Lecture 5: Random walk and sampling errors

Recap

Lecture 4 examined correlation, regression, and variance ellipses. We finished up by considering the statistics of a random walk in one dimension so that position at time step N is:

$$x_N = x_0 + \Delta t \sum_{n=1}^{N} v_n, \tag{1}$$

where x_0 is the initial position, where the velocities v_n are taken to be random numbers. We noted that over time, the variance of the position is:

$$\langle x_N^2 \rangle = (\Delta t)^2 \langle v^2 \rangle \sum_{n=1}^{N} \sum_{m=1}^{N} \delta_{nm} = (\Delta t)^2 \langle v^2 N = (\Delta t) \langle v^2 t_N. \tag{2}$$

Moreover, since x_N is a sum, the central limit theorem applies, and its distribution will be Gaussian:

$$F_{x_N}(r) = \frac{1}{\sqrt{2\pi \langle x_N^2 \rangle}} \exp\left(\frac{-r^2}{2\langle x_N^2 \rangle} \right). \tag{3}$$

Using the definition for $\langle x_N^2 \rangle$, we can define diffusivity k as:

$$k = \frac{1}{2} \frac{d\langle x^2 \rangle}{dt} = \frac{1}{2} \langle v^2 \rangle \Delta t, \tag{4}$$

which has units of diffusivity (although the factor of one half might seem a little unintuitive. As a result of this definition of diffusivity:

$$\langle x^2 \rangle = 2kt \tag{5}$$

We can test this out with numerical simulations, and the randwalk.m code posted in Canvas gives you a way to experiment. Numerical tests show that the random walk produces a dramatic spread. In a second numerical case let's ask how the variance of position would change if we normalized it by the number of time steps—that is if we asked about the average displacement per time step. In that case, randwalk.m shows that everything converges to zero—this is what we'd expect, and we'll explore this point a little further.

Spread of a tracer concentration

Now that we've defined diffusivity, we can ask how a patch of tracer might spread out. The diffusivity k, plugged into our equation linking $\langle x^2 \rangle$ and $\langle v^2 \rangle$ tells us that

$$\langle x^2 \rangle = 2kt. \tag{6}$$

Let's consider a tracer of mass M released from the origin. Over time, it is pushed around by multiple random velocity impulses. Since the sum of many random events has a Gaussian distribution that depends on its variance $\langle x^2 \rangle$. Thus the tracer concentration Γ:

$$\Gamma(r, t) = \frac{M}{\sqrt{2\pi k t}} \exp\left(\frac{-r^2}{4kt} \right). \tag{7}$$

If this works, then the tracer distribution should be consistent with the diffusion equation:

$$\frac{\partial \Gamma}{\partial t} = k\frac{\partial^2 \Gamma}{\partial x^2}. \tag{8}$$

Does this work? Plug (7) into (8) to check.

Of course, in real world examples, tracer is also advected by a mean flow, so we should augment our diffusion equation with an advection term.

$$\frac{\partial \Gamma}{\partial t} + \vec{v} \cdot \nabla \Gamma = k\frac{\partial^2 \Gamma}{\partial x^2}. \tag{9}$$

Sampling errors

We started out this class by defining the true mean of our data:

$$\langle x \rangle = \lim_{N \to \infty} \frac{1}{N} \sum_{n=1}^{\infty} x_n \tag{10}$$

This definition makes sense, but it defies the reality of our world: we never have infinite data, and the systems we sample are not necessarily stationary (with consistent, invariant statistics) over all time. This means that we need to compute not true statistics, but **sample statistics**. We now wish to determine the accuracy of our sample statistics to aid in interpretation. The difference between the true and sampling statistics is called the **sampling error**.

The **sampling mean** is defined as

$$\{x\} = \frac{1}{N} \sum_{n=1}^{N} x_n. \tag{11}$$

Note the use of curly braces as a notation for the sample mean. The distinguishing factor between (10) and (11) is that the sample mean is computed over a finite number of realizations.

