Necessary/very good for Quant Interviews. The best way to ace a quant interview is to articulate observations

CHAPTER 1:

1. Outcome Space and Events
    a. **Outcome space –** just a set, denoted by Ω, of all possible outcomes for a given experiment
    b. **Outcome –** ω is an element of Ω
    c. **Event –** a subset of the outcome space, and the empty set ∅, and the entire set Ω are both allowed as subsets
2. Equally Likely Outcomes
    a. Let A ∈ Ω, the **#(A)** denotes the number of elements inside of A
    b. P(A) denotes the probability that A occurs, or the chance that A occurs
        i. P(A) = #(A) / #(Ω)
3. Collisions in Hashing
    a. A Hash function assigns a code called a Hash to each set of individuals. If two individuals are assigned a similar values it causes a **collision** making issue when identifying
    b. It can be cumbersome to track all the different hash numbers and the individuals, but we pick randomly from a large enough set proportionally to the number of hash numbers we need, then the P(Collision) is relatively low
    c. If there are N potential hash values and *n* individuals that need a identification, then there are N^n different permutations available.
    d. What are the chances of No Collision?
        i. If n > N, then there has be a collision
        ii. We can simply do $\frac{N!}{n!N^n}$ as there are N potentially options for the first, N-1 for the next, etc
    e. Can find the probability of at least 1 collision as it is the complement
4. The Birthday Problem
    a. Classic example for finding the probability of having a "collision" of birthdays and shows how it increases rather sharply
5. An Exponential Approximation
    a. Although we have the equation $\prod_0^{n-1}\frac{N-i}{n}$ we don't have a good understanding of how this increasing or decreasing as N changes
    b. If we take the logarithm of a multiplicative series, it turns into a summation
        i. A*B = C, log(A*B) = log(C) = log(A) + log(B) = log(C)
        ii. Thus using the above logic, we now have log(P(No Collision)) = $\sum_0^{n-1}\log\left(\frac{N-i}{n}\right)$
        iii. Working with summations is much easier in general
    c. We utilize the knowledge that log(1 + x) is approximately equal to x as x approaches infinity. While this isn't necessarily accurate in our case as x (or in our case i/N) isn't zero, we are simply trying to reach an approximation for the most part
    d. Utilizing the point in 5c, we can see that the summation become
        i. $\log(P(No\ Collision)) = \sum_0^{n-1}\log\left(1\ -\frac{i}{n}\right)$

    ii. $\log\big(P(No\ Colliion)\big) = \sum_0^{n-1} \frac{i}{N}$
    iii. $\log\big(P(No\ Collision)\big) = \frac{n(n-1)}{-2N}$
    iv. Thus we have a general approximate for the P(No Collision) which we can get by exponentiating both sides to arrive at $e^{\left(\frac{n(n-1)}{-2N}\right)}$
    v. The distribution $1 - e^{(-cx^2)}$ is called the Rayleigh Distribution

CHAPTER 2: CALCULATING CHANCES

There are some basic axioms of probability that were outlined by Andrey Kolmogorov and are the basics for probability theory

Probability – the function that P defined on events in Ω

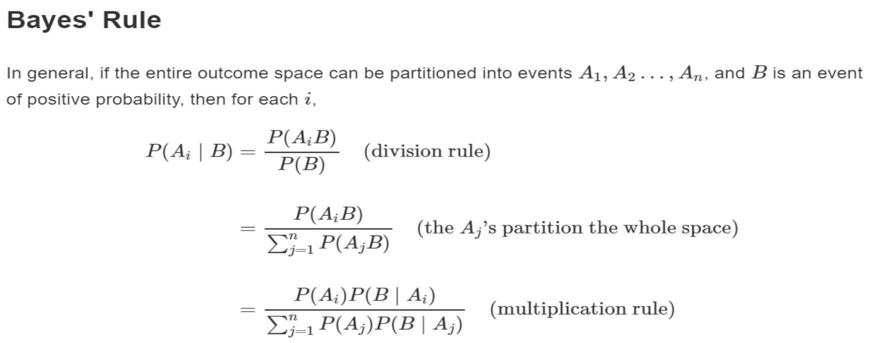
Axiom 1 – Probabilities are non negative: for each A, P(A) ≥ 0

Axiom 2 – Probability of an entire space Ω is 1, thus P(Ω) = 1

Axiom 3 – If two events A, B are mutually exclusive then

$A \cap B = \emptyset$ and $P(A \cup B) = P(A) + P(B)$

1. Addition
    a. Third axiom is about mutual exclusive events; A and B are mutually exclusive if at most one of them can happen
    b. If two events are mutually exclusive this makes addition easier as we know that the probability of either happening is just the sum of the individual probabilities
    c. We can generalize this to get the finite additivity which essentially states
        i. $P(\cup A_i) = \sum A_i \quad given\ that\ all\ A_i\ are\ disjoint\ to\ all\ others$
    d. Nested Events
        i. An event B is nested inside A if B is a subset of event A
        ii. $A\ \text{"\textbackslash"}\ B\ =\ A\ \cap\ B^C\ where\ B^C\ is\ the\ complement$
        iii. If B is a subset of A, then we have that A \ B = P(A) - P(B)
            1. However is a B is not a pure subset of A, we cannot make this generalization
    e. Complement
        i. For any event B, the complement B^C for which P(B^C) = equal to 1 – P(B)
2. Multiplication
    a. AB = $A \cap B$ and represents a union
    b. Conditional Probability is represented as $P(B\ |A\ ) = P\frac{(AB)}{P(A)}$ which just means the probability B will occur given that A occurs. IF they are disjoint, $P(B|A) = P(B)$
    c. This can simply be finessed to read that $P(B|A) * P(A) = P(AB)$ which can be helpful when multiplying probabilities if only particular knowledge is given
    d.
3. Updating Probabilities

a. 

Bayes' Rule

In general, if the entire outcome space can be partitioned into events $A_1, A_2 \ldots, A_n$, and B is an event of positive probability, then for each i,

$$\begin{aligned} P(A_i \mid B) &= \frac{P(A_i B)}{P(B)} \quad \text{(division rule)} \\ &= \frac{P(A_i B)}{\sum_{j=1}^{n} P(A_j B)} \quad \text{(the } A_j\text{'s partition the whole space)} \\ &= \frac{P(A_i)P(B \mid A_i)}{\sum_{j=1}^{n} P(A_j)P(B \mid A_j)} \quad \text{(multiplication rule)} \end{aligned}$$

b. The Effect of the Prior: You can update probabilities given new information and in a timeline, knowing an event can lead to changing the conditional probability

Chapter 3: Random Variables

A random variable – a numerical function defined on a outcome space, such that its domain is Ω and its range is the number line. Typically denoted by late upper case letters in the alphabet

1. Functions on an Outcome Space
   - a. Listing out all possible combinations for any particular event might become tedious as it grows so we can either use shorthands, abbreviations or a computer to do so
   - b. **Product Space –** set of all pairs (a, b) where $a \in A, b \in B$
     - i. The product of a single sample of a coin toss (H, T) and itself is the product of two coin tosses [(H,H), (T, H), (H, T), (T,T)] and multiplying it again gets us 3 tosses etc
     - ii. Python Library contains code to help us do this:
       1. from itertools import product
       2. base_case = np.array(‘H’, ‘T’)
       3. two_tosses_p = list(product(base_case, repeat = 2))
       4. three_tosses_p = list(product(base_case, repeat = 3))
   - **c.** **Probability Space** is simply the outcome space accompanied by the probabilities of each outcome. We can create a table as such with
     - **i.** three_tosses = 1/8 * np.ones(8)
     - **ii.** three_toss_space = Table().with_columns(‘omega’, three_tosses, ‘P (Omega)’ , three_tosses_p)
   - d. Can do the same type with a die, simply with die = np.arange(1,7,1)
2. A Function on the Outcome Space
   - a. What if we performed a function on the outcome space that we are given?
   - b. In this sense, lets say we rolled a dice 5 times, and we added up all the values that we saw. Thus we can essentially map each of the outcomes to a summation which we write as
     - i. $S : \Omega \rightarrow \{5, 6, \ldots .0\}$
     - ii. The above equation essentially means that the summation of any given vector of rolsl falls in the range {5, 30}
   - c. Can create this in Python similarly
     - i. new_column = Table().with_Columns(‘omega’, five_rolls, ‘S(Omega)’, five_roll_space.apply(sum, ‘omega’), ‘P(Omega)’, five_roll_probs)
3. Functions of Random Variables

a. A function of a random variable is also a random variable, thus the square of the sum, the cube of the sum, the add-one of a sum are all also random variables

4. Events Determined by S

a. For any subset of A of the range of S, define the event $\{S\ \in A\} = \{\ \omega : S(\omega) \in A\}$ which essentially means that the set includes all outcomes that have a summation included in A.

b. Five_rolls.where('S(Omega)', are.equal_to(10)) will get you all the places where 5 dice add up to 10

c. Thus the $P(S\ \in \{10\}) = \frac{126}{1776}\ \mathit{where}\ 126\ \mathit{is\ the\ number\ of\ entries\ returned\ by\ the\ table}$

5. Distributions

a. Often we care about the probability distribution for a particular Random Variable in which we ascertain the probability for each distinct value that the Random Variable can take

i. Python Code:

1. dist_S = five_rolls_sum.drop('omega').group*('S(Omega)', sum)
2. dist_S

ii. The produced table dist_S is the probability distribution table of S (which recall is the sum of 5 rolls of a 6 sided die)

iii. Probabilities in a distribution are non negative and must sum to 1

b. Visualizing Distributions

i. Prob140 library builds on the datascience library to provide some convenient tools for working with probability distributions and events

ii. We are going to extract both the values and their corresponding probabilities and set them equal to arrays

1. S = dist_S.column(0)
2. P_S = dist_S.column(1)
3. **Important Note: Plot in the Prob140 Library**

a. Plot(probability_distribution) will output a histogram as long as the distribution is valid

b. You can create a probability distribution by starting with an empty table and using value and probability table method

i. new_dist_s = Table().values(s).probability(P_S)

ii. Plot(new_dist_S)

c. Making binning decisions with non integer bounded random variables becomes substantially harder

iii. Notes on the Distribution of S

1. The new_dist_S is an exact distribution and assumes that we have iterated over all possible outcomes of the experiment.
2. The Central Limit Theorem says we need approximately 30 samples in order to ensure a bell shape curved for the sum of a large random sample. If you start with a uniform distribution, however, it approaches a bell curve much faster (within 5 is sufficient)

iv. Visualizing Probabilities of Events

1. We can use the event function of the Plot function to highlight a particular event

a. Plot(new_dist_S, event = np.arange(14, 22, 1))

2. We can also find the “exact” probability of this given the distribution table by using the prob_event method

a. new_dist_S.prob_event(np.arange(14,22,1))

v. Math and Code Correspondence

1. $P(14 \le S \le 21) = \sum_{s=14}^{21} P(S = s)$

2. You can equally represent this by doing

a. event_table = new_dist_S.where(0, are.between(14,22))

b. event_table.column(‘Probability).sum()

c. Equality

i. Two Random Variables are equal if their values are the same for every outcome of the same outcome space where both of them are defined in

1. $X(\omega) = Y(\omega)\ for\ all\ \omega\ \in\ \Omega$

2. Let N_H be the number of heads in N coin tosses and N_T be the number of tails, then N - N_T and N_H are equal random variables

3. N_H and N_T are not equal but have the same probability distribution and are noted as *equal in distribution*

4. If two Random Variables are equal they are equal in distribution but the converse does not necessarily hold true

ii. Extra Point:

1. Table().with_values(possible_i).probability_function(function)

a. This will apply a function to every value and place it as its probability

Chapter 4: Relations between Variables

It helps us understand the conditional behavior by understanding the relationships between different variables

1. Joint Distributions

a. Suppose X, Y are defined on the same outcome space: in this case, we utilize the notation $P(X = x, Y = y)$ for the probability that X = x and Y = y.

i. The **joint distribution** is the cumulation of all probabilities where (x,y) ranges over all the possible values of (X, Y)

ii. The call for this is

1. joint_table = Table().values(variable_name_1, values_1, variable_name_2).probability_function(function_name)

iii. ALthought this does serve us the purpose we need we can also visualize this is much more conventional way which would be:

1. Joint_dist = joint_table.to_joint()

iv. To Find the Particular Probability of a given relationship between X and Y, identify any cells that satisfy that condition and simply add them together

b. Marginal Distributions

i. We can partition by each of the variables inside of the joint distribution such that we have

ii. $\{X = x\} = \ \cup\{X = x,\ Y = y\}\ for\ all\ y\ \in Y$

iii. $\{X = x\} = \sum P(X = x,\ Y = y)\ for\ all\ y\ \in Y$

1. Given a joint distribution object, we can use its method .marginal as such
2. joint_dist.marginal('X'), where 'X' is a column name
3. OR joint_dist.both_marginals()

c. Conditional Distributions

i. Can create a conditional distribution using Python

1. Joint_dist.conditional_dist('Y', 'X')

a. Essentially produces a conditional distribution of Y given each different value of X

d. Updating Distributions

i. Conditioning gives us a way of updating our opinions based on new data

ii. You start out with a prior opinion about an unknown quantity. For every value of the unknown quantity, the data have a likelihood. (For N coin tosses we know the likelihood of getting 1 head, 2 heads etc based on our prior opinion – it should be 50, 50). After you see the data you apply Bayes Process and say based on the new information what is the probability of the true parameters. Then you keep going

e. Dependence and Independence

i. Conditional Distributions helps us formalize our intuitive ideas about whether two random variables are independent of each other

1. If the marginal distribution of Y changes depending on the X that we have selected, then we know that X has an influence on Y. Thus they are dependent. If knowing X has no affect on Y, then they are independent.

ii. **Independence –** P(B | A) = P(B). Or equivalently, P(AB) = P(A)P(B)

iii. **Random Variable Independence -** $P(X = x\ |Y = y) = P(X = x)$ for all x, y

iv. **Events of Random Variable Independence –** $P(X = x, Y = y\ ) = P(X = x)P(Y = y)$

f. Mutual Independence

i. Events $A_{1,} A_{2,} A_{3,} A_4$...... are mutually independent if given the values of any subset, chances of events determined by the remaining variables are unchanged

ii. Random Variables obey a similar law with $X_{1,} X_2$, X_3 , X_4 , … being mutually independent if given the values of any subset chances of events determined by the remaining variables are unchanged

g. i.i.d Random Variables

i. Suppose distribution of X is given by $P(X = i) = p_i\ for\ all\ i = 1{,}2 \dots n$. Suppose Y = X, then what is P(X = Y)

1. $P(X = Y) = \sum_{i=1}^{n} P(x = i, y = i)$
2. $P(X\ = y) = \sum_{i=1}^{n} P(x = i)P(y\ = i)$

3. $P(X = y) = \sum_{i=i}^{n} p_i^2$

Chapter 5: Collections of Events

1. Bounding the Chance of a Union
    a. $P(A \cup B) = P(A) + P(B\,'\backslash'AB)$
    b. $P(A \cup B) = P(A) + P(B) - P(AB)$
    c. Boole's Inequality – provides an upper bound for the probability of the union of n events
        i. $\max\{P(A_i): 1 \leq i \leq n\} \leq P(\bigcup A_i) \leq \sum A_i$
        ii. Essentially says that the Union of n different events is at the very least the size of the largest event and at the very max the sum of all the Probabilities of different events
        iii. Bonferroni Method –
            1. Assume we have 5 samples and want to make it so that all 5 are accurate to degree 95%. The complement of this situation is that least one is bad, which means that if P(A_i) represents A_i being accurate $A_i{}^C$ represents it being inaccurate. So if even one is bad that the chance that $\bigcup_{i=1}^{5} A_i{}^C$ is equal to .05 which we know has an upper bound of $\sum_{i=1}^{5} A^C{}_i$ which means each can be equal to .01 which entails each sample has to have an average accuracy of 99% to be certain the chance all 5 is accurate is equal to or greater than .95
2. Inclusion Exclusion
    a. $P(\bigcup_{i=1}^{n} A_i) = \sum_{i=1}^{n} P(A_i) - \sum\sum_{1 \leq i<j\leq n} P(A_i A_j) + \cdots..$
3. The Matching Problem
    a. Consider a random permutation of n elements each labeled from 1 to n. Whats the probability that a given element with the label i lands at position i
    b. This is equal to $\frac{1}{n}$ as each position is equally likely and there is exactly one position that satisfies the condition we have set
    c. For any number from 1 to *n* we can find the chance that that many envelopes land at corresponding position. We place *n* envelopes in their place and permutate the other n – k envelopes and divide by the total permutations (n!) and get the probability
    d. P(No Match) = 1 – P(At least One Match) = $= 1 - P(\bigcup_{i=1}^{n} A_i)$ *where* A_i *represents the chance that position i is matched*
    e. Since there are n elements each with probability $\frac{1}{n}$ and $\frac{n(n-1)}{2}$ elements with probability $\frac{1}{n(n-1)}$ and so on, the equation simplifies to $1 - \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!}$.. which is a *e* taylor series of exponent -1
4. Sampling without Replacement

a. Symmetry: For each fixed *i,* the coordinate X_i is an integer between 1 and n assuming a normal permutation from 1 to n. To find the marginal distribution of X_i we need to find $P(X_i = k)$ for each k in the range 1 to n.
    i. Equal to $\frac{1}{n}$ and it doesn't depend on i which means that it is uniform

b. Simple Random Samples: sample drawn at random without replacement from a finite population. The sample is a subset of the population not a rearrangement of the entire population
    i. This is essentially combinatorials and just choosing 5 particular, or 6 particular etc from a 50 cards etc
        1. Can do this using scipy import misc
            a. Misc.comb(52,5)

c. The Number of Simple Random Samples
    i. $\binom{N}{n}$ is choosing a sample of size n from size N and each one are equally likely

d. Counting Good Elements in a Simple Random Sample
    i. Suppose a population of N individual contains G good individuals and you take a simple random sample of size n. How many samples contain g good elements?
        1. Pick g individuals from a sample of G and do this in $\binom{G}{g}$
        2. For eac choice of these good there are $\binom{N-G}{n-g}$ choices of bad individuals
        3. The chance of getting g good elements in the sample is
            a. $\frac{\binom{G}{g}\binom{N-G}{n-g}}{\binom{N}{n}}$ is the chance of getting exactly g good elements in the sample
        4. These are called **hypergeometric** probabilities because the formula is related to the hypergeometric series of mathematics

