**Python**

Answer. 1 Python

from collections import Counter

data\_set = 'write write write all the number from from from 1 to 100'

split\_it = data\_set.split()

Counters\_found = Counter(split\_it)

most\_occur = Counters\_found.most\_common(1)

print(most\_occur)

len(most\_occur[0][0])

Explanation - From the given string we can note that the most frequent words are “write” and “from” and the maximum value of both the values is “write” and its corresponding length is 5

from collections import Counter

data\_set = 'Python is good Python dynamic'

split\_it = data\_set.split().

Counters\_found = Counter(split\_it)

most\_occur = Counters\_found.most\_common(1)

print(most\_occur)

len(most\_occur[0][0])

Explanation - From the given string we can note that the most frequent words is “Python” and

corresponding length is 6

Answer. 2 Python ->

def isValid(s):

d = Counter(s)

counts = Counter(d.values())

if len(counts) == 1:

return "YES"

elif len(counts) > 2:

return "NO"

else:

max\_v = max(counts.values())

k1, k2 = counts.keys()

if (max\_v == len(d) - 1):

if (abs(k1 - k2) == 1):

return "YES"

elif (min(k1, k2) == 1):

if counts[1] == 1:

return "YES"

else:

return "NO"

else:

return "NO"

else:

return "NO"

isValid('nnrrpp')

s = (“nnrrpp”). This is a valid string because frequencies are { “n”: 2, “r”: 2, “p”: 2 }-YES

s=('abcddc'). This string is not valid as we can remove only 1 occurrence of “c” & ‘d’. That leaves character frequencies of { “a”: 1, “b”: 1 , “c”: 2,’d’:2 }-NO

Answer. 3 Python ->

import wget

URL = "https://raw.githubusercontent.com/Biuni/PokemonGO-Pokedex/master/pokedex.json"

response = wget.download(URL,'data.json')

df = pd.read\_json("data.json")

df.to\_csv(‘data\_csv’,,index=False)

Answer. 4 Python ->

import wget

URL = "https://data.nasa.gov/resource/y77d-th95.json"

response = wget.download(URL,'data.json')

df = pd.read\_json("data.json")

df.to\_csv('data\_csv',index=False)

Answer. 5 Python ->

import requests

import json

response\_API = requests.get('http://api.tvmaze.com/singlesearch/shows?q=westworld&embed=episodes')

data = response\_API.text

parse\_json = json.loads(data)

Answer. 6 Python ->

df[df['spawn\_chance']<5]

df[df['weakness']<4]

df[df['multiplier']==NaN]

df[df['evolution']>2]

df[df['spawn time'].dt.second<300]

df[df['capabilities']>2]

Answer. 7 Python ->

df['year']=pd.to\_datetime(df.year,unit='ns',errors='coerce')

**Get all the Earth meteorites that fell before the year 2000**

**Plot:**

import seaborn as sns

fig = plt.figure(figsize =(50, 50))

sns.scatterplot(df,x=df['year'],y=df['name'])

**analysis**

df[df['year'].dt.year<2000]['name']

**● Get all the earth meteorites co-ordinates who fell before the year 1970**

**analysis**

new\_df=df[df['year'].dt.year<1970]['geolocation']

**Plot:**

import seaborn as sns

fig = plt.figure(figsize =(50, 50))

sns.scatterplot(new\_df,x=df['year'],y=df['name'])

● **Assuming that the mass of the earth meteorites was in kg, get all those whose mass was more than 10000kg**

**analysis**

df[df['mass']>10000]

**Plot:**

import seaborn as sns

fig = plt.figure(figsize =(50, 50))

sns.scatterplot(df,x=df['mass'],y=df['name'])

Answer. 8 Python ->

import pandas as pd

import requests

response = requests.get('http://api.tvmaze.com/singlesearch/shows?q=westworld&embed=episodes')

data = response.json()

df = pd.json\_normalize(data['\_embedded']['episodes'])

df=pd.DataFrame(df)

df.head()

**● Get all the overall ratings for each season and using plots compare the ratings for all the seasons, like season 1 ratings, season 2, and so on.**

**analysis**

df.groupby('season')['rating.average'].sum()

**Plot**

sns.barplot(data=df, x="season", y="rating.average")

**● Get all the episode names, whose average rating is more than 8 for every season**

**analysis**

df[df..groupby('season')& (df['rating.average']>8)]['name']

Or

df[(df['rating.average']>8)][['name','season']]

**analysis**

sns.barplot(data=df, x="name", y="rating.average")

● **Get all the episode names that aired before May 2019**

df['airdate']=pd.to\_datetime(df['airdate'])

df[(df['airdate'].dt.year)<2019]['name']

