Assignment 3

1. A travelling salesman has carpets in his van. He goes from door to door trying to sell them. Each time he sells one he gets excited and is able to be more convincing and each time he is rejected and cannot close a deal he gets less excited and is less convincing when dealing with the next customer. This salesman was blessed with 10 levels of excitement and each interaction either lifts or decreases his excitement by one level. The probabilities for the salesman to sell a carpet depending on how he feels are {0, 0.1, 0.2, 0.3, …0.9}. Write an algorithm to simulate this Markov chain. In b) - f) run simulations for all initial levels of excitement, **except the first level**, because business is downright non-existent when one feels this low for starters. Use one bin per timestep. **Run simulations over the time period (sales events) of $T = 1000$. For statistics, run 10 000 simulations**, which in the present story means that the salesman makes 10 000 trips.
   a) Write down the Markov matrix.
   b) **For every simulation of each initial level of excitement, compute the total number of sold carpets by the end of the simulation (cumulative number of sales). For each initial level of excitement plot as a normalised histogram (density=True) the probability mass density (PMD) of the cumulative number of sales** (9 histograms in total)**.** To make this clear, such a histogram has the cumulative number of sales on the x-axis and the (normalised) frequency with which each number of cumulative sales is realised in 10000 trials on the y-axis. Choose the appropriate scale to see the PMD as clearly as possible.
   c) For each different initial level, compute the average (over the 10 000 trials) number of carpets the salesman can sale within the given time T. **Plot these average numbers of carpets as a function of the salesman's initial level of excitement** (1 plot)**.**
   d) **Compute and plot the PMFs for time (the number of steps) required for the salesman to sell 5 carpets. Exclude the runs that ended up without selling at least 5 carpets**. At sufficiently high initial levels of excitement the PMFs should start showing a definite functional form. **Use appropriate scale so that you can name this functional form.**
   e) Compute the **average** times (for the different initial levels) it takes for the salesman to sell 5 carpets. Plot these times as a function of the salesman's initial level of excitement.
   f) In point d) and e), excluding the runs that didn't sell enough carpets is underestimating the number of steps needed to sell 5 carpets, as those "failed" runs could take anywhere from T+1 to an infinite number of steps to finish selling those 5 carpets. **Plot the probability of selling less than 5 carpets in T timesteps as a function of the initial level of excitement.**
   g) In d) you obtained some definite functional form(s) for certain level(s) of initial excitement. **Why does the current process produce these functional forms? (**If you did not get these forms, you can still argue for them based on what you know.)

2. Let's take the Poisson process to describe radioactive decay. The number of nuclei is initially $N_0 = 10000$. The nuclei decay (fission) at rate $\lambda = 0.2$ per second. We want to determine the half-time $t_{1/2}$, that is, the time it takes on average for the number of nuclei

to decay to $N(t) = N_0/2$. You can do this simulation in two ways, the first of which is what a statistician would do and that is presented in Lecture 3. There is an alternative way based on simulating the stochastic process in time steps, which is what for example a physicist would do. In this second way you should first run simulations to find appropriate time interval (time step) but let's pretend you have already done this and found that $\Delta t = 0.01$ s.

a) Implementation of the algorithm by either method. One method is enough.

b) By either simulation, determine the mean and variance for $t_{1/2}$ that you get by averaging over 100 simulations.

(Just to make sure: you only need to implement and simulate in one way.
In addition, needless to say, write the algorithms for the Poisson process, that is, don't use some all-inclusive black-box library function.)