Assignment Normal Distribution

Date: 04/05/2020 Name: D.Saravanan

1. Find the area under the standard normal curve which lie

   A normal distribution in a variate X with mean μ and variance σ^2 is a statistic distribution with probability density function

   $$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/(2\sigma^2)}$$

   The standard normal distribution is given by taking $\mu = 0$ and $\sigma^2 = 1$ in a general normal distribution. An arbitrary normal distribution can be converted to a standard normal distribution by changing variables to $Z \equiv (X - \mu)/\sigma$, so $dz = dx/\sigma$, yielding

   $$P(x)dx = \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz$$

   The normal distribution function $\Phi(z)$ gives the probability that a standard normal variate assumes a value in the interval $[0, z]$,

   $$\Phi(z) \equiv \frac{1}{\sqrt{2\pi}} \int_0^z e^{-x^2/2} dx$$

   The normal distribution is the limiting case of a discrete binomial distribution $P_p(n|N)$ as the sample size N becomes large, in which case $P_p(n|N)$ is normal with mean $\mu = Np$ and variance $\sigma^2 = Np(1-p)$.

   The distribution $P(x)$ is properly normalized since

   $$\int_{-\infty}^{\infty} P(x)dx = 1$$

   The cumulative distribution function, which gives the probability that a variate will assume a value $\leq x$, is then the integral of the normal distribution,

   $$\begin{aligned} D(x) &\equiv \int_{-\infty}^{x} P(x^{'})dx^{'} \\ &= \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} e^{-(x^{'}-\mu)^2/(2\sigma^2)} dx^{'} \end{aligned}$$

   Normal distributions have many convenient properties, so random variates with unknown distributions are often assumed to be normal. It is often a good approximation due to the central limit theorem. This theorem states that the mean of any set of variates with any distribution having a finite mean and variance tends to the normal distribution.

a) To the right of $Z = 2.70$

$$P(Z > 2.70) = \frac{1}{\sqrt{2\pi}} \int_{2.70}^{\infty} e^{-z^2/2} dz = 0.00347$$

Program:

```
#!/usr/bin/env python3
import numpy as np
from scipy.integrate import quad

def integrand(x): return 1/np.sqrt(2*np.pi) * np.exp(-x**2/2)

ans, err = quad(integrand, 2.70, np.inf)
print("Area under the standard normal curve P(Z > 2.70): {:.5f}".format(ans))
```

Output:

```
Area under the standard normal curve P(Z > 2.70): 0.00347
```

Program:

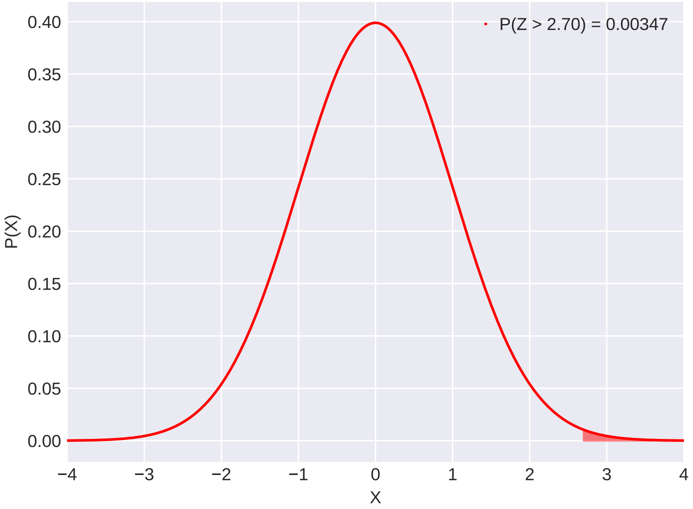
```
#!/usr/bin/env python3
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt
plt.style.use('seaborn-darkgrid')

# compute survival function (SF): P(Z > 2.70)
x = 2.70; loc = 0; scale = 1
pls = norm.sf(x, loc, scale)
print("Area under the standard normal curve P(Z > 2.70): {:.5f}".format(pls))

x1=np.linspace(x,10,1000); y1=norm.pdf(x1,loc,scale)
x2=np.linspace(-4,4,1000); y2=norm.pdf(x2,loc,scale)
ax = plt.figure().add_subplot(111); ax.minorticks_on()
ax.plot(x2,y2,c='r',label='P(Z > 2.70) = {:.5f}'.format(pls))
ax.set(xlabel='X',ylabel='P(X)'); ax.legend(handlelength=0)
ax.fill_between(x1,y1,alpha=0.5,color='r'); ax.set(xlim=[-4,4])
plt.savefig('norm1a.pdf',dpi=72,bbox_inches='tight'); plt.show()
```