Answer. 9 Python ->

**● Get all the cars and their types that do not qualify for clean alternative fuel vehicle**

import pandas as pd

df=pd.read\_csv('C:\\Users\\LOKESHRAJ P\\Downloads\\Electric\_Vehicle\_Population\_Data.csv')

df[df['Clean Alternative Fuel Vehicle (CAFV) Eligibility']!='Clean Alternative Fuel Vehicle Eligible'][['Make','Model']]

**● Get all TESLA cars with the model year, and model type made in Bothell City.**

df[(df['Make']=='TESLA')&(df['City']=='Bothell')][['Model Year','Model']]

**● Get all the cars that have an electric range of more than 100, and were made after**

**2015**

df[(df['Electric Range']>100) & (df['Model Year']>2015)]

**● Draw plots to show the distribution between city and electric vehicle type**

import seaborn as sns

sns.set(rc={'figure.figsize':(50,50)})

sns.countplot(data=df, x="City", hue="Electric Vehicle Type")

sns.displot(data=df, x="City", hue="Electric Vehicle Type")

Answer. 10 Python ->

import nltk

from nltk.tokenize import word\_tokenize

from nltk.tag import pos\_tag

def count\_pos(phrase):

tokens = word\_tokenize(phrase)

tags = pos\_tag(tokens)

counts = {'verb': 0, 'noun': 0, 'pronoun': 0, 'adjective': 0}

for word, tag in tags:

if tag.startswith('VB'):

counts['verb'] += 1

elif tag.startswith('NN'):

counts['noun'] += 1

elif tag.startswith('PRP'):

counts['pronoun'] += 1

elif tag.startswith('JJ'):

counts['adjective'] += 1

return counts

Test case-1:

phrase = "The quick brown fox jumps over the lazy dog."

counts = count\_pos(phrase)

print('DIC=',counts)

DIC= {'verb': 1, 'noun': 3, 'pronoun': 0, 'adjective': 2}

Explanation: verb-jumps

Noun- fox,dog,brown

Pronoun- Not there

Adjective- quick,lazy

Test case-2:

phrase = "Python is easy for programming"

counts = count\_pos(phrase)

print('DIC=',counts)

DIC= {'verb': 2, 'noun': 1, 'pronoun': 0, 'adjective': 1}

Explanation: verb-is,programming

Noun- Pyhton

Pronoun- Not there

Adjective- easy

**Statistics**

Answer. 1 statistics

SAT scores and college GPA are has positive correlation

Answer. 2 statistics

1. 68% (explanation-160-180 is come in 1 sigma range on each side from mean)
2. 30% [z score=(x-mu)/(sigma)]

x=175

mu=170

sigma=10

Poulation standard deviation given so choose Z-score

Z=(x-mu)/(sigma)

Z=0.5

At Z=0.5 in Z-table

Probability=0.3085=30%

1. Z=1.5

x=185

mu=170

sigma=10

# Poulation standard deviation given so choose Z-score

Z=(x-mu)/(sigma)

d. 153.6 cm

x=170

Lower CI= x-Z0.5\*sigma

Lower CI = 170-1.64\*10

=153.6 cm

e. 0.058

CV=sigma/mu

CV= 10/170

= 0.058

Answer. 3 statistics -> <https://github.com/Lokeshrajkp/Placement-Assignment_Lokeshraj-K-P/blob/main/Statistics/statistics_answer_3.ipynb>

Answer. 4 statistics

Probability=0.2

Total outcome=20

Favourable outcome=4

Probability=4/20=0.2

20%

Answer. 5 statistics

Probability that it belongs to Company A is 0.66 or 66%

Company A's success rate 95%

Company B's success rate 90%

Company A taxis=80%

Company B taxi=20%

According to Bayes theorem,

P(A|B)=(P(B|A)\*P(A))/P(B)

P(A)=0.8

P(B|A)=1-0.95=0.05

P(B)=0.2

P(B|B) = 1 - 0.90 = 0.10

P(B) = P(B|A) \* P(A) + P(B|B) \* P(B)

P(B) = 0.05 \* 0.80 + 0.10 \* 0.20

P(B)=0.06

Now substitute in Bayes theroem

P(A|B)=(0.05 \* 0.80) / 0.06

P(A|B)=0.66

= 66%

Answer. 6 statistics -> <https://github.com/Lokeshrajkp/Placement-Assignment_Lokeshraj-K-P/blob/main/Statistics/statistics_answer_6.ipynb>

Answer. 7 statistics

2𝑋 + 3 − 8 = 0 & 2𝑌 + 𝑋 − 5 = 0

2𝑋 + 3 − 8 = 0

2𝑋 − 5 = 0

2𝑋 = 5

𝑋 = 5/2

2𝑌 + 𝑋 − 5 = 0

2𝑌 + (5/2) − 5 = 0

2𝑌 − 5/2 = 0

2𝑌 = 5/2

𝑌 = 5/4

a. Variance of Y= 4

Var(𝑌) = Var(𝑋) × (1 - R²)

