Statistics for Astronomers
Solutions to Homework #2

Prof. Sundar Srinivasan

October 6, 2020

1. (a) Both S_1 and S_2 are computed by summing together random deviates. They are, therefore, also random numbers.
   (i) We implicitly assume that the random number generator is able to generate independent random deviates. Therefore, S_1 and S_2 must be independent since they are derived from independent runs of the generator. (ii) Due to the Central Limit Theorem, the sum of iid random numbers also derives from the same distribution as the original deviates. Therefore, S_1 and S_2 are identically distributed.

   (b) The mean of the sums should be 10 times the population mean of the original random deviates. Therefore, $\mu_{\rm mean} = 35$. From Bienaymé's Identity, the variance of a sum of independent random variables is the sum of their variances. Therefore, $\mathrm{Var}(S_1) = \mathrm{Var}(S_2) = 10 \times \mathrm{Var}(X_i) = 0.25$.

2. The total passenger load is the sum of 100 normal random numbers drawn from $\mathcal{N}(\mu, \sigma^2)$. The sum is therefore distributed according to $\mathcal{N}(100\mu, 100\sigma^2) = \mathcal{N}(100\mu, (10\sigma)^2)$ from Bienaymé's Identity and the Central Limit Theorem[1].
   The problem asks for the probability that this sum is greater than the passenger load limit of 8450 kg; that is, $P(\sum X > 8450) = 1 - P(\sum X \leq 8450)$. The second term on the RHS is nothing but the CDF for the Normal distribution with mean 100μ and variance $100\sigma^2$.
   We can use the `scipy.stats.norm` module to calculate the necessary probability, setting $\mu = 80$ kg and $\sigma = 15$ kg:

   ```
   from scipy.stats import norm
   print(1 - norm.cdf(8450, loc = 100 * 80, scale = 10 * 15))
   ```

   The result is close to 0.00135.

   Note that the passenger load limit is exactly 3 standard deviations higher than the mean passenger load, so we could also have solved the problem using the Standard Normal:

   ```
   print(1 - norm.cdf(3))
   ```

3. We assume independence here – *i. e.*, that the occurrence of one binary does not affect the occurrence of any other binary. In such a case, the detection or non-detection of a single binary is the result of a Bernoulli trial. Since we are interested in the **sum** of many such trials, the problem requires use of the Binomial distribution.

[1] In fact, as long as the masses are independent and identically distributed, the actual distribution from which they are drawn is immaterial – their sum will still be normally distributed according to the Central Limit Theorem.

(a) We first estimate the probability p of finding a single binary in the dataset: $\hat{p} = 0.6$ (given). If X is the total binary count, we are asked to find $P(X = 7)$ (7 binaries $\equiv$ 3 non-binaries) in a random sample of $N = 10$ stars. The requisite probability is

$P(X = 7) = \binom{10}{7} p^7 (1-p)^3 \approx 0.215.$

This result can also be obtained using the `scipy.stats.binom` package:

```
from scipy.stats import binom
print(binom.pmf(7, 10, 0.6))
```

(b) The event "at least 2 non-binaries" is the same as "at most N-2 binaries". The problem states that the probability associated with this event is 0.99. That is, $P(X \leq N-2) \geq 0.99$. The term on the LHS is just the CDF of the Binomial distribution. We are given the smallest value that the CDF can have, and are asked to find the argument $N-2$ such that this is true. Using the `cdf` method of the `scipy.stats.binom` module,

```
from scipy.stats import binom
prob = 0.0; i = 2
while prob < 0.99:
    prob = binom.cdf(i-2, i, 0.6)
    i += 1
print(i-1, prob) #because i was updated when exiting the loop
```

The result is $N = 14$.

The problem can also be solved by root-finding methods. The probability associated with the complementary event "at least N-1 binaries" is $0.01 = \alpha$ (say). If X is the number of binaries found in a sample of N stars,

$$P(X \geq N-1) = Np^{N-1}(1-p) + p^N = p^N\left(1 + N\frac{1-p}{p}\right) \leq \alpha.$$

While we can guess the value of N quite easily by substitution, I'll describe a couple of methods here that will be applicable to more general problems. Since $p < 1$, p^N rapidly decreases. In order to reduce the dynamic range, let's work with the logarithmic version of the above inequality:

$$N \log p + \log\left(1 + N\frac{1-p}{p}\right) - \log\alpha \leq 0, \text{ with } p = 0.6, \alpha = 0.01. \quad (1)$$

