∫ probability density = probability distribution

John Hodson
EEL4930
04/17/19

Homework 4

1. Consider the function
$$f_X(x) = \begin{cases} cx^4, & -1 \le x \le 1 \\ 0, & \text{otherwise} \end{cases}$$

   (a) Find c for $f_X(x)$ to be a valid density function. $c\int_{-1}^{1} x^4 = 1 \rightarrow \frac{1}{5}x^5\Big|_{-1}^{1} = \frac{1}{c}$

   a.) $\frac{2}{5} = \frac{1}{c}$, $c = 2.5$

   (b) Let X be a random variable with density function $f_X(x)$. Find the distribution function of X.

   b.) $= \begin{cases} \frac{1}{2}x^5 & -1 \le x \le 1 \\ 0 & \text{otherwise} \end{cases}$

   (c) Find the mean of X.

   (d) Find the variance of X. on Scratch

   (e) Find the expected value of $(X-1)^2$. Scratch

2. A professor offers an exam for which a randomly chosen student's grade can be modeled as a Gaussian random variable, X. The following grading scheme is used:

   - if $X \ge 85$, the grade is A
   - if $70 \le X < 85$, the grade is B
   - if $X < 70$, the grade is C

   $f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$

   $\mu \Rightarrow$ Mean $\sigma^2 \Rightarrow$ Var.

   Gaussian Model

   (a) If the mean exam score is 82 and the variance is 64, find the probabilities of A, B, and C. $P(A) = .354$ $P(B) = 0.579$ $P(C) = 0.067$

   (b) If a student is confident that he made at least an 80 on the exam, what is the probability that the student made an A on the exam? $= 0.354/(1-.4001) = 0.591$

   (c) Suppose that the professor instead wants to adjust the difficulty of the problems of the exam so that the probability of an A is 0.25 and the probability of a B is 0.52. What should the values of the mean and variance be to achieve these probabilities?

   $\mu = 78$, Variance $= 112$. See attached Jupyter noteBook for work

3. Four boxes of computer chips arrive. Each box is from a different manufacturer. The boxes contain, respectively, 250, 300, 400, and 500 chips. One of the boxes is chosen at random. Let X denote the total number of chips in the chosen box. The boxes are combined, and one of the chips is selected at random. Let Y denote the number of chips that were in the box from the selected chip.

   (a) Which of $E[X]$ or $E[Y]$ do you think is larger? Why? $E[Y]$, you have a higher chance to draw a chip from the 500 box, and thus 500 is weighted more heavily in this expected value.

   (b) Compute $E[X]$ and $E[Y]$.

   on Scratch paper

4. The Laser Interferometer Gravitational-Wave Observatory (LIGO) is very sensitive – it is designed to detect a change in distance between its mirrors 1/10,000th the width of a proton! Thus, it also detects very minor disturbances, such as a car driving by, or someone walking down the hall. On average, the LIGO inferometer at one of the two locations detects an average of 30 disturbances per hour that are not gravitational waves.

$V = \lambda t$

$\lambda = \dfrac{30 \text{ disturbances}}{60 \text{ minutes}} = .5 \text{ disturbances/min}$

$P_N(n) = \dfrac{(0.5t)^n}{n!} e^{-0.5t}$

Poisson Model

(a) What is the probability that there will be at least one disturbance in a 1 minute period? 0.393

(b) What is the probability that there will be at least one disturbance in a 5 minute period? 0.918

(c) What is the probability that there will be fewer than 5 disturbances in a 15 minute period? 0.132

(d) What is the probability that there will be more than 10 disturbances in a 15 minute period? 0.224

5. (Ross) If 65% of the population of a large community is in favor of a proposed rise in school taxes, approximate the probability that a random sample of 100 people will contain:

Binomial Model

(a) at least 50 who are in favor of the proposition; Scratch Paper

(b) between 60 and 70 who are in favor of the proposition; Scratch Paper

(c) fewer than 75 in favor. Scratch paper

6. Samples from a certain low noise amplifier are distributed Gaussian with mean 0 and variance 1. Samples from the same company's power amplifier are Gaussian with mean 0 and variance 4. During a particular day, the chip labeler breaks down, but nobody noticed until chips of both types have been mixed together.

Recall that by the "weighted density functions", we refer to the numerator of the *a posteriori* probabilities, which is the product of the likelihoods and the *a priori* probabilities.

This problem requires both plotting work (in a Jupyter notebook) and analytical work (that I recommend you do by hand on paper).

✓ (a) First consider that chips of both types are equally likely. Plot the weighted density functions in Jupyter notebook. Vary the regions plotted to zoom in on any points necessary to give a reasonable estimate (at least accurated to within ±0.25) of the MAP decision regions.

✓ (b) The MAP decision regions in this problem are determined by a set of thresholds. (Unlike the case covered in class, there is not just a single threshold for each value of the *a priori* probabilities.) Give a formula for the MAP decision thresholds

✓ (c) Write a function in your Jupyter notebook that inputs the probability that a randomly chosen chip is a low noise amplifier. The function should

- plot (using `plt.plot`) both the weighted density functions
- calculate the numerical value of the MAP decision thresholds
- plot (using `plt.scatter`) the points where the x-values is a MAP decision threshold and the y-value is the value of the weighted densities at that MAP decision threshold

✓ (d) Show the output of your function when there are equal numbers of each type of amplifier

✓ (e) Show the output of your function when there are twice as many power amplifiers as low-noise amplifiers

✓ (f) Show the output of your function when the probability of a randomly chosen chip being a low-noise amplifier is 0.9

(g) What happens when the probability of a random chosen chip being a low-noise amplifier is 0.2?

(h) Plot the weighted density functions using a logarithmic y-axis (using `plt.semilogy`) when the probability of a random chosen chip being a low-noise amplifier is 0.2

(i) Show where the math breaks down when the probability of a random chosen chip being a low-noise amplifier is low

1.)

c.) $E(x) = \mu = \int_{-\infty}^{\infty} x f_X(x)\,dx$

$= \int_{-1}^{1} x\left(\frac{5}{2}x^4\right)dx = 0$

d.) $\sigma^2 = E[(X - \mu_x)^2]$

$= \int_{-1}^{1} (x-\mu_x)^2 f_x(x)\,dx$

$= \int_{-1}^{1} x^2 \cdot \frac{5}{2}x^4\,dx = \int_{-1}^{1} \frac{5}{2}x^6\,dx$

$= 0.714$

e.) $E[(X-1)^2]$

$= \int_{-1}^{1} (x-1)^2 f_x(x)\,dx$

$E[a] = \int_{-\infty}^{\infty} a f_x(a)\,da$ $= \int_{-1}^{1} (x-1)^2 \frac{5}{2}x^4\,dx$

