بسم الله الرحمان الرحیم

گزارش پروژه یک درس آمار و احتمالات مهندسی

پاییز ۱۴۰۲

دکتر توسلی پور

مهدی وجهی

۸۱۰۱۰۱۵۵۸

فهرست

سوال ۱

پرسش ۱

در این بخش می بایست تابعی طراحی کنیم که به تعداد داده شده نمونه تولید کنیم. برای این کار به ابتدا با استفاده از numpy می آییم m * n نمونه از برنولی تولید کرده و با تقسیم آن به قسمت های n تایی نمونه های دو جمله ای را درست می کنیم و در نهایت با شمارش آنها خروجی را تولید می کنیم.

```
def create_samples(m : int, n : int, p : float):
    '''
    Create binomial samples
    args:
    m: samples count
    n: count of bernoulli test in binomial
    p: probability of bernoulli
    '''
    samples = np.reshape(np.random.choice([0, 1], n*m, p=[1 - p, p]), (m, n))
    return np.sum(samples, axis=1)
```

پرسش ۲

برای محاسبه توزیع احتمال با موارد داده شده لازم است مقادیر m,n را مطابق با اطلاعات تنظیم کنیم و با یک حلقه از p صفر تا ۱۰۰ را جمع بزنیم. برای محاسبه مقدار تنها کافیست از فرمول های زیر استفاده کرده و مقادیر را حساب کنیم.

$$E[x] = np$$

$$var(x) = npq$$

کد قسمت نمونه گیری به شرح زیر است:

```
real_output = dict()
for p in range(0, 101, 1):
    tmp = create_samples(m, n, p / 100)
    avg = np.average(tmp)
    var = np.var(tmp)
    real_output[p / 100] = {"avg": avg, "var": var}
real_output
```

که خروجی ای به شکل زیر دارد:

```
{0.0: {'avg': 0.0, 'var': 0.0},
```

```
0.01: {'avg': 5.042, 'var': 5.017436},
0.02: {'avg': 10.0658, 'var': 10.19987036},
0.03: {'avg': 14.9182, 'var': 14.357908760000003},
0.04: {'avg': 19.8966, 'var': 19.497108439999998}}
```

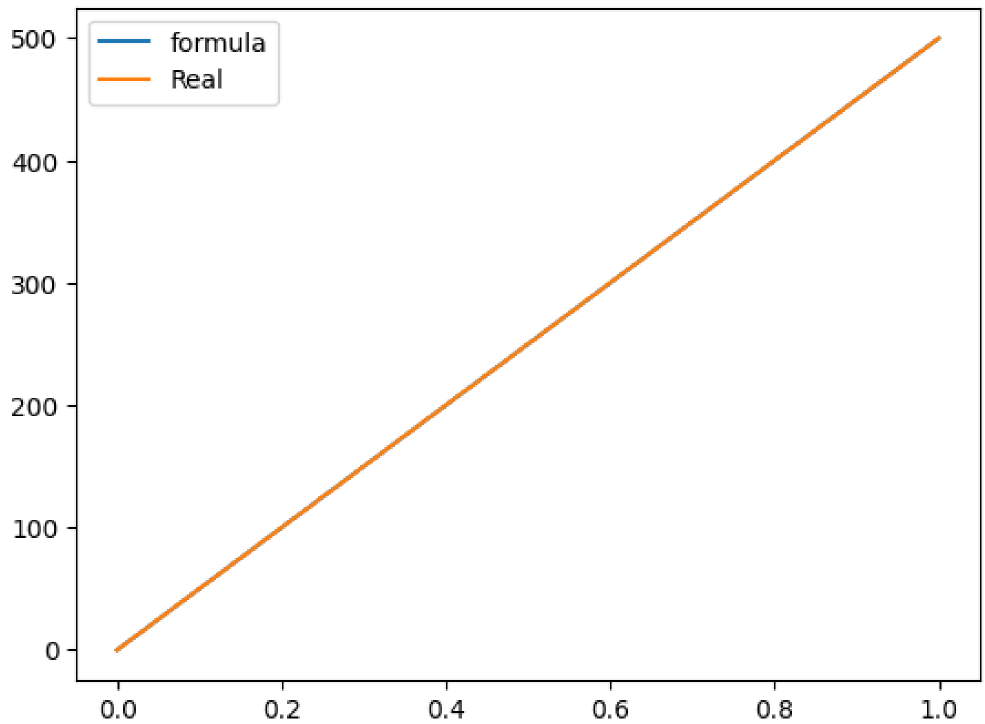
همچنین برای بخش فرمولی کد به صورت زیر است و خروجی مشابه قسمت قبل:

```
formula_output = dict()
for p in range(0, 101, 1):
   formula_output[p / 100] = {"avg": n * p / 100,"var": n * (p / 100) * (1 - (p / 100))}
formula_output
```

