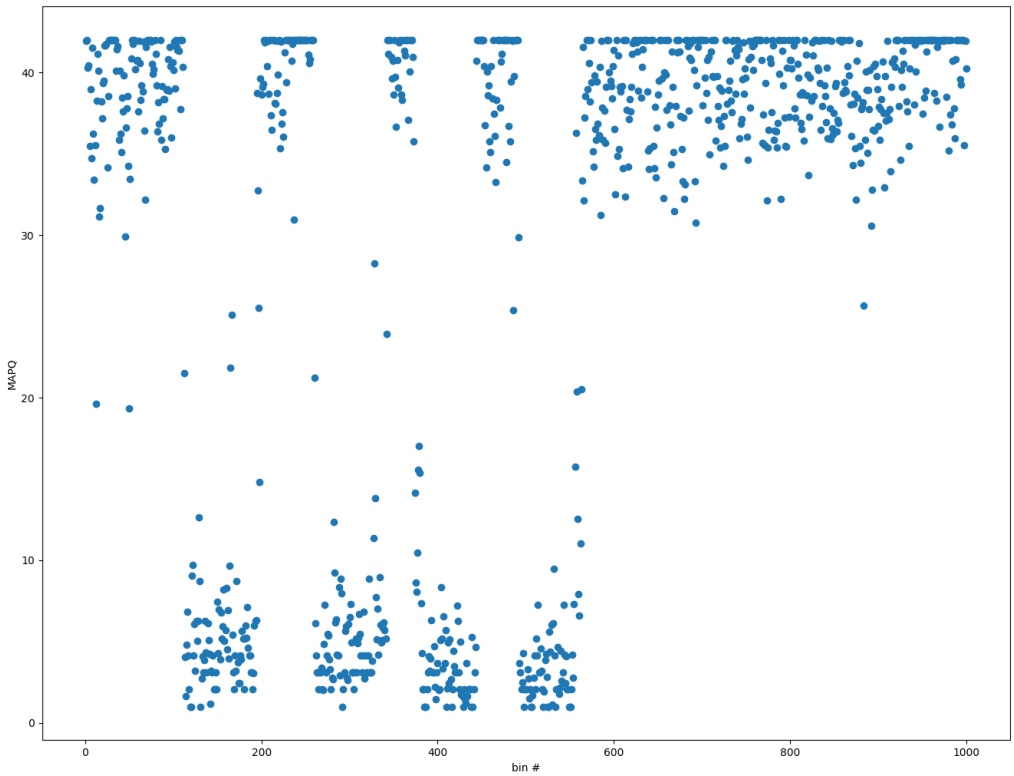
```
import matplotlib.pyplot as plt

f = open('3c.txt','r')
dic = {}
for lines in f:
    line =  lines.strip().rstrip('\n').split('\t')
    a = int(line[0])//1000 + 1
    if a in dic:
        dic[a] = dic[a] + int(line[1])
    else:
        dic[a] = int(line[1])
for i in dic:
    dic[i] = dic[i]/1000

f.close()

plt.scatter(dic.keys(),dic.values())
plt.xlabel('bin #')
plt.ylabel('MAPQ')
plt.show()
```

Results:


MAPQ
bin #
0
10
20
30
40
0
200
400
600
800
1000

3d

Commands:

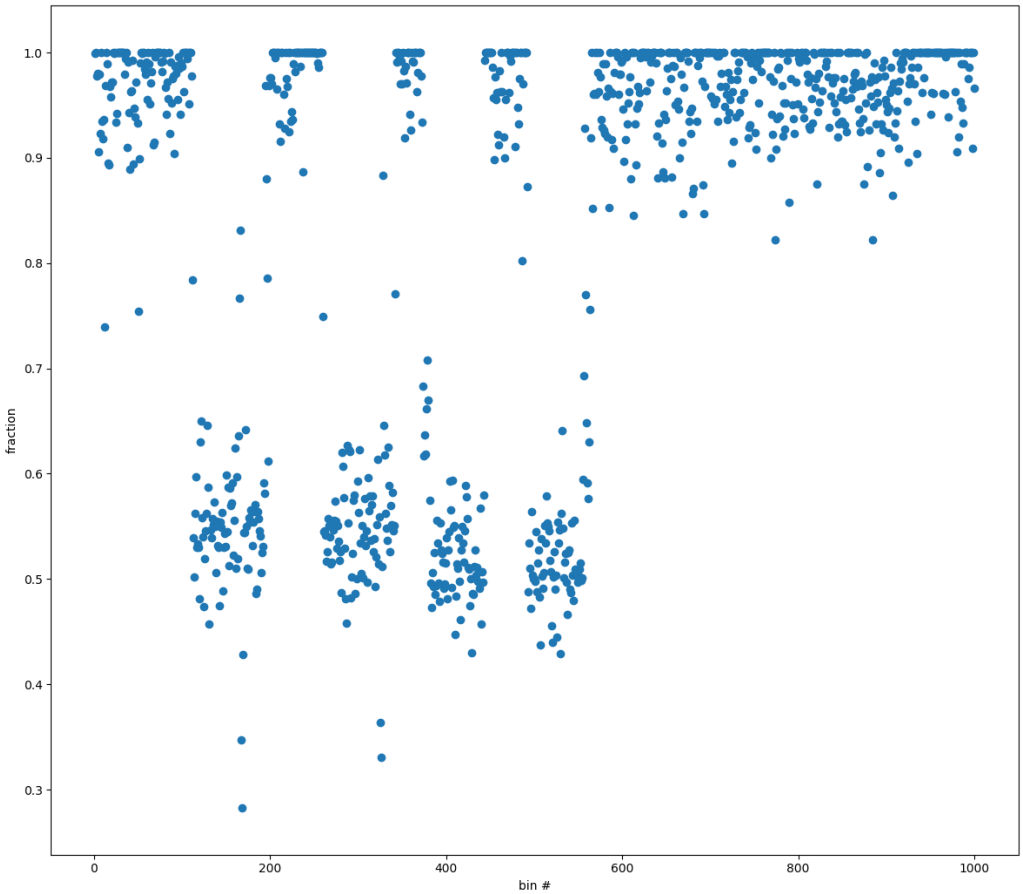
```
cat sub.sam|awk 'NR >3'|awk '{print $1"\t"$4}'> 3d.txt
```