$= 1.714$

3.) b.) $E[x] = \frac{250 + 300 + 400 + 500}{4}$

$= 362.50$

$E[y] = \frac{250 \cdot 250 + 300 \cdot 300 + 400 \cdot 400 + 500 \cdot 500}{}$

$= 387.9$

5.) $P[X=k] = \begin{cases} \binom{n}{k} p^k (1-p)^{n-k}, & k = 0, 1, 2 \ldots n \\ 0 & \text{o.w.} \end{cases}$

n trials, k successes

$pmf(k) = \binom{100}{k} (0.65)^k (1-0.65)^{100-k}$

a) $0.999 = \sum_{k=50}^{100} pmf(k)$

b) $0.702 = \sum_{k=60}^{70} pmf(k)$

c.) $0.965 = \sum_{k=0}^{74} pmf(k)$

In [1]:
```
import scipy.stats as stats
import scipy.special as sp

import numpy as np

import matplotlib.pyplot as plt
%matplotlib inline
```

Problem 2

In [2]:
```
gauss = stats.norm(loc=82,scale=8)
```

Part A

In [3]:
```
gauss.cdf(70)
```

Out[3]: 0.06680720126885807

In [4]:
```
gauss.cdf(85)-gauss.cdf(70)
```

Out[4]: 0.5793625654038657

In [5]:
```
1-gauss.cdf(85)
```

Out[5]: 0.35383023332727626

Part B

In [6]:
```
lis = gauss.rvs(1000000)
greater_than_80 = lis >= 80
greater_than_85 = lis >= 85
np.sum(greater_than_85)/np.sum(greater_than_80)
```

Out[6]: 0.5911556069997511

Part C

In [7]:
```
def gauss_pdf(mu, var, x):
    return 1/(2*np.pi*var)**(1/2)*np.exp(-(x-mu)**2/(2*var))
```

In [8]:
```
for mu in range(30,100):
    for var in range(30,200):
        prob_a = 0
        prob_b = 0
        for x_a in range(85,100): # calculate the cdf of a
            prob_a+=gauss_pdf(mu,var,x_a)
        for x_b in range(70,85): # calculate the cdf of b
            prob_b+=gauss_pdf(mu,var,x_b)
        if (abs(prob_a-.25) <= .0025 and (abs(prob_b-.52) <= .0025)):
            print("Mu = ",mu,". Var = ",var,". Prob of A = ",prob_a,". P
rob of B = ",prob_b,".",sep="")
```

```
Mu = 78. Var = 111. Prob of A = 0.2479537739631005. Prob of B = 0.52163
53936665281.
Mu = 78. Var = 112. Prob of A = 0.24840772370227115. Prob of B = 0.5196
808507166172.
Mu = 78. Var = 113. Prob of A = 0.24884873008939257. Prob of B = 0.5177
478397299403.
```

In [9]:
```
new_gauss = stats.norm(loc=78, scale = 112**.5) # create a new gaussian
 with the paramaters found above
```

In [10]:
```
new_gauss.cdf(85)-new_gauss.cdf(70) # probability of a B
```

Out[10]: 0.5209883142394818

In [11]:
```
1-new_gauss.cdf(85) # probability of an A
```

Out[11]: 0.2541657867760727

Problem 5

a. Define the pmf function of the binomial random variable

In [12]:
```
def binom_pdf(k):
    return sp.binom(100,k)*(.65**k)*(1-.65)**(100-k)
```

In [13]:
```
sum = 0
for i in range(60,70):
    sum+=binom_pdf(i)
sum
```

Out[13]: 0.701957712263844

b. Random variable - summing the pmf

In [14]:
```
rv = stats.binom(n=100,p=.65)
```

In [15]:
```
sum = 0
for i in range(60,70):
    sum+=rv.pmf(i)
sum
```

Out[15]: 0.7019577122638231

c. Random variable - using the cdf

In [16]:
```
rv.cdf(70)-rv.cdf(60)
```

Out[16]: 0.7039868846425189

Problem 6

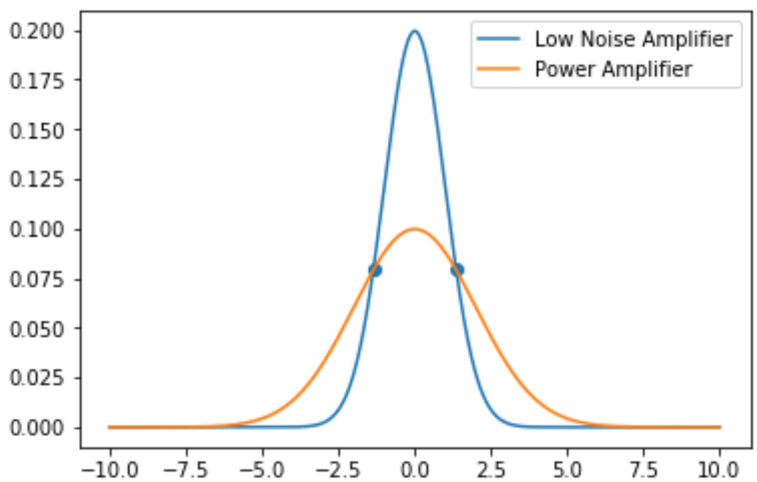
Part A

In [17]:
```
def solve(m1,m2,std1,std2,s1,s2):
  a = 1/(2*std1**2) - 1/(2*std2**2)
  b = m2/(std2**2) - m1/(std1**2)
  c = m1**2 /(2*std1**2) - m2**2 / (2*std2**2) - np.log((std2*s1)/(std1*
s2))
  return np.roots([a,b,c])
```

In [18]:
```
def plot_curves(m1,m2,std1,std2,s1,s2,l1,l2):
    low_noise = stats.norm(loc=m1,scale=std1)
    power = stats.norm(loc=m2,scale=std2)

    fig, ax = plt.subplots()

    x=np.linspace(-10,10,100000)
    ax.plot(x,s1*low_noise.pdf(x),label=l1)
    ax.plot(x,s2*power.pdf(x),label=l2)

    intersections_x = solve(m1,m2,std1,std2,s1,s2)
    intersections_y = s1*low_noise.pdf(intersections_x)

    ax.scatter(intersections_x,intersections_y)

    ax.legend()

    plt.show()

    return intersections_x,intersections_y
```

In [19]:
```python
def map_decide(intersections_x, curr_x, decision_1, decision_2):
    if ((intersections_x[1] < curr_x and curr_x < intersections_x[0]) or
(intersections_x[1] < curr_x and curr_x < intersections_x[0])):
        return decision_1
    else:
        return decision_2
```