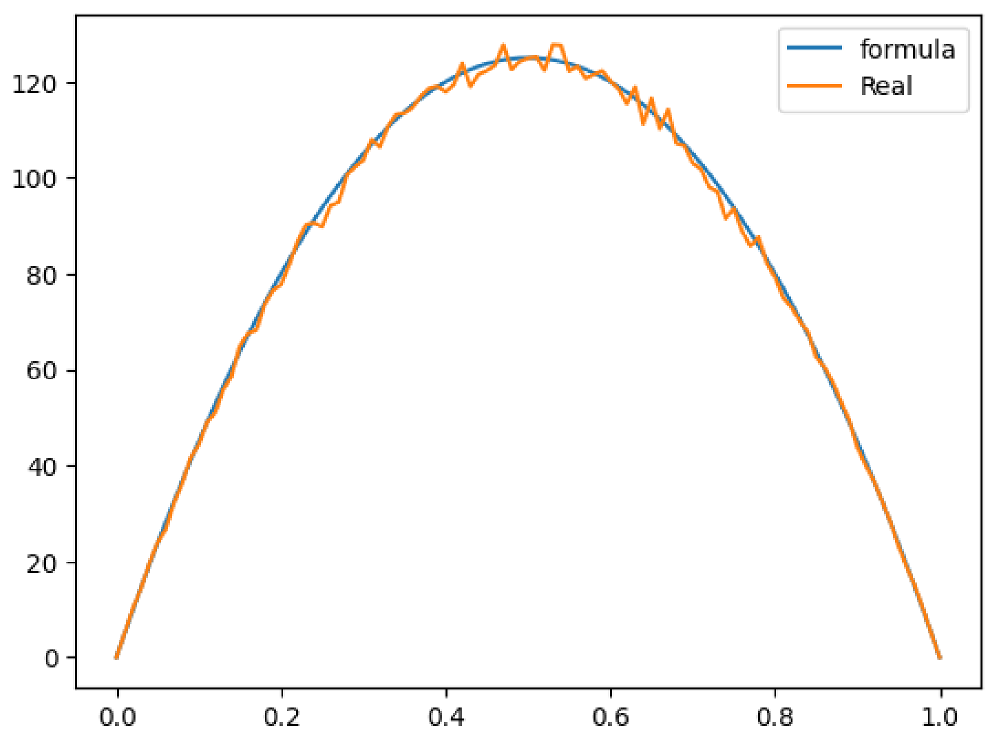
پرسش ۳

در این قسمت کافیست نتایج قسمت بالا را روی نمودار نمایش دهیم که پیدا سازی پیچیده ای ندارد و کد و نمودار آن به شکل زیر است:

```
fig, ax = plt.subplots()
line1, = ax.plot(formula_output.keys(), [i["avg or var"] for i in formula_output.values()], label='formula')

line2, = ax.plot(real_output.keys(), [i["avg or var"] for i in real_output.values()], label='Real')
ax.legend()
plt.show()
```



نمودار میانگین



نمودار واریانس

پرسش ۴

به طرز جالبی میانگین ها تقریبا برهم منطبق هستند. در واریانس در احتمالات نزدیک به صفر و یک نمودار های منطبق می شوند اما در مقادیر میانی (۰.۵) واریانس ها با هم اختلاف دارند. علت این موضوع این است که هرچه به حاشیه نزدیکتر می شویم حالت های ممکن محدود تر می شوند و این موضوع باعث افزایش دقت برنامه می شوند اما در حالت هایی میانی به علت حالت های زیاد این امکان وجود ندارد.

سوال ۲

پرسش ۱

در این پرسش باید احتمال دوجمله ای و تقریب های نرمال و پواسون آن را نمایش دهیم. با جست و جو در اینترنت نحوه کار با توابع را یاد گرفتم و به صورت زیر پیاده شد:

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
import math
```

```
n = 250
p = 0.008
m = 10000
mu = 2
variance = 1.984
sigma = math.sqrt(variance)

fig, ax = plt.subplots()
```

نمایش تابع دوجمله ای:

