Bazi Olasilik Dagilimlari 1

Bu bolumde olasilik dagimlarinin uzerinde duracagiz ama baslamdan once biraz sohbet etmek gerekirse size biraz daha olasiligin onemini anlatmak istiyorum. Olasilik aslinda evrenin kendisidir. Atom alti parcaciklardan koca galaksilere kadar uzanan bu evren tamamen bir olasiliklar silsilesidir. Bugune kadar bulunan butun fizik formulleri olasiliklar evrenin bir parcasidir. Olasilik hayatimizin bir parcasi ve bunu kullanmayi ogrenmeli ve bu bilgilerimizi gunluk yasamlarimizda kullanabilmeliyiz. Zaten olasiligi anlamaya basladiktan sonra hayatimizin hangi kisimlarinda olasiligin daha etkili oldugu konusunda daha cok farkindaliga sahip olacaksiniz. Bunlara nazaran modern teknolijinin de her alaninda mutlaka olasiliga bagli bolumler vardir. Benim fikrimi soracak olursaniz olasiligi anlamak demek kesinlikle hayatta bir adim one gecmek demektir. Simdi konuya dalmadan once tanimlamak istedigim bir kac terim var. Bu 2 terimi anlamak bu yazi ve geri kalan yazilari anlamak icin cok onemli oldugundan oturu mutlaka uzerine kafa yormanizi ve anlamadan devam etmemenizi rica edecegim.

1. Rastgele Degisken(Random Variable) : Bir olayin sonucunda olusan olasiliklar kumesi olarak degerlendirebiliriz. Hepimizin bildigi uzere bir olasilik kumesinde her sonucun bir olasiligi vardir. Iste rastgele degisken diye adlandirdigimiz terim ise olayin sonucunda olusabilecek herhangi bir durumun sonucunu simgeler. Klasik olarak bilinen zar ornegini vermek istiyorum. Bir zari attigimizda 6 tane olasi sonucu vardir. Bu sonuclari {1, 2, 3, 4, 5, 6} kumesiyle ifade edebiliriz ve bunlarin her biri rastgele degisken kumesi olabilir. Iki zarimiz olursa olasi sonuclara esdeger su sekilde bir rastgele degisken kumesi olusturabiliriz : (1, 1) = 1, (1, 2) = 2, (1, 3) = 3 … (6, 5) = 35, (6, 6) = 36. Elimizde 36 farkli olasi sonuc olacaktir. Tabi ki olayin tanimlanmasina gore de rastgele degiskenlerin tanimlari degisebilir. Bu matematikte fonksiyon olarak gecer. Aslinda bir olasi bir sonucu olayin olasilik kumesinden bizim tanimladigimiz rastegele degisken kumesine ait bir degisken ile yer degistiriyor.
2. Frekans Dagilimi(Frequency Distribution) : Frekans dagilimi aslinda bir olayin sonuclarinin sayisidir. Bu dagilimda hangi olayin kac kere oldugunu gormemizi saglayan bir dagilimdir. Mesela gecen yazida konustugumuz ornegi vermek gerekirse bir frekans dagilimi yapmistik. Bu frekans dagiliminda 1000 tane olayin sonucunda kac tane yazi kac tane tura geliyor onu gormustuk. Bu bir frekans dagilimidir. Baska bir frekans dagilimi ornegi daha vermek istiyorum. Bir zar atalim.

import random

import matplotlib.pyplot as plt

bir = 0; iki = 0; uc = 0; dort = 0; bes = 0; alti = 0

olaylar\_kutuphanesi = {"1":bir, "2":iki, "3":uc, "4":dort, "5":bes, "6":alti}

olaylar = ['1', '2', '3', '4', '5', '6']

for i in range(1000):

    rastgele\_secim = random.choice(olaylar)

    for olay in olaylar:

        if olay == rastgele\_secim:

