STATISTICAL METHODS FOR THE PHYSICAL SCIENCES

Week 3 tutorial: Using random numbers

Change of variables: the transformation relation: discrete variables

- Consider a random variable X, with a probability mass function $p_X(x)$. We want to know the pmf for a transformation of this, i.e. $p_Y(y)$ for a new variable $Y = f(X)$.
- For discrete variables the transformation is simple, we simply map each value of x on to the corresponding y:

$$\Pr(Y = y_i) = \Pr(Y = f(x_i)) = \Pr(X = x_i)$$

- E.g. summing the roll of 2 6-sided dice to get X and transforming to $Y=1/X$:

$$\Pr(X = 2) = \frac{1}{36}, \Pr(X = 3) = \frac{2}{36} \cdots \longleftrightarrow \Pr(Y = 1/2) = \frac{1}{36}, \Pr(Y = 1/3) = \frac{2}{36} \cdots$$

- More generally, we can allow for the possibility that Y maps on to multiple values of X, by defining the inverse function $X = g(Y)$ and writing:

$$\Pr(Y = y) = \sum_{x \in g(Y)} \Pr(X = x)$$

Change of variables: the transformation relation: continuous variables

- Consider a continuous random variable *X*, with a probability density function $p_X(x)$. The simplest case is where $Y = f(X)$ is an increasing or decreasing function – one-to-one correspondence of *X* to *Y*:

$$|p(x)\mathrm{d}x| = |p(y)\mathrm{d}y|$$

- However, we need to consider also cases where $f(X)$ has a more complex shape. Defining again the inverse function $X = g(Y)$, we can state that:

$$\Pr\left(a < Y < b\right) = \Pr\left(g(a) < X < g(b)\right)$$

$$\rightarrow \quad \int_a^b p_Y(y)\mathrm{d}y = \left| \int_{g(a)}^{g(b)} p_X(x)\mathrm{d}x \right| = \int_a^b p_X(x) \left| \frac{\mathrm{d}x}{\mathrm{d}y} \right| \mathrm{d}y$$

- We have to use the absolute value to ensure the integral is positive when $f(X)$ is a decreasing function. E.g., consider the simple case of $Y=1/X$ again.
- Removing the integrals (which are now identical) we have:

$$p_Y(y) = p_X(x) \left| \frac{\mathrm{d}x}{\mathrm{d}y} \right| = p_X(x) \left| \frac{\mathrm{d}f(x)}{\mathrm{d}x} \right|^{-1} = \frac{p_X(g(y))}{|f'(g(y))|}$$

where:

$$f'(x) = \frac{\mathrm{d}y}{\mathrm{d}x}$$

Change of variables: example

- If the transformation is not one-to-one, e.g. more than one *X* maps to one *Y*, we need to sum over corresponding patches of the distribution
- Consider an example where *X* is distributed as a standard normal: $X \sim N(0,1)$, and we want the pdf of $Z = X^2$
- *X* is symmetric about zero, so first define: $Y = |X|$

$$\rightarrow p(y) = p(x) + p(-x) = 2N(0,1) = \frac{2}{\sqrt{2\pi}} \mathrm{e}^{-y^2/2} \quad \text{(for } y > 0\text{)}$$

- And we sub in: $z = y^2$ and use $\frac{\mathrm{d}y}{\mathrm{d}z} = z^{-1/2}/2$ to get:

$$p(z) = p(y) \left| \frac{\mathrm{d}y}{\mathrm{d}z} \right| = \frac{2}{\sqrt{2\pi}} \mathrm{e}^{-z/2} \left| \frac{z^{-1/2}}{2} \right| = \frac{z^{-1/2} \mathrm{e}^{-z/2}}{\sqrt{2\pi}}$$

which is the χ^2_1 distribution (chi-squared for 1 d.o.f.), as we would expect!

Generating (pseudo-)random numbers

- Most higher-level programming languages have their own pseudo-random number generators.
- The numbers generated by these functions are almost always distributed as $U(0,1)$
- They are ‘pseudo’-random because they are generated from algorithms (genuine random numbers can be obtained from random physical processes, e.g. radioactive decay).
- Typically they generate a sequence initiated by a ***seed*** number (which the user may specify). Each successive call of the generator function within the same run of code remembers the previous call, such that a sequence of (to all intents and purposes) random numbers is generated.
- Starting with the same seed will repeat the same sequence!
- Many functions will use, e.g. a system file or the system clock, to generate the seed. Be sure you know what your code is doing!