```
x = np.arange(-2, 6, 1)
dist = [stats.binom.pmf(r, n, p) for r in x ]
ax.bar(x, dist)
```

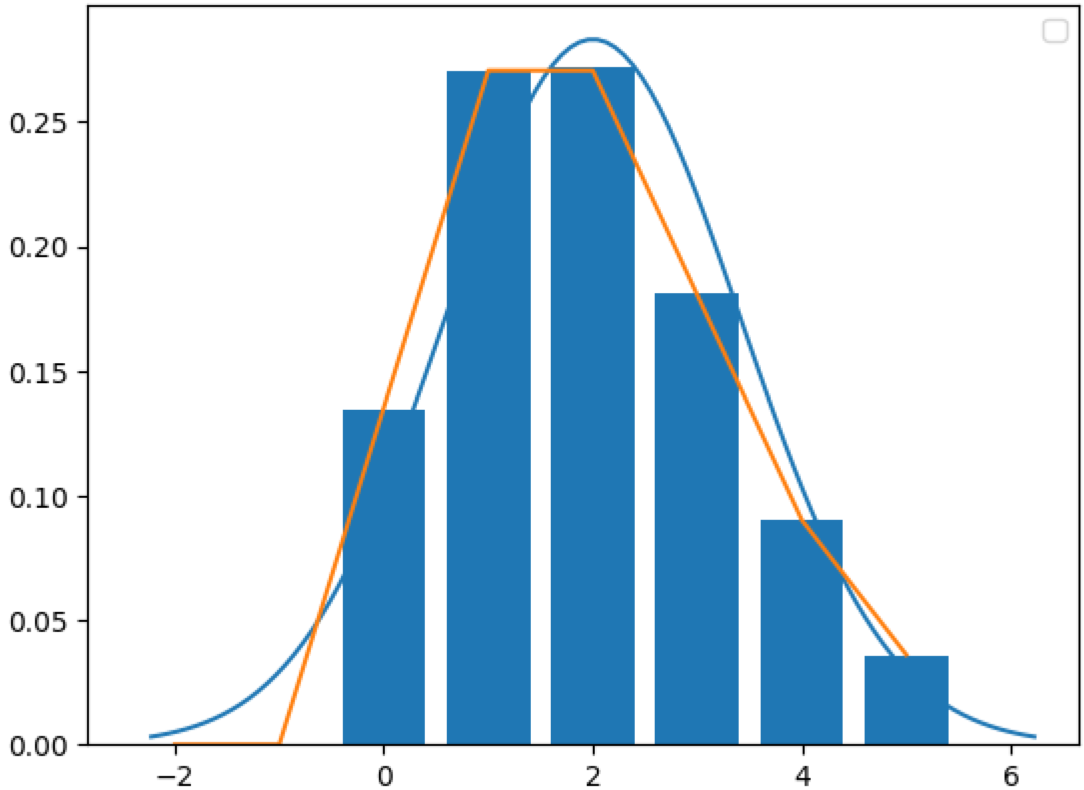
نمایش تابع نرمال:

```
x = np.linspace(mu - 3*sigma, mu + 3*sigma, 100)
norm, = ax.plot(x, stats.norm.pdf(x, mu, sigma))
```

نمایش تابع پواسون:

```
x = np.arange(-2, 6, 1)
y = stats.poisson.pmf(x, mu=mu)
ax.plot(x, y)
```

نمودار خروجی:



نمودار مقایسه توزیع دوجمله ای (میله ای آبی)، پواسون (نارنجی)، نرمال (آبی)

پرسش ۲

با توجه به نمودار واضح است که پواسون دقت بیشتری دارد اما دقت نرمال هم خوب است. دلیل این موضوع این است که برای این که پواسون دقت کافی را داشته باشد باید n * p بین ۰ تا ۱۰ باشد که اینجا ۲ است و صدق می کند اما برای نرمال باید $p \simeq 0.5$ که در اینجا صادق نیست پس نمی توان دقت بالا را انتظار داشت.

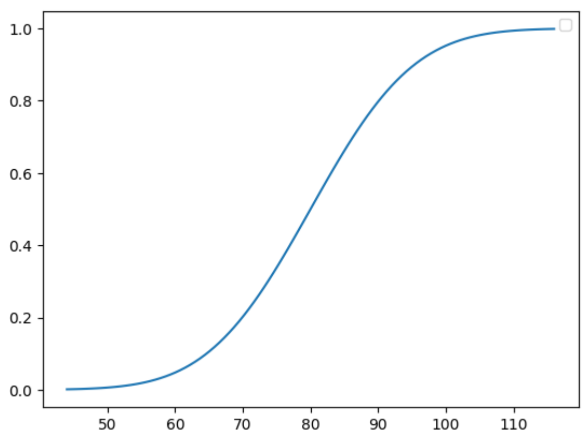
سوال ۳

پرسش ۱

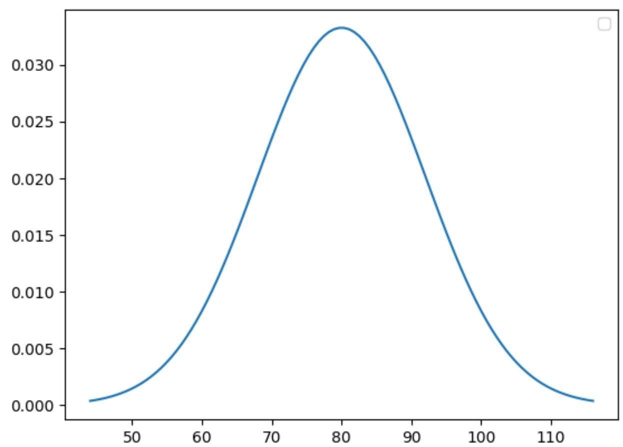
برای پیدا کردن حداقل نمره ابتدا نمودار ها را رسم می کنیم.

