# Basic Statistics (Module – 4 (Part – 1))

Q1) Calculate probability from the given dataset for the below cases

Data\_set: Cars.csv

Calculate the probability of MPG of Cars for the below cases.

MPG <- Cars$MPG

1. P(MPG>38)

Ans:-

Number of events MPG\_[<=38] = 48

Number of events MPG\_(>38] = 33

P(MPG>38) = 33/(33+48) = 0.407

1. P(MPG<40)

Ans:-

Number of events MPG\_[<40] = 61

Number of events MPG\_(>=40] = 20

P(MPG<40) = 61/(61+20) =0.753

1. P (20<MPG<50)

Ans:-

Number of events MPG\_(20<MPG<50] = 69

Number of events MPG\_[<=20] = 7

Number of events MPG\_[>=50] = 5

P (20<MPG<50)= 69/(69+7+5) = 0.852

Python Code:-

############## proability part 1 ###############

import pandas as pd

import numpy as np

from sklearn.preprocessing import LabelEncoder

import seaborn as sn

import matplotlib.pyplot as plt

data = pd.read\_csv("C://Users//user//Downloads//Prob,CLT//cars.csv")

data.isnull().sum() # no null values

data.dropna()

data.columns

# creating copy

data1= data.copy(deep=True)

data2= data.copy(deep=True)

data3= data.copy(deep=True)

#converting into categorical & binary

lb = LabelEncoder()

## discretize ouput column ;;MPG > 38

data1['MPG'].describe() # max value = 53.700681

bins=[0,38,54] # [<=38],(>38]

group\_names= ['MPG\_[<=38]','MPG\_(>38]'] # (a,b] => a not included; but b included

data1['MPG']= pd.cut(data1['MPG'],bins, labels = group\_names)

#summary

data1.dropna()

data1.head(10)

data1.describe()

# counts

data1['MPG'].value\_counts()

len(data1['MPG'])

## discretize ouput column ;;MPG < 40

data2['MPG'].describe() # max value = 53.700681

bins=[0,39.9999,54] # [<40],(>=40]

group\_names= ['MPG\_[<40]','MPG\_(>=40]'] # (a,b] => a not included; but b included

data2['MPG']= pd.cut(data2['MPG'],bins, labels = group\_names)

#summary

data2.dropna()

data2.head(10)

data2.describe()

# counts

data2['MPG'].value\_counts()

len(data2['MPG'])

## discretize ouput column ;;P (20<MPG<50)

data3['MPG'].describe() # max value = 53.700681

bins=[0,20,49.9999,54] # [<40],(>=40]

group\_names= ['MPG\_[<=20]','MPG\_(20<MPG<50]','MPG\_[>=50]'] # (a,b] => a not included; but b included

data3['MPG']= pd.cut(data3['MPG'],bins, labels = group\_names)

#summary

data3.dropna()

data3.head(10)

data3.describe()

# counts

data3['MPG'].value\_counts()

len(data3['MPG'])