Consider the problem of estimating the true mean with the sample mean. Suppose the sample mean were calculated an infinite number of times, then we could consider the mean of the sample mean, and compare it to the true mean. The **bias** is defined to be the mean difference between the sample mean and the true mean:

$$E_1 = \langle \{x\} - \langle x \rangle \rangle = \left\langle \frac{1}{N} \sum_{n=1}^{N} x_n \right\rangle - \langle x \rangle = \frac{1}{N} \sum_{n=1}^{N} \langle x_n \rangle - \langle x \rangle = \frac{1}{N}(N\langle x \rangle) - \langle x \rangle = 0. \tag{12}$$

This tells us that the sample mean is unbiased relative to the true mean: in the limit of many samples, the sample mean converges to the true mean.

Now think about the variance of the the sample mean:

$$E_2 = \langle [\{x\} - \langle x \rangle]^2 \rangle = \left\langle \left(\frac{1}{N} \sum_{n=1}^{N} x_n - \langle x \rangle \right)^2 \right\rangle. \tag{13}$$

Moving the true mean inside the sum results in

$$E_2 = \left\langle \left(\frac{1}{N} \sum_{n=1}^{N} [x_n - \langle x \rangle] \right)^2 \right\rangle . \tag{14}$$

The quantity in square brackets in (**??**) is the fluctuation, so

$$E_2 = \frac{1}{N^2} \left\langle \left(\sum_{n=1}^{N} [x'_n] \right)^2 \right\rangle . \tag{15}$$

The average squared sum of fluctuations is the sum over all elements of the covariance matrix of the fluctuations

$$E_2 = \frac{1}{N^2} \sum_{n=1}^{N} \sum_{m=1}^{N} \langle x'_n x'_m \rangle . \tag{16}$$

Assuming the fluctuations are independent with equal variance then

$$E_2 = \frac{1}{N^2} \sum_{n=1}^{N} \sum_{m=1}^{N} \delta_{nm} \sigma^2 = \frac{\sigma^2}{N} . \tag{17}$$

So the variance decreases as the inverse of N. The square root of E_2 is called the **root-mean-square error** or the **standard error**, and decreases as the inverse of the square root of N:

$$\sqrt{E_2} = \frac{\sigma}{\sqrt{N}} . \tag{18}$$

Recognizing this improvement of error by $N^{-1/2}$ is key to understanding many measures of statistical uncertainty. This result also illustrates a parallel between sampling error and the random walk (which showed that variance of position expanded like N rather than N^2).

The rms error defined by E_2 depends on the true variance. Now let's consider the thornier situation when we are constrained by limited sampling. Our estimate of the variance of anomalies x' is:

$$\hat{\sigma}^2 = \left\{ x'^2 \right\} . \tag{19}$$

The mean of this estimate is:

$$\langle \hat{\sigma}^2 \rangle = \frac{1}{N} \sum_{n=1}^{N} \langle x'^2 \rangle = \sigma^2 , \tag{20}$$

so this estimate of variance is unbiased. The variance of the estimate of variance is:

$$F_2 \equiv \left\langle \left(\left\{ x'^2 \right\} - \sigma^2 \right)^2 \right\rangle = \frac{1}{N^2} \sum_{n=1}^{N} \sum_{m=1}^{N} \langle x'^2_n x'^2_m \rangle - \sigma^4 . \tag{21}$$

In estmating F_2, we've used (20) which means that the cross terms ($\langle \left\{ x'^2 \right\} \sigma^2 \rangle$) produce $2\sigma^4$. Now assuming that the fluctuations x'_n and x'_m are independent,

$$F_2 = \frac{1}{N^2} \left[N(N-1)\sigma^4 + N \langle x'^4 \rangle \right] - \sigma^4 = \frac{1}{N} \left(\langle x'^4 \rangle - \sigma^4 \right) . \tag{22}$$

Thus the error in the variance depends on the fourth moment. This is a problem of closure. To determine the error of any statistic we need to know a higher order statistic.