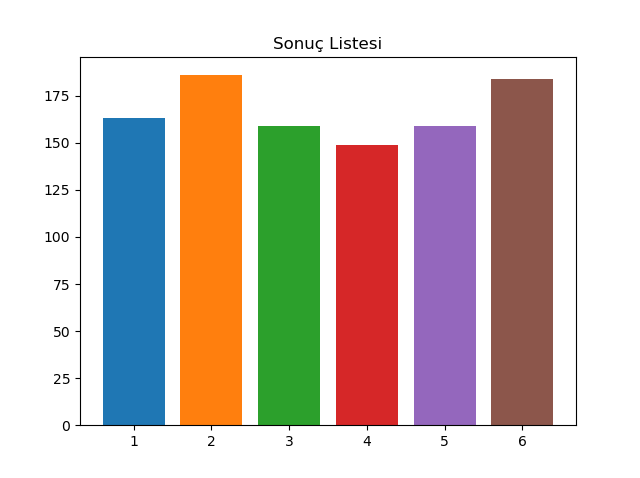
            olaylar\_kutuphanesi[olay] += 1

for olay in olaylar\_kutuphanesi:

     plt.bar(olay, olaylar\_kutuphanesi[olay])

plt.title("Sonuç Listesi")

plt.show()



Gordugunuz uzere her olasi sonucun kac kere oldugunu gostedigimiz bir cizim yaptik. Bu cizim bir frekans dagilimidir.

1. Populasyon(Population) : Populasyon bir olayin olabilecek butun sonuclarini kapsayan veri setidir. Mesela dunyadaki tum insanlarin sayisi bir populasyon ornegidir.
2. Ornek(Sample) : Ornek ise olabilecek olaylarin bir kismini kapsayan veri setidir. Mesela Turkiye’deki insanlar bir ornek ornegidir.
3. Beklenen Deger(Expected Value) : Basit bir sekilde tanimlamak gerekirse beklenen deger bir dagilimin olasiliklarina bagli ortalamasidir diyebiliriz. Aslinda bu bir olay gerceklestiginde olmasi en yuksek olasiliga sahip olan sonucu bize gosterecektir. Su sekilde formule edilir :

Burada olayin sonucunda olabilecek sonuclar iken, bu sonuclarin karsilikli olasiliklaridir.

1. Varyans(Variance) : Varyans olculen degerlerin ortalama degerden ne kadar uzakta oldugunun bir olcusudur. Varyans genel olarak standart sapmayi hesaplamak icin kullanacagimiz bir arac olacaktir. Belirsizlikleri hesaplarken en onemli araclarimizdan biri standart sapmadir yani dolayisiyla varyans. Formulasyonu bu sekilde yapilir:

or

,

1. Standart Sapma(Standard Deviation) : Standart sapma ise elimizdeki verideki dagilmayi olcen bir aractir. Varyansi anlatirken daha cok standart sapmayi kullanacagimi soylemistim. Sebebi ise eger dikkat ettiyseniz varyansin birimi verinin biriminin karesi. Yani sicaklik olcerken gibi ilginc bir dagilim olcusu olacak. Bunu kullanamayiz cunku aslinda boyle bir birim fiziksel bir niceligi yansitmiyor. Standart sapma ise varyans kare kokune esit oldugu icin verilerin ortalama degere gore nasil dagildini gostermek icin standart sapmayi kullaniriz cunku elimizdeki veri setiyle ayni birime sahiptir ve uygulama olarak cok daha kullanislidir. Standart sapma kisaca istatistikteki en onemli araclarin basinda gelir.

Bu tur kavramlar istatistik ve olasilik bilimi icin onemlidir ve bizimde bu kavramlara asina olmamiz ve bunlari kendi analizlerimizde kullanabiliyor olmamiz gerekmektedir. Yukarida verdigim formullerin nereden geldigini ogrenmek yararli bir aktivite olacaktir. O yuzden vaktiniz varsa ayirmanizi oneriririm. Cunku genelde konularin ozunu anlamak onlari daha etkin kullanabilmek ve hatta bir adim ileriye tasimak icin nereden ve hangi varsayimlar yapilarak turetildigini bilmemiz gerekir. Simdi olasilik dagilimlarina gecebiliriz. Olasilik dagilimlari aslinda bir olayin olasi sonuclarinin olasiliklarini temsil eden dagilimlardir. Hem istatistikte hem de olasilik biliminde cok onemli bir yeri vardir. Surekli dagilimlar ve sureksiz dagilimlar olmak uzere iki cesidi vardir. Hadi sureksiz dagilimlardan baslayalim:

1. Sureksiz Dagilimlar(Discrete Distributions): Sureksiz dagilimlar aslinda olaylarin kesikli olarak gerceklestigini ve iki olasi sonuc arasinda bir sureklilik olmadigi zaman kullandigimiz dagilimdir. Ornek vermek gerekirse; yazi tura olayinin sonuclari, zar atma olayinin sonuclari, bir desteden kart cekme olayinin sonuclari … Sureksiz olaylarinda kendi icinde farkli olay sonuclarina hitap eden farkli dagilimlari vardir. Haydi bunlara bir goz atalim.
2. Tekduze Dagilim(Uniform Distribution): Bir olaydaki sonuclarin olma olasiligi birbirine esitse eger olayin tekrarlanmasi durumunda tekduze dagilimi elde ederiz. Buna bir ornek vermek istiyorum. Ornegin elimizde bir zar var ve bu zari 1000 kere atiyoruz. Bu deneyin sonucunda nasil bir dagilim elde etmemiz gerekir.

import random

import matplotlib.pyplot as plt

bir = 0; iki = 0; uc = 0; dort = 0; bes = 0; alti = 0

olaylar\_kutuphanesi = {"1":bir, "2":iki, "3":uc, "4":dort, "5":bes, "6":alti}

olaylar = ['1', '2', '3', '4', '5', '6']

for i in range(1000):

    rastgele\_secim = random.choice(olaylar)

    for olay in olaylar:

        if olay == rastgele\_secim:

            olaylar\_kutuphanesi[olay] += 1

for olay in olaylar\_kutuphanesi:

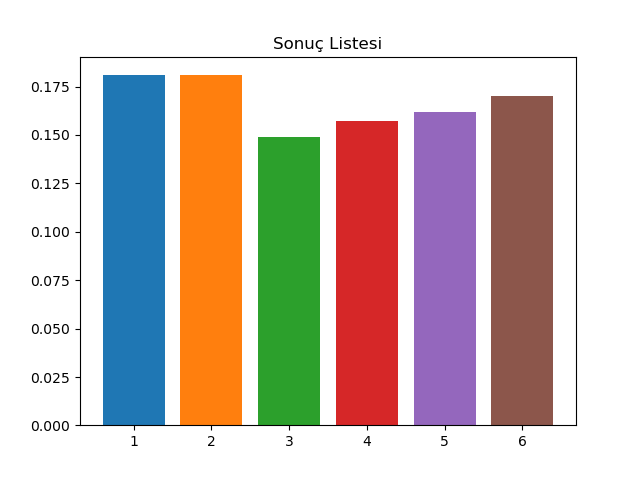
    olaylar\_kutuphanesi[olay] = olaylar\_kutuphanesi[olay] / 1000

for olay in olaylar\_kutuphanesi:

    plt.bar(olay, olaylar\_kutuphanesi[olay])

plt.title("Sonuç Listesi")

plt.show()



Dikkatiniz ceken bir farklilik oldugunu umit ediyorum. Aslinda frekans dagilimindaki koda cok benzemesine ragmen ekstra ama bu dagilimi elde etmemiz icin gerekli bir adim var. Cikan sonuclari toplam olay sayisina bolduk. Aslinda basitce her olayin olasiligini bulduk. Fakat tekduze dagilimlarin karakteristigi dikdortgen seklinde olmasidir. Bizim dagilimimiz dikdortgen degil. Bunu nasil dikdortgen yapariz? Tabiki de onceki yazimizda bahsettigimiz gibi olay sayisini arttirarak!

import random

import matplotlib.pyplot as plt

bir = 0; iki = 0; uc = 0; dort = 0; bes = 0; alti = 0

olaylar\_kutuphanesi = {"1":bir, "2":iki, "3":uc, "4":dort, "5":bes, "6":alti}

olaylar = ['1', '2', '3', '4', '5', '6']

for i in range(1000000):

    rastgele\_secim = random.choice(olaylar)

    for olay in olaylar:

        if olay == rastgele\_secim:

            olaylar\_kutuphanesi[olay] += 1

for olay in olaylar\_kutuphanesi:

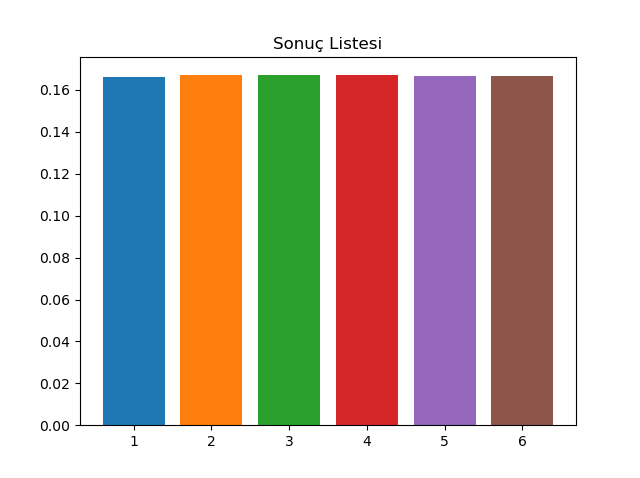
    olaylar\_kutuphanesi[olay] = olaylar\_kutuphanesi[olay] / 1000000

for olay in olaylar\_kutuphanesi:

    plt.bar(olay, olaylar\_kutuphanesi[olay])

plt.title("Sonuç Listesi")

plt.show()



Olay sayisini 1 milyon yaparak aslinda gozlemimizin dogrulugunu arttirmis oluyoruz. Gozlem sayisi arttikca dogru olasilik degerine yaklasiriz.

b.) Bernoulli Dagilimi(Bernoulli Distribution) : Bir olayin sonuclarinin sadece bir deney ile gozlemlenmesiyle olusan dagilimdir. Mesela elimizde bir tane adil olmayan tavla zari olsun. Bu zari attigimizda 6 gelme olasiligi 0.6 iken diger 5 sonucun olasiligi 0.4 olsun. 6 nin gelme olasiligina p dersek eger olmama olasiligi q = 1 – p = 0.4 olacaktir. Bu olayda bir dagilimdan ziyade daha cok mantik araclarini kullariz. Eger 6 gelirse olayin sonucu 1, eger gelmezse olayin sonucu 0 olacaktir. Bir nevi evet, hayir gibi dusunulebilir. Her zamanki gibi deney sayisini arttirirsak eger dogru sonuca olusmus oluruz. Bunu 1000 kere denersek kodumuzla neler olacagina bakalim.

from scipy.stats import bernoulli

import seaborn as sns

import matplotlib.pyplot as plt

# Bernoulli dagilimi icin rastgele sayi ureten bir fonksiyon

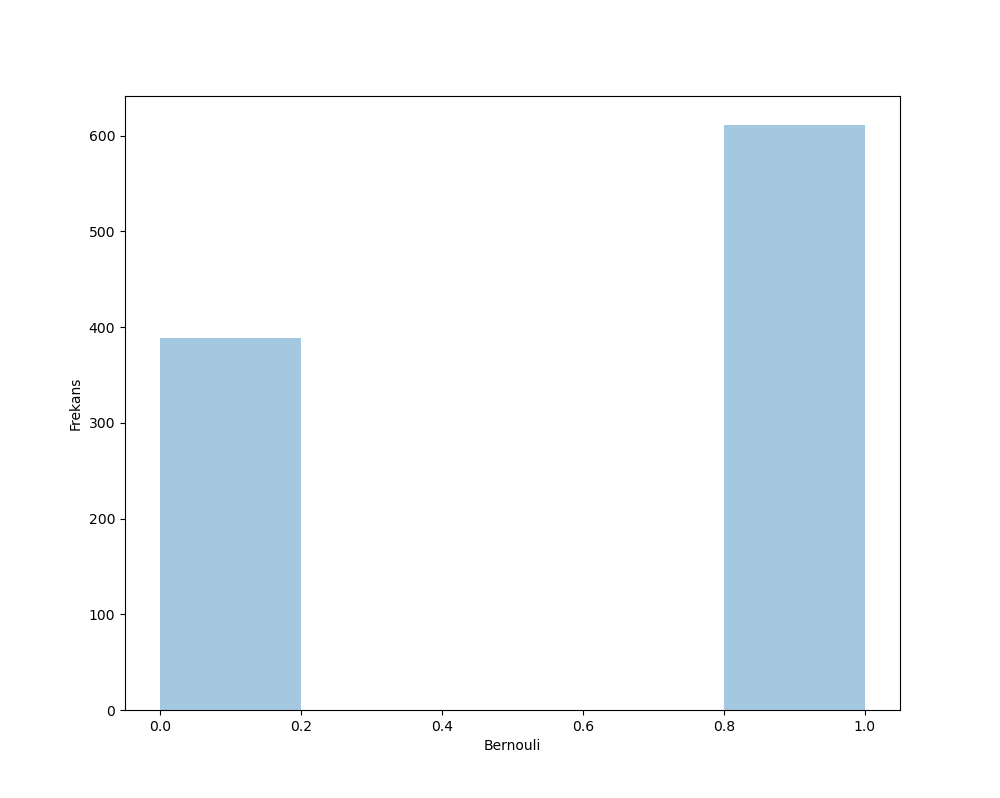
olay = bernoulli.rvs(size = 1000, p = 0.6)