Q2) Check whether the data follows normal distribution

1. Check whether the MPG of Cars follows Normal Distribution Dataset: Cars.csv

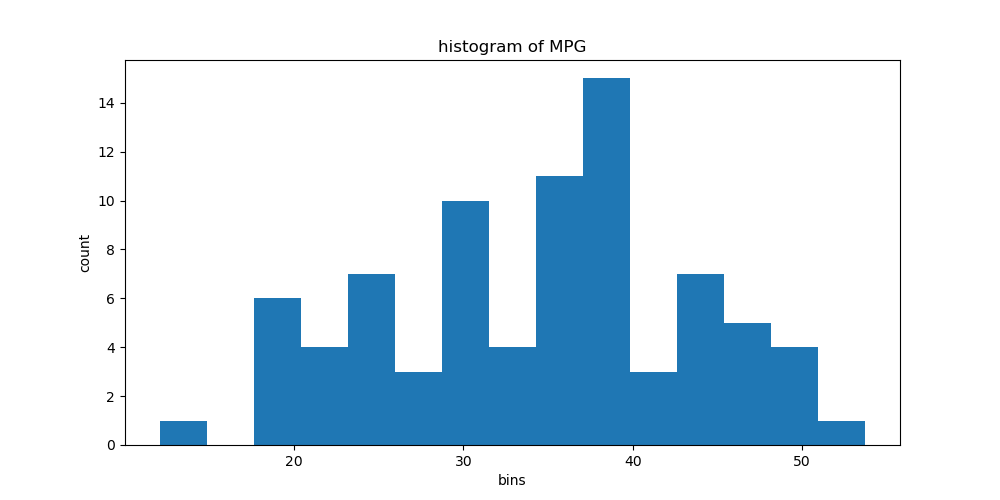
Ans:-

min. MPG = 12.10126289

max. MPG = 53.70068138

mean MPG = 34.422075728024666

std.dev MPG = 9.131444731795982



**Python Code:-**

###################### normal distribution check ##################

mean= data['MPG'].mean()

std= data['MPG'].std()

med = data['MPG'].median() # 35.15

print('min. MPG = ',data['MPG'].min())

print('max. MPG = ',data['MPG'].max())

print('mean MPG = ',mean)

print('std.dev MPG = ',std)

**min. MPG = 12.10126289**

**max. MPG = 53.70068138**

**mean MPG = 34.422075728024666**

**std.dev MPG = 9.131444731795982**

# histogram

fig, axs = plt.subplots(figsize=(10,5))

axs.hist(data['MPG'],bins=15)

axs.set\_title('histogram of MPG')

axs.set\_xlabel("bins")

axs.set\_ylabel('count')

df = data.MPG

x=[]

# verifying normal distribution within one std

within\_one\_std\_deviation = data[(mean-std) > data['MPG']]

within\_one\_std\_deviation = data[data['MPG'] < (mean+std)]

new\_data = [item for item in data.MPG if item > (mean-std)]

new\_data = [item for item in new\_data if item < (mean+std)]

df = pd.DataFrame(new\_data)

df.describe()

# percentage of data points falling within one std.dev

percn\_one\_std = len(df)/len(data.MPG) \* 100

percn\_one\_std

**Out[14]: 61.72839506172839**

# verifying normal distribution within 2 std

new\_data = [item for item in data.MPG if item > (mean-(2\*std))]

new\_data = [item for item in new\_data if item < (mean+(2\*std))]

df2 = pd.DataFrame(new\_data)

df2.describe()

# percentage of data points falling within two std.dev

percn\_two\_std = len(df2)/len(data.MPG) \* 100

percn\_two\_std

**Out[16]: 97.53086419753086**

# verifying normal distribution within 3 std

new\_data = [item for item in data.MPG if item > (mean-(3\*std))]

new\_data = [item for item in new\_data if item < (mean+(3\*std))]

df3 = pd.DataFrame(new\_data)

df3.describe()

# percentage of data points falling within 3 std.dev

percn\_three\_std = len(df3)/len(data.MPG) \* 100

percn\_three\_std

Out[18]: 100.0

**Summary:-**

* + Mean(34.42) & median (35.15) are not equal but close in value
  + Percentage of values within one standard deviation away from mean is 61.72% but for normal distribution it would be 68.26%
  + Percentage of values within two standard deviation away from mean is 95.53% but for normal distribution it would be 95.46%. its almost equal
  + Percentage of values within three standard deviation away from mean is 100% but for normal distribution it would be 99.73%
  + **So its not satisfying not satisfying all of the normal distribution criterias we cant treat it as a normal distribution**

1. Check Whether the Adipose Tissue (AT) and Waist Circumference (Waist) from wc-at data set follows Normal Distribution

Dataset: wc-at.csv

**Ans:-**

##b)check Whether the Adipose Tissue (AT) and Waist Circumference (Waist) from wc-at data set follows Normal Distribution

d = pd.read\_csv("C://Users//user//Downloads//Prob,CLT//wc-at.csv")

d.isnull().sum() # no null values

d.dropna()

d.columns

#### for waist #######

## normal distribution check

mean= d['Waist'].mean()

std= d['Waist'].std()

med = d['Waist'].median() #90.8

print('min. Waist = ',d['Waist'].min())

print('max. Waist = ',d['Waist'].max())

print('mean Waist = ',mean)

print('std.dev Waist = ',std)

