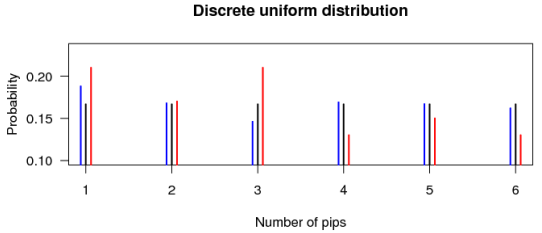
Session 4
Probability Distributions

Probability distributions

- Discrete versus Continuous
- Examples, generating processes, shapes, and properties
  - Uniform distribution
  - Poisson distribution
  - Binomial distribution
  - Gaussian or Normal distribution
  - Log-normal distribution
- Why is the Normal distribution normal?

Uniform distribution

- Generate a discrete uniform distribution by rolling a die



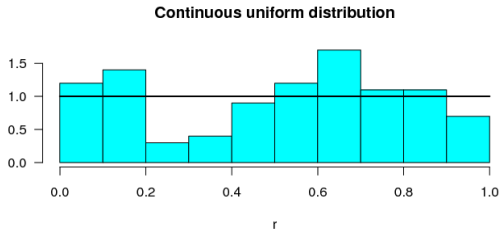
- The theoretical distribution (P=1/6) is in black in the middle
- Empirical distributions of die rolls are less perfect, but approach the theoretical distribution with increasing sample size, 100 die rolls (red) or 1000 (blue)

Uniform distribution

- Law of large numbers:
  - The sample mean *m* approaches the expected value (the population mean or true mean μ) with increasing sample size
  - The larger the sample size, the closer the empirical distribution gets to the theoretical distribution
    - This holds generally, not just for the uniform distribution

Uniform distribution

- Normalized histogram of a continuous uniform distribution



- The theoretical distribution with uniform probability is the black line
- Empirical distribution of a sample, n=100

Poisson distribution

- Counts of 'events'
  - If you can count how many times some event happened, but not how many times it did not happen
  - In other words, if there is no limit, or maximum count
  - Which means you cannot calculate a proportion as in the Binomial distribution where you either have one event (red marble) or another (blue marble), so you can calculate the proportion of red or blue marbles from the total number of marbles drawn from an urn

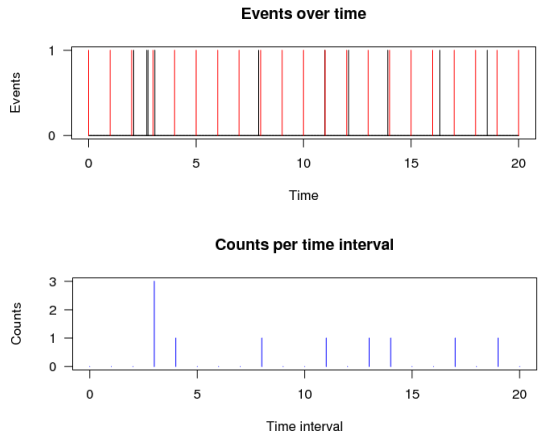
Poisson distribution

- Examples
  - number of accidents on Bristol Road per week (you don't know how many non-accidents happened, so can't calculate a proportion!)
  - number of radioactive decays per time interval (there is no upper limit, you can't define the number of non-decays)
  - number of leukocytes per grid element in a counting chamber (can be more than one cell per grid element)
  - always a count of something per time interval, per length interval, per area, per volume

Poisson distribution

- Generating process: any process where the probability of an event occurring per unit time (or area or volume) is constant
  - implies that events are independent (no correlation or interaction)
  - Next slide gives a simulated example: radioactive decay events over time

Poisson distribution



Poisson distribution

- Do the same for 200 intervals (to get better statistics, c.f. law of large numbers)
- In R, you can fit the Poisson distribution to your data
- We start here with vector c (counts) containing the number of events in each interval (from experiment or simulation)