Chapter 6: Random Counts

1. These form a class of random variables that are of utmost importance. General setting is that there a number of trials each which can be a success or a failure. Random count is the number successes among the trials.
2. Indicators ad Bernoulli (p) Distribution
    a. Consider a trial that can only result in a success or a failure. The number of successes X is thus a zero one valued random variable and is said to have a Bernoulli distribution.
    b. Counting is the Same as Adding Zeros and Ones
3. Binomial Distribution
    a. Let X_1, $X_{2,}$ X_3….. be i.i.d Bernoulli random variables and let $S_n = X_1 + X_2 + X_3 + \cdots X_n$. That's a formal way of saying:

i. Suppose you have a fixed number (n) of trials

ii. Trials are independent

iii. The probability of success is each to p

b. Let S_n be number of successes in n independent Bernoulli(p) trials. It then has a binomial distribution with parameters n and p.

i. $P(S_n = k) = \binom{n}{k} p^k (1-p)^{n-k}$

ii. $(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^k b^{n-k}$

iii. from scipy import stats, stats.binom.pmf(3,7, 1/6)

1. can also pass an array as the first argument and it will return probability for each of the

c. Binomial Histograms

i. First create a table:

1. K = Np.arange(n+1)

2. Binom_prob = stats.binom.pmf(k, n, p)

3. Binom_dist = Table().values(k).probability(binom_prob)

4. Plot(Binom_dist)

5. Plt.xlim(x, y) will zoom in on a specific range (x,y)

ii. Both the number of samples and the probability influence the shape of the graph

d. Binomial Distribution is valid only with: finding the number of successes in a known number of independent trials with the same probability of success for each trial

e. HyperGeometric Distribution

i. Let N = G + B, where G represents the good elements and B represents the bad elements.

ii. Number of Good Elements in a simple Random Sample

1. $P(X = g) = \frac{\binom{G}{g}\binom{B}{b}}{\binom{N}{n}}, b + g = n$

2. Can calculate using stats.hypergeom.pmf(values_we_want, Pop_size, Good_elements, sample_size)

f. Relation with Binomial

i. Suppose you sample with replacement from a population, then there is the chance G/N of selecting a good individual each time, with a population of n.

g. Odds Ratio

i. Binomial(n, p) involves powers and factorials which become irritating to calculate once the numbers become large enough

1. We can simply the process a little bit

ii. Consecutive Odds Ratio

1. The kth consecutive odds ratio $= R(k) = \frac{P(k)}{P(k-1)}$

2. P(0) = $(1-p)^n$, P(1) = P(0)R(1), P(2) = P(0)R(1)R(2)

3. Utilizes the common factorial elements of all the numbers, which results in $R(k) = \frac{\left(\frac{n+1}{k}-1\right)p}{1-p}$
4. This allows for no factorials to be involved at the expense of a little less clean formula
5. R(k) is a pure function of k as the rest of the variables are constant, which means that as k increases, R(K) decreases. This implies that once the binomial distribution passes a certain *k* it will continually decrease from that point onwards
6. Mode of Binomial – **mode of a binomial distribution is the possible value that has the highest probability.**
    a. The largest K for which R(K) > 1 has to be a mode; after this the histogram is falling and before this it was rising
    b. $\frac{n+1}{k} - 1 \geq 1 * \frac{1-p}{p}$
    c. $\frac{n+1}{k} - 1 \geq \frac{1}{p} - 1$
    d. $\frac{n+1}{k} \geq \frac{1}{p} \rightarrow k \leq p(n+1)$
    e. There can be two modes if the R(K) = 1, because that means P(K) = P(K – 1) and thus they have the same probability of being chosen
7. The Law of Small Numbers
    a. Consecutive odd ratio helps us to derive an approximation for the distribution when *n* is large and *p* is small and is called the Law of Small Numbers because we are expecting a small number of successes -> it approximates a distribution given the probability of success is small resulting in a tightly distributed model
    b. Binomial formula is rather clunky which results in the fact that we would rather approximate to some degree of certainty but quickly then relying on exactness and inefficiency.
    c. Let n → infinity and p_n → infinity but in a way such that $np_n > 0$.
        i. Let $P_n(K)$ be the probability of k successes given the binomial distribution *bin(n, p_n)*
            1. Then $P_n(0) = (1 - p_n)^n = (1 - \frac{np_n}{n})^n = (1 - \frac{u}{n})^n = e^{-u}$
            2. We can solve for the general k equation using the consecutive odds ratio

3.
$$\begin{aligned} P_n(k) &= P_n(k-1)R_n(k) \\ &= P_n(k-1)\frac{n-k+1}{k}\cdot\frac{p_n}{1-p_n} \\ &= P_n(k-1)\left(\frac{np_n}{k}-\frac{(k-1)p_n}{k}\right)\frac{1}{1-p_n} \\ &\sim P_n(k-1)\cdot\frac{\mu}{k} \end{aligned}$$

4. And then use induction to conclude that the distribution is $P(k) = \frac{e^{-u}u^{k}}{k!}$

d. Poisson Approximation to the Binomial

   i. Let n → infinity and p_n → 0, in a way that their product approaches a value u greater than zero. For large n, we can approximate the probability by using a Poisson Distribution

   ii. Can use stats.poisson.pmf in the exact same way as stats.binom.pmf

      1. Poisson only has one parameter (equal to the product of n and p_n)

Chapter 7: Poissonization

1. Poissonization – binomial (n, p) random variable has a finite number of values; it can only be between 0 and n. However, we should expand this to include infinite spaces especially as *n* approaches very large numbers

   a. $P(k) = \frac{e^{-u}(u^{k})}{k!}$

   b. We have to state the additivity axiom of probability theory in terms of countably many outcomes: If A1, A2 … are mutually exclusive, then:

      i. $P(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} A_i$

      ii. This is only infinite but can be reduced to finite additivity

   c. Poisson Distribution

      i. A R.V X has a Poisson distribution if $P(X = k) = e^{-\mu}\frac{(\mu^{k})}{k!}, k = 0, 1, 2 \ldots$

      ii. The terms of proportional to the Taylor Series expansion of $e^{-\mu}$

   d. The Mode of the Poisson Distribution is the integer part of the μ. If it is an integer both μ and $\mu - 1$ are modes

   e. The Cumulative Distribution Function

      i. If we want to find $P(X < 5)$ or $P(X \geq 4)$ we can use the c.d.f

         1. Can utilize the stats library to do so

a. Stats.binom.cdf, or stats.poisson.cdf followed by the parameters and then a number is equal to $P(X \leq number)$

2. Poissonizing the Binomial

a. Sums of Independent Poisson Variables

i. Let X have the Possion(μ) distribution and let Y be an independent Poisson(λ). Then $S = X + Y, and\ S\ has\ a\ Poison(\mu\ +\ \lambda)$

ii. An important application of this is that if you have *n* I.i.d Poisson(p) variables then their sum becomes a Poisson with parameter np.

b. Randomizing the Number of Bernoulli Trials

i. In a fixed number of Bernoulli Trials the number of successes and failures are directly proportional; if you know one then you can find the other and they are not independent

1. However, if the number of trials is random and has a Possion Distribution we come across a interesting finding

2. Let N have the Poisson (μ) distribution, Let S be the number of successes in N i.i.d Bernoulli trials with parameter p.

a. $P(N = n, S = s) = e^{-\mu}\frac{(\mu^n)}{n!} * \frac{(n!)}{s!(n-s)!}\ p^s\ (1-p)^{n-s}$

b. $P(S = s) = \sum_{n-s=0}^{\infty} P(N = n,\ S = s)$

c. $P(S = s) = e^{-\mu p}\frac{(up)^s}{s!}$

d. Thus if the number of trials is fixed, we know the distribution of the Bernoulli i.i.d R.V's follows a binomial distribution, but if the number of trials is variable as per a Poisson distribution, the distribution of successes follows a Poisson distribution

e. You can repeat this process with the "failures" of th same distribution and come to the conclusion that both the successes and the failures are independent

ii. Summary of Poissonization of Binomial

1. Suppose you run N i.i.d Bernoulli(p) trials where N has the Poisson(μ) distribution. Let S be the number of successes and F be the number of failures.

a. S has the Poisson(μp) distribution

b. F has the Poisson(μ(1-p)) distribution

c. S and F are independent

3. Multinomial Distribution

a. Extension of the binomial distribution to include scenarios in which multiple different outcomes are possible, not just 2

b. Suppose we are running n i.i.d trials where each trials can result in one of k different classes. For each *i = 1,2,3…k* let the chance of getting Class i on each trial be p_i such that $\sum_{i=1}^{k} p_i = 1$. Let N_i be the number of trials that result in Class i , so that $\sum_{i=1}^{k} N_i = n$

c. Thus this results in the joint distribution of N_1, N_2, N_3 … N_k being:

i. $P(N_1 = n_1, N_2 = n_2 \dots . N_k = n_k) = \frac{n!}{n_1!n_2!n_3! ..n_k!} p_1^{n_1} p_2^{n_2} \dots p_k^{n_k}$

ii. The distribution of any N_i is binomial as we are simply selecting a portion from N each with a certain probability of success while the rest need to not meet that requirement

d. Poissonization

i. For each *i = 1,2….k* the distribution of N_i is Poisson(μp_i)

1. The counts N_1, N_2, N_3 …. In the k different categories are mutually independent
2. Creating a random sample allows us to simply multiple independent probabilities to get the equation we want

Chapter 8: Expectation

1. Expectation

a. The distribution of a random variable gives us detailed information about how probability is distributed over all the possible values of a variable. The expectation is a way of gaining insight into a distribution by looking at its "middle".