**min. Waist = 63.5**

**max. Waist = 121.0**

**mean Waist = 91.90183486238533**

**std.dev Waist = 13.559115982678826**

# histogram

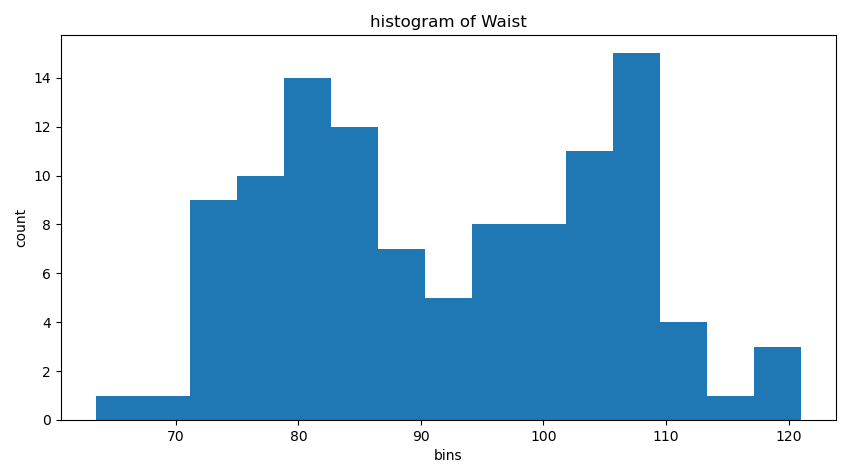
fig, axs = plt.subplots(figsize=(10,5))

axs.hist(d['Waist'],bins=15)

axs.set\_title('histogram of Waist')

axs.set\_xlabel("bins")

axs.set\_ylabel('count')



df = d.Waist

x=[]

# verifying normal distribution within one std

new\_data = [item for item in d.Waist if item > (mean-std)]

new\_data = [item for item in new\_data if item < (mean+std)]

df = pd.DataFrame(new\_data)

df.describe()

# percentage of data points falling within one std.dev

percn\_one\_std = len(df)/len(d.Waist) \* 100

percn\_one\_std

**Out[23]: 59.63302752293578**

# verifying normal distribution within 2 std

new\_data = [item for item in d.Waist if item > (mean-(2\*std))]

new\_data = [item for item in new\_data if item < (mean+(2\*std))]

df2 = pd.DataFrame(new\_data)

df2.describe()

# percentage of data points falling within two std.dev

percn\_two\_std = len(df2)/len(d.Waist) \* 100

percn\_two\_std

**Out[24]: 96.3302752293578**

# verifying normal distribution within 3 std

new\_data = [item for item in d.Waist if item > (mean-(3\*std))]

new\_data = [item for item in new\_data if item < (mean+(3\*std))]

df3 = pd.DataFrame(new\_data)

df3.describe()

# percentage of data points falling within two std.dev

percn\_three\_std = len(df3)/len(d.Waist) \* 100

percn\_three\_std

**Out[25]: 100.0**

**Summary:-**

* + Mean(90.8) & median (91.9) are not equal but close in value
  + Percentage of values within one standard deviation away from mean is 59.63% but for normal distribution it would be 68.26%
  + Percentage of values within two standard deviation away from mean is 93.33% but for normal distribution it would be 95.46%
  + Percentage of values within three standard deviation away from mean is 100% but for normal distribution it would be 99.73%
  + **So its not satisfying not satisfying all of the normal distribution criterias we cant treat it as a normal distribution**

### for AT ##########

## normal distribution check

mean= d['AT'].mean()

std= d['AT'].std()

med = d['AT'].median() # 96.54

print('min. AT = ',d['AT'].min())

print('max. AT = ',d['AT'].max())

print('mean AT = ',mean)

print('std.dev AT = ',std)