plt.figure(figsize = (10, 8))

ax = sns.distplot(olay, kde = False)

ax.set(xlabel = 'Bernouli', ylabel = 'Frekans')

plt.show()



Grafiktende anlasilacagi uzere elimizde bir frekans dagilimi Bernoulli dagilimi var. Hazir fonksiyonlari kullanmanin kotu yanlarindan biri de bazen kendi istedigimiz seyi yapamiyor olmamizdir. Mesela ben bu yaziyi yazarken bu fonksiyonu nasil normalize edecegimi bulamadim. Fonksiyonun kendisi bir vektor seklinde ve bu vektorde rastgele 1 ve 0 lar bulunuyor. Aynisini biz de yazabiliriz. Genelde yazilmasi kolay ve basit deneyleri kendimiz yaparsak daha dogru olur diye dusunuyorum. O yuzden hadi kendi deneyimizi yapalim.

import random

import matplotlib.pyplot as plt

# Bernoulli dagilimi icin rastgele sayi ureten bir fonksiyon

olaylar = ["6", "6 gelmeme durumu"]

alti\_gelme\_durumu = 0

alti\_gelmeme\_durumu = 0

for i in range(100):

    olay = random.random()

    if olay <= 0.6:

        alti\_gelme\_durumu += 1

    else:

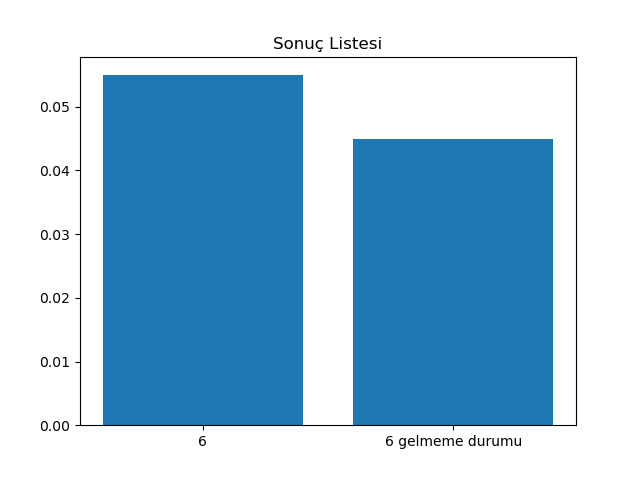
        alti\_gelmeme\_durumu += 1

sonuclar = [alti\_gelme\_durumu/1000, alti\_gelmeme\_durumu/1000]

plt.bar(olaylar, sonuclar)

plt.title("Sonuç Listesi")

plt.show()



c.) Binomial Dagilim(Binomial Distribution) : Bernoulli olasilik dagilimini bir seri seklinde gozleme sonucunda elde edilen dagilimdir. Mesela bir yazili sinavinda karsimizda 10 tane dogru yanlis sorusu olsun. Bu 10 tane dogru yanlis sorusundan bir tanesini cevaplamak bir Bernoulli olayidir. 10 tanesini birden cevaplarsak eger Binomial dagilimi elde ederiz. Biraz daha kafanizda canlandirmak gerekirse eger, bu olayi kod ile tekrar edelim. 100 tane sinavda butun dogru yanlislari sorularini sallarsak eger basari akibetimiz ne olur?