```
fig, ax = plt.subplots()
fig, bx = plt.subplots()

x = np.linspace(mu - 3*sigma, mu + 3*sigma, 100)
norm, = ax.plot(x, stats.norm.pdf(x, mu, sigma))
norm, = bx.plot(x, stats.norm.cdf(x, mu, sigma))

ax.legend()
bx.legend()
plt.show()
```

نتیجه:



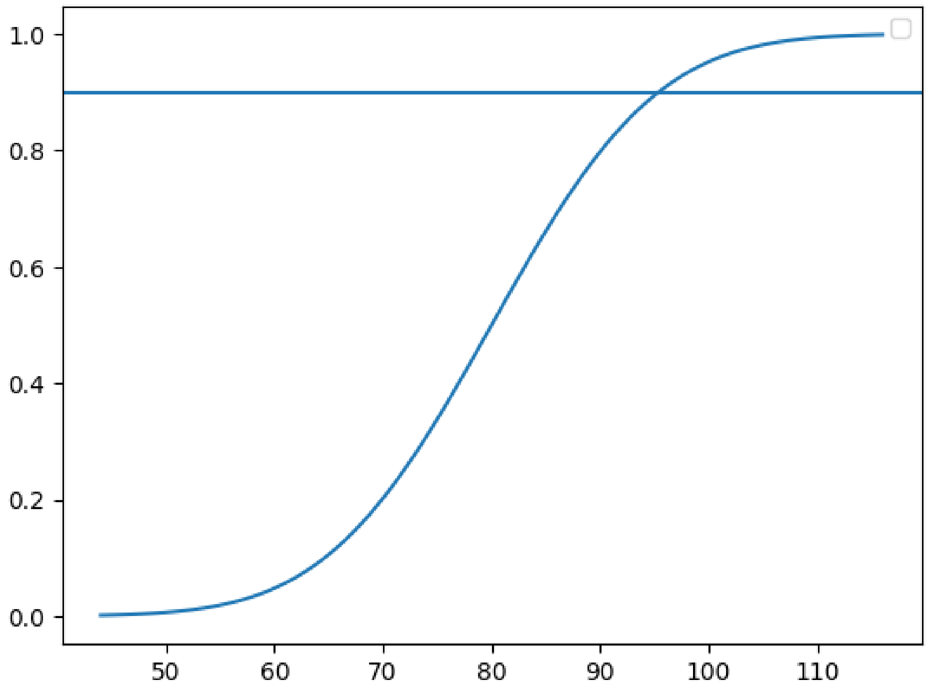
CDF



PDF

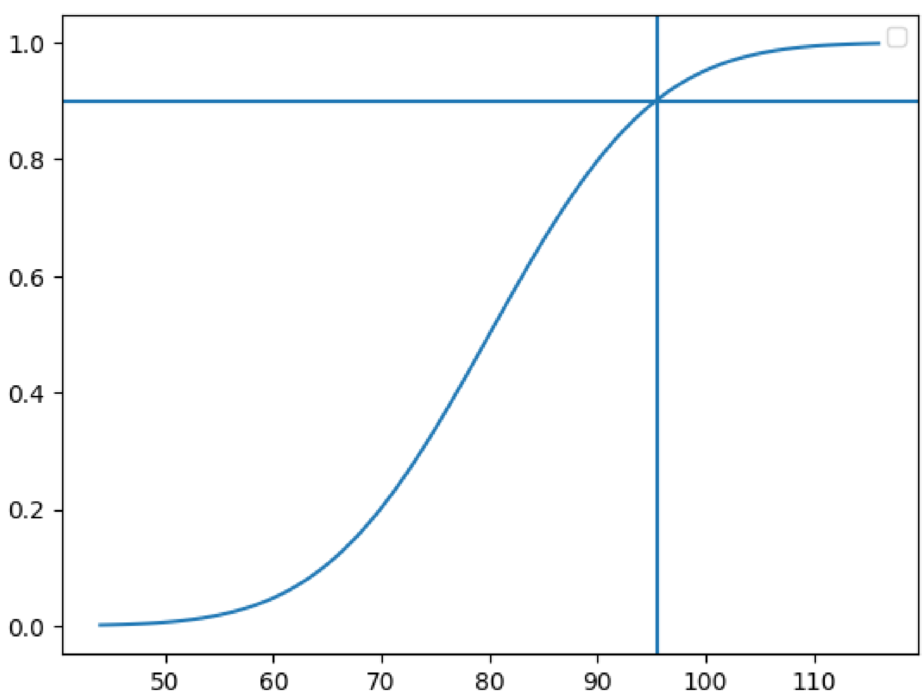
حال خط $y = 0.9$ رسم می کنیم:

```
bx.axhline(y=0.9)
```



با حدس جواب مقدار حدودی را پیدا می کنیم:

```
bx.axvline(x=95.5)
```



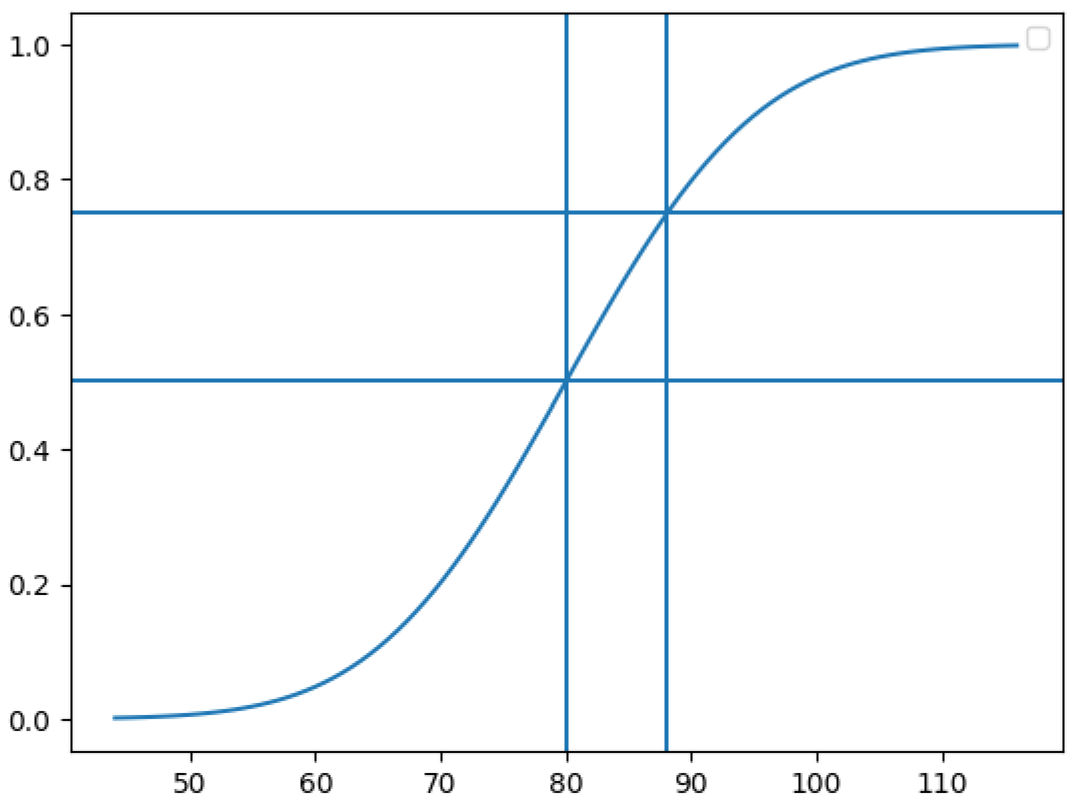
جواب تقریبا برابر ۹۵.۵ است.

پرسش ۲

مانند قسمت قبل نمودار را کشیده و مقدار را به صورت تقریبی حدس زده.

```
fig, bx = plt.subplots()

x = np.linspace(mu - 3*sigma, mu + 3*sigma, 100)
norm, = bx.plot(x, stats.norm.cdf(x, mu, sigma))
bx.axhline(y=0.50)
bx.axhline(y=0.75)

bx.axvline(x=80)
bx.axvline(x=88)

bx.legend()
plt.show()
```

نمودار خروجی:



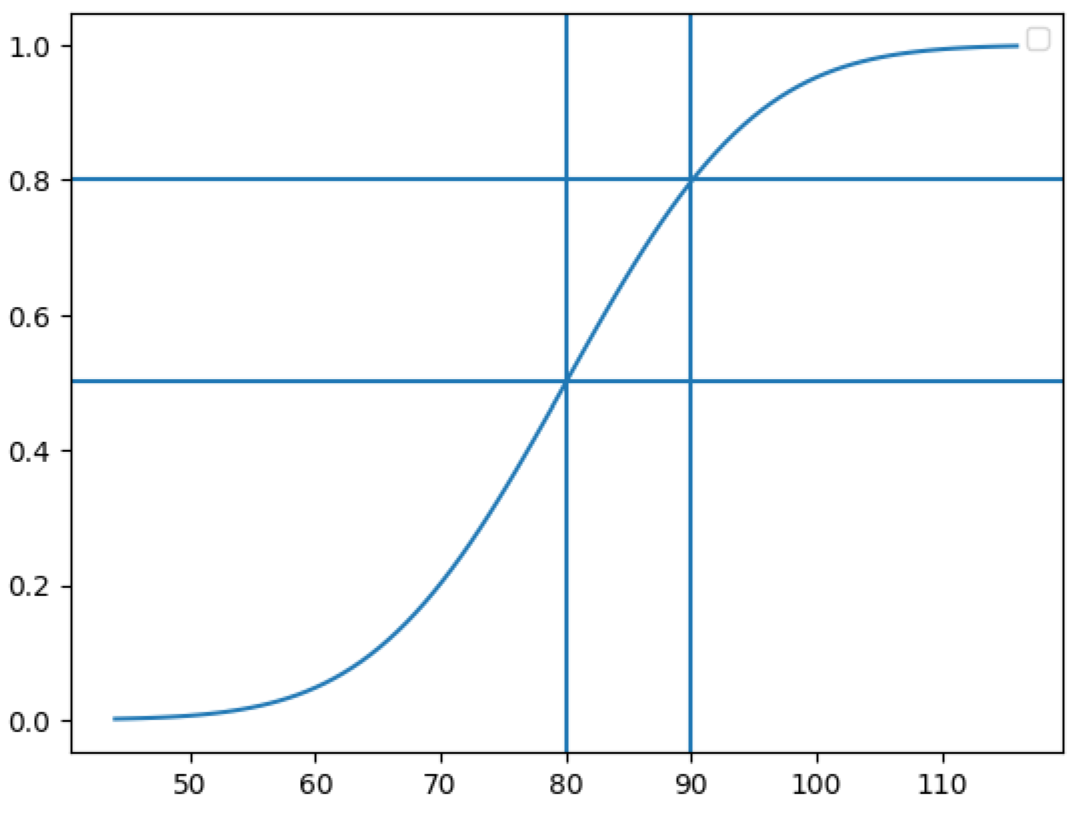
پس می توان متوجه شد که نمرات در ۲ چارک میانی بین ۸۰ و ۸۸ می باشد.

پرسش ۳

مانند دو قسمت قبل با رسم نمودار و حدس جواب پیش می رویم.

