Homework Assignment Number Five Solutions

Problem 1. (6 points)

Given a standard normal distribution, find the area under the curve which lies
(a) to the left of z = 1.43
(b) to the right of z = -0.89
(c) between z = -2.16 and z = -0.65

Solution:

In Python:
script:

```
from scipy.stats import norm

# standard normal parameters
mean = 0.0
stdev = 1.0

#(a) to the left of z = 1.43
x = 1.43
cdfp = norm.cdf(x, loc=mean, scale=stdev)
print(f"(a) cdf at {x} is {cdfp}")


#(b) to the right of z = -0.89
x = -0.89
cdfp = norm.cdf(x, loc=mean, scale=stdev)
partb = 1.0 - cdfp
print(f"(b) prob from {x} is {partb}")

#(c) between z = -2.16 and z = -0.65
xlower = -2.16
xupper = -0.65
cdflower = norm.cdf(xlower, loc=mean, scale=stdev)
cdfupper = norm.cdf(xupper, loc=mean, scale=stdev)
prob = cdfupper - cdflower
print(f"(c) prob from {xlower} to {xupper} is {prob}")
```

output:

```
run hw5_p01_norm_script.py
(a) cdf at 1.43 is 0.9236414904632608
(b) prob from -0.89 is 0.8132670569628273
(c) prob from -2.16 to -0.65 is 0.24245977602193927
```

In Matlab:

(a) to the left of z = 1.43

Using the standard normal cumulative pdf in Matlab

```
>> p = normcdf(1.43)
```

```
p =    0.923641490463261
```

$$P(z < 1.43) = 0.9236$$

(b) to the right of z =-0.89

```
>> p = 1 - normcdf(-0.89)

p =    0.813267056962827
```

$$P(z > -0.89) = 1 - P(z \leq -0.89) = 1 - 0.1867 = 0.8133$$

(c) between z=-2.16 and z=-0.65.

```
>> p = normcdf(-0.65) - normcdf(-2.16)

p =    0.242459776021939
```

$$P(-2.16 < z < -0.65) = P(z < -0.65) - P(z < -2.16) = 0.2578 - 0.0154 = 0.2424$$

Problem 2. (6 points)

The daily amount of coffee, in liters, dispensed by a machine located in an airport lobby is a random variable X having a continuous uniform distribution with A = 7 and B = 10. Find the probability that on a given day the amount of coffee dispensed by this machine will be

(a) at most 8.8 liters
(b) more than 7.4 liters but less than 9.5 liters
(c) at least 8.5 liters.

Solution:

Analytical solution:

This problem requires the continuous uniform distribution with A=7 and B=10, and x<8.8 for part (a), 7.4<x<9.5 for part (b), and x>8.5 for part (c).

$$f(x; A, B) = \begin{cases} \dfrac{1}{B-A} & \text{for A} \leq x \leq B \\ 0 & \text{otherwise} \end{cases}$$

(a)

$$P(X < 8.8) = \int_7^{8.8} f(x)dx = \int_7^{8.8} \frac{1}{10-7}\,dx = \frac{8.8-7}{3} = 0.6$$

(b)

$$P(7.4 < X < 9.5) = \int_{7.4}^{9.5} f(x)dx = \int_{7.4}^{9.5} \frac{1}{10-7}\,dx = \frac{9.5-7.4}{3} = 0.7$$

(c)

$$P(X > 8.5) = \int_{8.5}^{10} f(x)dx = \int_{8.5}^{10} \frac{1}{10-7} dx = \frac{10-8.5}{3} = 0.5$$