In [20]:
```python
m1 = 0 # mean
std1 = 1**.5 # std_dev
s1 = 1/2 # scale factor
l1 = "Low Noise Amplifier"

m2 = 0 # mean
std2 = 4**.5 # std_dev
s2 = 1/2 #scale factor
l2 = "Power Amplifier"

intersections_x,intersections_y = plot_curves(m1,m2,std1,std2,s1,s2,l1,l
2)

print("MAP decision boundary:\nChoose the Power Amplifier for any point
 with an x less than ",intersections_x[0],",",intersections_y[0]," or an
x greater than ",intersections_x[1],",",intersections_y[1],". For the re
gion inbetween these two points, choose the Low Noise Amplifier.",sep=""
)
```




```
MAP decision boundary:
Choose the Power Amplifier for any point with an x less than 1.35955598
68917453,0.07916017444797833 or an x greater than -1.3595559868917453,
0.07916017444797833. For the region inbetween these two points, choose
the Low Noise Amplifier.
```

Part B

In [21]:
```
# Provided a set of intersections, one can determine
# the decision boundary for any scaling of these two
# normal distributions

m1 = 0 # mean
std1 = 1**.5 # std_dev
s1 = 1/2 # scale factor
l1 = "Low Noise Amplifier"

m2 = 0 # mean
std2 = 4**.5 # std_dev
s2 = 1/2 #scale factor
l2 = "Power Amplifier"

for x_int in range(-75,75,2):
    x = x_int/10
    print("Decide ", map_decide(solve(m1,m2,std1,std2,s1,s2), x, l1, l2
), ", given x=", x, sep="")
```

```
Decide Power Amplifier, given x=-7.5
Decide Power Amplifier, given x=-7.3
Decide Power Amplifier, given x=-7.1
Decide Power Amplifier, given x=-6.9
Decide Power Amplifier, given x=-6.7
Decide Power Amplifier, given x=-6.5
Decide Power Amplifier, given x=-6.3
Decide Power Amplifier, given x=-6.1
Decide Power Amplifier, given x=-5.9
Decide Power Amplifier, given x=-5.7
Decide Power Amplifier, given x=-5.5
Decide Power Amplifier, given x=-5.3
Decide Power Amplifier, given x=-5.1
Decide Power Amplifier, given x=-4.9
Decide Power Amplifier, given x=-4.7
Decide Power Amplifier, given x=-4.5
Decide Power Amplifier, given x=-4.3
Decide Power Amplifier, given x=-4.1
Decide Power Amplifier, given x=-3.9
Decide Power Amplifier, given x=-3.7
Decide Power Amplifier, given x=-3.5
Decide Power Amplifier, given x=-3.3
Decide Power Amplifier, given x=-3.1
Decide Power Amplifier, given x=-2.9
Decide Power Amplifier, given x=-2.7
Decide Power Amplifier, given x=-2.5
Decide Power Amplifier, given x=-2.3
Decide Power Amplifier, given x=-2.1
Decide Power Amplifier, given x=-1.9
Decide Power Amplifier, given x=-1.7
Decide Power Amplifier, given x=-1.5
Decide Low Noise Amplifier, given x=-1.3
Decide Low Noise Amplifier, given x=-1.1
Decide Low Noise Amplifier, given x=-0.9
Decide Low Noise Amplifier, given x=-0.7
Decide Low Noise Amplifier, given x=-0.5
Decide Low Noise Amplifier, given x=-0.3
Decide Low Noise Amplifier, given x=-0.1
Decide Low Noise Amplifier, given x=0.1
Decide Low Noise Amplifier, given x=0.3
Decide Low Noise Amplifier, given x=0.5
Decide Low Noise Amplifier, given x=0.7
Decide Low Noise Amplifier, given x=0.9
Decide Low Noise Amplifier, given x=1.1
Decide Low Noise Amplifier, given x=1.3
Decide Power Amplifier, given x=1.5
Decide Power Amplifier, given x=1.7
Decide Power Amplifier, given x=1.9
Decide Power Amplifier, given x=2.1
Decide Power Amplifier, given x=2.3
Decide Power Amplifier, given x=2.5
Decide Power Amplifier, given x=2.7
Decide Power Amplifier, given x=2.9
Decide Power Amplifier, given x=3.1
Decide Power Amplifier, given x=3.3
Decide Power Amplifier, given x=3.5
Decide Power Amplifier, given x=3.7
```

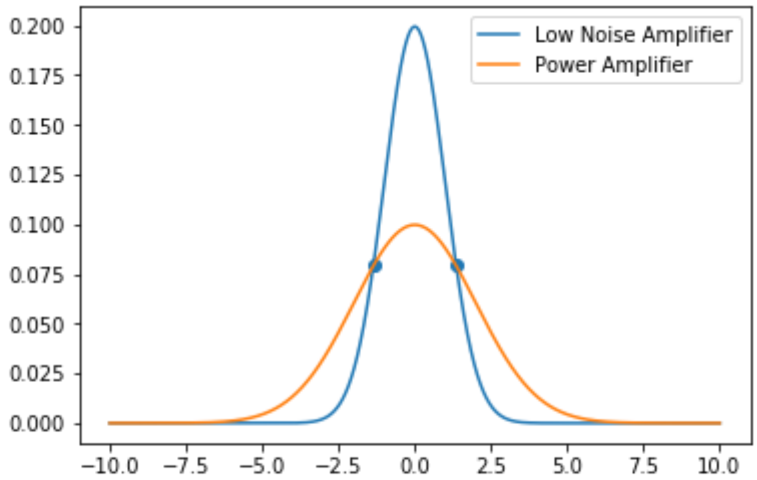
```
Decide Power Amplifier, given x=3.9
Decide Power Amplifier, given x=4.1
Decide Power Amplifier, given x=4.3
Decide Power Amplifier, given x=4.5
Decide Power Amplifier, given x=4.7
Decide Power Amplifier, given x=4.9
Decide Power Amplifier, given x=5.1
Decide Power Amplifier, given x=5.3
Decide Power Amplifier, given x=5.5
Decide Power Amplifier, given x=5.7
Decide Power Amplifier, given x=5.9
Decide Power Amplifier, given x=6.1
Decide Power Amplifier, given x=6.3
Decide Power Amplifier, given x=6.5
Decide Power Amplifier, given x=6.7
Decide Power Amplifier, given x=6.9
Decide Power Amplifier, given x=7.1
Decide Power Amplifier, given x=7.3
```