2. Definition

a. The **expectation** of a R.V X denoted E[X] is the average of all possible values of X weighted by their probabilities

i. Either on X's domain: $E(X) = \sum_{w \in \Omega} X(\omega)P(\omega)$

ii. Or on it's range: $E(x) = \sum_{all\ x}(x * P(X = x))$

iii. If two R.V have the same distribution then they have the same Expectation

b. You can either sum the multiplication of the value array and the probability array OR you can put it into a distribution table and call .ev() to get the value

i. Example_dist.ev() & Plot(example_dist, show_ev = True)

c. The Expectation is akin to the center of mass for the distribution

d. Long Run Average

i. The Expectation can also be viewed as a long run average if we sampled a large amount of times from the sample

1. Simulated = example_dist.sample_from_dist(1000000)
2. np.mean(simulated_x) approximate E(X)

e. Constant

i. E(c) = c

f. Bernoulli and Indicators

i. If X has Bernoulli (p) indicators then P(X = 1) = p and P(X = 0) = 1- p. So, E(X) = 1p = p

ii. Let A be an event, and the indicator of A is the random variable I_A that is 1 if A occurs and 0 if A doesn't occur. Thus I_A has the Bernoulli Distribution (P(A)) which means that $E(I_A)$ = P(A)

    g. Uniform on an Interval
        i. If a R.V is uniform on an interval then $E(X) = \frac{(a+b)}{2}$
    h. Poisson
        i. E(X) = μ (already done the calculations before)
    i. Existence
        i. If X has countably many values then we are taking partial sums in order to find the Expectation. However, expectation doesn't exist if the series doesn't converge which can happen
            1. OOS but we should know that there are not well defined and infinite expectations

3. Additivity
    a. Calculating expectation by plugging and chugging works but often can be cumbersome; thus we want shortcuts
    b. Additivity of Expectation:
        i. E(X + Y) = E(X) + E(Y)
    c. Simplest way to apply the AOE is to break apart large R.V into indicators or more manageable distributions
    d. Linearity of Expectation
        i. E(ax + B) = aE(X) + B
    e. Unbiased Estimator
        i. Assume that λ is a parameter of X and that E(X) = λ, then X is an unbiased estimator of λ.
            1. This means that in the long run, if we keep estimating the parameter based on X, the average will approach the true value of the parameter
    f. Method of Indicators
        i. If we have a N that measures the number of successes in k trials then we can code each of the k trials as an individual indicator. Thus the Expectation for N becomes the sum of the Expectation for each indicator

4. Expectations of Functions
    a. If Y = g(X) then we can calculate the domain of Y in three ways;
        i. $E(Y) = \sum_{all\ y} y * P(Y = y)$
        ii. $E(Y) = E\big(g(X)\big) = \sum_{w \in \Omega} (g \circ X)(\omega)P(\omega)$
        iii. $E(Y) = E\big(g(X)\big) = \sum_{all\ x} g(x)P(X = x)$
    b. Tips and Tricks
        i. Python Code
            1. If you have a dist_table for X, then you can simply add a new column to it that transforms it:
                a. Dist_y = dist.withColumn('name', np.abs(dist.column('value')))

b. Then simply sum(dist_y.column(‘name’)*dist_y.column(‘probability))

ii. If you know E(X) and E(X(X-1)) then you can find E(X^2) as per linearity of additivity

1. E(X^2 – X) = E(X^2) – E(X), E(X^2) = E(X) + E(X(X-1))

Chapter 9 : Conditioning

1. Conditioning Revisited: we’re gonna be looking at random processes indexed by time.
2. Probability by Conditioning
    a. Suppose we have a gambler who plays a game; if a flipped coin lands head, he’ll win one dollar but if its tails he loses a dollar. He starts with a dollars and will play till he either has zero or gains b dollars.
        i. If p_k represents the probability he will face ruin on the kth flip, then we have $p_k = \frac{p_{k-1}}{2} + \frac{p_{k+1}}{2}$ and then we can simplify this to achieve $\frac{p_k}{2} + \frac{p_k}{2} = \frac{p_{k-1}}{2} + \frac{p_{k+1}}{2}$ = $p_k - p_{k-1} = p_{k+1} - p_k$
        ii. This means that the space between every succession is equal thus it is linear function which means that we can use our endpoints to construct a function
        iii. Our endpoints are (-a, 1) and (b, 0) where we are assuming 0 is our starting position. Thus our line’s slope = $\frac{-1}{a+b}$ with intercept $\frac{b}{a+b}$ which means that the probability of ruin at 0 is equal to $\frac{b}{a+b}$
    b. Same Scenario but with a unfair coin:
        i. $p_k = p(p_{k+1}) + (1-p)(p_{k-1})$
        ii. $p(p_k) - p(p_{k+1}) = (1-p)(p_k) + (1-p)(p_{k-1})$
        iii. $p(p_k - p_{k-1}) = (1-p)(p_k + p_{k-1})$
        iv. This means that each term has a constant ratio between them which means we can still construct a geometric equation
        v. If $r = \frac{p}{1-p}\ then\ p_k = (r^{a+k} - r^{a+b})\ /\ (1 - r^{a+b})$
        vi.
3. Expectation with Conditioning
    a. If T and S are R.V’s conditioned on the same space, conditioning on S might be a good way to find the probability of T if S and T are related.
        i. Given a value of S, we can find what the expectation of T should be for that, such that we have E(T | S =s) , along with P(S = s)
        ii. We can thus interpret E(T) as the average of the conditional expectations of T given the different values of S, weighted by the probability of those values
        iii. Thus we now have an R.V as for each value of s, there is an associated function for the same probability, with the function being = E(T | S= s)
    b. Iterated Expectations

i. E(E(T|S)) = E(T)

c. Other Properties of Conditional Expectation

i. Additivity: E(T + U |S) = E(T |S) + E(U | S)

ii. Linear Transformation: E(aT + b | S) = a E(T |S) + b

iii. **A variable can be treated as if a constant in the conditional expectation environment because we are simply conditioning on a variety of constnats**

1. E(g(S) | S) = g(S)

2. E(g(S)T | S) = g(S) E(T|S)

4. Expected Waiting Times

a. Waiting till H

b. Examples…Examples etc

Chapter 10: Markov Chains

1. Markov Chain Overview

a. Stochastic process is a collection of random variables on a probability space. We study a kind of process that evolves over discrete intervals of time. It starts at X_0 and at time n, it is at R.V X_n

b. Markov Chains form a class of stochastic processes and essentially the distribution in the future depends on the present value, but not how it arrived at the present value

c. Formally:

i. For each $n \geq 1$,
$conditional\ distributon\ of\ X_{n+1}\ given\ X_0, X_1, X_2 \dots X_n depends\ only\ on\ X_n$

ii. $P(X_{n+1} = i_{n+1} | X_0 = i_o, X_1 = i_1 \dots X_n = i_n) = P(X_{n+1} = i_{n+1} | X_n = i_n)$

d. The state space is the set of possible values of the random variables in the chain

i. We will restrict the state space to be discrete and typically finite

e. Conditional Independence:

i. R.V X, Y are said to be *conditionally independent of Z* if:

1. P(X | Y, Z) = P(X | Z), thus Y provides no affect when Z's influence is already counted

ii. Markov Property says that the past and future are conditionally independent given the present

2. Transitions

a. Conditional Probabilities in the product are called transitional probabilities. For states, *i,j* the conditional probability $P(X_{n+1} = j \,|X_n = i) : \ One\ Step\ Transition\ Probability$

b. Stationary Transition Probabilities

i. When one step transition probabilities don't depend on n , they are calel stationary or time-homogenous; all of the ones we study will be time homogenous

ii. $P(i,j) = P(X_{n+1} = j \mid X_n = i) = P(X_1 = j \mid X_0 = i)$

c. One Step Transition Matrix

i. Matrix P whose (*i, j)* element is $P(i,j) = P(X_1 = j \,|X_0 = i)$

1. Square matrix which is indexed by the state space
2. Each **row** of P is a distribution; for each state *i* and each *n,* Row *i* of the transition matrix is the conditional distribution of X_{n+1} given that X_n = i

ii. We can construct this inside Python by doing the following:

1. Create an array for all the states
2. Create a function that takes in a i and a j and then outputs the transition probability
3. Use the Markov Chain object with method:

a. MarkovObject = MarkovChain.from_transition_function(state_array, markovfunc)

b. Can predict the probability of a certain path by using the method prob_of_path

i. MarkovObject.prob_of_path(start, [list of paths])

c. Simulate paths of the chain using the simulate path method:

i. MarkoObject.simulate_path(start, num_steps)

1. Returns a path and can be plotted by passing plot = True into the method

d. N-Step Transition Matrix

i. For state i and j, the chance of getting from i to j in n steps is called the n -step transition probability

1. $P_n(i,j) = P(X_n = j \,|X_0 = i)$
2. Can use the .transition_matrix(n) to get the n-step matrix given a one step matrix
3. The n step matrix is simply equal to $(One\ Step\ Matrix)^n$
4. The MarkovObject isn't actually a matrix however and we can use .get_transition_matrix(n) to get an numpy matrix
5. Can use np.linalg.matrix_power(Numpy_Matrix, power) to raise a matrix to a certain power

ii. The Long Run

1. If you increase n to infinity, the chain will exhibit memory loss in the sense that the starting state of the matrix no longer matters