In Python:
script:

```
from scipy.stats import uniform

# standard normal parameters
xlo = 7.0
xhi = 10.0
dx = xhi - xlo

#(a) less than 8.8
x = 8.8
cdfp = uniform.cdf(x, loc=xlo, scale=dx)
print(f"(a) cdf at {x} is {cdfp}")

#(b) between 7.4 and 9.5
xlower = 7.4
xupper = 9.5
cdflower = uniform.cdf(xlower, loc=xlo, scale=dx)
cdfupper = uniform.cdf(xupper, loc=xlo, scale=dx)
prob = cdfupper - cdflower
print(f"(c) prob from {xlower} to {xupper} is {prob}")

#(c) greater than
x = 8.5
cdfp = uniform.cdf(x, loc=xlo, scale=dx)
partc = 1.0 - cdfp
print(f"(b) prob from {x} is {partc}")
```

output:

```
run hw5_p02_uniform_script.py
(a) cdf at 8.8 is 0.6000000000000002
(c) prob from 7.4 to 9.5 is 0.7
(b) prob from 8.5 is 0.5
```

Problem 3. (4 points)

In the November 1990 issue of Chemical Engineering Progress a study discussed the percent purity of oxygen from a certain supplier. Assume that the mean was 99.61 with a standard deviation of 0.08. Assume that the distribution of percent purity was approximately normal.

(a) What percentage of product would you expect to be between 99.5 and 99.7?
(b) What purity value would you expect to exceed exactly 5% of the population?

Solution:

(a) What percentage of the purity values would you expect to be between 99.5 and 99.7?

$$P(99.5 < z < 99.7) = P(z < 99.7) - P(z < 99.5)$$

(b) What purity value would you expect to exceed exactly 5% of the population?

$$P(x \leq x_{hi}) = 0.05 = \int_{-\infty}^{x_{hi}} f(x)dx$$

In Python:
script:

```
from scipy.stats import norm

# part (a)
mean = 99.61
stdev = 0.08

xlower = 99.5
xupper = 99.7
cdflower = norm.cdf(xlower, loc=mean, scale=stdev)
cdfupper = norm.cdf(xupper, loc=mean, scale=stdev)
prob = cdfupper - cdflower
print(f"(a) prob from {xlower} to {xupper} is {prob}")

# part(b)

prob = 0.05
xhi = norm.ppf(prob, loc=mean, scale=stdev)
print(f"(b) the cdf is {prob} at {xhi} ")
```

output:

```
run hw5_p03_norm_script.py
(a) prob from 99.5 to 99.7 is 0.7851397605118633
(b) the cdf is 0.05 at 99.47841170984388
```

(a) You would expect the purity of 78.5% of product to be between 99.5 and 99.7.
(b) Only 5% of product has purity less than 99.48.

Problem 4. (2 points)

The average life of a certain type of small motor is 10 years with a standard deviation of nine months. The manufacturer replaces all free all motors that fail while under guarantee. If he is willing to replace only 3% of the motors that fail, how long a guarantee should he offer? Assume that the lifetime of a motor follows the normal distribution.

Solution:

This problem requires the standard normal distribution with μ=10 and σ=0.75.

We must find the failure lifetime that corresponds to 3% of the motors. This is an "inverse cdf" problem.

$$P(x \leq x_{hi}) = 0.03 = \int_{-\infty}^{x_{hi}} f(x)dx$$

In Python:
script:

```
from scipy.stats import norm

mean = 10.0
stdev = 0.75
prob = 0.03
xhi = norm.ppf(prob, loc=mean, scale=stdev)
print(f"the cdf is {prob} at {xhi} ")
```

output:

```
run hw5_p04_norm_script.py
the cdf is 0.03 at 8.589404793886562
```

If the manufacturer wants to replace no more than 3% of the batteries, the guarantee should cover batteries that fail before 8.59 years.

Problem 5. (4 points)

You flip a coin 100 times.

(a) What is the probability that you obtain at least 40 heads? Use the binomial distribution.

(b) What is the probability that you obtain at least 40 heads? Use the normal approximation to the binomial distribution.