Part C

In [22]:
```
def vary_densities(low_noise_prob, plot_logarithmic=False):
    m1 = 0 # mean
    std1 = 1**.5 # std_dev
    s1 = low_noise_prob # scale factor
    l1 = "Low Noise Amplifier"

    m2 = 0 # mean
    std2 = 4**.5 # std_dev
    s2 = 1-low_noise_prob #scale factor
    l2 = "Power Amplifier"

    plot_curves(m1,m2,std1,std2,s1,s2,l1,l2)
```

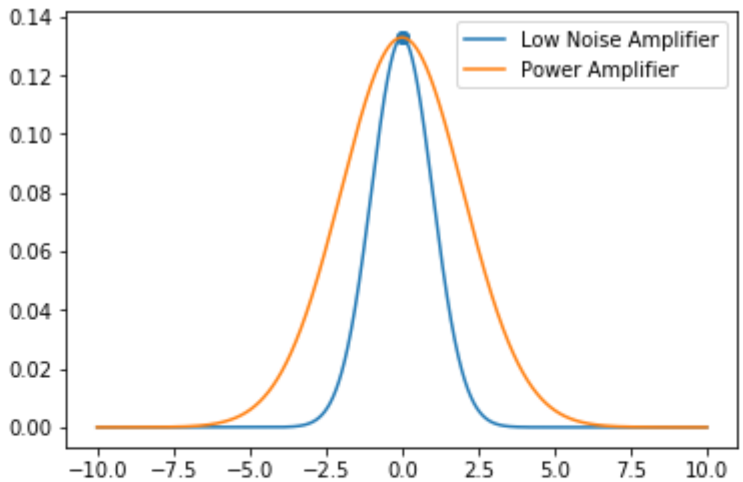
Part D

In [23]:
```
vary_densities(0.5)
```



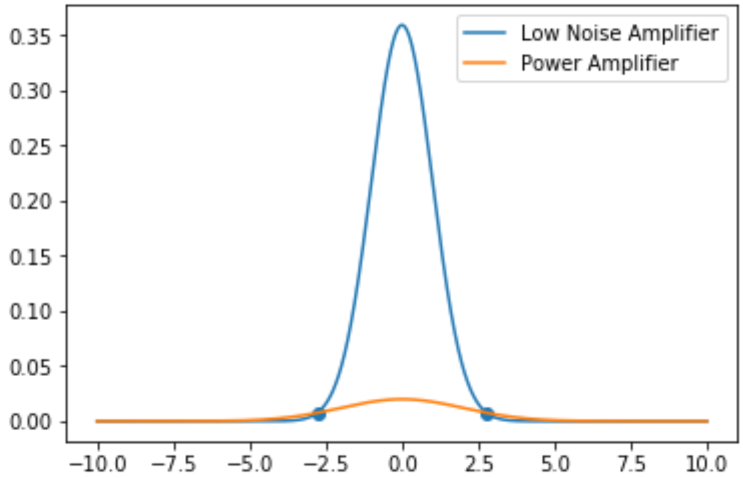
Part E

In [30]:
```
vary_densities(1/3)
```



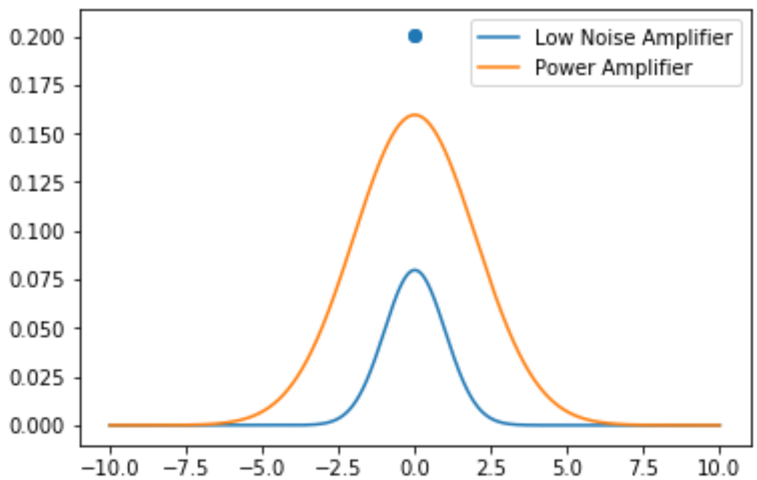
Part F

In [25]:
```
vary_densities(0.9)
```



Part G

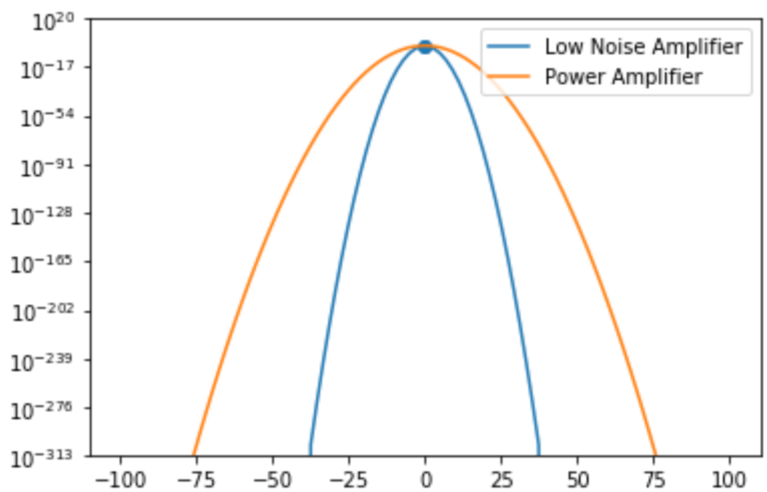
In [26]:
```
vary_densities(0.2)
```



When the probability of a random chosen chip being a low noise amplifier is 0.2, the math breaks down and we always decide that it's a power amplifier and never decide that it's a low noise amplifier.