Reminder: pdf [$p(x)$] and cdf [$F(x)$]

$$\Pr(X \leq x) = F(x) = \int_{-\infty}^{x} p(x')dx'$$

(where x' is a dummy variable)

- Also we have:

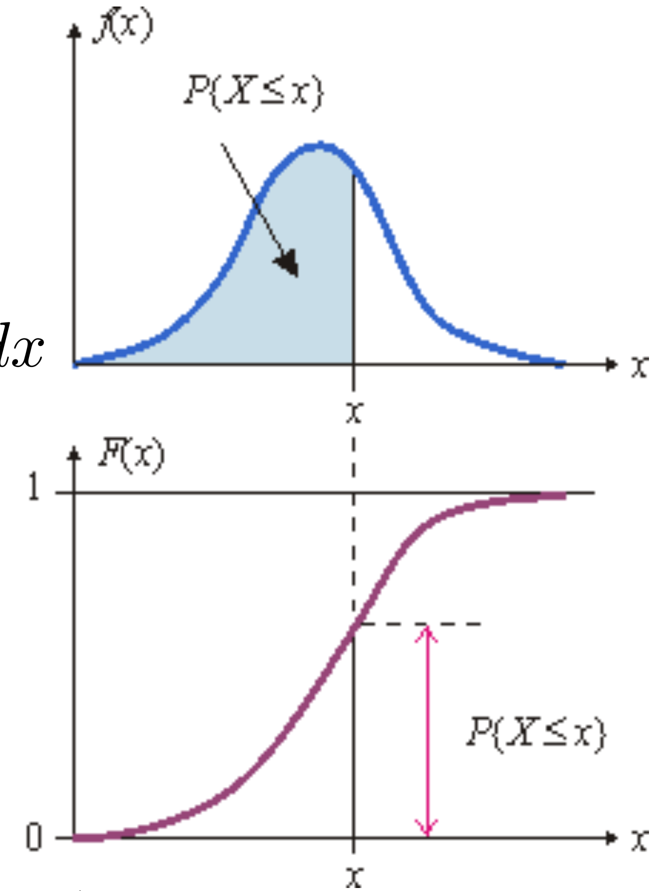
$$\Pr(a \leq X \leq b) = F(b) - F(a) = \int_{a}^{b} p(x)dx$$

which means that:

$$\int_{-\infty}^{+\infty} p(x)dx = 1$$

- We can also define quantiles α :

$$F(x_\alpha) = \int_{-\infty}^{x_\alpha} p(x)dx = \alpha \Longleftrightarrow x_\alpha = F^{-1}(\alpha)$$



But note that the quantiles α are distributed uniformly between 0 and 1!

From *U(0,1)* to a different distribution: the inverse transformation method

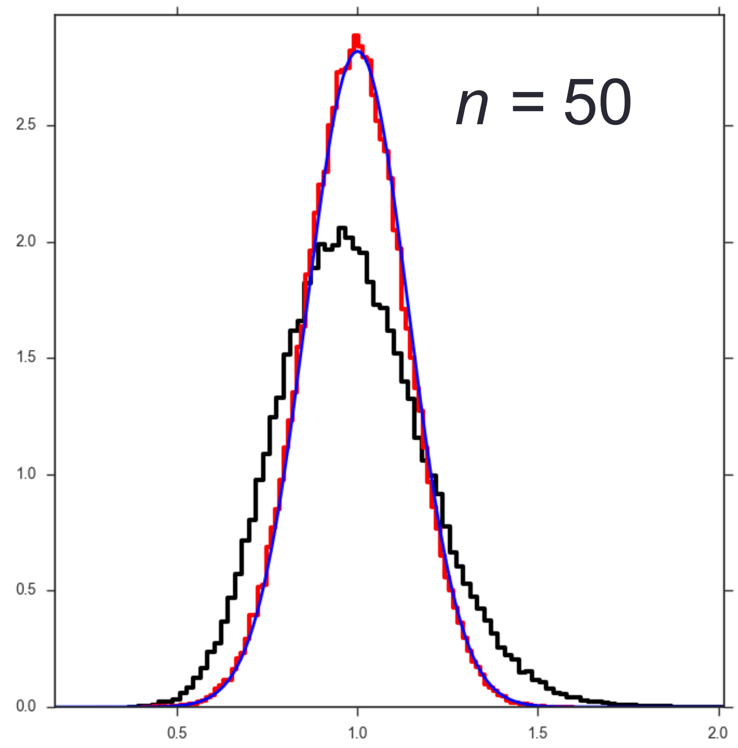
- Quantiles are distributed as $U(0,1)$
- Thus for a variable $p_X(x)$ the cdf, $u = F_X(x)$ is distributed as $U(0,1)$
- If it can be found, the inverse function $F_X^{-1}(u)$ can be used to transform from a $U(0,1)$ random variable back to x.
- Hence random numbers can be generated that are drawn from ***any*** pdf, provided we know the cdf and can find the inverse function (if not solvable analytically it can be computed using numerical integration…)