3. Deconstructing Chains

a. Let S be a countably infinite or finite set of states; Any stochastic matrix indexed by the state space S is a transition matrix of some markov chain with state space S

b. Communication:

i. We say that i leads to j, I → j, or formally if

1. There is a path of positive probability that starts at *i* and ends at *j*
2. OR, (equivalently) there is some *n* > 0 ,such that $P_n(i,j) > 0$

        ii. If both i and j communicate between each other, they then are said to "communicate" and if all states in a state space communicate with each other then it is called irreducible

4. Period
    a. States can be periodic if we are working within discrete time.
        i. A state has period *d* if starting at *i* the chain can only come back to *i* only at times that are multiples of d. That is *d* is the greatest common divisor of the set of all *n* such that $P_n(i, j) > 0$
        ii. Periodic functions (ones with a d > 1) cause issues with long run behavior as they have zero probabilities for many *n*'s thus causing issues with limit statements
        iii. We will focus on aperiodic functions for the purpose of simplicity for scope. Thus if we know a markov chain is irreducible then we know that all the states have the same period, and we simply have to find one of them
5. Long Run Behavior
    a. Every irreducible aperiod markov chain exhibits regularity over a long run
    b. $Let\ X_0, X_1 \dots be\ an\ irreducible, aperiodic\ Markov\ Chain\ on\ a\ finite\ Space\ S$
    c. $For\ all\ states\ i\ and\ j, P_n(i, j) \rightarrow \pi(j)\ as\ n \rightarrow\ \infty$
        i. Essentially the n-step transition probability converges to a distribution equvalent across all possible values of i
    d. Moreover
        i. $\pi(j) > 0\ for\ all\ j$
        ii. $\sum_{j\ \in S} \pi(j) = 1$
    e. Properties
        i. Vector π is the unique solution to the balance equation π = Pπ
        ii. If for some n, the distribution of X_n is π then the distribution for X_m is also π if m > n. Thus π is called the stationary or steady state distribution of the chain
        iii. For each state j, the j^{th} entry of the π vector of π(j) is the expected long run proportion the time chain spends at *j*
    f. Uniqueness
        i. A finite, aperiodic, irreducible markov chain has exactly one stationary distribution
    g. Can use the .steady_state() method for any given Markov Object

Chapter 11: Reversing Markov Chains

1. Overviews
    a. Analyzing long run behavior of markov chains helps us quantify random phenomenon and particularly in data science are used to draw random samples from a complex distribution. They can also be used to approximate expectations of random quantities whose distributions are either overly complicated or involve too many unknowns

b. Sometimes we can do this using the Markov Chain Monte Carlo (MCMC) in which we create a markov chain with a complicated distribution as its stationary distribution and then running it multiple times

   i. Have to know how to reverse a Markov chain

2. Detailed Balance

   a. If the chain is in a steady state that we know that is balanced; or essentially the number of particles leaving any state j is the same as the number of particles entering it

      i. $\pi(j) = \sum_{k \,\epsilon\, S} \pi(k)P(k,j)$ where π(k) is the proportion of particles leaving *k*

      ii. We also have detailed balance which goes more in depth to state the relationship between each of the states and j

         1. $\pi(i)P(i,j) = \pi(j)P(j,i)$

      iii. The detailed balance implies the balance equations but in particular the detailed balance provides us key advantages

         1. Balance equations are simple

         2. There are lots of them, for s states there $\binom{s}{2}$ equations

3. Reversibility

   a. Reversed Process

      i. Let X_0 , X_1 , X_2 ….X_n be an irreducible Markov Chain with stationary distribution π. Let's consider a reversed sequence Y_1, Y_2 , Y_3 … Y_n where $Y_k = X_{n-k}$

      ii. $P(Y_1 = j \,|Y_o = i) = \frac{\pi(j)P(j,i)}{\pi(i)}$

      iii. The forwards chain (X chain) is reversible for all *n* if the reversed sequence has the same one -step transition probabilities as the original

         1. $\frac{\pi(j)P(j,i)}{\pi(i)} = P(i,j), for\ all\ i,j$

         2. $\pi(i)P(i,j) = \pi(j)P(j,i)$ which the balance equations which means the chain is reversible if all the detailed balance equations have a positive solution

4. Markov Chain Monte Carlo → → ERROR I DON'T UNDERSTAND

Chapter 12: Standard Deviation

1. Overview

   a. Allows us to measure the distance of a random variable from the Expected Value (μ) a.k.a the mean

      i. Can't just do E(X - μ_x) because that results in E(X) - μ_x which means its zero

2. Definition

   a. Let X be a R.V with expectation μ_x. The Standard Deviation (SD) denoted a SD(X) or σ_X is the root mean square deviations from the mean

      i. $SD(X) = \sigma_X = \sqrt{E((X - \mu_X)^2)}$

      ii. $Var(X) = {\sigma_X}^2$ and is closely related to the Pythagorean Theorem

      iii. Given a distribution object (table) you can use method .sd() to get St. Dev

b. $SD(aX + b) = |a|\sigma_X$

c. Computational Formula for Variance

   i. ${\sigma_X}^2 = E((X - \mu_X)^2) = E(X^2) - {\mu_X}^2$

3. Prediction and Estimation

   a. If we wanted to make a guess among all the choices of c when guessing the value of a Random Variable, it would make sense to pick μ_X as it would minimize the standard deviation of your answer, meaning you would be closest to you answer the majority of the time.

      i. The Mean as a Least Squares Predictor

         1. The predictor μ_X has the smallest mean squared error among all the possible choices c and that value is the Variance of X, a.k.a the square of the standard deviation

      ii. Comparing Estimates

         1. If we have two competing estimators of a parameter, we can use expected values and standard deviations in order to figure out which one is more accurate probabilistically

         2. Many times even though an estimator can be more accurate in the sense that its expectation is closer to the parameter mean, it can have a larger standard deviation causing impreciseness.

            a. Bias-Variance trade off, a.k.a Accuracy vs Precision

      iii. Tail Bounds

         1. If you know E(X) and SD(X) then it is possible to figure out the tail bounds of the random variable.

         2. Suppose g(X) and h(X) are two functions such that g(X) ≥ H(X) for all possible values of x, then E(g(X)) ≥ E(h(X))

            a. Lets now consider two functions g(x) = x and h(X) = c I(x ≥ c)

            b. E(g(X)) = E(X) while E(h(X)) = c E(I) = c P(X ≥ c) =

               i. E(X) ≥ E(h(X)) → E(X) ≥ cP(X ≥c)

         3. Markov's Inequality

            a. Let X be an non negative random variable and for any c ≥ 0

               i. $P(X \geq c) \leq \frac{E(X)}{c}$

               ii. $OR\ P\left(X \geq k\mu_X\right) \leq \frac{1}{k}\ for\ all\ k > 0$

         4. Chebyshev's Inequality

            a. $P(|X - \mu_X| \geq z\sigma_X) = \frac{1}{z^2}$

            b. Chebyshev's Inequality makes no assumption about how a distribution looks and whether it is poisson ,geometric uniform etc…, the inequality still applies no matter what

            c. HOWEVER, if we know the shape of the distribution often we can do better than what is required

         5. Standard Units

a. We can create a R.V Z for any random variable X, which abides by the following translation

i. $Z = \frac{X - \mu_X}{\sigma_X}$

b. By linear function rules, E(Z) = 0 and SD(Z) = 1

c. Thus chebsyshev's for Z states $P(|Z| \geq z) \ \leq \frac{1}{z^2}$

d. Another form is

i. $P(|X - \mu_X| \geq c) \leq \frac{{\sigma_X}^2}{c^2}$

6. Heavy Tails

a. Expectations and Standard Deviations aren't useful when we encounter elongated and heavy tail skew

b. Often times we encounter distributions such as Zipf's Law which states an inverse probability to rank which means that as n gets large the Expectation and standard deviations approach infinity

Chapter 13: Variance via Covariance

1. Overview

a. μ_X = E(X), σ_X = SD(X) and D_X = X - μ_X or the deviation of X from its mean such that Var(X) = E($D^2{}_X$)

b. Variance of Sum

i. S = X + Y → E(S) = μ_X + μ_Y and D_S is the sum of the deviations of X and Y

ii. Var(S) = E($D^2{}_S$) = E[(D_X + D_y)^2] = Var(X) + Var(Y) + 2E($D_X D_Y$)

iii. The extra term in that equation is twice the covariance of X and Y Cov(X, Y), and is the expected product of the deviations of X and Y

1. Cov(X, Y) = E[(X - μ_X) (Y - μ_y)]
2. $Var(X + Y\ ) = Var(X) + Var(Y) + 2Cov(X, Y)$
3. $Cov(X, y) = E[(\ X - \mu_X)(Y - \mu_X)]$