Part H

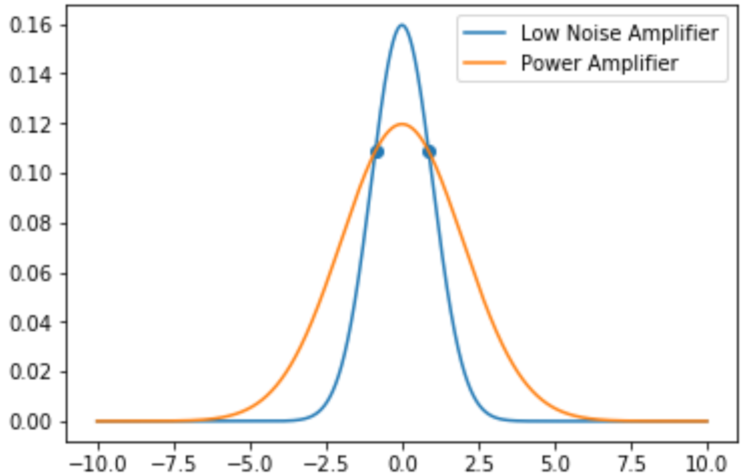
In [27]:
```
low_noise_prob = 0.2

m1 = 0 # mean
std1 = 1**.5 # std_dev
s1 = low_noise_prob # scale factor
l1 = "Low Noise Amplifier"

m2 = 0 # mean
std2 = 4**.5 # std_dev
s2 = 1-low_noise_prob #scale factor
l2 = "Power Amplifier"

low_noise = stats.norm(loc=m1,scale=std1)
power = stats.norm(loc=m2,scale=std2)

x=np.linspace(-10**2,10**2,10**2)
plt.semilogy(x,s1*low_noise.pdf(x),label=l1)
plt.semilogy(x,s2*power.pdf(x),label=l2)

intersections_x = solve(m1,m2,std1,std2,s1,s2)
intersections_y = s1*low_noise.pdf(intersections_x)

plt.scatter(intersections_x,intersections_y)

plt.legend();
```



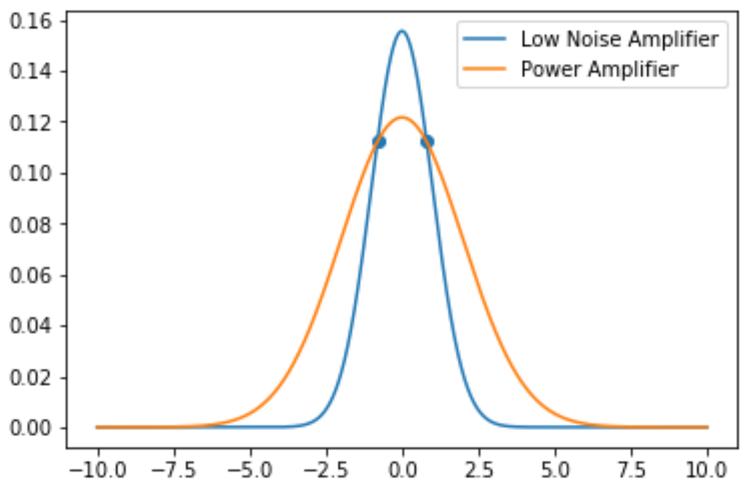
Part I

In [29]:
```
for x_int in range (40,20,-1):
    x = x_int/100
    print("Probability of a randomly chosen chip being a low-noise ampli
fier=",x,".",sep="")
    vary_densities(x)
```

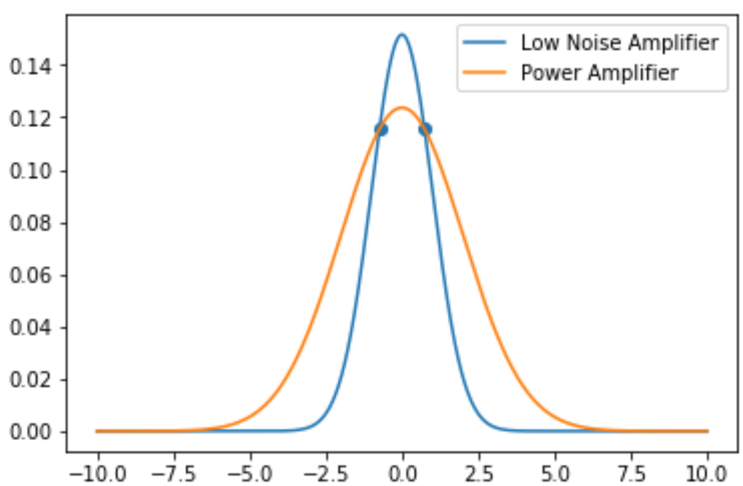
```
Probability of a randomly chosen chip being a low-noise amplifier=0.4.
```



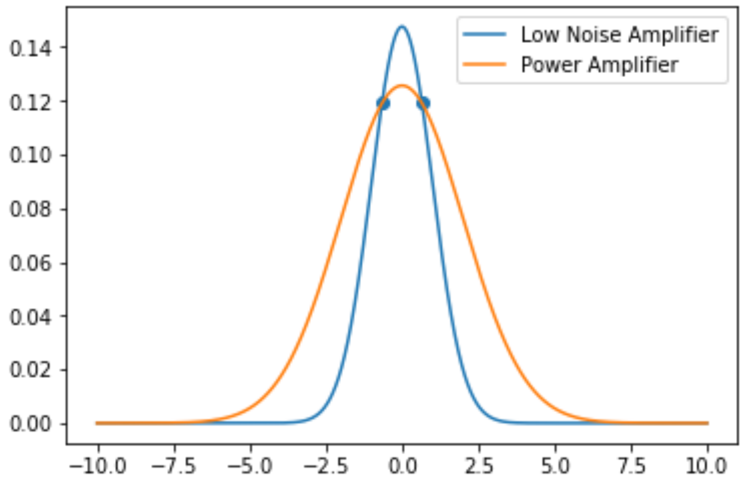

```
Probability of a randomly chosen chip being a low-noise amplifier=0.39.
```



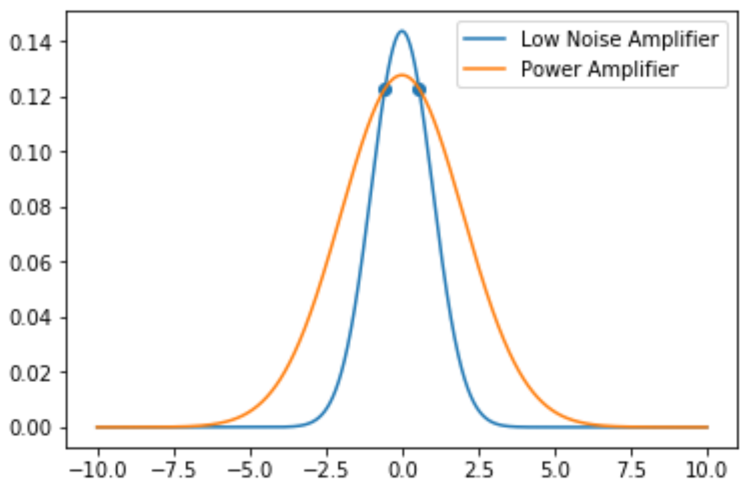

```
Probability of a randomly chosen chip being a low-noise amplifier=0.38.
```