We can solve Equation (1) for N in many ways. I'll discuss three possibilities below.
(a) Precise solution via a root finder: the `scipy.optimize.root_scalar` method is suitable for this problem, using the bisection method (`method = "bisect"`). Please find this implementation in the sample python script here . Solution: $N \geq 14$.
(b) Approximate solution by finding the minimum absolute value: since we are interested in an integer, we can evaluate the function on a relatively low-resolution grid of x values and find the integer such that the absolute value of the function is closest to zero for $x < N$. This is implemented in the sample script. Solution: $N \geq 14$.
(c) Approximate solution by visual estimation from plot: the sample script also generates Fig. (1). Solution: $N \geq 14$.

4. According to the problem, the rate **per square degree** of occurrence of quasars in the BOSS survey area is $87822/3275 \approx 26.82$. Therefore, $\lambda = 26.82 \text{ deg}^{-2} \times 1 \text{ deg}^2 = 26.82$.

(a) We can assume a Poisson distribution if (a) the occurrence of each quasar is independent of the occurrence of other quasars and (b) the average rate of occurrence is independent of the region

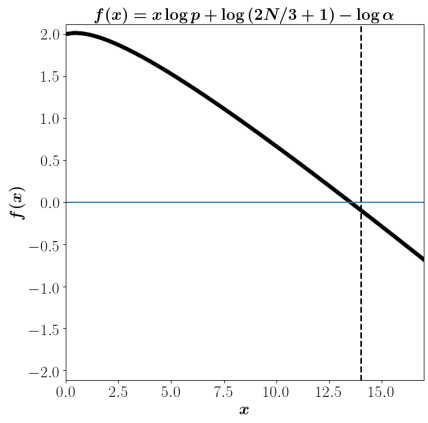

Figure 1: The function from Equation (1) (thick curve) hits zero (blue line) at $x > 13$. The smallest integer higher than this value is also indicated (dashed line).

of the survey area picked. If X is the number of quasars detected, then the problem requires

$$\begin{aligned} P(X < 4) &= P(X = 0) + P(X = 1) + P(X = 2) + P(X = 3) \\ &= e^{-\lambda}\left(\frac{\lambda^0}{0!} + \frac{\lambda^1}{1!} + \frac{\lambda^2}{2!} + \frac{\lambda^3}{3!}\right) \approx 8.11 \times 10^{-9}. \end{aligned}$$

The required result is the CDF for $X = 3$, which can also be computed using the `scipy.stats.poisson` module:

```
from scipy.stats import poisson
print(poisson.cdf(3, 26.82))
```

The low probability is consistent with the fact that the expected number of quasars per square degree is 26.82.

(b) The expected number of quasars in an area of A square degrees is 26.82 A. The probability of finding zero quasars in this area is $P(X = 0) = e^{-26.82\ A}$, which the problem requires to be < 0.01. The inequality becomes

$$e^{-26.82\ A} < 0.01 \implies A < -\frac{\ln 0.01}{26.82} \approx 0.172 \text{ sq. deg.}$$

5. Let $Y = \cos\phi$. For $\phi \in [0, \pi)$, $Y \in (-1, 1]$, and the inverse is not multi-valued. We have $y = \cos\phi \Longrightarrow \phi = \cos^{-1} y$ and $\dfrac{dy}{d\phi} = \sin\phi = \sqrt{1 - y^2}$.

   (a) Using the above relations and the PDF method described in the notes, we get $p_Y(y) = \dfrac{p_\Phi(\phi)}{\left|\dfrac{dy}{d\phi}\right|} = \dfrac{1}{\pi\sqrt{1-y^2}}$, since $\Phi \sim$ Uniform$[0, \pi)$. This PDF is valid for for $-1 < y \leq 1$.

   (b) The population mean equals the expectation value, since we know the underlying distribution in this case: $\mathbb{E}[Y] \equiv \dfrac{1}{\pi}\displaystyle\int_{-1}^{1} \frac{y\ dy}{\sqrt{1-y^2}} = 0$

   (c) The variance is $\mathbb{E}[Y^2] - (\mathbb{E}[Y])^2 = \mathbb{E}[Y^2]$, since the expectation value is zero from above. The second moment, and therefore the variance, is $\mathbb{E}[Y^2] \equiv \dfrac{1}{\pi}\displaystyle\int_{-1}^{1} \frac{y^2\ dy}{\sqrt{1-y^2}} = \frac{1}{2}$

6. (a) It is convenient to work with polar coordinates (r, ϕ) for this part of the problem. In the second part, we can relate these coordinates to their Cartesian equivalents for plotting purposes.