2. Properties of CoVariance

a. $Cov(X, c) = 0\ , thus\ a\ constant\ doesn't\ vary$

b. $Cov(X, X) = E(D_X D_X) = E({D_X}^2) = Var(X)$

c. $Cov(X, Y) = Cov(Y, X)$

d. $Cov(X, Y) = E\big[(X - \mu_X)\big(Y - \mu_y\big)\big] = E[(XY - X\mu_Y - Y\mu_X + u_X\mu_Y)] =$

e. $E(XY) - E(Xu_Y) - E(Y\mu_X) + E(\mu_X\mu_Y) = E(XY) - \mu_X\mu_Y$

f. $Cov(X + Y, Z) = Cov(X, Z) + Cov(Y, Z)$

g. Main Property : Bilinearity

i. $Cov(aX, bY) = abCov(X, Y)$

ii. And by induction $Cov(\sum_{i=1}^{n} a_i X_i, \sum_{i=1}^{n} {}_{\_}b_i Y_i) = \ \sum_{i=1}^{n} a_i \sum_{i=1}^{n} b_i Cov(X, Y)$

iii. Such allows us to solve equations such as these

1. $Cov(10X - Y, 3Y + Z) = 30Cov(X,Y) + 10Cov(X,Z) - 3Cov(Y,Y) - Cov(Y,Z)$

h. Independence Implies No Correlation

i. $E(XY) = E(X)E(Y) \rightarrow E(XY) - \mu_X\mu_Y = 0 \rightarrow Independent\ R.V\ have\ Cov = 0$

3. Sums of I.I.D Samples

a. $Let\ X_1, X_2 \ldots X_N be\ random\ variables, then\ S_n = \sum_{i+1}^{n} X_i$

b. $Var(S_n) = Cov(S_n, S_n) = \sum_{i=1}^{n}\sum_{j=1}^{n} Cov(X_i, X_j) = \sum_{i=1}^{n} X_i + \sum\sum_{1<i<j<n} Cov(X_i, X_j)$

c. Therefore if all X_1 ..X_n are independent then the covariances become 0, leaving us with

i. $Var(S_n) = \sum_{i=1}^{n} Var(X_i)$

d. Sum of I.I.D Samples

i. $E(S_n) = n\mu, Var(S_n) = n\sigma^2, SD(S_n) = \sqrt{n}\sigma$

e. Variance of the Binomial

i. Let X = *Bin(n, p),* then we know that $X = \sum_{i=1}^{n} I_j$ where each I is an i.i.d that takes on a value 1 with probability p, thus having $E(I) = p, Var(I) = p(1-p)$

ii. Thus $E(X) = np, Var(X) = np(1-p)$

4. Sums of Simple Random Samples

a. Indicator CoVariance

i. Let A & B be two events and let I_A and I_B be the indicators of A and B respectively

ii. $Cov(I_A, I_B) = E(I_A, I_B) - E(I_A)E(I_B) = P(AB) - P(A)P(B)$

iii. If the covariance is positive we can rearrange to find that P(B|A) > P(A) which entails that the probability A occurs is higher if B has occurred

b. Variance of the HyperGeometric

i. We know that $X = \sum_{i=1}^{n} I_i\ where\ I_i is\ the\ indicator\ that\ i\ draws\ a\ good\ element$

ii. $E(I_i) = \frac{G}{N}\ for\ each\ j, \rightarrow E(X) = \frac{nG}{N},$

iii. $Var(I_j) = \frac{G}{N} * \frac{B}{N}, where\ B = N - G\ and\ the\ Cov\ is\ equivalent\ between\ any\ two\ indicators$

iv. $Cov(I_j, I_k) = E(I_j, I_k) - E(I_j)E(I_k) = \frac{G}{N} * \frac{G-1}{N-1} - \frac{G}{N} * \frac{G}{N}$

v. $Var(X) = \sum_{i=1}^{n} Var(I_i) + \sum\sum_{1 \le j \ne i \le n} Cov(I_j, I_i) \rightarrow \frac{np(1-p)(N-n)}{N-1},\ p = \frac{G}{N}$

c. Variance of a Simple Random Sample Sum

i. $Var(S_n) = n\sigma^2 + n(n-1)Cov(X_1, X_2)$

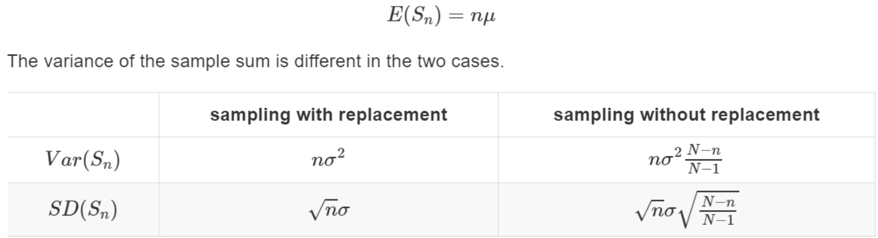
ii. If we sample every single element from a population (i.e n = N) then we will have a variability of 0, thus we have a new equation

$$0 = N\sigma^2 + N(N-1)Cov(X_1, X_2)$$

$$Cov(X_1, X_2) = -\frac{N\sigma^2}{N(N-1)} = \frac{\sigma^2}{N-1}$$

Thus our final equation becomes $Var(S_n) = \frac{n\sigma^2(N-n)}{N-1}$

5. Finite Population Correction

   a. 

   $$E(S_n) = n\mu$$

   The variance of the sample sum is different in the two cases.

	sampling with replacement	sampling without replacement
$Var(S_n)$	$n\sigma^2$	$n\sigma^2\frac{N-n}{N-1}$
$SD(S_n)$	$\sqrt{n}\sigma$	$\sqrt{n}\sigma\sqrt{\frac{N-n}{N-1}}$

   b. There is only one difference between the two method which is the finite population correction = $\frac{N-n}{N-1}$, with the name arising because sampling with replacement is the same as samping without replacement from an infinite population
   c. When N is moderately large (> 100) we have $\frac{N-n}{N-1} \approx \frac{N-n}{N} = 1 - \frac{n}{N}$, which means that if n is very small relative to N, then our FPC is approximately 1
   d. Non Effect of the Population Size
      i. SD of a simple random sample depends only on the sample size and the population SD provided that FPC is close enough to 1
         1. $SD(S_n) \approx \sqrt{n}\sigma$

Chapter 14: The Central Limit Theorem

1. Exact Distribution
   a. $P(X+Y=k) = \sum_j P(X=j,\ Y=k-j) \rightarrow X, Y\ are\ ind = \sum_j P(X=j)P(Y=k-j)$
   b. However, this can be hard to expand out to multiple different R.V's instead of just two as it involves a lot of overlapping terms
   c. Probability Generating Functions
      i. Let X be a R.V with possible values 1,2… N for some fixed integer . Let P(X =k) = p_k
      ii. $G_X(s) = \sum_{k=0}^{N} p_k s^k,\ -\infty < s < \infty$ where G_X is the probability generating function of X such that you can plug in any value of s and get the values through this function and that we could get all possible values and probability from it
      iii. We can convert the G_X into an expectation as it follows the same form of $\sum_j f(x)P(X=x), where\ f(x) = s^x \rightarrow G_X(s) = E(s^X)$
      iv. By the linearity of expectation we know that
         1. $G_{X+Y}(s) = G_X(s)G_Y(s)$
         2. The Probability Generating Function of The sum of Independent Random Samples is the product of their PGF's
   d. PGF of the Sum of I.I.D Sample
      i. Let X_1, X_2, X_3…X_N be i.i.id with a distribution of 0, 1….N. Let $S_n = X_1 + X_2 .. + X_n$, then the PGF of S_n = $G_{S_n}(s) = \left(G_{X_1}(s)\right)^n, -\infty < s < \infty$
      ii. As G_{X1} is a polynomial of degree N, the entire PGF for a sum of i.i.d variables is of degree nN, and as with any PGF

        1. $P(S_n = k) = coefficent\ of\ s^k\ in\ G_{S_n}(s)$

    iii. Our methodology now follows as such:

        1. Start with PGF of X_1
        2. Raise it to the power of n, and get the pgf of S_n
        3. Read the distribution of S_n off the pgf

2. PGF's in Numpy
   - a. We can use Numpy to raise a polynomial to an *nth* power fairly easily
     - i. Probs_X1 = np.array(0.1, 0.5, 0.4)
     - ii. Coeffs_X1 = np.fluidud(probs_X1) (reverses the array)
     - iii. Pgf_X1 = np.poly1d(coeffs_X1) (creates a polynomial from an array, with highest power first)
     - iv. Pgf_s3 = pgf_X1**3 (Raises polynomial to the third power)
     - v. Coeffs_s3 = pgf_s3.c (Gets the coefficients from the polynomial)
     - vi. Probs_S3 = np.fluidud(coeffs_s3)
   - b. A Function to Calculate The Distribution of S3
     - i. Can create a function to automate all the stuff above, but no point writing it all down
3. Central Limit Theorem
   - a. Central to the fields of probability, statistics and data science
   - b. Few Things to Go Over
     - i. Standard Units
       1. $Z = \frac{X - \mu_X}{\sigma_X} \rightarrow E(Z) = 0\,, SD(Z) = 1$
     - ii. Standard Normal Curve
       1. $\phi(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2}, -\infty < z < \infty$
     - iii. Terminology
       1. Curve has location parameter 0, akin to a mean
       2. Curve has scale parameter 1, akin to Standard Deviation
     - iv. Normal Curves
       1. General Formula is $\phi(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-u}{\sigma}\right)^2}$
   - c. Central Limit Theorem
     - i. Let X_1 , X_2 …X_n be i.i.d each with mean μ and SD σ. Let $S_n = X_1 + X_2 \ldots + X_n$, then we know that $E(S_n) = n\mu$ and $Var(S_n) = n\sigma^2$
     - ii. The Theorem states:
       1. When n is large, the distribution of the standard sum $\frac{S_n - n\mu}{\sigma\sqrt{n}}$ approximately follows the standard normal curve, regardless of the distribution of the individual X's
       2. Plot_norm(plot_interval, mean, sd)
       3. Stats.norm.cdf(right bound, mean, sd)