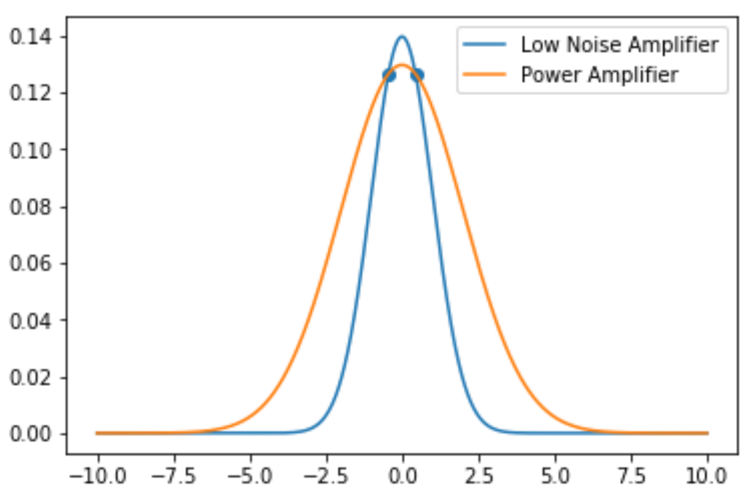

```
Probability of a randomly chosen chip being a low-noise amplifier=0.37.
```



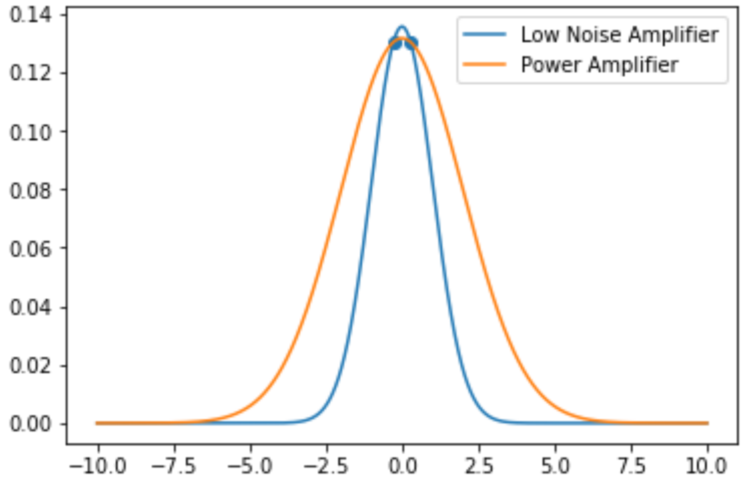

```
Probability of a randomly chosen chip being a low-noise amplifier=0.36.
```



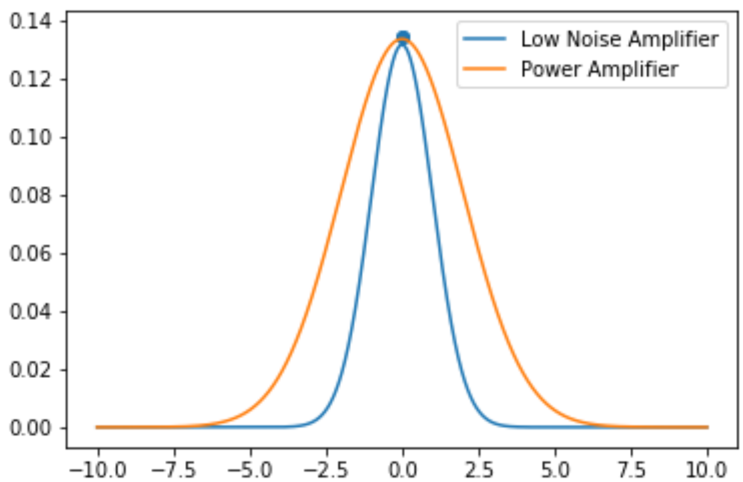

```
Probability of a randomly chosen chip being a low-noise amplifier=0.35.
```



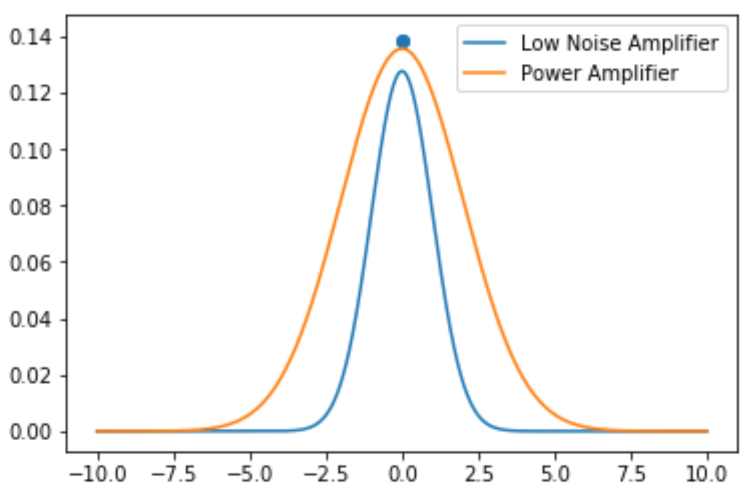

```
Probability of a randomly chosen chip being a low-noise amplifier=0.34.
```



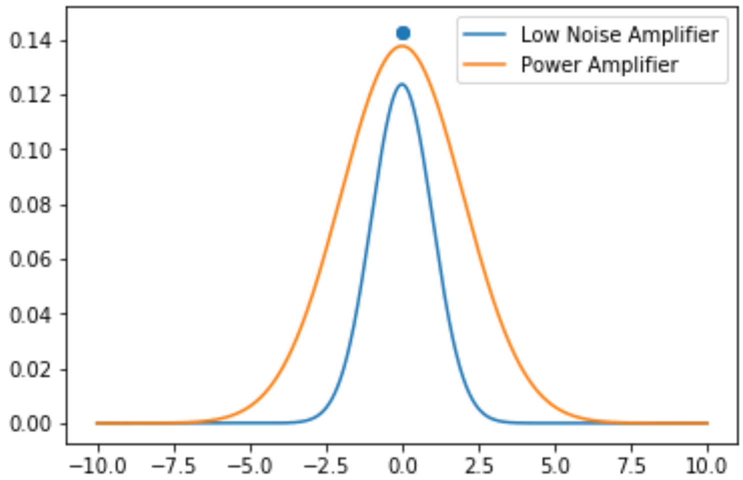
Probability of a randomly chosen chip being a low-noise amplifier=0.33.