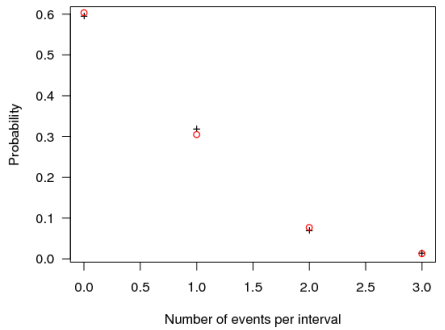
```
cs <- table(c) # produces a table object (a kind of list)
  summarizing all counts
> cs
c
  0   1   2   3
119  64  14   3
cs <- cs[] # convert list into vector of counts
k <- min(c):max(c)
```


Poisson distribution

```
# fit Poisson distribution
require(MASS) # fitdistr() function is in package MASS
f <- fitdistr(c,"poisson") # fit counts to Poisson
  distribution
fc <- coef(f) # get coefficients (parameters) from the
  fit, Poisson has only 1!
d <- dpois(k,lambda=fc) # calculate (probability)
  densities of theoretical Poisson distribution with
  fitted lambda
# plot data and then fit on top of data
plot(k,cs/sum(cs),xlab="Number of events per
  interval",ylab="Probability",las=1,pch="+") # normalize
  counts by total
points(k,d,col="red")
```

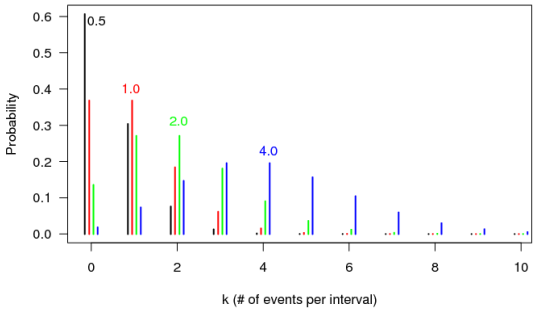

Poisson distribution

- Simulated data (200 events) (+)
- Poisson distribution fitted to data (o)



Poisson distribution

- Equation: $P(k) = \frac{\mu^k}{k!} e^{-\mu}$
- Parameters: mean (μ): only one because mean = variance
  - in other words: s = sqrt(mean)
- Shape: from highly skewed to symmetric with increasing μ



Exercise

- Poisson distribution: counts on a grid
  - Example: counting cells in grid elements of a counting chamber
  - Step 1: Generate uniform random pattern in space. Make 100 points, x and y coordinates in interval (0,10), plot these 100 points)
    - use runif() to generate x and y coordinates
    - check help on runif() if needed: ?runif
    - plot x and y coordinates
  - Step 2: Draw grid on top of the points like this

```
# draw grid for counting
abline(h=0:n) # horizontal lines
abline(v=0:n) # vertical lines
```


Exercise continued

- Step 3: Count number of points in each grid element from plot or automatically

```
# count automatically
c <- matrix(numeric(10^2),nrow=10)
# visit all points and convert coordinates into integers
for (i in 1:length(x)) {
  xi <- ceiling(x[i])
  yi <- ceiling(y[i])
  c[yi,xi] <- c[yi,xi] + 1
}
```

- Step 4: Summarize count data: 0 counts in x grid elements, etc.
- Step 5: Fit count data to Poisson distribution
- Step 6: Plot summarized count data and fitted Poisson distribution
  - Steps 4-6 are described in the slides on Poisson in time

Exercise: Counting colonies on agar plates

- You have plated out 0.1 ml aliquots of a 1:10 and a 1:100 dilution of a culture on two plates each
  - Results of counting colonies:
  - Dilution 1:10 200 and 180 colonies
  - Dilution 1:100 20 and 18 colonies
- The more data we include, the better, so we sum all counts regardless of dilution, and we sum all volumes that have been counted. This is better than just using counts from one dilution
  - That's why we sum all counts
  - And sum all volumes counted
- Calculate the mean CFU per ml of culture from these sums
- Calculate the error from the total count (remember for Poisson data, SD = sqrt(count))
- From this, calculate the SD of CFU per ml of original culture

Why do we include as many counts as possible?

