MATH 1060   Lab 4 - Discrete distributions

Binomial Distribution

The following code block generates a binomial distribution:

```
1  #Binomial ditribution
2  n <- 100 #number of trials (e.g. coin tosses) in an experiment (sample)
3  p <- 0.5 # probability of getting success (head) in a single trial (coin toss)
4  ns <- 1000 #number of experiments (sample size)
5  x <- rep(0,len=ns) # each component is the number of successes in a single experiment.
6  for(i in 1:ns){
7    x[i] <- sum( ifelse(runif(n) < p, 1, 0) )
8  }
9  rf <- table(x)/ns
10 plot(rf, ylim=c(0, 1.3*max(rf)))
11 xlabels <- 0:ns
12 Ptheory <- choose(n,xlabels)*p^(xlabels)*(1-p)^(n-xlabels)
13 points(xlabels, Ptheory)
14
15 c(mean(x), n*p)
16 c(sd(x), sqrt(n*p*(1-p)) )
```

The code is very similar to the ones we had in Lab 3, but there are some new ones. Let's go over these new commands first:

Line 7: ifelse(*conditional statement*, *statement 1*, *statement 2*) outputs the result of *statement 1* if *conditional statement* is True, otherwise the result of *statement 2.* It works on vectors as well. Try running ifelse(1:10 > 5, 1, 0) in the console. What do you think it will output?

runif(n) creates a vector of n random numbers sampled from a uniform distribution of real numbers between 0 and 1. Inspect ifelse(runif(n) < p, 1, 0) in the console. Can you explain what it does?

Line 9: table(x) creates a frequency table for the values in x. Inspect the output of table(x) in the console. Here, it is divided by ns, the sample size, to make them relative frequencies (probabilities).

Line 10: The lower and upper bounds of plot can be specified by setting the option

ylim=c(*lower*, *upper*)

Line 12: This line computes the theoretical probability distribution. choose(n, k) is the binomial coefficient ${}_nC_k$.

Lines 15 and 16: The outputs are formatted as vectors using c( ) command.

Now you understand what each line does, let's modify it and do some experiments.

1) Increase the number of trials (n) and the sample size (ns). Then observe the shape of the distribution. The histogram should look like a so-called *bell curve*. The theoretical values (circles) should be more or less right on the experimental values (vertical lines).
2) Compare the mean and the standard deviation calculated from the simulation to the theoretical values $\bar{x} = n \cdot p$ and $s = \sqrt{n \cdot p \cdot (1-p)}$. The simulated values should get very close to the theoretical values as n and ns become large.
3) Copy and paste the above code, then modify the value of p, the probability of getting a success in a single trial, to a number smaller than 0.5, then to a number larger than 0.5. Observe that the peak of the distribution shifts and its width changes. You should see the peak shifts to the left for p<0.5, and to the right for p>0.5. Also, note that the standard deviation will decrease for both p<0.5 and p>0.5.
4) Now copy the code block and paste it just below the previous one. We will modify it to see the Poisson distribution arising from the binomial distribution. At the top of the block let us add a line specifying the mean; `mu <- 2` for example. The value of the mean should be fairly small for the theoretical Poisson probability function to be a good approximation. Since mu is given p can be calculated as `p <- mu/n`. Change the theoretical probability function, `Ptheory` to the one for the Poisson distribution. Observe that the distribution is very skewed for small mu, and the theoretical values should match the simulated ones.

Well done. Submit your script to the Learning Hub by next lecture.