Solution:

(a) What is the probability that you obtain at least 40 heads? Use the binomial distribution.

$$P(x \geq 40) = 1 - P(x \leq 39)$$

(b) What is the probability that you obtain at least 40 heads? Use the normal approximation to the binomial distribution.

$$\mu_x = np = 100 \cdot 0.5 = 50$$

$$\sigma_x = \sqrt{npq} = \sqrt{100 \cdot 0.5 \cdot 0.5} = 5$$

In the binomial distribution, the lower limit is 40 and the upper limit is 100. In the normal approximation, this translates into a lower limit of 39.5 (since we have to take the midpoint between 40 and the next lower value, which is 39) and infinity (since 100 is as large as you can have).

In Python:
script:

```
import numpy as np
from scipy.stats import norm
from scipy.stats import binom

# task (a):  evaluate the binomial probability

x = 39 # number of successes
n = 100 # number of trials
p = 0.5 # probability of success
cdfp = binom.cdf(x, n, p)
parta = 1.0 - cdfp
print(f"binomial pdf at {x} is {parta}")

# task (b):  evaluate with the normal probability

mean = n*p
var = n*p*(1.0-p)
stdev = np.sqrt(var)
x = 39.5
cdfp = norm.cdf(x, loc=mean, scale=stdev)
partb = 1.0 - cdfp

print(f"cdf at {x} is {partb}")
```

output:

```
run hw5_p05_binom_script.py
binomial pdf at 39 is 0.9823998998911476
cdf at 39.5 is 0.9821355794371834
```

(a) The probability using the binomial distribution is 0.9824.
(b) The probability using the normal approximation to the binomial distribution is 0.9821.

Problem 6. (6 points)

The life of a certain type of device has an advertised lifetime of 100 hours. The failure rate is constant and the exponential distribution applies.

(a) What is the mean time to failure?
(b) What is the probability that 200 hours will pass before a failure is observed?
(c) What is the probability that exactly 3 of 5 devices are functioning after 200 hours?

Solution:

(a) What is the mean time to failure?

The mean of the exponential distribution is the beta parameter.

$\mu = \beta = 100$ hours

(b) What is the probability that 200 hours will pass before a failure is observed?

$$P(x > 200) = 1 - P(x < 200)$$

(c) What is the probability that exactly 3 of 5 devices are functioning after 200 hours?

Use the binomial distribution with the probability of success given by the solution to part (b).

$$b(x = 3; n = 5, p = 0.4554)$$

In Python:
script:

```
from scipy.stats import expon
from scipy.stats import binom

# exponential PDF parameters
beta = 100.0

# task (b):  evaluate the CDF at a given value of random variable
x = 200.0
cdfp = expon.cdf(x, scale=beta)
partb = 1.0 - cdfp
print(f"cdf at {x} is {partb}")

# task (c):  evaluate the binomial probability

x = 3 # maximum number of successes
n = 5 # number of trials
p = partb # probability of success
f = binom.pmf(x, n, p)
print(f"binomial pdf at {x} is {f}")
```

output:

```
run hw5_p06_expon_script.py
cdf at 200.0 is 0.1353352832366127
binomial pdf at 3 is 0.01853226850623819
```

In Matlab:

(b) What is the probability that 200 hours will pass before a failure is observed?

```
>> p = 1 - cdf('exp',200,100)

p =   0.135335283236613
```

(c) What is the probability that exactly 3 of 5 devices are functioning after 200 hours?

```
>> p = pdf('binomial',3,5,0.1353)

p =   0.018519288944000
```

Problem 7. (4 points)

Using the gamma distribution with $\alpha = 4$ and $\beta = 1/3$, find

(a) $P(x < 1/2)$
(b) C where $P(x < C) = 0.05$

Solution:

In Python:
script:

```
from scipy.stats import gamma

# gamma PDF parameters
a = 4.0
b = 1.0/3.0

# task 3:  evaluate the CDF at a given value of random variable
x = 0.5
cdfp = gamma.cdf(x, a=a, scale=b)
print(f"cdf at {x} is {cdfp}")

# task 4:  evaluate the inverse CDF at a given probability
cdfp = 0.05
icdfx = gamma.ppf(cdfp, a=a, scale=b)
print(f"for probability of {cdfp}, x is {icdfx}")
```

output:

```
run hw5_p07_gamma_script.py
cdf at 0.5 is 0.06564245437845008
for probability of 0.05, x is 0.45543946558327697
```

