STATISTICS PRACTICAL 1 SOLUTIONS

(1) 100 students each perform an experiment to estimate a parameter μ, and each one independently computes a 99% confidence interval for μ. What is the probability that there will be at least 3 students whose confidence intervals do not include μ? (Hint: Use the binomial distribution or the Poisson distribution.)

The probability of a confidence interval not including μ is 0.01 and there are 100 of them, so the expected number X of "bad confidence intervals" is 1. Assuming they are independent, we may model the number X as Poisson with parameter 1. The probability mass function is

$$\mathbb{P}\{X = k\} = e^{-1} \cdot \frac{1^k}{k!}.$$

Thus

$$\begin{aligned}\mathbb{P}\{X \geq 3\} &= 1 - \mathbb{P}\{X \leq 2\} \\ &= 1 - e^{-1}\left(1 + 1 + \frac{1}{2}\right) \\ &= 0.080.\end{aligned}$$

We could also compute this with the `R` command `1-ppois(2,1)`. The more exact solution would use the binomial distribution with parameters $(100, 0.01)$. In `R` this would be `1-pbinom(2,100,.01)`, yielding 0.079.

(2) Suppose we have a new IQ test, which we wish to compare to a standard IQ test, to see whether the scores on the new test have the same average as the scores on the old test. We take 200 subjects, and pick 120 at random to take the new test; the other 80 take the old test. The results are as follows:

Test	# subjects	Mean	SD
Old	80	101	11
New	120	98	8

We wish to do a Z test at the 0.01 level to determine whether the means on the tests could be the same.

(a) State the test hypotheses.

Null: Means are equal. Alternative: Means are different.

(b) State any assumptions you have to make.

Only that the tests are independent. Normality does not need to be assumed.

(c) Calculate the test statistic, compute the p-value.

The standard error for the difference is

$$\sqrt{\frac{11^2}{80} + \frac{8^2}{120}} = 1.43.$$

The test statistic is

$$Z = \frac{101 - 98}{1.43} = 2.10.$$

The p-value is `2-2*pnorm(2.1)=0.036`.

(d) Conclude.

We are testing at the 0.01 level. Since the p-value is higher, we do not reject the null hypothesis. The difference between the two test means is not significant at the .01 level.

(3) Suppose you have 4 independent observations from a normal distribution with unknown mean and unknown variance. You estimate the mean and SD from the data, and use them to compute the Student t statistic $T = (\bar{X} - 1)/(\hat{\sigma}/\sqrt{n})$. You report it to someone, who mistakenly believes that the variance is known, and thus performs a Z test for the hypothesis that the mean of the distribution is 0. What will the impact be on the statistical conclusions? What if the error goes in the other direction (that is, you compute from a known variance, but someone believes you have delivered a T statistic)?

(a) If the person is performing hypothesis tests at level 0.05, calculate the size of the error in each case.

(b) Carry out some simulations to demonstrate this.

The person who thinks your T statistic is a Z statistic will be overconfident about rejecting the null hypothesis. Her type I error probability will be higher than the official level. The person who thinks your Z statistic is a T statistic will have a type I error probability that is lower than she thinks, meaning that the type II error probability is raised, hence the power of the test is lowered.

(a) If the person is performing two-tailed hypothesis tests at level 0.05, calculate the size of the error in each case.

In the first case, she will set the critical value at 1.96. The true type I error probability will be `2*(1-pt(1.96,3))=.145`. In the second case she will set the critical value at `qt(.975,3)=3.18`, and so will have a Type I error probability of `2*(1-pnorm(3.18))=.0014`.

(b) Carry out some simulations to demonstrate this.

```
> n=1000 # Number of T statistics
> X=matrix(rnorm(n*4),n,4)
> barX=apply(X,1,mean)
> sigma=apply(X,1,sd)
> T=barX/(sigma/sqrt(4))
> reject=mean(abs(T)>1.96) # Fraction rejected
> reject
[1] 0.148
```

.

(4) Count the number of Adenines, Guanines, Cytosines and Thymines (As, Gs, Cs and Ts). Then there is the dinucleotide composition, that is, the number of AAs, AGs etc. (along one strand). There further is the trinucleotide composition, that is, the number of AAAs, AAGs etc.

```
> table(ecp)
ecp
A       C       G       T
1142742 1180091 1177437 1141382
>
> table(rbind(ecp[-n],ecp[-1]))

> table(ecp[-n],ecp[-1])

A      C      G      T
A 338006 256773 238013 309950
C 325327 271821 346793 236149
G 267384 384102 270252 255699
T 212024 267395 322379 339584

> table(ecp[-c(n-1,n)],ecp[-c(1,n)],ecp[-c(1,2)])
, ,  = A


A      C      G      T
```

```
A 108964  58664  56659  63721
C  76654  86491  70971  26770
G  83530  96071  56222  52688
T  68858  84101  83532  68845

, ,  = C


A      C      G      T
A  82616  74935  80909  86523
C  66782  47807 115734  42746
G  54764  93028  92189  54247
T  52611  56051  95270  83879

, ,  = G


A      C      G      T
A  63405  73288  50653  76282
C 104850  87076  86904 102957
G  42503 114670  47515  66142
T  27254  71759  85180  76998

, ,  = T


A      C      G      T
A  83021  49886  49792  83424
C  77041  50447  73184  63676
G  86587  80333  74326  82622
T  63301  55483  58397 109862
```

.