- The mean increases with the number of particles counted, but the SD increases only with the square root of the number of particles counted! Therefore, your relative SD (called Coefficient of Variation CV), decreases the more you have counted. If you count 9 times more colonies, your mean is 9 times higher, your SD is only 3 times higher, so your relative error (CV) is reduced to 1/3!
- You should count at least several hundred cells/colonies/etc.

Binomial distribution

- Proportion data
  - Events fall into two categories, either one or the other can occur
    - Head or Tail
    - Girl or Boy
    - Red or Green marble drawn from an urn
    - mnemonic: binomial is Latin for 'two names'
  - There is a total number of events
    - Total number of marbles drawn = red marbles + green marbles
  - That's why we can calculate a proportion!
    - Proportion red (red marbles/total) and green (green marbles/total)
  - Compare Poisson distribution (count data) where you have only one type of event and an unknown number of non-events

Binomial distribution

- Examples
  - Proportion of people cured by treatment (others not cured)
  - Proportion of animals surviving (others dead)
  - Proportion of votes for Obama (others voted for McCain)
  - Proportion of students who know what "significance" means (the others don't)
- Generally: two classes of events and both the number of A events and non-A events are known so the total and the proportions can be calculated

Binomial distribution

- Generating processes:
  - (Biased) coin flips (head or tail)
  - (Biased) random walk in 1D (step left or right)
  - p: probability for stepping to the left
  - q: probability for stepping to the right, q=1-p
  - n: number of steps in total, i.e.
    - k steps to the left
    - (n-k) steps to the right
  - One possible sequence is : LLRRRL
  - Probability for this: pp qqq p = $p^3q^3 = p^kq^{n-k} = p^k(1-p)^{n-k}$

Binomial distribution

- Generating processes:
  - Part 1: Probability for any sequence of k steps to the left, n-k to the right is $p^k q^{n-k}$
  - Part 2: How many sequences will contain k steps to the left, n-k to the right?
    - Number of combinations of k elements taken from a total of n elements: binomial coefficient, pronounced n choose k

$$\binom{n}{k} = \left(\frac{n!}{k!(n-k)!}\right)$$

  - R functions of note: factorial(), choose(), combn()
    - The factorial of 4 is 1 * 2 * 3 * 4 = 24, of n is 1 * 2 * 3 *...* (n-1) * n
    - The factorial of 0 (an empty product) is 1

Binomial distribution

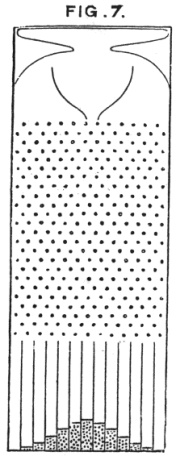
- Generating processes:
  - Probability of k steps to the left =
    probability of any sequence including k steps (Part 1) times
    number of sequences (combinations) yielding k steps (Part 2)

$$P(k) = \binom{n}{k} p^k q^{n-k} = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k}$$

  - Parameters:
    - p: probability of 'success' (probability of 'failure' is q = 1-p)
    - n: total number
    - (k is not a parameter, k is where the function lives)
  - Mean $\mu = np$
  - Variance $\sigma^2 = np(1-p) = npq$ = mean * q

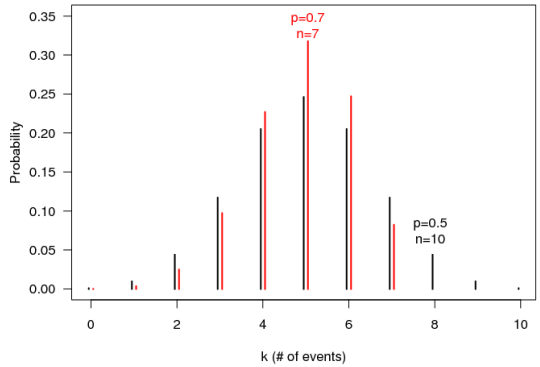
Binomial distribution

- Generating processes:
  - Bean machine invented by Sir Francis Galton (born 1822 in B'ham)
  - Physical (biased) random walk, a bean hitting a peg bounces left or right with probability p and q
  - With a large number of beans, the Binomial distribution converges towards the Normal distribution