4. Plot_norm(plot_interval, mean, sd, left_end = left_end, right_end = rightend) highlights the marked area in gold

d. Standard Normal CDF

i. There is no mathematical approximation for the standard curve and it has to be calculated via approximation but we still give it a general notation which is

ii. $\Phi(x) = \int_{-\infty}^{x} \phi(z)dz, -\infty < z < \infty$ thus under this we can say that

iii. $IF\ n\ is\ large, P(S_n < x) \approx \Phi\left(\frac{\mathrm{x-n\mu}}{\sigma\sqrt{n}}\right)$

e. Binomial Distribution

i. Is is Poisson or Normal? We said that as n approaches infinity and p approaches zero, then binomial is poisson

1. We need to see whether the data is crunched around zero or not and we assume a general threshold of
2. Np and n(1-p) > 10 OR np(1-p) > 9

4. The Sample Mean

a. So for any given sum of i.i.d variables we see that as n increases the mean of the sum increases proportionally ($E(S_n) = n\mu$) and the $SD(S_n)$ is more spread out as it is = root(n)σ. However, the average of the Sum of i.i.d behaves much more differently

b. $E\left(\frac{S_n}{n}\right) = \frac{n\mu}{n} = \mu, SD\left(\frac{S_n}{n}\right) = \frac{\sqrt{n}\sigma}{n} = \frac{\sigma}{\sqrt{n}}$

i. The relation the SD has to the sample size means that as we increase our sample size, we will decrease our variability, by a factor of a power of .5 This mean if we increase our sample size by 9, we decrease our variability by 3

5. Weak Law Of Large Numbers

a. The mean of a large sample is close to the population mean with high probability

b. $Let\ X_1, X_2 \ldots X_n be\ i.i.d\ with\ mean\ \mu\ and\ SD\ \sigma\ and\ let\ \ X_n\ be\ sample\ mean.$

c. $P(|X_n - \mu| > \epsilon) < \frac{\sigma_{X_n}^2}{\epsilon^2} = \frac{\sigma^2}{n\epsilon^2} \to 0\ as\ n \to \infty$

6. Confidence Intervals

a. CLT implies that with 95% confidence, the sample mean is within 2 SD's of our population mean which means that with 95% confidence we can say that the population mean is within 2 standard deviations of our sample mean

b. General Definition

i. $Let\ \lambda\ be\ any\ confidence\ level\ from\ 0 - 100\ such\ that\ (-z_\lambda, z_\lambda)$

ii. $contains\ \lambda\ \%\ of\ the\ data\ \ then\ if\ n\ is\ large\ then$

iii. $\frac{\lambda}{100} \approx P\left(\mu \in X_n \pm \frac{z_\lambda \sigma}{n^{.5}}\right)$

Chapter 15: Continuous Distributions

1. Density and CDF

a. Let f be a non-negative function on the real number line and suppose

i. $\int_{-\infty}^{\infty} f(x)dx = 1$

    ii. Then f is a Probability Density Function (PDF) or just density

  b. Density is not the Same as Probability

    i. f(x) is not necessarily equal to P(x)

  c. Areas are Probabilities

    i. A R.V X is said to have density f if

$$for\ all\ a,b\colon\ P(a < X < b) = \int_a^b f(x)dx$$

  d. Cumulative Density Function

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(s)ds$$

$$f(x) = \frac{d}{dx}F(x)$$

2. The Meaning of Density
   a. When X can take on an infinite amount of numbers, then each probability value is equal to zero
   b. f(x)dx is the chance that X is around x, and the integral takes care of summing all possible values a such
   c. The function f represents the probability per unit length
3. Expectation

$$E\big(g(x)\big) = \int_{-\infty}^{\infty} g(x)f(x)dx\ \ , provided\ that\ the\ integral\ converges$$

4. Exponential Distribution
   a. A R.V has an exponential distribution with parameter λ if the density of T is given by

$$f_T(t) = \lambda e^{-\lambda t}, t\ \geq 0$$

   b. CDF and Survival Function
      i. $P(T\ \leq t) = F_T(t) = 1 - e^{-\lambda t}$
   c. Expectation and SD
      i. $E(X) = \frac{1}{\lambda}, Var(X) = \frac{1}{\lambda^2}$
   d. Median
      i. It is equal to where the survival function and the CDF intersect which is at log(2)E(X)
   e. Memoryless Property
      i. $P(T > t + s \mid T > t) = P(T > s)$
      ii. Essentially the contextualization of the probability doesn't matter
   f. The Rate
      i. λ is the instantaneous death rate for a reason that is irrelevant of me writing down and it just is as such
5. Calculus in Sympy
   a. Can do calculus in SymPy and it's a hell of a lot easier

i. From sympy import *
ii. Init_printing() (essentially creates a better formatting for when we print)
iii. Declare(‘x’, interval = (0,1)) (declares a variable and the range it can take)
iv. Density = 105* x**2 + (1-x)**4 (sets Density to a polynomial with x)
v. Total_area = Integral(density, (x,0,1)) (creates an integral called total area)
vi. Total_area.doit() (actually executes the program)
vii. Indefinite = Integral(density).doit() (will return the indefinite integral)
viii. Indefinite.subs(x, 0) (allows you to substitute a value for a variable)
ix. Declare(‘lambda’, positive = True) (declares a variable as only taking positive values)
x. Integral(polynomial, (t, 0, oo)).doit() (oo is the same as infinity)

Chapter 16: Transformations

1. Overview
   a. We will work on finding the density of Y if Y = g(X)
2. Linear Transformations
   a. Let X be a R.V with density f_X and Y = aX + b, then:

$$f_Y(y) = f_X\left(\frac{y-b}{a}\right)\frac{1}{|a|}$$

3. Monotone Functions
   a. The Formula
      i. $f_Y(y) = f_X(x) * \frac{1}{g'(x)}\ at\ x = g^{-1}(y)$
   b. Understanding the Formula
      i. When we apply the density functiojn to y, we have the stretch of whatever the rate of change is at point x at that moment, which is given by g’(x) which is why we have to compensate by dividing by g’(x)
   c. Applying the Formula
      i. Make sure g(x) is continuous and increasing
      ii. Find the derivative of g(x)
      iii. Find the inverse function of g(x)
      iv. Utilize the known PDF of X
   d. Generalization
      i. If G is monotone we can simply replace the g’(x) in the denominator to the absolute value of g’(x) and still have the equation hold true
4. Two to One Functions
   a. For Y = X^2 we have the issue that the function is not monotone and for every value of y, there are in fact two values of X, which means that the density of Y given density of X must take that into account
      i. Hence Y = a + b where a is conversion formula at positive x and b is conversion formula at negative y

b. Square of the Standard

i. Let Z be standard normal and W = Z^2

ii. $f_W(w) = f_Z(\sqrt{w}) * \frac{1}{2\sqrt{w}} + f_Z(\sqrt{-w}) * \frac{1}{2\sqrt{w}} = \frac{1}{\sqrt{2\pi}} w^{-\frac{1}{2}} e^{-\frac{1}{2}w}$

Chapter 17: Joint Densities

1. Probabilities and Expectations

a. A function f on a plane is said to be a joint density if:

i. $f(x,y) \geq 0, for\ all\ x, y$

ii. $\int_x \int_y f(x,y)dydx = 1$

b. A function f is said to be a joint density of R.V X and Y if:

$$P\big((X,Y) \in A\big) = \int \int_A f(x,y)dy\,dx, \qquad for\ all\ A$$

c. Can plot a joint density function by using

i. Plot_3d(x_limits=(0,1), y_limits=(1,2), f= joint, cstride=4, rstride=4)

ii. Using Sympy

1. Declare(‘x’, interval= (0,1))
2. Declare(‘y’, interval=(0,1))
3. f = 120*x *(y-x)*(1-y)
4. Integral(f, (x,0,y), (y,0,1)).doit()

a. Function, inner integral, outer integral

d. Expectation

i. $E\big(g(X,Y)\big) = \int_y \int_x g(x,y)f(x,y)dxdy$

2. Independence

a. X, Y are independent if

i. $P(X \in A, Y \in B) = P(X \in A)P(Y \in B), \ \ \forall\ A, B$

ii. Thus if X,Y and independent

1. $f(x,y) = f_X(x)f_Y(y)$

b. Independent Standard Normal Random Variables

i. Suppose X, Y and i.i.d standard R.V, then their joint density is given by their product which is

$$f(x,y) = \frac{1}{2\pi}e^{-\frac{1}{2}(x^2+y^2)}, -\infty < y < \infty$$

3. Marginal and Conditional Densities

a. To find the marginal density of a variable, you need to sum all its density across the different values of y

b. Thus our equation becomes $f_X(x) = \int_y f(x,y)dy \ \ for\ all\ x$

c. Conditional Densities

i. $f_{(Y|X=x)}\,(y) = \frac{f(x,y)}{f_X(x)}$ otherwise interpreted as as the ratio of the density of X and Y divided by the density of X at a point x,y