from scipy.stats import binom

import seaborn as sns

import matplotlib.pyplot as plt

"""

Bernoulli dagilimi icin rastgele sayi ureten bir fonksiyon yazalim.

Burada 10 bir deneyde kac tane kart cektigimizin sayisi.

p ise olma olasiligi.

Size ise kac kere bu deneyi tekrarladigimizin sayisi

"""

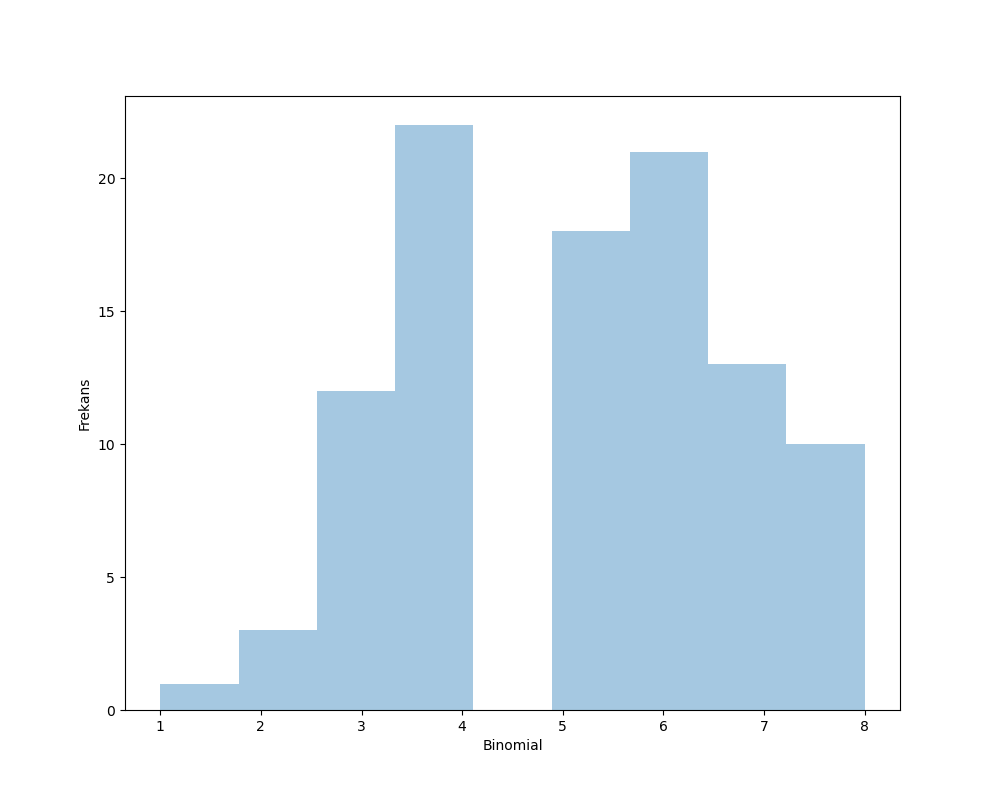
olay = binom.rvs(n = 10, p = 1/2, size = 100)

plt.figure(figsize = (10, 8))

ax = sns.distplot(olay, kde = False)

ax.set(xlabel = 'Binomial', ylabel='Frekans')

plt.show()



Bu dagilimin kodunu kendiniz yazmaniz icin size birakiyorum. Simdi son sureksiz dagilimimiz olan Poisson dagilimina bakalim.