Binomial distribution

- Shape symmetric if unbiased (p=0.5), skewed otherwise
  - Mean= np is about the same for both distributions

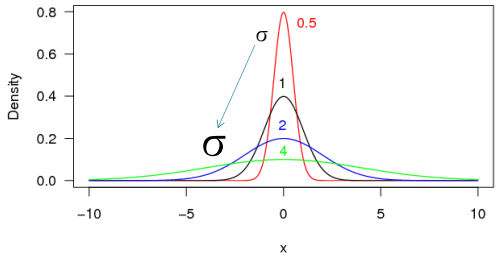


k is were the function lives (has values, is defined)

Normal or Gaussian distribution

- Bell-shaped curve described by two parameters (mean and standard deviation)
  - All you need to describe 'normal' data is mean ± standard deviation (μ ± σ)
- Continuous
- Symmetric
- Lives on the 'interval' −∞ to +∞

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



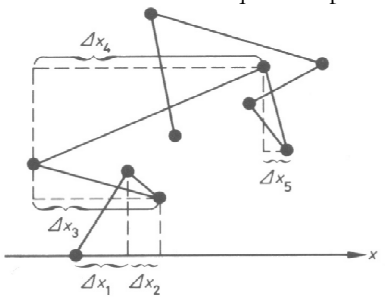
Exercise

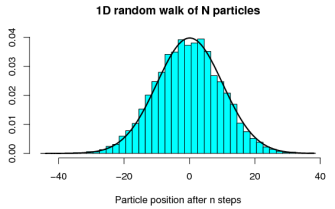
- Plot normal distributions with two different standard deviations into the same plot
  - Say mean = 0, sd=1 and sd=2
  - Use dnorm
  - Check out ?dnorm (the help file on dnorm)
  - To make a plot, you need to make an x vector, and then calculate the y vector, and then plot(x,y)
- Same can be done for other distributions replacing density function dnorm for the normal distribution with dlnorm for log-normal, etc.

Normal or Gaussian distribution

- Generating a normal distribution
  - Making many little additive errors leads to the Normal distribution
  - Example: random walk
  - Variance = mean-square displacement

$$s^2 = \frac{SS}{d.f.} = \frac{\sum(x_i - \bar{x})^2}{n-1} \cong \overline{\Delta x^2} = 2Dt$$





Normal or Gaussian distribution

- Example: random walk of many particles
  - Diffusion from a point source generates a concentration gradient that has the same bell shape as a Normal distribution
  - This is the law of large numbers in action: for a few random walking particles, you get the Binomial distribution, for zillions of particles, you get Diffusion and the Normal distribution
- Generally, a process where lots of small independent errors add up (additive errors)

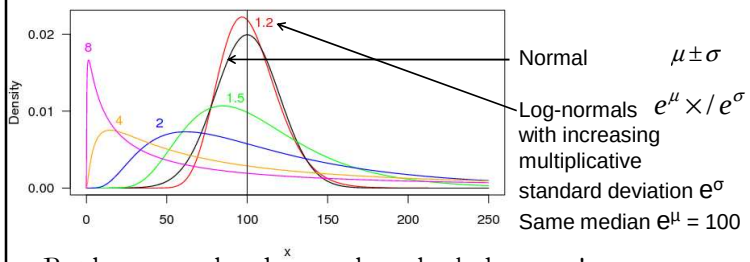
Exercise

- Simulate a 1D random walk of N particles for n steps
  - Need random numbers, from which distribution?
- Plot positions of all particles after n steps
- Make histogram of particle positions after n steps

- Extra:
  - Calculate mean and sd from data
  - Fit normal distribution with these mean and sd to data
  - Plot fitted distribution on top of histogram (see above)

Log-normal distribution

- The log-normal distribution is
  - skewed (not symmetric)
  - restricted to positive values (actually, most measures such as sizes, concentrations, abundances, etc., can't be negative)



- But log-normal and normal can look the same!