Output:

```
Area under the standard normal curve P(Z > 2.70): 0.00347
```



b) To the left of $Z = 1.73$

$$P(Z < 1.73) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{1.73} e^{-z^2/2} dz = 0.95818$$

Program:

```
#!/usr/bin/env python3
import numpy as np
from scipy.integrate import quad

def integrand(x): return 1/np.sqrt(2*np.pi) * np.exp(-x**2/2)

ans, err = quad(integrand, -np.inf, 1.73)
print("Area under the standard normal curve P(Z < 1.73): {:.5f}".format(ans))
```

Output:

```
Area under the standard normal curve P(Z < 1.73): 0.95818
```

Program:

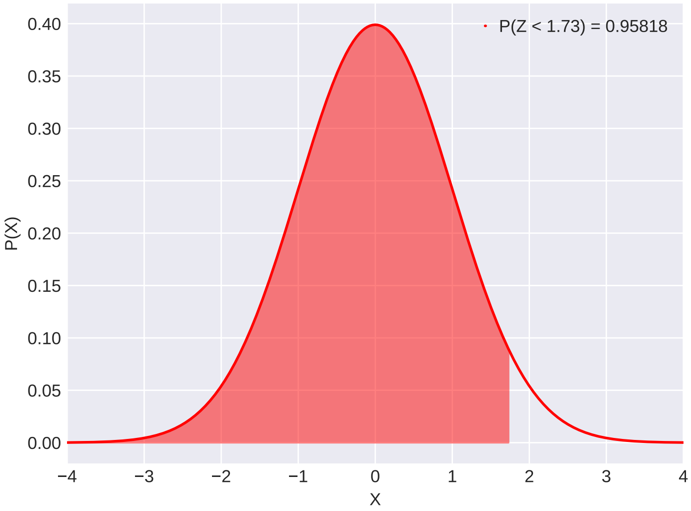
```
#!/usr/bin/env python3
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt
plt.style.use('seaborn-darkgrid')

# compute cumulative distribution function (CDF): P(Z < 1.73)
x = 1.73; loc = 0; scale = 1
pls = norm.cdf(x, loc, scale)
print("Area under the standard normal curve P(Z < 1.73): {:.5f}".format(pls))

x1=np.linspace(-10,x,1000);y1=norm.pdf(x1,loc,scale)
x2=np.linspace(-4,4,1000); y2=norm.pdf(x2,loc,scale)
ax = plt.figure().add_subplot(111); ax.minorticks_on()
ax.plot(x2,y2,c='r',label='P(Z < 1.73) = {:.5f}'.format(pls))
ax.set(xlabel='X',ylabel='P(X)'); ax.legend(handlelength=0)
ax.fill_between(x1,y1,alpha=0.5,color='r'); ax.set(xlim=[-4,4])
plt.savefig('norm1b.pdf',dpi=72,bbox_inches='tight'); plt.show()
```

Output:

```
Area under the standard normal curve P(Z < 1.73): 0.95818
```



c) To the right of $Z = -0.66$

$$P(Z > -0.66) = \frac{1}{\sqrt{2\pi}} \int_{-0.66}^{\infty} e^{-z^2/2} dz = 0.74537$$

Program:

```
#!/usr/bin/env python3
import numpy as np
from scipy.integrate import quad

def integrand(x): return 1/np.sqrt(2*np.pi) * np.exp(-x**2/2)

ans, err = quad(integrand, -0.66, np.inf)
print("Area under the standard normal curve P(Z > -0.66): {:.5f}".format(ans))
```

Output:

```
Area under the standard normal curve P(Z > -0.66): 0.74537
```

Program:

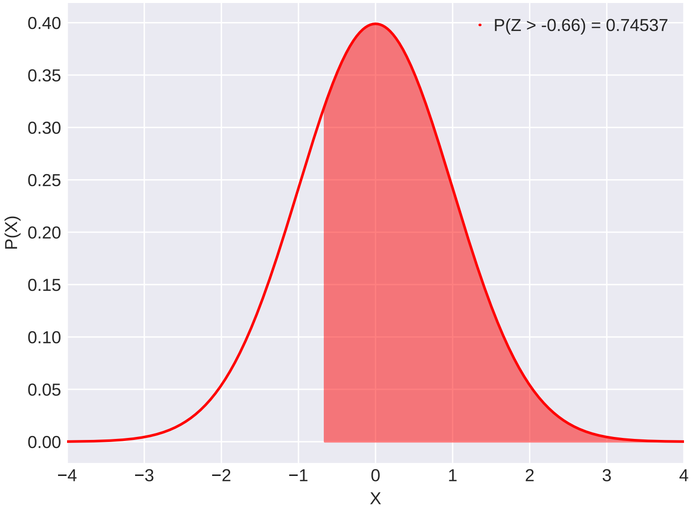
```
#!/usr/bin/env python3
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt
plt.style.use('seaborn-darkgrid')

# compute survival function (SF): P(Z > -0.66)
x = -0.66; loc = 0; scale = 1; pls = norm.sf(x, loc, scale)
print("Area under the standard normal curve P(Z > -0.66): {:.5f}".format(pls))

x1=np.linspace(x,10,1000); y1=norm.pdf(x1,loc,scale)
x2=np.linspace(-4,4,1000); y2=norm.pdf(x2,loc,scale)
ax = plt.figure().add_subplot(111); ax.minorticks_on()
ax.plot(x2,y2,c='r',label='P(Z > -0.66) = {:.5f}'.format(pls))
ax.set(xlabel='X',ylabel='P(X)'); ax.legend(handlelength=0)
ax.fill_between(x1,y1,alpha=0.5,color='r'); ax.set(xlim=[-4,4])
plt.savefig('norm1c.pdf',dpi=72,bbox_inches='tight'); plt.show()
```

Output:

```
Area under the standard normal curve P(Z > -0.66): 0.74537
```



d) To the left of $Z = -1.88$

$$P(Z < -1.88) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{-1.88} e^{-z^2/2} dz = 0.03005$$

Program:

```
#!/usr/bin/env python3
import numpy as np
from scipy.integrate import quad

def integrand(x): return 1/np.sqrt(2*np.pi) * np.exp(-x**2/2)

ans, err = quad(integrand, -np.inf, -1.88)
print("Area under the standard normal curve P(Z < -1.88): {:.5f}".format(ans))
```

Output:

```
Area under the standard normal curve P(Z < -1.88): 0.03005
```

Program:

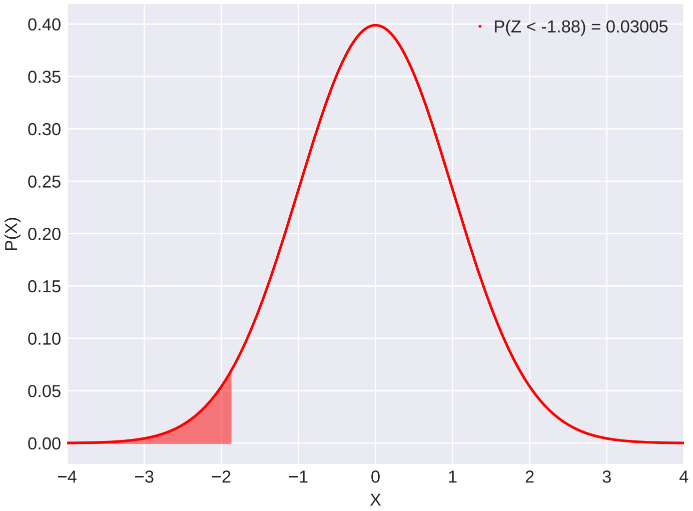
```
#!/usr/bin/env python3
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt
plt.style.use('seaborn-darkgrid')

# compute cumulative distribution function (CDF): P(Z < -1.88)
x = -1.88; loc = 0; scale = 1; pls = norm.cdf(x, loc, scale)
print("Area under the standard normal curve P(Z < -1.88): {:.5f}".format(pls))

x1=np.linspace(-10,x,1000);y1=norm.pdf(x1,loc,scale)
x2=np.linspace(-4,4,1000); y2=norm.pdf(x2,loc,scale)
ax = plt.figure().add_subplot(111); ax.minorticks_on()
ax.plot(x2,y2,c='r',label='P(Z < -1.88) = {:.5f}'.format(pls))
ax.set(xlabel='X',ylabel='P(X)'); ax.legend(handlelength=0)
ax.fill_between(x1,y1,alpha=0.5,color='r'); ax.set(xlim=[-4,4])
plt.savefig('norm1d.pdf',dpi=72,bbox_inches='tight'); plt.show()
```