Now, let's come back to the potential bias in our estimate $\hat{\sigma}^2$ in the real-world scenario when the true mean $\langle x\rangle$ is unknown, and we're stuck working with $\{x\}$. In this case, we know that we want $\hat{\sigma}^2$ to be unbiased relative to the true σ^2, but we might need to do a little adjustment. We'll define:

$$\hat{\sigma}^2 = A\left\{(x-\{x\})^2\right\}, \tag{23}$$

where A is an unknown adjustment parameter that we want to find. Then

$$\begin{aligned} F_1 &= \langle \hat{\sigma}^2 - \sigma^2 \rangle & (24)\\ &= A\left\langle\left\{(x-\{x\})^2\right\}\right\rangle - \sigma^2 & (25)\\ &= A\left\langle\left\{\left[(x-\langle x\rangle)-(\{x\}-\langle x\rangle)\right]^2\right\}\right\rangle - \sigma^2 & (26)\\ &= A\left\langle\left\{x'^2\right\} - (\{x\}-\langle x\rangle)^2\right\rangle - \sigma^2, & (27) \end{aligned}$$

where we have assumed no correlation between errors in the mean ($\{x\} - \langle x\rangle$) and the individual anomalies ($x - \langle x\rangle$). This leads to

$$\begin{aligned} F_1 &= A\langle \sigma^2 - \frac{\sigma^2}{N}\rangle - \sigma^2 & (28)\\ &= A\sigma^2\left(1-\frac{1}{N}\right) - \sigma^2 & (29)\\ &= 0, & (30) \end{aligned}$$

implying that in order to obtain zero bias,

$$A = \frac{N}{N-1}. \tag{31}$$

In essence, when we compute the mean from the original data, we lose a degree of freedom, so we need to compute the standard deviation by assuming $N-1$ degrees of freedom as a normalization, rather than N degrees of freedom. Software packages that compute standard deviation (e.g. Matlab's "std" function) take this into account.

We can test out the success of these calculations using a large number of random data, with N large enough to begin to converge toward a true value. Here's the case that I showed in class:

```
% first create 100,000 ensembles, each with 100 random data points
x=randn(100000,100);

% now we can compare the mean of the ensemble with the sample mean
% E1 above
true_mean=mean(x(:))

plt(mean(x,2))
title('sample means')

% now consider the variance of the mean (E2)
```

```
xbar=mean(x,2);  %  computes the mean of each sample
% here we print the variance of the full data set ("true" variance)
%  and the variance of the means
[var(x(:))  var(xbar(:))]
% you'll see that the variance of the sample means is the true variance/N

% finally consider F1: variance from real-world data vs true variance
variance_estimated = mean((x-mean(x,2)).^2,2);
variance_true = mean(x(:).^2);

N=100;
[mean(variance_estimated) variance_true N/(N-1) variance_true/mean(variance_estimat
```

or in python

```
import numpy as np
import numpy.matlib

#  First create 100,000 ensembles, each with 100 random data points
N=100000
M=100
x=np.random.normal(size=[N,M])

# now we can compare the mean of the ensemble with the sample mean
true_mean=np.mean(x.flatten())
true_mean
# vs the sample means
plt.plot(np.mean(x,axis=1))
plt.title('sample means')

# now consider the variance of the mean (E2)
xbar=np.mean(x,axis=1)  # computes the mean of each sample
# display the variance of the full data set ("true" variance) and the variance of t

# this shows that the variance of the sample means is the true variance/N
np.var(x.flatten()),  np.var(xbar)

# finally consider F1: variance from real-world data vs true variance
xbar_matrix=np.matlib.repmat(xbar,M,1).T
variance_estimated = np.mean((x-xbar_matrix)**2,axis=1)
variance_true = np.mean(x.flatten()**2)
variance_formal = np.var(x.flatten())

print(np.mean(variance_estimated), variance_true,
        variance_formal, M/(M-1),
        variance_true/np.mean(variance_estimated))
```