(a) If you had to assign a probability to observing an A at a stated position on the *E. coli* genome, what figure would you use and why? Under what assumptions does this seem appropriate?
Assuming A's are evenly spread through the genome, $1142742/4641652 = 0.246193$.

(b) Your are told there is an A at a position along the *E. coli* genome. What probability would you assign to it being followed by a G, and why? The fraction of A's that are followed by G's is $238013/1142742 = 0.2082824$.

(c) Does the *E. coli* composition data suggest that the event we observe a G at one site is independent (in some suitable sense) of the previous two bases? Explain fully, illustrating with appropriate data.

```
> load(url('http://steinsaltz.me.uk/DTC/ecoli.rda'))
> n=length(ecp)
> table(ecp)
ecp
A       C       G       T
1142742 1180091 1177437 1141382
> table(ecp[-c(n-1,n)],ecp[-c(1,n)],ecp[-c(1,2)])
, ,  = A


A      C      G      T
A 108964  58664  56659  63721
C  76654  86491  70971  26770
G  83530  96071  56222  52688
```

```
T  68858  84101  83532  68845

, ,  = C


A      C      G      T
A  82616  74935  80909  86523
C  66782  47807 115734  42746
G  54764  93028  92189  54247
T  52611  56051  95270  83879

, ,  = G


A      C      G      T
A  63405  73288  50653  76282
C 104850  87076  86904 102957
G  42503 114670  47515  66142
T  27254  71759  85180  76998

, ,  = T


A      C      G      T
A  83021  49886  49792  83424
C  77041  50447  73184  63676
G  86587  80333  74326  82622
T  63301  55483  58397 109862

> ft=table(ecp[-c(n-1,n)],ecp[-c(1,n)],ecp[-c(1,2)])
> dim(ft)
[1] 4 4 4
> for(i in 1:4){
+   ft[,,i]=ft[,,i]/sum(ft[,,i])
+ }
> ft
, ,  = A


A           C           G           T
A 0.09535319 0.05133622 0.04958166 0.05576154
C 0.06707907 0.07568732 0.06210594 0.02342613
G 0.07309618 0.08407067 0.04919925 0.04610669
T 0.06025687 0.07359585 0.07309793 0.06024550

, ,  = C


A           C           G           T
A 0.07000816 0.06349934 0.06856166 0.07331892
C 0.05659055 0.04051128 0.09807210 0.03622263
G 0.04640659 0.07883121 0.07812025 0.04596849
T 0.04458216 0.04749718 0.08073106 0.07107842

, ,  = G


A           C           G           T
A 0.05385006 0.06224372 0.04301975 0.06478654
C 0.08904943 0.07395391 0.07380783 0.08744170
G 0.03609793 0.09738958 0.04035463 0.05617460
```

```
T 0.02314691  0.06094514  0.07234363  0.06539464

,  ,   = T


A           C           G           T
A 0.07273726  0.04370666  0.04362431  0.07309034
C 0.06749800  0.04419817  0.06411876  0.05578851
G 0.07586154  0.07038222  0.06511930  0.07238768
T 0.05545996  0.04861037  0.05116341  0.09625349

> ft=table(ecp[-c(n-1,n)],ecp[-c(1,n)],ecp[-c(1,2)])
> t2/sum(t2)
 [1] 0.02905434 0.06078956 0.08237753 0.06569789 0.05374411 0.07180936
     0.07972736 0.14208318 0.05883309 0.02881407
[11] 0.03239641 0.03375130 0.06220407 0.05160699 0.03784669 0.05850069
     0.05076335
> t2=table(ecp[-n],ecp[-1])
> t2/sum(t2)

A           C           G           T
A 0.07282021  0.05531932  0.05127766  0.06677581
C 0.07008864  0.05856127  0.07471329  0.05087608
G 0.05760536  0.08275116  0.05822325  0.05508794
T 0.04567857  0.05760773  0.06945352  0.07316018
```

.

At least to the naked eye it appears that the proportions of the 16 possible pairs preceding a G (the first array) are almost the same as the lower array, which has the overall proportions of those pairs.