Python:

```
import matplotlib.pyplot as plt
f = open('3d.txt','r')
dic = {}
for lines in f:
    line =  lines.strip().rstrip('\n').split('\t')
    a = int(line[0])//1000 + 1
    if a in dic:
                dic[a] = dic[a] + (line[0] == line[1])
    else:
                dic[a] = (line[0]==line[1])
for i in dic:
    dic[i] = dic[i]/1000

f.close()
plt.scatter(dic.keys(),dic.values())
plt.xlabel('bin #')
plt.ylabel('fraction')
plt.show()
```

Results:

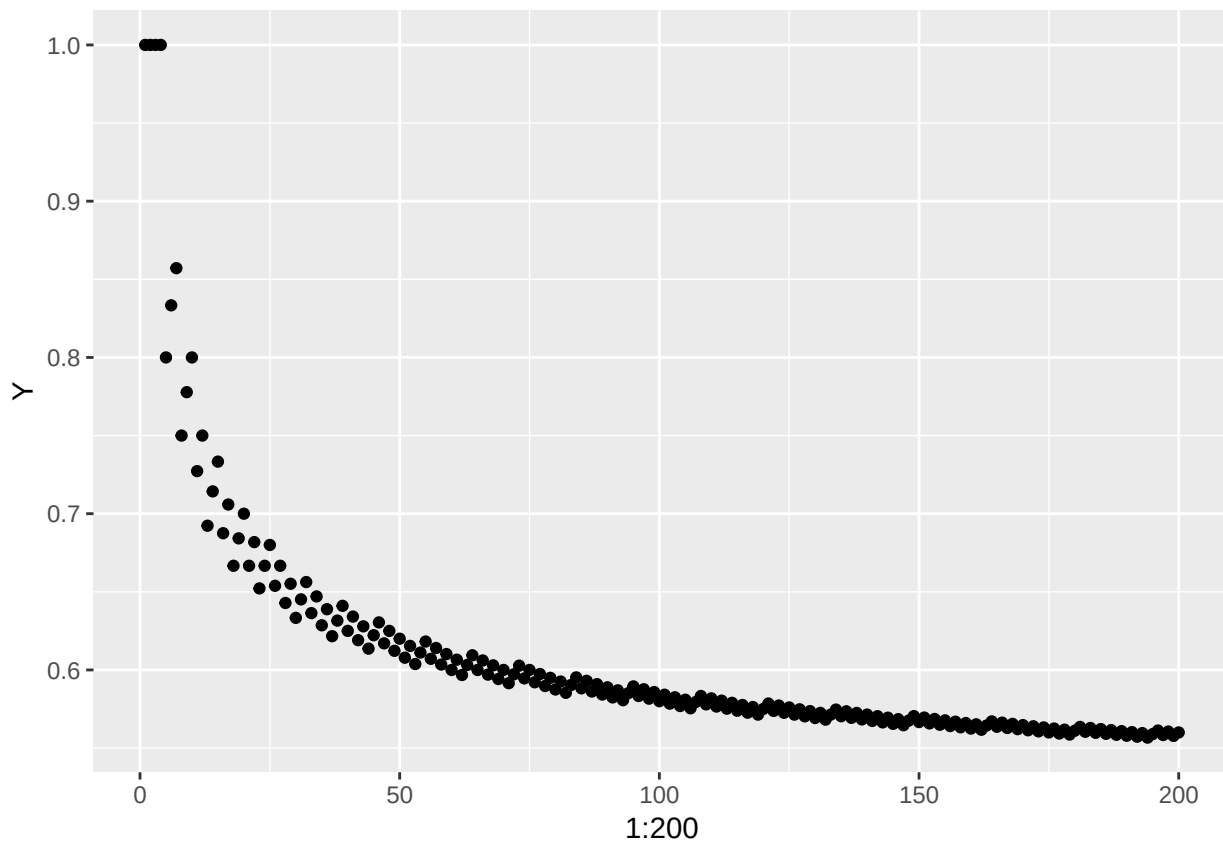


Looks similar to the plot in 3c.

Q4

4a

```
X = 1:200
Y = vector()
for (size in X)
  {
    Y[size] = qbinom(0.95,size,0.5)/size
}
library(ggplot2)
ggplot(data = as.data.frame(Y),aes(x=1:200,y=Y))+geom_point()
```



4b

The plot seems to approach 0.5 because the larger the size is, according to the binormial distribution(p=0.5), the closer 95% quantile is to the 50% of the size.