Output:

```
Area under the standard normal curve P(Z < -1.88): 0.03005
```



e) Between $Z = -0.90$ and $Z = -1.85$

$$P(-1.85 < Z < -0.90) = \frac{1}{\sqrt{2\pi}} \int_{-1.85}^{-0.90} e^{-z^2/2} dz = 0.15190$$

Program:

```
#!/usr/bin/env python3
import numpy as np
from scipy.integrate import quad

def integrand(x): return 1/np.sqrt(2*np.pi) * np.exp(-x**2/2)

ans, err = quad(integrand, -1.85, -0.90)
print("Area under the standard normal curve P(-1.85 < Z < -0.90): {:.5f}".format(ans))
```

Output:

```
Area under the standard normal curve P(-1.85 < Z < -0.90): 0.15190
```

Program:

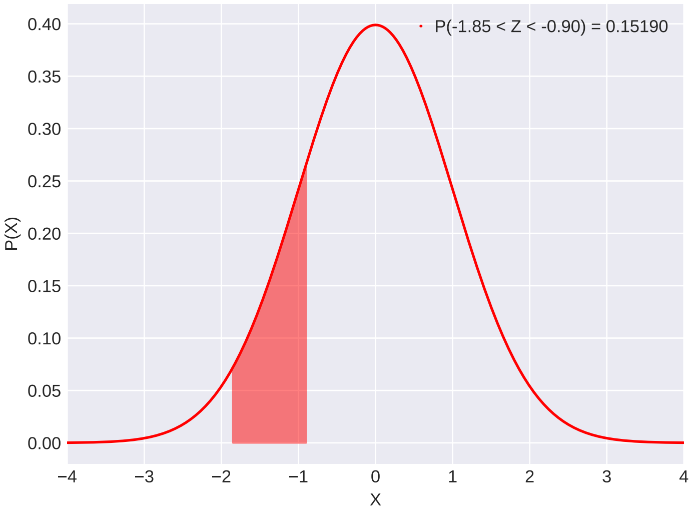
```
#!/usr/bin/env python3
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt
plt.style.use('seaborn-darkgrid')

# compute cumulative distribution function (CDF): P(-1.85 < Z < -0.90)
x = -1.85; y = -0.90; loc = 0; scale = 1
pls = norm.cdf(y, loc, scale) - norm.cdf(x, loc, scale)
print("Area under the standard normal curve P(-1.85 < Z < -0.90): {:.5f}".format(pls))

x1=np.linspace(x,y, 1000); y1=norm.pdf(x1,loc,scale)
x2=np.linspace(-4,4,1000); y2=norm.pdf(x2,loc,scale)
ax = plt.figure().add_subplot(111); ax.minorticks_on()
ax.plot(x2,y2,c='r',label='P(-1.85 < Z < -0.90) = {:.5f}'.format(pls))
ax.set(xlabel='X',ylabel='P(X)'); ax.legend(handlelength=0)
ax.fill_between(x1,y1,alpha=0.5,color='r'); ax.set(xlim=[-4,4])
plt.savefig('norm1e.pdf',dpi=72,bbox_inches='tight'); plt.show()
```

Output:

```
Area under the standard normal curve P(-1.85 < Z < -0.90): 0.15190
```



f) Between $Z = -1.45$ and $Z = 1.45$

$$P(-1.45 < Z < 1.45) = \frac{1}{\sqrt{2\pi}} \int_{-1.45}^{1.45} e^{-z^2/2} dz = 0.85294$$

Program:

```
#!/usr/bin/env python3
import numpy as np
from scipy.integrate import quad

def integrand(x): return 1/np.sqrt(2*np.pi) * np.exp(-x**2/2)

ans, err = quad(integrand, -1.45, 1.45)
print("Area under the standard normal curve P(-1.45 < Z < 1.45): {:.5f}".format(ans))
```

Output:

```
Area under the standard normal curve P(-1.45 < Z < 1.45): 0.85294
```