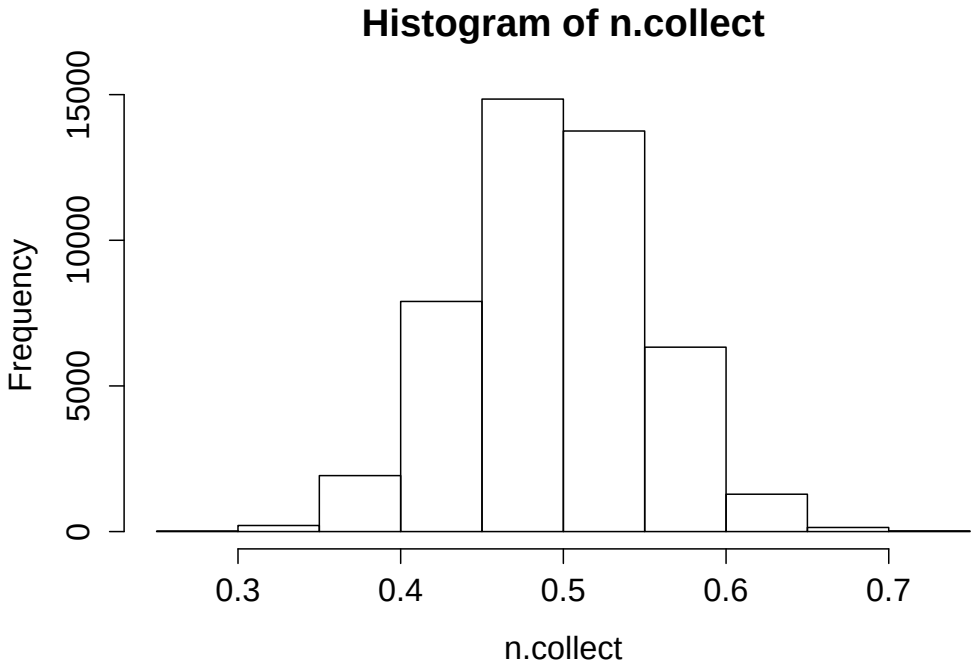
(d) **Purine counts.**

Divide up the data into about 46,400 blocks of 100 base pairs (bp). Count the number of purines (i.e. A or G). Do the same for the 4,640 blocks of 1,000 bp, and 464 blocks of 10,000 bp.

(i) For each set of counts, calculate the mean and standard deviation of the number of purines per block, and draw histograms of these numbers.

```
> k=100
> n=length(ecp)%/%k -1
> n.collect=NULL
> for (i in 0:n){
+   a=ecp[i*k+(1:100)]
+   n.collect=c(n.collect,sum(a=='A'|a=='G'))
+ }
> sd(n.collect)
 [1] 5.678099
> mean(n.collect)
 [1] 49.986
> hist(n.collect)
```

.



(ii) Compare the results of (i) across the different block sizes and comment.

```
> k=500
> n=length(ecp)%/%k -1
> n.collect2=NULL
> for (i in 0:n){
+    a=ecp[i*k+(1:100)]
+    n.collect2=c(n.collect2,sum(a=='A'|a=='G'))
+ }
>
> sd(n.collect2)
[1] 5.673332
> mean(n.collect2)
[1] 50.01077
```

.

(iii) For each block size, calculate the fraction of the counts within 1, 2 and 3 standard deviations of the mean.

```
> sapply(1:3,function(i) sum(abs(n.collect-mean(n.collect))/sd(n.
    collect)<i))/length(n.collect)
[1] 0.6647492 0.9588719 0.9978456
> #Block size 500
> sapply(1:3,function(i) sum(abs(n.collect2-mean(n.collect2))/sd(
    n.collect2)<i))/length(n.collect2)
[1] 0.6656253 0.9599267 0.9974146
```

.

(iv) Repeat (i), (ii) and (iii) for *proportions* (rather than counts) of purines in each block.

Just change `sum` to `mean` in defining `n.collect` and `n.collect2`.

(e) Compute counts of TATAAT in blocks of 5,000 bp. Assuming that these counts follow a Poisson distribution, estimate the parameter of this distribution and obtain an estimate

of the standard error of your parameter estimate. This can be done by either a formula or by simulation.

```
> count.pattern=function(source=ecp,target=c('T','A','T','A','A','T')){
+   k=length(target)
+   n=length(source)-k+1
+   sum(sapply(1:n,function(i) prod(source[i:(i+k-1)]==target)))
+ }
> g=sapply(1:u,function(i) count.pattern(ecp[((i-1)*n+1):(i*n)]))
> table(g)
g
0   1   2   3   4   5   6   7
611 213  56  25  15   5   2   1
> mean(g) # is the estimate of the Poisson parameter
[1] 0.5431034
# This is the proportion of each count number among the blocks
> round(table(g)/length(g),3)
g
0     1     2     3     4     5     6     7
0.658 0.230 0.060 0.027 0.016 0.005 0.002 0.001
> round(dpois(0:7,.5431034),3)
# This is what the proportions of each count would be if Poisson
[1] 0.581 0.316 0.086 0.016 0.002 0.000 0.000 0.000
```

.

If the distribution of TATAAT were Poisson — that is, if they were scattered among the 5000-base blocks like independent trials with small probability of success at every point, the Poisson parameter would be the mean number of "successes" per block, which is 0.5431. In fact, over the 928 blocks the distribution of different count numbers is very different from Poisson.