   Step 1: determine the joint PDF for R and Φ

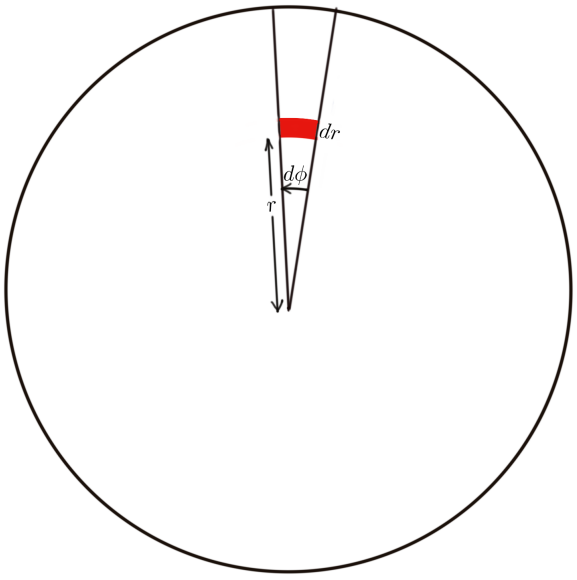

Figure 2: The infinitesimal area at distance r from the centre of the circle.

The problem requires N points uniformly distributed inside a circle of radius R_0 (say). This is equivalent to requiring a constant surface density $\sigma = \dfrac{N}{\pi\ R_0^2}$. An infinitesimal strip at distance

r from the centre of the circle has area $dA = r\ dr\ d\phi$ (see Fig. 2). The number of points in this infinitesimal area is given by

$$dn(r,\phi) = dn(r) \text{ (isotropy required!)} = \sigma\ dA = \sigma\ r\ dr\ d\phi$$

However, by definition, the fraction of points $\dfrac{dn(r,\phi)}{N}$ in this infinitesimal area equals the joint probability density of R and Φ, so that $dn(r,\phi) = N p_{R\Phi}(r,\phi)\ dr\ d\phi$. Therefore, a constant surface density requires $p_{R\Phi}(r,\phi) = \dfrac{\sigma}{N}\ r$.

Step 2: split the joint PDF into the individual PDFs for R and Φ
One possible solution is obtained by requiring that R and Φ be independent random variables. The joint distribution is then the product of the PDFs for R and Φ:

$$p_{R\Phi}(r,\phi) = p_R(r)\ p_\Phi(\phi) = \frac{\sigma}{N}\ r \implies p_R(r) = 2\pi\ \frac{\sigma}{N}\ r,\ p_\Phi(\phi) \sim \text{Uniform}[0, 2\pi] \tag{2}$$

The last two relations are the only way to ensure that Φ is independent of R, and the proportionality constant for the PDF of R is chosen such that it cancels the normalisation factor of $(2\pi)^{-1}$ in the Uniform distribution for Φ.

Note that, at first glance, the naïve choice of PDFs would be to draw both R and Φ from Uniform distributions. From Equation 2, it is clear that this is only possible if $\sigma \propto r^{-1}$, so that there is a concentration of points close to the centre of the circle. The left panel in Fig 3 demonstrates the result of this erroneous choice.

Step 3: connect R to a Uniform random variable through a transformation
The remainder of the problem is figuring out how we can draw random numbers from a distribution that is proportionate to r. Let us exploit the transformation properties for functions of random variables. We set R equal to a function of a Uniform random variable Z, then use the transformation properties of PDFs to determine the form of this function. That is,

$$R \equiv R(Z) \text{ such that } p_R(r) = 2\pi\ \frac{\sigma}{N}\ r, \text{ with } Z \sim \text{Uniform}(a,b) \text{ for } a, b \in \mathbb{R}.$$

Using the PDF method described in class, we have

$$p_Z(z)\Big(= \frac{1}{b-a}\Big) = p_R(r)\ \left|\frac{dr}{dz}\right| = 2\pi\ \frac{\sigma}{N}\ r\ \left|\frac{dr}{dz}\right| \implies \frac{d}{dz}r^2 = \frac{N}{\pi\ \sigma\ (b-a)}$$