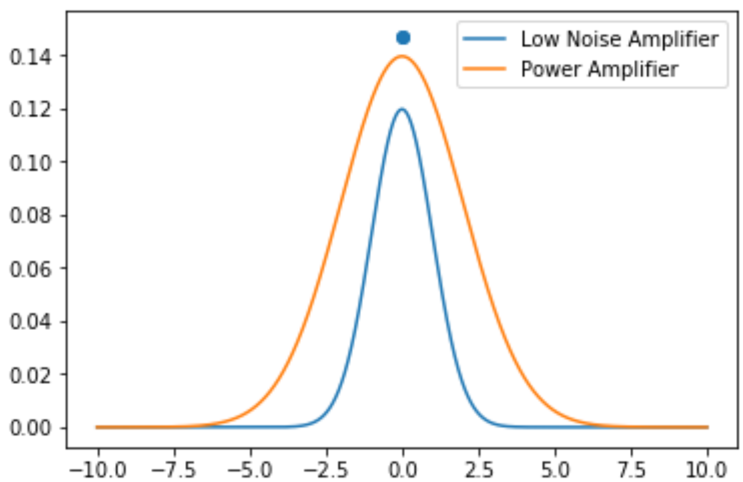
Probability of a randomly chosen chip being a low-noise amplifier=0.32.



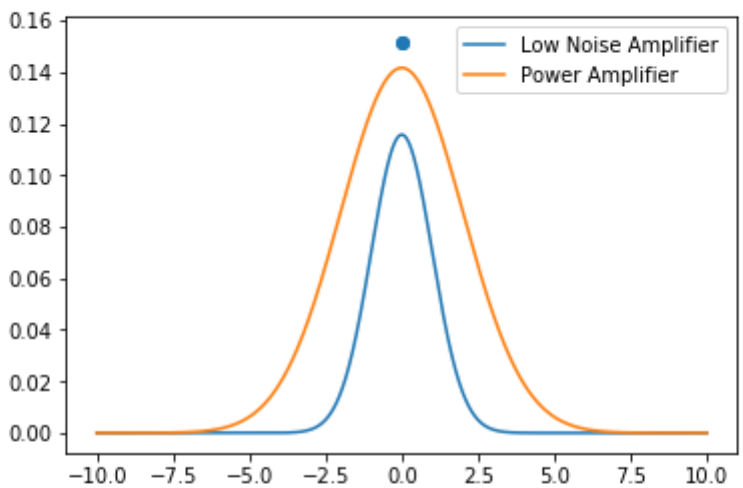
Probability of a randomly chosen chip being a low-noise amplifier=0.31.



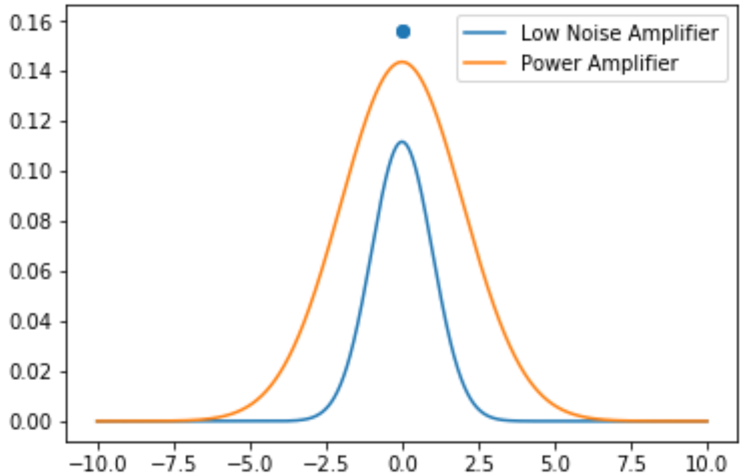

```
Probability of a randomly chosen chip being a low-noise amplifier=0.3.
```



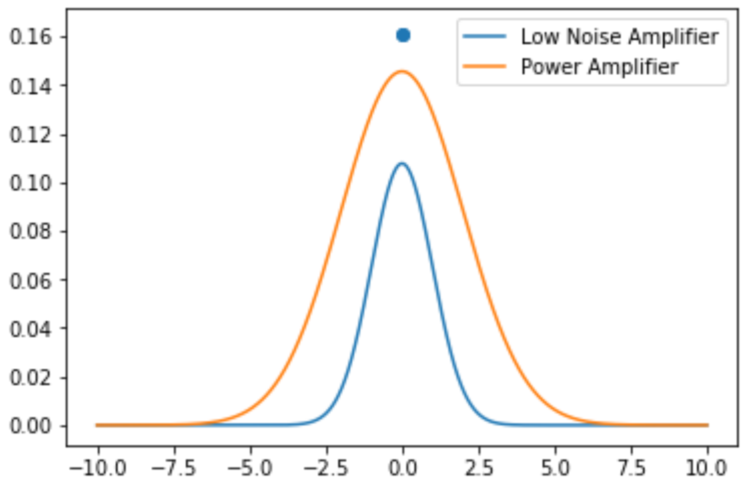

```
Probability of a randomly chosen chip being a low-noise amplifier=0.29.
```



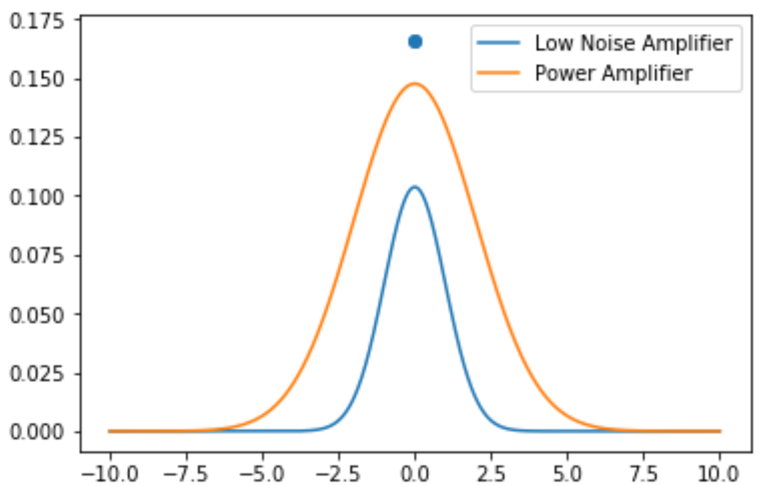

```
Probability of a randomly chosen chip being a low-noise amplifier=0.28.
```