Program:

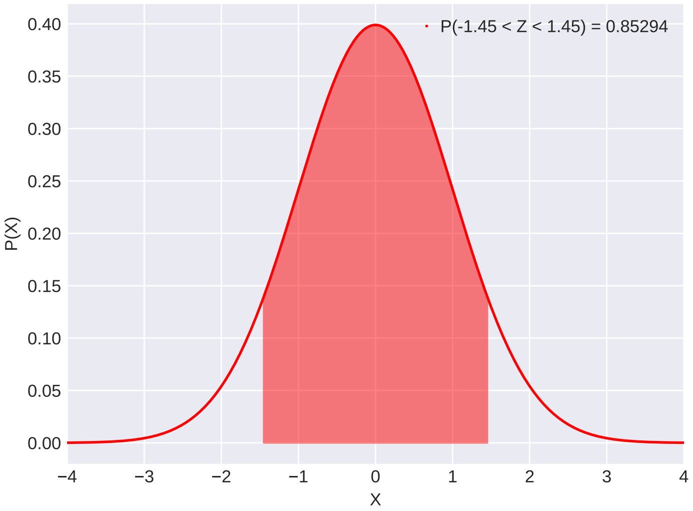
```
#!/usr/bin/env python3
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt
plt.style.use('seaborn-darkgrid')

# compute cumulative distribution function (CDF): P(-1.45 < Z < 1.45)
x = -1.45; y = 1.45; loc = 0; scale = 1
pls = norm.cdf(y, loc, scale) - norm.cdf(x, loc, scale)
print("Area under the standard normal curve P(-1.45 < Z < 1.45): {:.5f}".format(pls))

x1=np.linspace(x,y, 1000); y1=norm.pdf(x1,loc,scale)
x2=np.linspace(-4,4,1000); y2=norm.pdf(x2,loc,scale)
ax = plt.figure().add_subplot(111); ax.minorticks_on()
ax.plot(x2,y2,c='r',label='P(-1.45 < Z < 1.45) = {:.5f}'.format(pls))
ax.set(xlabel='X',ylabel='P(X)'); ax.legend(handlelength=0)
ax.fill_between(x1,y1,alpha=0.5,color='r'); ax.set(xlim=[-4,4])
plt.savefig('norm1f.pdf',dpi=72,bbox_inches='tight'); plt.show()
```

Output:

```
Area under the standard normal curve P(-1.45 < Z < 1.45): 0.85294
```



g) Between $Z = -0.90$ and $Z = 1.58$

$$P(-0.90 < Z < 1.58) = \frac{1}{\sqrt{2\pi}} \int_{-0.90}^{1.58} e^{-z^2/2} dz = 0.75889$$

Program:

```
#!/usr/bin/env python3
import numpy as np
from scipy.integrate import quad

def integrand(x): return 1/np.sqrt(2*np.pi) * np.exp(-x**2/2)

ans, err = quad(integrand, -0.90, 1.58)
print("Area under the standard normal curve P(-0.90 < Z < 1.58): {:.5f}".format(ans))
```

Output:

```
Area under the standard normal curve P(-0.90 < Z < 1.58): 0.75889
```

Program:

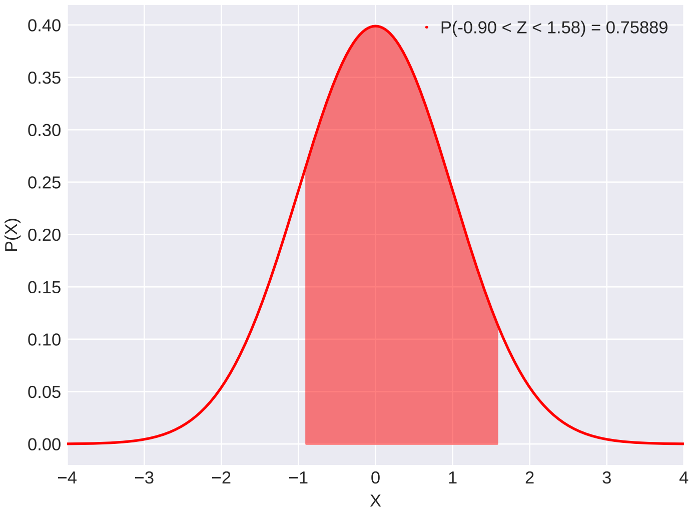
```
#!/usr/bin/env python3
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt
plt.style.use('seaborn-darkgrid')

# compute cumulative distribution function (CDF): P(-0.90 < Z < 1.58)
x = -0.90; y = 1.58; loc = 0; scale = 1
pls = norm.cdf(y, loc, scale) - norm.cdf(x, loc, scale)
print("Area under the standard normal curve P(-0.90 < Z < 1.58): {:.5f}".format(pls))

x1=np.linspace(x,y, 1000); y1=norm.pdf(x1,loc,scale)
x2=np.linspace(-4,4,1000); y2=norm.pdf(x2,loc,scale)
ax = plt.figure().add_subplot(111); ax.minorticks_on()
ax.plot(x2,y2,c='r',label='P(-0.90 < Z < 1.58) = {:.5f}'.format(pls))
ax.set(xlabel='X',ylabel='P(X)'); ax.legend(handlelength=0)
ax.fill_between(x1,y1,alpha=0.5,color='r'); ax.set(xlim=[-4,4])
plt.savefig('norm1g.pdf',dpi=72,bbox_inches='tight'); plt.show()
```

Output:

```
Area under the standard normal curve P(-0.90 < Z < 1.58): 0.75889
```



2. The life of a certain kind of electronic device has a mean of 300 hours and a standard deviation of 25 hours. Assuming that the distribution of life times which are measured to the nearest hours can be approximated closely with a normal curve.

   a) Find the probability that any one of these devices will have a lifetime of more than 350 hours.

   given, $\mu = 300$ and $\sigma = 25$

$$\begin{aligned} P(X > 350) &= \frac{1}{\sigma\sqrt{2\pi}} \int_{350}^{\infty} e^{-(x-\mu)^2/(2\sigma^2)} dx \\ &= \frac{1}{25\sqrt{2\pi}} \int_{350}^{\infty} e^{-(x-300)^2/(2\times 25^2)} dx \\ &= 0.02275 \end{aligned}$$

   Program:

```
#!/usr/bin/env python3
import numpy as np
from scipy.integrate import quad

def integrand(x):
    return 1/(sigma*np.sqrt(2*np.pi)) * np.exp(-(x-mu)**2/(2*sigma**2))

mu = 300; sigma = 25
ans, err = quad(integrand, 350, np.inf)
print("Probability, P(X > 350): {:.5f}".format(ans))
```

   Output:

```
Probability, P(X > 350): 0.02275
```

   Program:

```
#!/usr/bin/env python3
from scipy.stats import norm

# compute survival function (SF)
# P(X > 350)
pls = norm.sf(x=350, loc=300, scale=25)
print("Probability, P(X > 350): {:.5f}".format(pls))
```

   Output:

```
Probability, P(X > 350): 0.02275
```

   b) What percentage will have life time from 220 to 260 hours?

$$\begin{aligned} P(220 < X < 260) &= \frac{1}{\sigma\sqrt{2\pi}} \int_{220}^{260} e^{-(x-\mu)^2/(2\sigma^2)} dx \\ &= \frac{1}{25\sqrt{2\pi}} \int_{220}^{260} e^{-(x-300)^2/(2\times 25^2)} dx \\ &= 0.05411 \end{aligned}$$

Program:

```
#!/usr/bin/env python3
import numpy as np
from scipy.integrate import quad

def integrand(x):
    return 1/(sigma*np.sqrt(2*np.pi)) * np.exp(-(x-mu)**2/(2*sigma**2))

mu = 300; sigma = 25
ans, err = quad(integrand, 220, 260)
print("Probability, P(220 < X < 260): {:.5f}".format(ans))
print("Percentage,  P(220 < X < 260): {:.2f}".format(ans*100))
```

Output:

```
Probability, P(220 < X < 260): 0.05411
Percentage,  P(220 < X < 260): 5.41
```

Program:

```
#!/usr/bin/env python3
from scipy.stats import norm

# compute cumulative distribution function (CDF)
# P(220 < X < 260)
x=220; y=260; loc=300; scale=25
pls = norm.cdf(y,loc,scale) - norm.cdf(x,loc,scale)
print("Probability, P(220 < X < 260): {:.5f}".format(pls))
print("Percentage,  P(220 < X < 260): {:.2f}".format(pls*100))
```