d) Poisson Dagilimi(Poisson Distribution) : Poisson Dagilimi simdiye kadar gordugumuz dagilimlardan biraz daha farklidir. Poisson dagilimini genelde ortalamasini bildigimiz veri setleri icin hazirlariz. Mesela bir matbaadan gunluk ortalama 5 tane kitap basiliyor diyelim. Matbaa sahibi 2 yillik kagit stogu yapmak istiyor ve bu yuzdende ekstrem kosullari ongormek istiyor diyelim. Ekstrem kosuldan kastim mesela bir gunde 15 tane kitap basilma olasiligi veya 5 tane kitap basilma olasiligi nedir? Bu tur durumlarda kullanabilecegimiz dagilimimi Poisson dagilimi olacaktir. Hadi bir ornek yapalim.

from scipy.stats import poisson

import seaborn as sns

import matplotlib.pyplot as plt

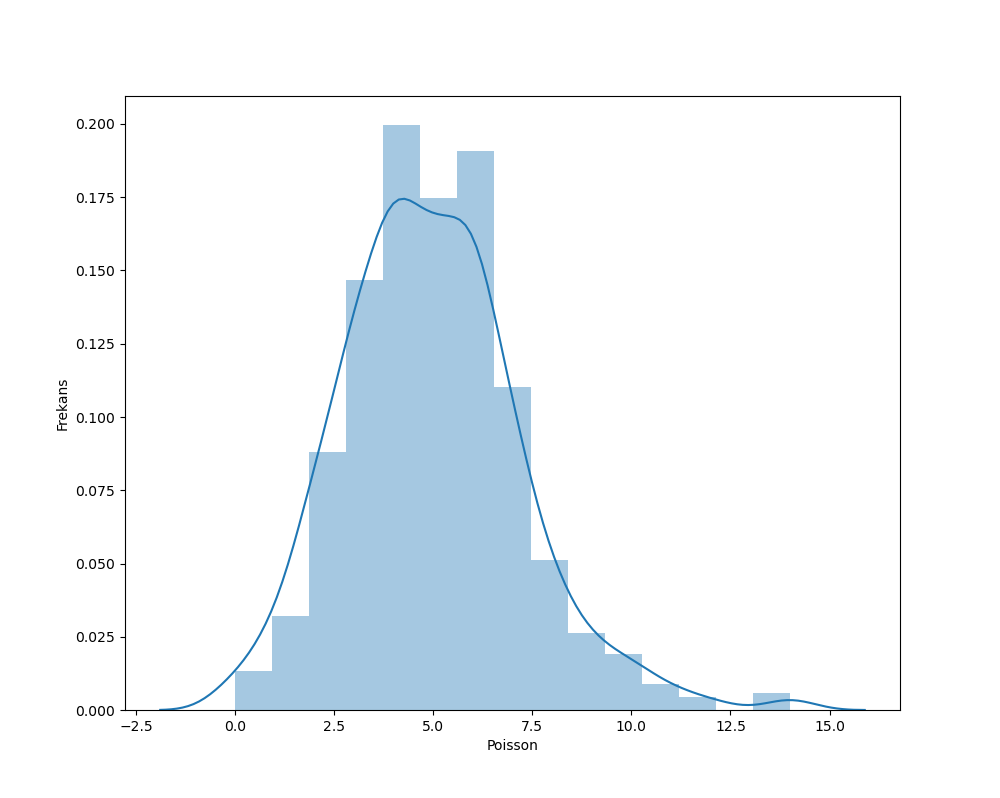
veri = poisson.rvs(mu = 5, size = 730)

plt.figure(figsize = (10, 8))

ax = sns.distplot(veri, kde = True, bins = 15)

ax.set(xlabel = 'Poisson', ylabel = 'Frekans')

plt.show()



Gordugunuz gibi 15 tane kitap basilma olasiligi cok dusuk. O yuzden matbaanin 15 tane kitap basma olasiligini hesaba katmamasi bile gerektigini goruyoruz. O zaman ortalama olarak 6 tane kitaplik mi kagit koymali garanti olsun diye? Buna nasil karar verecegiz? Tabi ki bunun icin istatistigi kullanmamiz gerekiyor! Beklenen deger(Expectation value), varyans(Variance) ve standart sapma(Standard deviation) kavramlarini ogrenmemiz gerek. Bu haftalik bu kadar arkadaslar. Saglicakla kalin.