Log-normal distribution

- Generating processes
  - Growth processes: size increase is proportional to size (positive feedback), which leads to exponential growth
    - If growth last year was a bit stronger by chance, growth this year will be proportional to the size reached last year, multiplying the 'error' from last year
  - Generally, whenever errors are multiplicative rather than additive
- Examples
  - Body weight (height is usually considered to be normally distributed)
  - Abundance of bacteria on plant leaves
  - Latent periods (time from infection to first symptoms)

Log-normal distribution

- Survival times after cancer diagnosis
- Species abundance (also other distributions)
- Farm sizes in England & Wales
- Income
- Size of ice crystals in ice cream
- Age at first marriage

- We tend to neglect the log-normal because
  - additive errors are simpler than multiplicative errors
  - it is not symmetric

Why is the Normal Distribution so Normal?

- Is the generating process more common or typical???
  - Normal distributions arise from many additive 'errors', e.g. an (unbiased) random walk generates a Normal distribution with variance increasing with time
  - There are many processes that don't generate a Normal distribution
  - It makes no sense to say one is more typical than the other, it depends on your research topic which distributions are appropriate

Why is the Normal Distribution so Normal?

- The Normal distribution is so important thanks to the Central Limit Theorem
  - For large *n*, the Binomial and other distributions converge to the Normal distribution
  - Even if the distribution of something in a population is anything but normal, the distribution of the means of samples from that population tends to the Normal distribution
  - Convergence of sample means to the Normal distribution is slower if the underlying distribution is skewed (not symmetric)
  - All of this is also true for other statistics than the mean, e.g. the standard deviation

Why is the Normal Distribution so Normal?

- Example: x is uniformly distributed
  - Make small population of uniformly distributed random numbers

```
x <- runif(10,0,20) # 10 random numbers
  between 0 and 20
xr <- round(x) # round numbers to nearest
  integer for simplicity
xr
 [1]  1 13 10 17 20  5 13  3 10  8
mean(xr) # mean of population, i.e. of all
  numbers
[1] 10
```


Why is the Normal Distribution so Normal?

- Now draw samples with n=3 from this population and calculate the means of these samples
- Choosing 3 elements from a set of 10 gives 120 combinations, calculate with binomial coefficient (n choose k)

```
choose(10,3)
[1] 120
```

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

- R not only calculates the number of combinations with choose(), but also makes a table (matrix) listing all these 120 combinations with the function combn()

```
c <- combn(xr,3)
```

Why is the Normal Distribution so Normal?

- Will only list the first 3 and last 3 combinations, not all 120!

1	1	1	…	13	13	3
13	13	13	…	3	10	10
10	17	20	…	8	8	8

- Now compute the sample means of all these 120 combinations (one column is one combination, so use colMeans() to compute)

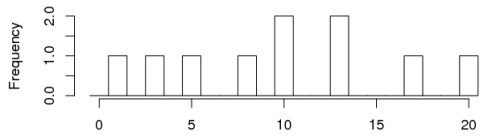
```
mc <- colMeans(c)
```

- Now we are ready to compare the distribution of the population with the distribution of the means of samples from that population, using hist()

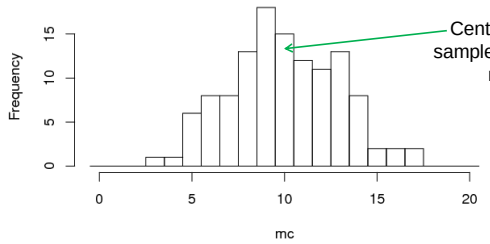
```
hist(xr,-.5:20.5)
hist(mc,-.5:20.5)
```


Why is the Normal Distribution so Normal?

Distribution of data is approx. uniform



Distribution of means of samples from that population is approx. Normal!



Central limit theorem: sample means tend to be more central