Proof that it works:

$$\begin{aligned}
\Pr(X \leq x) &= \Pr(F_X^{-1}(U) \leq x) \\
&= \Pr(F_X(F_X^{-1}(U)) \leq F_X(x)) \\
&= \Pr(U \leq F_X(x)) \\
&= \Pr(0 \leq U \leq F_X(x)) \\
&= F_X(x) - 0 \\
&= F_X(x)
\end{aligned}$$

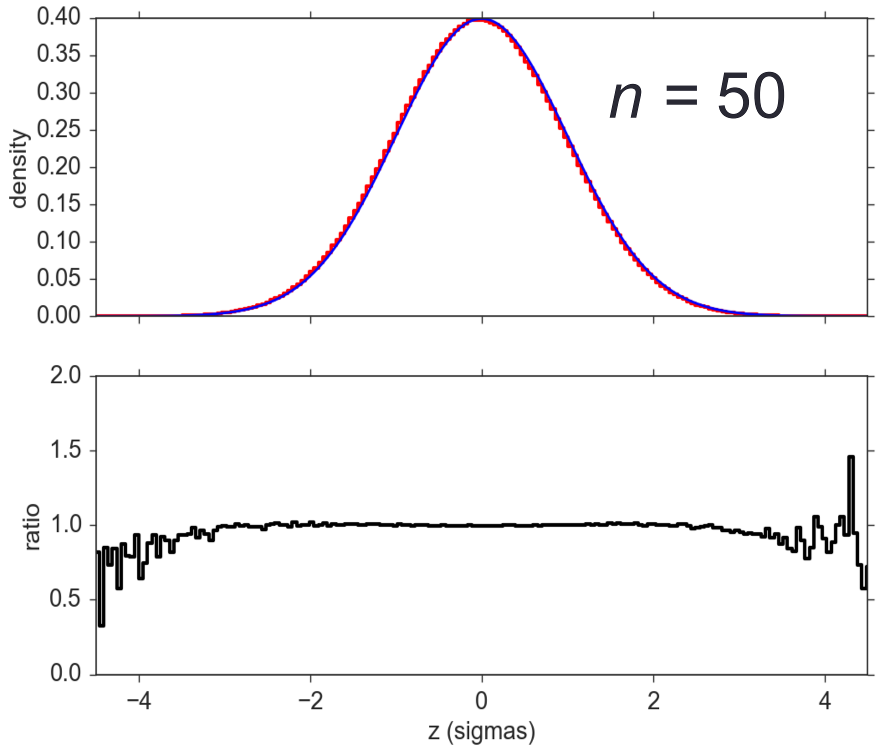
Example: testing the central limit theorem

- We can generate random numbers drawn from any distribution using, e.g. the Python scipy stats package
- Now try averaging sequences of *n* random numbers from a ***uniform*** distribution, or a ***chi-squared (with 1 d.o.f.)*** distribution.
- Repeat 10^5+ times and compare the distribution of averages with a normal distribution of the same variance and mean…
- Try for different *n*!



The limits of the central limit theorem

- As the name suggests, the theorem only holds strictly in the limit of very large *n*…
- The more skewed the averaged distribution, the higher *n* needed to approximate normal (the effect can be dramatic!)
- Even if the normal distribution is well-matched in the centre, it may be a poor match in the tails of the distribution
- So the significance of extreme values (estimated under the assumption that the sample is normally distributed) should be treated with caution, simulations may be needed for a rigorous test!



Upper: simulated pdf from averaging *n* uniformly distributed values (red) and normal pdf (blue)
Lower: ratio of the simulated to the normal pdf