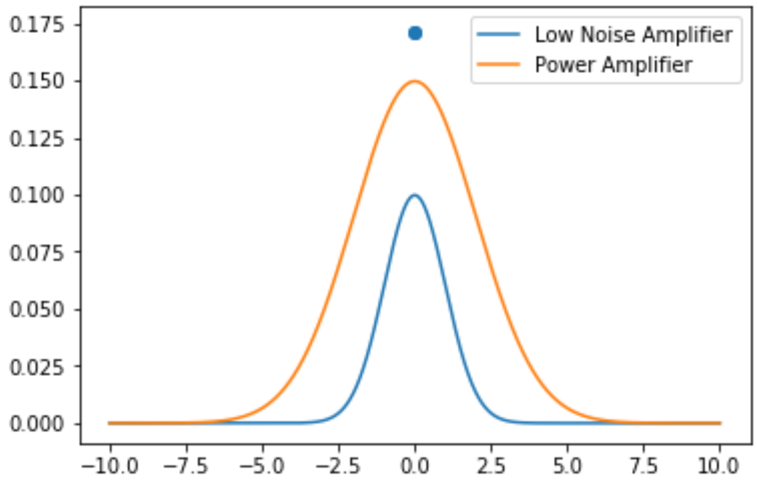

```
Probability of a randomly chosen chip being a low-noise amplifier=0.27.
```



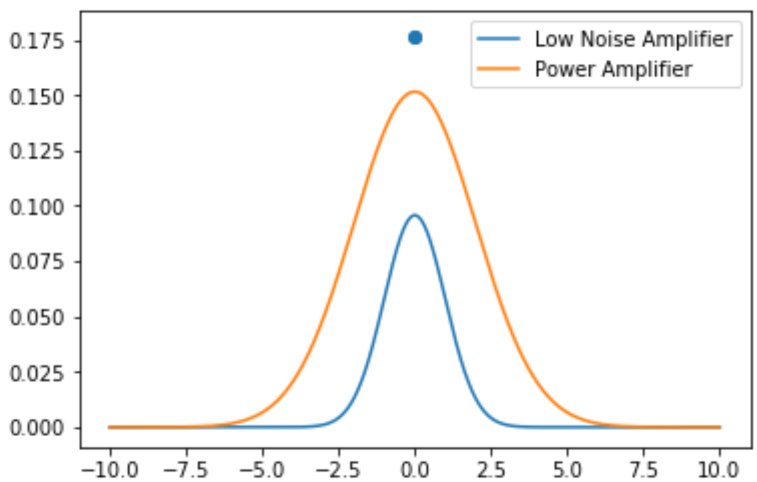

```
Probability of a randomly chosen chip being a low-noise amplifier=0.26.
```



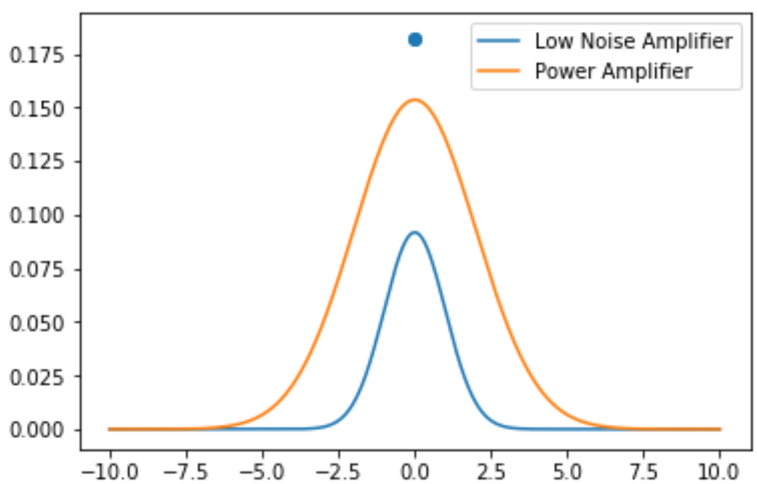

```
Probability of a randomly chosen chip being a low-noise amplifier=0.25.
```



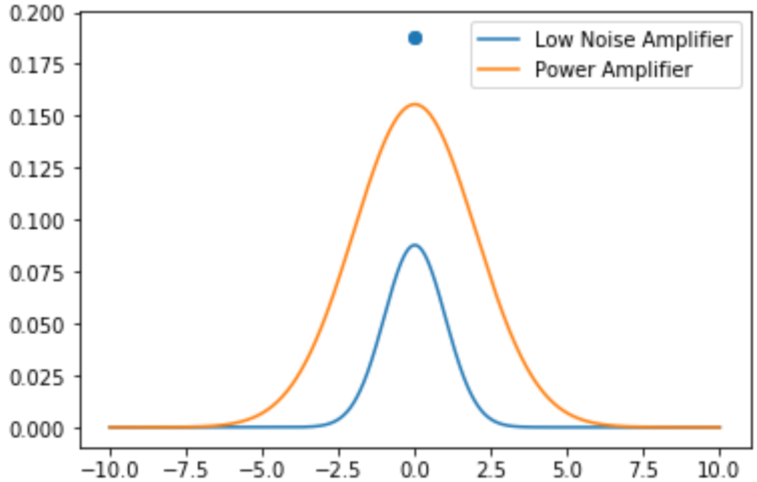

```
Probability of a randomly chosen chip being a low-noise amplifier=0.24.
```



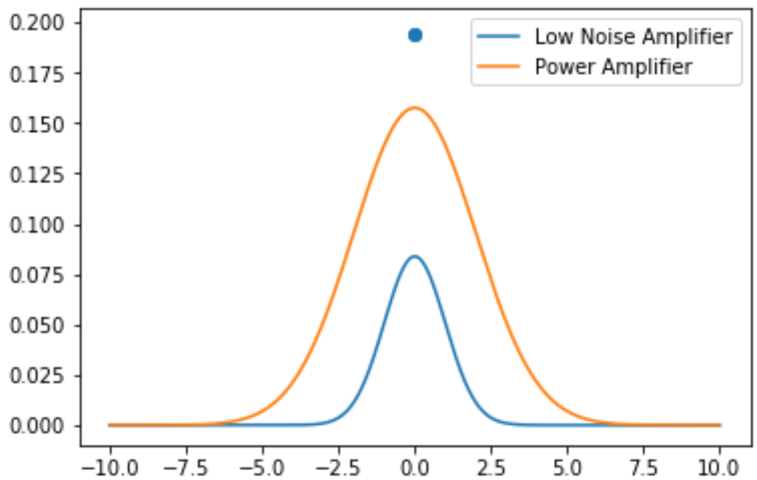

```
Probability of a randomly chosen chip being a low-noise amplifier=0.23.
```




```
Probability of a randomly chosen chip being a low-noise amplifier=0.22.
```




```
Probability of a randomly chosen chip being a low-noise amplifier=0.21.
```



As seen above, the logic breaks down when the probability of a randomly chosen chip being a low-noise amplifier is approximately one third.