**min. AT = 11.44**

**max. AT = 253.0**

**mean AT = 101.89403669724771**

**std.dev AT = 57.29476272231215**

# histogram

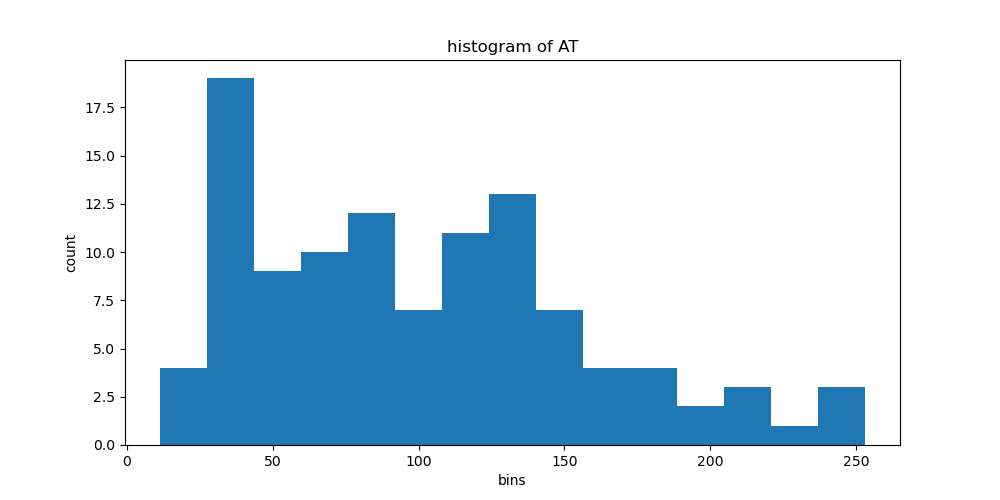
fig, axs = plt.subplots(figsize=(10,5))

axs.hist(d['AT'],bins=15)

axs.set\_title('histogram of AT')

axs.set\_xlabel("bins")

axs.set\_ylabel('count')



df = d.Waist

x=[]

# verifying normal distribution within one std

new\_data = [item for item in d.AT if item > (mean-std)]

new\_data = [item for item in new\_data if item < (mean+std)]

df = pd.DataFrame(new\_data)

df.describe()

# percentage of data points falling within one std.dev

percn\_one\_std = len(df)/len(d.AT) \* 100

percn\_one\_std

**Out[32]: 63.30275229357798**

# verifying normal distribution within 2 std

new\_data = [item for item in d.AT if item > (mean-(2\*std))]

new\_data = [item for item in new\_data if item < (mean+(2\*std))]

df2 = pd.DataFrame(new\_data)

df2.describe()

# percentage of data points falling within two std.dev

percn\_two\_std = len(df2)/len(d.AT) \* 100

percn\_two\_std

**Out[33]: 95.41284403669725**

# verifying normal distribution within 3 std

new\_data = [item for item in d.AT if item > (mean-(3\*std))]

new\_data = [item for item in new\_data if item < (mean+(3\*std))]

df3 = pd.DataFrame(new\_data)

df3.describe()

# percentage of data points falling within two std.dev

percn\_three\_std = len(df3)/len(d.AT) \* 100

percn\_three\_std

**Out[35]: 100.0**

**Summary:-**

* + Mean(101.89) & median (96.54) are not equal
  + Percentage of values within one standard deviation away from mean is 63.30% but for normal distribution it would be 68.26%
  + Percentage of values within two standard deviation away from mean is 95.41% but for normal distribution it would be 95.46%
  + Percentage of values within three standard deviation away from mean is 100% but for normal distribution it would be 99.73%
  + **So its not satisfying not satisfying all of the normal distribution criterias we cant treat it as a normal distribution**

Q3) Calculate the Z scores of 90% confidence interval,94% confidence interval, 60% confidence interval

Ans:-

90%

Compute alpha (α): α = 1 - (confidence level / 100)

= 1 - (90 / 100) = .1

Find the critical probability (p\*): p\* = 1 - α/2 = 1 - 0.1/2 = 0.95

Corresponding z score from the table = +1.645 & -1.645

94%

Compute alpha (α): α = 1 - (confidence level / 100)

= 1 - (94 / 100) = .06

Find the critical probability (p\*): p\* = 1 - α/2 = 1 - 0.06/2 = 0.97

Corresponding z score from the table = +1.89 & -1.89

60%

Compute alpha (α): α = 1 - (confidence level / 100)

= 1 - (60 / 100) = 0.4

Find the critical probability (p\*): p\* = 1 - α/2 = 1 - 0.4/2 = 0.8

Corresponding z score from the table = +0.84 & -0.84

Q4) Calculate the t scores of 95% confidence interval, 96% confidence interval, 99% confidence interval for sample size of 25

Ans:-

95%

Compute alpha (α): α = 1 - (confidence level / 100)

= 1 - (95 / 100) = 0.05

Find the critical probability (p\*): p\* = 1 - α/2 = 1 - 0.05/2 = 0.975

Sample size = 25

Degree of freedom = n-1 = 25 -1 = 24

Corresponding t score from the table = 2.064

96%

Compute alpha (α): α = 1 - (confidence level / 100)