We have reduced the problem to solving a first-order differential equation for r. The solution is

$$r^2 = \frac{N}{\pi\ \sigma}\ \frac{z-a}{b-a} = R_0^2\ \frac{z-a}{b-a} \implies r = R_0\ \sqrt{\frac{z-a}{b-a}}$$

Since this is true for any values of a and $b \neq 0$, we can conveniently pick $a = 0, b = 1$ so that Z is a Standard Uniform random variable. We then have

$$R = R_0\ \sqrt{Z}, \text{ with } Z \sim \text{Uniform}(0,1). \tag{3}$$

Procedure summary
The following steps guarantee that the N points will be uniformly distributed inside a circle of radius R_0:

i. Draw N values for the angle ϕ from Uniform$(0, 2\pi)$.

ii. Draw N Standard Uniform deviates.

iii. Compute N radii from these Standard Uniform deviates according to Equation 3.

This is not the only way to solve the problem; an approximate solution can be obtained by the following procedure instead for the Cartesian coordinates:

i. Draw N points (X, Y) uniformly distributed inside a square of size $2R_0$ – that is, both X and Y are drawn from Uniform$(-R_0, R_0)$.

ii. Compute $R = \sqrt{X^2 + Y^2}$.

iii. Remove any points that have $R > R_0$.

The problem with this method is that, due to randomness, the final number of points inside the circle is not guaranteed to be exactly equal to N. In fact, in the first step above, we are drawing N points that will be uniformly distributed over a **square of size** $2R_0$. Truncating these points to a circle of radius R_0 then reduces the total number of points by a factor $\frac{\pi}{4}$. We can compensate for this by drawing $\frac{4N}{\pi}$ points in the first step; due to randomness, however, the resulting number of points inside the circle will still only be approximately equal to N (and, due to the randomness associated with the problem, will change each time the procedure is repeated). Of course, this discrepancy vanishes as $N \to \infty$.

(b) The script here implements the procedure described in the previous part of the problem for $R_0 = 4$ and $N = 1000$ points. This result is compared to the naïve (and erroneous) method in which both R and Φ are drawn from a Uniform distribution and converted to their Cartesian equivalents (see Fig. 3).

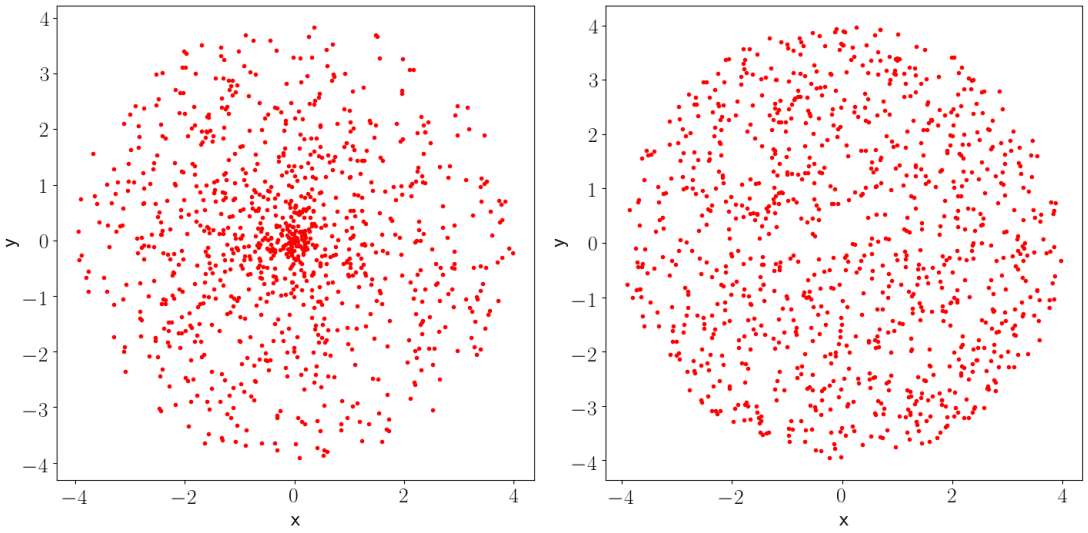

Figure 3: The incorrect (left) and correct (right) ways of generating a uniform distribution of $N \approx 1000$ points on a circle. In the first case, the is a higher concentration of points near the centre of the circle of radius $R = 4$. See text for details.