Output:

```
Probability, P(220 < X < 260): 0.05411
Percentage,  P(220 < X < 260): 5.41
```

3. The customer accounts of a certain departmental store have an average balance of Rs.120 and standard deviation of Rs.40 Assuming that the account balances are normally distributed, find

   a) What proportion of accounts is over Rs.150?

   given, $\mu = 120$ and $\sigma = 40$

$$\begin{aligned} P(X > 150) &= \frac{1}{\sigma\sqrt{2\pi}} \int_{150}^{\infty} e^{-(x-\mu)^2/(2\sigma^2)} dx \\ &= \frac{1}{40\sqrt{2\pi}} \int_{150}^{\infty} e^{-(x-120)^2/(2\times 40^2)} dx \\ &= 0.22663 \end{aligned}$$

Program:

```
#!/usr/bin/env python3
import numpy as np
from scipy.integrate import quad

def integrand(x):
    return 1/(sigma*np.sqrt(2*np.pi)) * np.exp(-(x-mu)**2/(2*sigma**2))
```

```
mu = 120; sigma = 40
ans, err = quad(integrand, 150, np.inf)
print("Probability, P(X > 150): {:.5f}".format(ans))
```

Output:

```
Probability, P(X > 150): 0.22663
```

Program:

```
#!/usr/bin/env python3
from scipy.stats import norm

# compute survival function (SF)
# P(X > 150)
pls = norm.sf(x=150, loc=120, scale=40)
print("Probability, P(X > 150): {:.5f}".format(pls))
```

Output:

```
Probability, P(X > 150): 0.22663
```

b) What proportion of accounts is between Rs.100 and Rs.150?

$$\begin{aligned}
P(100 < X < 150) &= \frac{1}{\sigma\sqrt{2\pi}} \int_{100}^{150} e^{-(x-\mu)^2/(2\sigma^2)} dx \\
&= \frac{1}{40\sqrt{2\pi}} \int_{100}^{150} e^{-(x-120)^2/(2\times 40^2)} dx \\
&= 0.46484
\end{aligned}$$

Program:

```
#!/usr/bin/env python3
import numpy as np
from scipy.integrate import quad

def integrand(x):
    return 1/(sigma*np.sqrt(2*np.pi)) * np.exp(-(x-mu)**2/(2*sigma**2))

mu = 120; sigma = 40
ans, err = quad(integrand, 100, 150)
print("Probability, P(100 < X < 150): {:.5f}".format(ans))
```

Output:

```
Probability, P(100 < X < 150): 0.46484
```

Program:

```
#!/usr/bin/env python3
from scipy.stats import norm

# compute cumulative distribution function (CDF)
# P(100 < X < 150)
x=100; y=150; loc=120; scale=40
pls = norm.cdf(y,loc,scale) - norm.cdf(x,loc,scale)
print("Probability, P(100 < X < 150): {:.5f}".format(pls))
```

Output:

```
Probability, P(100 < X < 150): 0.46484
```

c) What proportion of accounts is between Rs.60 and Rs.90?

$$
\begin{aligned}
P(60 < X < 90) &= \frac{1}{\sigma\sqrt{2\pi}} \int_{60}^{90} e^{-(x-\mu)^2/(2\sigma^2)} dx \\
&= \frac{1}{40\sqrt{2\pi}} \int_{60}^{90} e^{-(x-120)^2/(2\times 40^2)} dx \\
&= 0.15982
\end{aligned}
$$

Program:

```
#!/usr/bin/env python3
import numpy as np
from scipy.integrate import quad

def integrand(x):
    return 1/(sigma*np.sqrt(2*np.pi)) * np.exp(-(x-mu)**2/(2*sigma**2))

mu = 120; sigma = 40
ans, err = quad(integrand, 60, 90)
print("Probability, P(60 < X < 90): {:.5f}".format(ans))
```

Output:

```
Probability, P(60 < X < 90): 0.15982
```

Program:

```
#!/usr/bin/env python3
from scipy.stats import norm

# compute cumulative distribution function (CDF)
# P(60 < X < 90)
x=60; y=90; loc=120; scale=40
pls = norm.cdf(y,loc,scale) - norm.cdf(x,loc,scale)
print("Probability, P(60 < X < 90): {:.5f}".format(pls))
```

Output:

```
Probability, P(60 < X < 90): 0.15982
```