= 1 - (96 / 100) = 0.04

Find the critical probability (p\*): p\* = 1 - α/2 = 1 - 0.04/2 = 0.98

Sample size = 25

Degree of freedom = n-1 = 25 -1 = 24

Corresponding t score from the table = 2.172

99%

Compute alpha (α): α = 1 - (confidence level / 100)

= 1 - (99 / 100) = 0.01

Find the critical probability (p\*): p\* = 1 - α/2 = 1 - 0.01/2 = 0.995

Sample size = 25

Degree of freedom = n-1 = 25 -1 = 24

Corresponding t score from the table = 2.797

Q5**)** A Government company claims that an average light bulb lasts 270 days. A researcher randomly selects 18 bulbs for testing. The sampled bulbs last an average of 260 days, with a standard deviation of 90 days. If the CEO's claim were true, what is the probability that 18 randomly selected bulbs would have an average life of no more than 260 days

Hint:

rcode → pt(tscore,df)

df → degrees of freedom

Ans:-

t - statistics for the data is given as follows:

https://tex.z-dn.net/?f=t%3D%5Cdfrac%7Bx-%5Cmu%7D%7B%5Cfrac%7Bs%7D%7B%5Csqrt%20n%7D%7D

x = mean of the sample of bulbs =  260

μ = population mean = 270

s = standard deviation of the sample = 90

n = number of items in the sample = 18

https://tex.z-dn.net/?f=t%3D%5Cdfrac%7B260-270%7D%7B%5Cfrac%7B90%7D%7B%5Csqrt%2018%7D%7D

https://tex.z-dn.net/?f=t%20%3D%20%5Cdfrac%7B-10%7D%7B%5Cfrac%7B90%7D%7B3%5Csqrt%202%7D%7D

https://tex.z-dn.net/?f=t%20%3D%20%5Cdfrac%7B-10%7D%7B%5Cfrac%7B30%7D%7B%5Csqrt%202%7D%7D

https://tex.z-dn.net/?f=t%20%3D%20%5Cdfrac%7B-1%20%5Ctimes%20%5Csqrt%202%7D%7B3%7D

t = - 0.471

For probability calculations, the number of degrees of freedom is n - 1, so here you need the t-distribution with 17 degrees of freedom.

The probability that **t < - 0.471 with 17 degrees of freedom** assuming the population mean is true, the t-value is less than the t-value obtained With 17 degrees of freedom and a t score of - 0.471, the probability of the bulbs lasting less than 260 days on average of **0.3218** assuming the mean life of the bulbs is 300 days.

Q6) The time required for servicing transmissions is normally distributed with  = 45 minutes and  = 8 minutes. The service manager plans to have work begin on the transmission of a customer’s car 10 minutes after the car is dropped off and the

customer is told that the car will be ready within 1 hour from drop-off. What is the probability that the service manager cannot meet his commitment?

A. 0.3875

B. 0.2676

C. 0.5

D. 0.6987

Ans:-

The work begin after 10 min, so the average time increase from 45min to 55min.

for normal distribution :-

z = (X-μ)/б

= (60-55)/8

= 0.625

In R software for probability finding we use function called pnorm

As we want to find the probability of service manager cannot meet his commitment, So we should write below command.

1-pnorm(0.625)

=0.2659

**OR**

In R software we can directly write code

1-pnorm(60, mean=55, sd=8)

=0.2659

Q7) The current age (in years) of 400 clerical employees at an insurance claims processing center is normally distributed with mean  = 38 and Standard deviation

 =6. For each statement below, please specify True/False. If false, briefly explain why.

1. More employees at the processing center are older than 44 than between 38 and 44.

**Ans:-**

Mean = 38

SD = 6

Z score = (Value - Mean)/SD

Z score for 44  = (44 - 38)/6  = 1  =>  84.13 %

=> People above 44 age = 100 - 84.13 =  15.87%  ≈  63    out of 400

Z score for 38  = (38 - 38)/6 = 0 => 50%

Hence People between 38 & 44  age = 84.13 - 50 = 34.13 % ≈  137 out of 400

Hence More employees at the processing center are older than 44 than between 38 and 44. is F**ALSE**

1. A training program for employees under the age of 30 at the center would be expected to attract about 36 employees.

**Ans:-**

Z score for 30  = (30 - 38)/6 =  -1.33  =  9.15  %   ≈ 36 out of 400

Hence A training program for employees under the age of 30 at the center would be expected to attract about 36 employees - **TRUE**

Q8) If X1 ~ N(μ, σ2) and X2 ~ N(μ, σ2) are iid normal random variables, then what is the

difference between 2 X1 and X1 + X2? Discuss both their distributions and parameters.