4. Beta Densities with Integer Parameters
   a. Order Statistics of IID Uniform (0,1) Variables
      i. Let U_1, U_2…U_n be i.i.id uniform on (0,1). Imagine each U_i as the position of a dart thrown at the unit interval
         1. You can't tell which one is U_1 or U_4 but it is possible to determine which one is lowest, second lowest etc… which we can rename $U_{(1)}$, $U_{(2)}$…$U_{(n)}$ also known as the order statistics of U_1…U_n
         2. The k^th^ order statistic is the k^th^ element if we list all the i.i.d in increasing order and is denoted as $U_{(k)}$
   b. Joint Density of Two Order Statistics
      i. Let n = 5 and we want to work out the joint density of $U_{(2)}$ and $U_{(4)}$, so essentially we want $P\big(U_{(2)} \in dx, U_{(4)} \in dy\big)$
         1. There are 5 variables we can choose to be in dx
         2. There are 4 varaibles we can then chose to in dy
         3. There are 3 variables that can be placed from (0 to x)
         4. There are 2 variables that can be placed from (x to y)
         5. There is 1 variabel that can be placed from(y to 1)
      ii. Density of $U_{(k)}$
         1. One variable has to be at dx
         2. K-1 must be from 1 to x and n-k must be from x to 1
         3. Thus we have the total distribution as

$$P\big(U_{(k)} \in dx\,\big) \sim n * dx * \binom{n-1}{k-1} x^{k-1}(1-x)^{n-k}$$

         4. This is the binomial distribution as we can count a success as landing beyond x and failure as landing below an we have n-1 different uniform variables that need to satisfy that
         5. We can take each probability and show that the density is

$$f_{U(k)}(x) = \frac{n!}{(k-1)!\,(n-k)!} x^{k-1}(1-x)^{n-k}$$

   c. Beta Densities
      i. Using the above equation as a basis, we can see that if we have two positive numbers r and s then
      ii. $f(x) = \frac{(r+s-1)!}{(r-1)!(s-1)!} x^{r-1}(1-x)^{s-1}, 0 < x < 1$
      iii. Is a probability density function, known as a beta density with parameters r and s
      iv. The shape is determined by r and s and we can manipulate them to skew the beta densities to have concentrated masses in certain areas

v. We know that the beta density integrates to 1 so:

$$\int_0^1 \frac{(r+s-1)!}{(r-1)!\,(s-1)!} x^{(r-1)}(1-x)^{s-1} = 1$$

$$\frac{(r+s-1)!}{(r-1)!\,(s-1)!}\int_0^1 x^{(r-1)}(1-x)^{s-1} = 1$$

$$\int_0^1 x^{r-1}(1-x)^{s-1} = \frac{(r-1)!\,(s-1)!}{(r+s-1)!}$$

vi. Let X have Beta(r, s), then the expectation can be found which ends up being $\frac{r}{r+s}$

vii. We can also find the E(X^2) and then find Var(X)

1. Knowing this allows us to chose parameters that satisfy the concentration of the beta distribution

Chapter 18: Normal and Gamma Families

1. Standard Normal: The Basics
   a. Lets use the Rayleigh distribution = $R = \sqrt{T}, where\ T = Expo\left(\frac{1}{2}\right)$
      i. $f_R(r) = re^{-\frac{1}{2}r^2}, r > 0$
      ii. We want to prove that the constant of integration is $\frac{1}{\sqrt{2\pi}}$ so right now we'll call it c and instead have the joint distribution of X and Y be
      iii. $f(x,y) = c^2e^{-\frac{1}{2}(x^2+y^2)}$, and we know that is has circular symmetry so we can define a new variable $R = \sqrt{x^2+y^2}$ and solve for its density
      iv. $f(R) = 2\pi r * c^2e^{-\frac{1}{2}r^2}, where\ r = x^2 + y^2$
      v. This looks the same to the Rayleigh distribution except for 2πc^2 which means that this must equal 1 in order to maintain the integral equal to 1 requirement arriving at our constant of integration
2. Sums of Independent Random Variables
   a. Sum of independent random variables is normal
   b. Sums of I.I.D Random Variables
      i. The sum of any number of i.i.d random normal variables will be normal
3. The Gamma Family
   a. Non-negative R.V X has a distribution gamma(r, λ) for two positive parameters r and λ if the density of X is given by

$$f_X(x) = \frac{\lambda^r}{\Gamma(r)} x^{r-1}e^{-\lambda x}, x \geq\ 0$$

where $\Gamma(r) = \int_0^\infty x^{(r-1)}e^{-x}\,dx$

   b. Key Fact about Gamma Recursion is that $\Gamma(r+1) = r\,\Gamma(r)$

i. Which implies that $\Gamma(r) = (r-1)!$

1. This ends up allowing us to solve for E(X) and SD(X) arriving at

a. $E(X) = \frac{r}{\lambda}, SD(X) = \frac{\sqrt{r}}{\lambda}$

c. The Rate λ

i. For a fixed r, the larger λ implies a smaller E(X) and essentially identifies the units of measurement Y = cX has a gamma (r, λ/c) distribution

d. The Shape Parameter r

i. When r = 1, the density is exponential but as r increases the density's mass moves to the right and flattens out, and when we reach r = 10, the graph looks approximately normal

e. Sums of Independent Gamma Variables with Same Rate

i. IF X has the gamma distribution (r, λ) and Y independent of X has the gamma distribution (s, λ)   then X + Y has the gamma distribution (r + s, λ) distribution

ii. Proof?

iii. The sum of r i.i.id Exponential Variables(λ) has the distribution gamma(r, λ) where r is a positive integer

4. Chi-Squared Distributions

a. Chi-squared density with 1 degree of freedom (Chi-Squared(1)) has the density:

$$f_V(v) = \frac{1}{\sqrt{2\pi}} v^{-\frac{1}{2}} e^{-\frac{1}{2v}}, \qquad v > 0$$

b. From Chi-Squared (1) to Chi-Squared (n)

i. A standard normal variable squared has the gamma distribution (1/2 , ½) which means that the sum of those two would be (1, ½) which is the same as Expo(1/2) distribution

ii. If we sum n different squares of standard normal variables we arrive at the gamma distribution (n/2, ½) which we call a chi-squared distribution of n degrees of freedom

c. Chi-Squared with n Degrees of Freedom

i. If we have a Chi-Squared(n) = Gamme(n/2, ½) then we can show that the density is

$$f_X(x) = \frac{\left(\frac{1}{2}\right)^{\frac{n}{2}}}{\Gamma\left(\frac{n}{2}\right)} x^{\frac{n}{2}-1} e^{-\frac{x}{2}}, \qquad x > 0$$

ii. Mean and Variance

1. The E(Chi-Squared(n)) = r/λ = n/2/(1/2) = **n**

2. SD(X) = root(n/2)/(1/2) = 2 root(n/2) = $\sqrt{2n}$

d. Estimating the Normal Variance

i. Need to review more about the Chi-Square Distributions and Degrees of Freedom

Chapter 19: Distributions of Sums

1. Section is concerned with providing some general methods for working with sums of random variables, whether discrete or continuous
2. The Convolution Formulas
    a. Let X and Y be discrete random variables and let S = X + Y . Then w know that the easiest way to find the distribution of S is by

    $$P(S\ = s\ ) = \sum_{all\ x} P(X = x)P(Y = s - x)\,, if\ X, Y\ are\ independent$$

    b. This can be applied to continuous distributions as well

    $$f_S(s) = \int_{\infty}^{\infty} f_X(x) f_Y(s - x)dx$$

    c. Sum of Two IID Exponential Random Variables
        i. Using the convolution formula we can find that the integral we need to solve for is

        $$f_s(s) = \int_{-\infty}^{\infty} \lambda e^{-\lambda x} \lambda e^{-\lambda(s-x)} dx = \int_{0}^{s} \lambda^2 e^{-\lambda s} = \lambda^2 e^{-\lambda s} s$$

3. Moment Generating Functions
    a. Probability Mass Function, Cummulative Density Function, Probability Density Function and survival functions are all examples of specifying the probability distributions of a random variable
        i. We can become more abstract with this and generate powerful tools for studying distributions specifically in this case: Moment Generating Function
        ii. $M_X(t) = E(e^{tX}), where\ X\ is\ a\ R.V, \forall\ t \rightarrow Expectation\ is\ convergent$
            1. The probability generating function is actually a very specific case of the Moment Generating Function in which s = e^t
    b. Generating Moments
        i. For non-negative functions k, the expectations $E(X^k)$ is called the kth moment of X. The first moment is called the Center of Mass
        ii. We can expand the moment equation using the expansion series of E to find that

        $$M_X(t) = E(1 + \frac{tX}{1!} + \frac{t^2X^2}{2!} + \frac{t^3X^3}{3!} \dots. = E(1) + \frac{tE(X)}{1!} + \frac{t^2E(X^2)}{2!} \dots.$$

        iii. We can utilize the differentiation of M_X to find the Expectations and it correlates that the n^{th} derivate of M_X taken at zero = $E(X^n)$
    c. Identifying the Distribution
        i. If two distributions have the same mgf then they must be the same distribution
    d. Working well with Sums
        i. If X and Y are independent then

        $$M_{X+Y}(t) = M_X(t)M_Y(t)$$