```
fig, bx = plt.subplots()

x = np.linspace(mu - 3*sigma, mu + 3*sigma, 100)
norm, = bx.plot(x, stats.norm.cdf(x, mu, sigma))
bx.axvline(x=80)
bx.axvline(x=90)

bx.axhline(y=0.5)
bx.axhline(y=0.8)

bx.legend()
plt.show()
```

نمودار خروجی:



در نتیجه بین ۵۰٪ و ۸۰٪ می باشد پس در نتیجه احتمال در بین بازه مذکور ۰.۳ است.

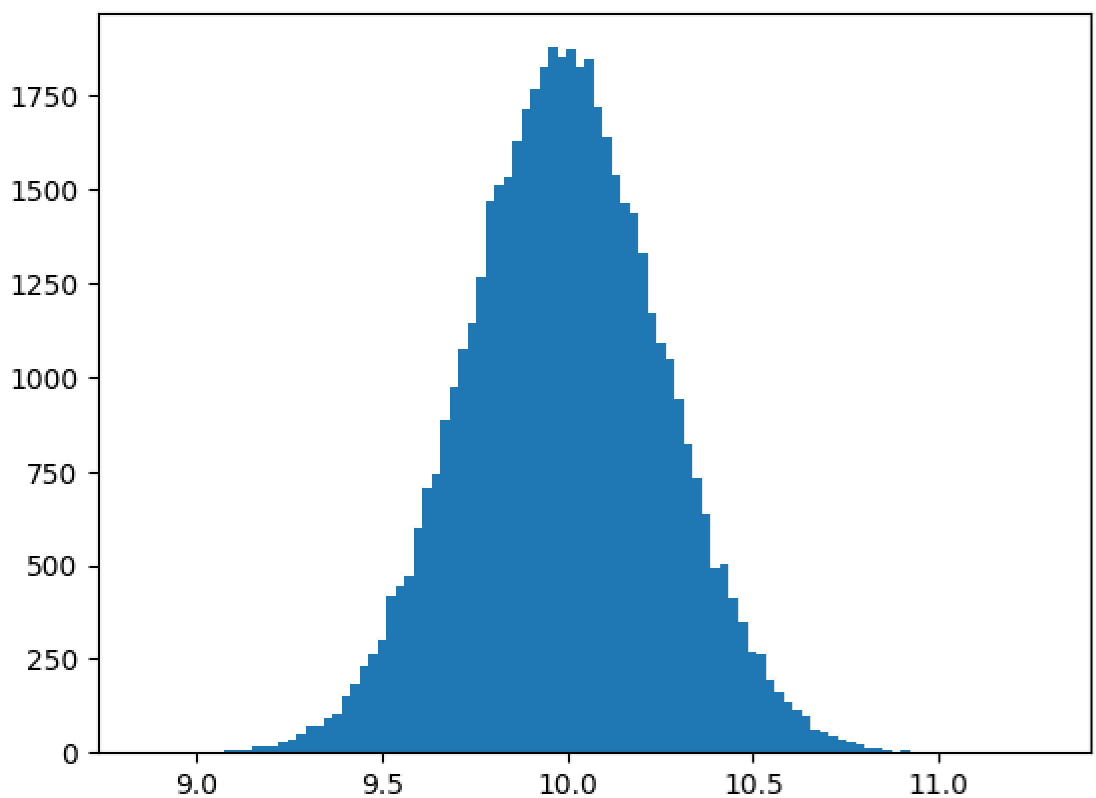
پرسش ۴

مفروضات:

```
lam = 17
grads = [i / 4 for i in range(81)]
sample_count = 10000
avg_sample_count = 50000
sample_count_per_avg = 500
```

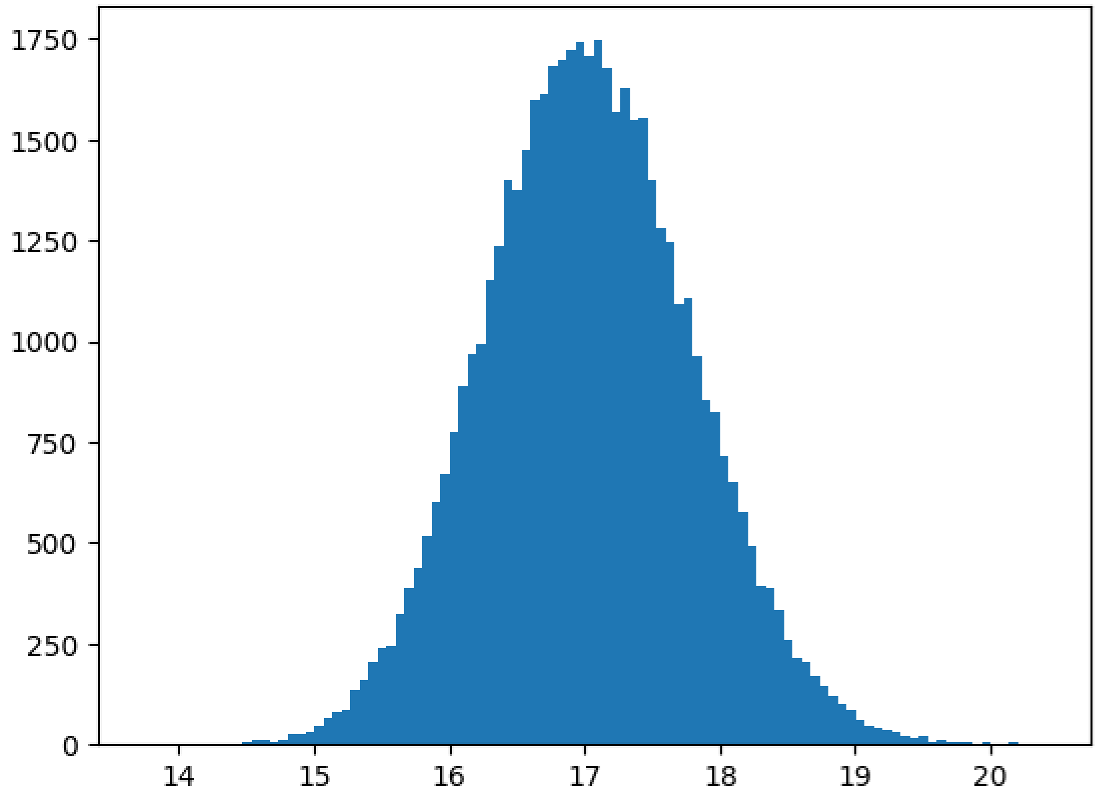
حال برای یکنواخت می کشیم:

```
uniform_sample = np.random.choice(grads, sample_count)
uniform_avg_sample = [np.mean(np.random.choice(uniform_sample,
sample_count_per_avg)) for i in range(avg_sample_count)]
plt.hist(uniform_avg_sample, bins=100)
plt.show()
```



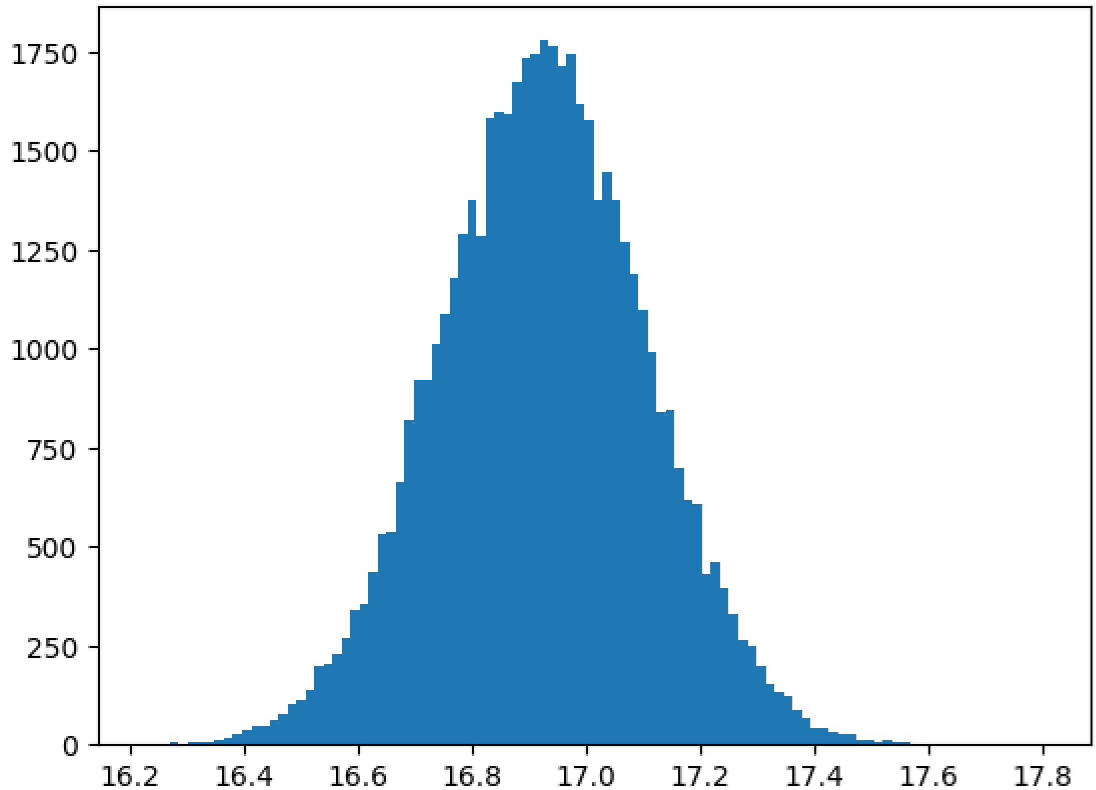
برای نمایی نیز به همین صورت زیر است:

```
exp_samples = np.random.exponential(lam, sample_count)
exp_avg_samples = [np.mean(np.random.choice(exp_samples, sample_count_per_avg))
for i in range(avg_sample_count)]
plt.hist(exp_avg_samples, bins=100)
plt.show()
```



در آخر هم برای پواسون داریم:

```
poi_samples = np.random.poisson(lam, sample_count)
poi_avg_samples = [np.mean(np.random.choice(poi_samples, sample_count_per_avg))
for i in range(avg_sample_count)]
plt.hist(poi_avg_samples, bins=100)
plt.show()
```



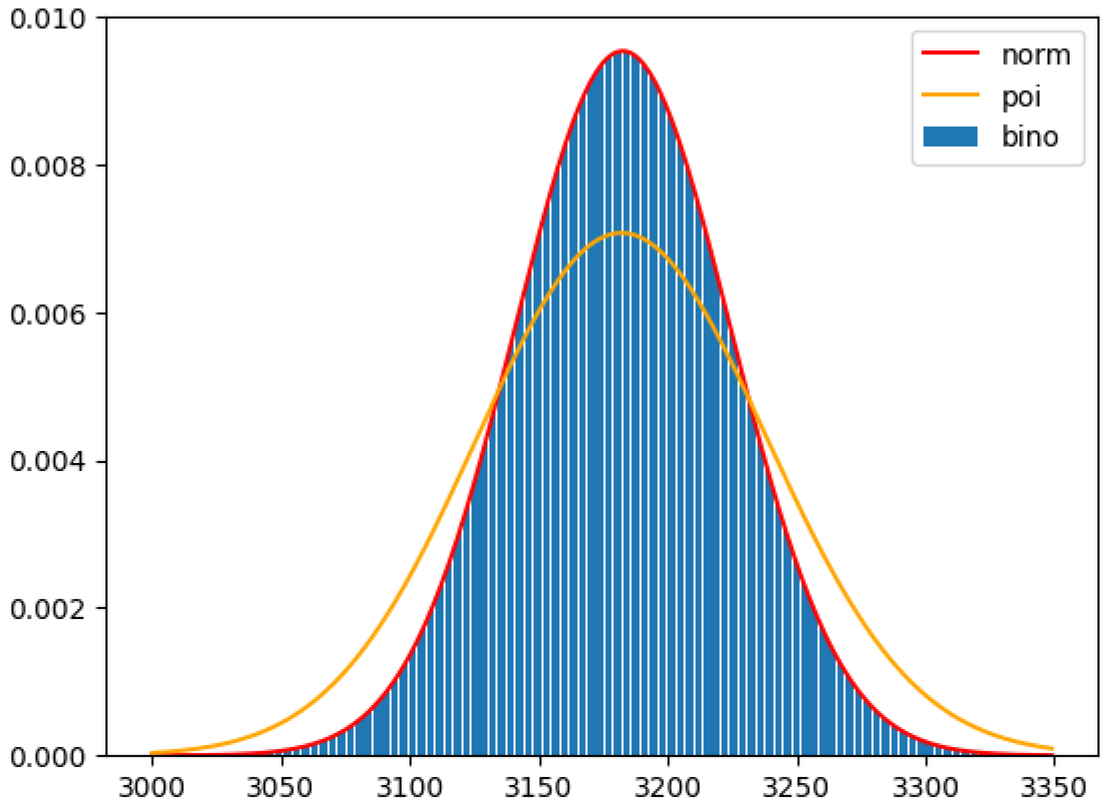
لازم به ذکر است که در مفروضات فقط count ها مهم هستند و بقیه هر مقدار دلخواه هستند. برای تعداد sample ها باید گفت که تعداد آنها می بایست به میزان کافی زیاد باشد تا دقت مسئله خوب باشد. لازم به ذکر است که sample_count_per_avg اگر مقدار کمی باشد نمودار poi دقت کافی را ندارد.

سوال ۴

پرسش ۱

همانند سوال ۲ پرسش ۱ عمل می کنیم تنها اعداد را عوض می کنیم.

خروجی:



پرسش ۲

تقریب نرمال کاملا بر توزیع دوجمله ای منطبق است. دلیل این موضوع این است که برای این که پواسون دقت کافی را داشته باشد باید n * p بین ۰ تا ۱۰ باشد که اینجا ۳۲۰۰ است و صدق نمی کند اما برای نرمال باید $p \simeq 0.5$ که در اینجا صادق است پس نرمال تعقیب بهتری است.