**Ans:-**

If X and Y are independent, and X1 ∼ Normal(µ1, σ1 2 ), X2 ∼ Normal(µ2, σ2 2 )

then X1 + X2 ∼ Normal(µ1+ µ2 , σ1 2 +σ2 2 )

here µ1 = µ2, = µ && σ1 2 = σ2 2  = σ2

therefore 2 X1 = X1 + X2 ∼ Normal(µ+ µ, σ2 + σ2 )

∼ Normal(2µ + 2σ2 )

Q9) Let X ~ N(100, 20^2) its (100, 20 square).Find two values, a and b, symmetric about the mean, such that the probability of the random variable taking a value between them is 0.99.

A.

90.5, 105.9

B. 80.2, 119.8 C.

22, 78

D. 48.5, 151.5

E. 90.1, 109.9

Ans:-

Compute alpha (α): α = 1 - (confidence level / 100)

= 1 - (99 / 100) = .01

Find the critical probability (p\*): p\* = 1 - α/2 = 1 - 0.01/2 = 0.995

Corresponding z score from the table = +2.58 & -2.58

Z = (X – μ) / σ 🡺 X = (z \* σ )+ μ

Suppose a & b are extreme points of the interval

Then,

a = (-z \* σ )+ μ & b = (+z \* σ )+ μ

= (-2.58\*20) + 100 = (2.58\*20) + 100

= 48.4 = 151.6

Q10) Consider a company that has two different divisions. The annual profits from the two divisions are independent and have distributions Profit1 ~ N(5, 3^2) and Profit2 ~ N(7, 4^2) respectively. Both the profits are in $ Million. Answer the following questions about the total profit of the company in Rupees. Assume that $1 = Rs. 45

1. Specify a Rupee range (centered on the mean) such that it contains 95% probability for the annual profit of the company.
2. Specify the 5th percentile of profit (in Rupees) for the company
3. Which of the two divisions has a larger probability of making a loss in a given year?

Ans:-

A) X1 + X2 ∼ Normal(µ1+ µ2 , σ1 2 +σ2 2 ) 🡺 ∼ Normal(5+ 7, 3^2+4^2)

∼ Normal(12, 5^2)

Compute alpha (α): α = 1 - (confidence level / 100)

= 1 - (95 / 100) = .05

Find the critical probability (p\*): p\* = 1 - α/2 = 1 - 0.05/2 = 0.975

Corresponding z score from the table = +1.96 & -1.96

Z = (X – μ) / σ 🡺 X = (z \* σ )+ μ

Suppose a & b are extreme points of the interval

Then,

a = (-z \* σ )+ μ & b = (+z \* σ )+ μ

= (-1.96 \*5) + 12 = (1.96\*5) + 12

= 2.2 = 21.8

Range 🡺 [2.2, 21.8]

**B)**

5th Percentile 🡺 below 5%

Find the critical probability (p\*): p\* = α = 0.05

Corresponding z score from the table = -1.64

X = (z \* σ )+ μ = (-1.64\*5) + 12 = 3.8

**C)**

There have only smaller difference in the mean of profit. So the loss depends on the std. deviation(σ). The second division have high std.deviation compare to the first one. So it have more chances of go for loss

**Hints:**

1. Business Problem
   1. Objective
   2. Constraints (if any)
2. For each assignment the solution should be submitted in the below format
3. Research and Perform all possible steps for obtaining solution
4. For Basic Statistics explanation of the solutions should be documented in black and white along with the codes.

One must follow these guidelines as well:

* 1. Be thorough with the concepts of Probability, Central Limit Theorem and Perform the calculation stepwise
  2. For True/False Questions, explanation is must.
  3. R & Python code for Univariate Analysis (histogram, box plot, bar plots etc.) for data distribution to be attached

1. All the codes (executable programs) should execute without errors