In Matlab:
(a) $P(x < 1/2)$

```
>> p = cdf('gamma',1/2,4,1/3)

p =   0.065642454378450
```

(b) C where $P(x < C) = 0.05$

```
>> x = icdf('gamma',0.05,4,1/3)

x =   0.455439465583277
```

Problem 8. (4 points)
Using the Chi-Squared distribution with v=10, find
(a) $P(\chi^2 < 4)$
(b) C where $P(\chi^2 < C) = 0.05$

Solution:

In Python:
script:

```
from scipy.stats import chi2

# chi squared PDF parameter
dof = 10.0

# task (a):  evaluate the CDF at a given value of random variable
x = 4.0
cdfp = chi2.cdf(x, df=dof)
print(f"cdf at {x} is {cdfp}")

# task (b):  evaluate the inverse CDF at a given probability
cdfp = 0.05
icdfx = chi2.ppf(cdfp, df=dof)
print(f"for probability of {cdfp}, x is {icdfx}")
```

output:

```
run hw5_p08_chi2_script.py
cdf at 4.0 is 0.052653017343711125
for probability of 0.05, x is 3.9402991361190605
```

In Matlab:

(a) $P(\chi^2 < 4)$

```
>> p = chi2cdf(4,10)

p =   0.052653017343711
```

(b) C where $P(\chi^2 < C) = 0.05$

```
>> X = chi2inv(0.05,10)

X =   3.940299136119060
```

Problem 9. (4 points)

Using the t-distribution with v=10, find
(a) $P(t < -1)$
(b) C where $P(t < C) = 0.05$

Solution:

In Python:
script:

```
from scipy.stats import t
```

```
# t PDF parameters
dof = 10.0

# task (a):  evaluate the CDF at a given value of random variable
x = -1.0
cdfp = t.cdf(x, df=dof)
print(f"cdf at {x} is {cdfp}")

# task (b):  evaluate the inverse CDF at a given probability
cdfp = 0.05
icdfx = t.ppf(cdfp, df=dof)
print(f"for probability of {cdfp}, x is {icdfx}")
```

output:

```
run hw5_p09_t_script.py
cdf at -1.0 is 0.17044656615103004
for probability of 0.05, x is -1.8124611228107341
```

In Matlab:

(a) $P(t < -1)$

```
>> p = tcdf(-1,10)

p =   0.170446566151030
```

(b) C where $P(t < C) = 0.05$

```
>> X = tinv(0.05,10)

X =  -1.812461122811675
```

Problem 10. (4 points)

Using the F-distribution with v_1=8 and v_2 = 12, find

(a) $P(F < 1)$

(b) C where $P(F < C) = 0.05$

Solution:

In Python:

script:

```
from scipy.stats import f

# F PDF parameters
dof1 = 8.0
dof2 = 12.0

# task (a):  evaluate the CDF at a given value of random variable
x = 1.0
cdfp = f.cdf(x, dfn=dof1, dfd=dof2)
print(f"cdf at {x} is {cdfp}")
```

```
# task (b):  evaluate the inverse CDF at a given probability
cdfp = 0.05
icdfx = f.ppf(cdfp, dfn=dof1, dfd=dof2)
print(f"for probability of {cdfp}, x is {icdfx}")
```

output:

```
run hw5_p10_f_script.py
cdf at 1.0 is 0.5173903360000001
for probability of 0.05, x is 0.3045123549055689
```

In Matlab:

(a) $P(F < 1)$

```
>> p = fcdf(1,8,12)

p =   0.517390336000000
```

(b) C where $P(F < C) = 0.05$

```
>> X = finv(0.05,8,12)

X =   0.304512354905569
```