Var(𝑋) = 4

2𝑌 + 𝑋 − 5 = 0

2𝑌 + (5/2) − 5 = 0

2𝑌 = 5/2 - 5/2

2𝑌 = 0

𝑌 = 0/2

𝑌 = 0

𝑌 is constant and doesn't depend on 𝑋. Hence, R² =0

Var(𝑌) = 4 × (1 - R²)

Var(𝑌) = 4 × (1 - 0)

Var(𝑌) = 4 × 1

**Var(𝑌) = 4**

b. Coefficient of determination of C and Y= 0

Coefficient of determination (R²)=0

c. Standard error of estimate of X on Y and of Y on X.

Standard error of estimate of X on Y=2

Standard error of estimate of Y on X=2

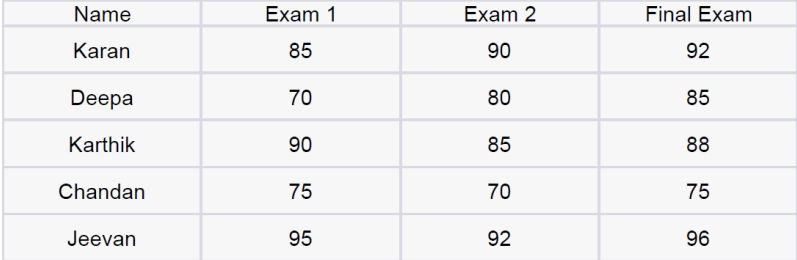
SE(estimate) = sqrt((Var(𝑌) × (1 - R²)))

Standard error of estimate of X on Y= sqrt(4\*(1-0))= 2

SE(estimate) = sqrt((Var(X) × (1 - R²)))

Standard error of estimate of Y on X= sqrt(4\*(1-0))= 2

Answer. 9 statistics -> <https://raw.githubusercontent.com/Lokeshrajkp/Placement-Assignment_Lokeshraj-K-P/main/Statistics/statistics_9.ipynb>



Mean 83 83.4 87.2

Null hypothesis:

H0:μ1=μ2=μ3

Means are equal

Alternate hypothesis:

H1: μ1≠μ2≠μ3

Means are not equal

Total mean=( 83+83.4+87.7)/3=84.5

n1=n2=n3=5

k=3

SSB = 6(5 - 8)2 + 6(9 - 8)2 + 6(10 - 8)2

SSB=53.72

df1 = k - 1 = 2

SSE = 430+311.2+254.8=996

N = 15

df2 = N - k = 15 - 3 = 12

MSB = SSB / df1 = 53.72 / 2 = 26.86

MSE = SSE / df2 = 996 / 12 = 83

ANOVA test statistic, f = MSB / MSE = 26.86 /83 = 0.323

Using the f table at α= 0.05 the critical value is given as F(0.05,2,12)=3.88

As f < F, thus, the null hypothesis is accepted and it can be concluded that there is no difference in the mean.

**Answer:** Fail to reject the null hypothesis

Answer. 10 statistics -><https://raw.githubusercontent.com/Lokeshrajkp/Placement-Assignment_Lokeshraj-K-P/main/Statistics/Statistics_10.ipynb>

Number of trials (n) = 500 (batch of light bulbs)

Probability of success (p) = 0.05 (probability of a bulb being defective)

Probability of failure (q) = 1 - p = 0.95 (probability of a bulb not being defective)

a. What is the probability that exactly 20 bulbs are defective?

P(X = 20) = (500C20) \* (0.05)^20 \* (0.95)^(500 - 20)

P(X = 20) =0.05162

b. What is the probability that at least 10 bulbs are defective?

P(X ≥ 10) = P(X = 10) + P(X = 11) + ... + P(X = 500)

P(X ≥ 10)=0.99983

c. What is the probability that at max 15 bulbs are defective?

P(X ≤ 15) = P(X = 0) + P(X = 1) + ... + P(X = 15)

P(X ≤ 15)=0.01986

d. On average, how many defective bulbs would you expect in a batch of 500?

E(X) = n \* p =500\*0.05

E(X)=25 bulb

Answer. 12 statistics ->

<https://raw.githubusercontent.com/Lokeshrajkp/Placement-Assignment_Lokeshraj-K-P/main/Statistics/statistics_12.ipynb>

a.

Null hypothesis (H₀): There is no significant difference in the mean improvement scores between Group A and Group B.

Alternative hypothesis (H₁): There is a significant difference in the mean improvement scores between Group A and Group B.

mean of Group A ==2.5

mean of Group B = =2.2

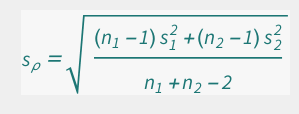
standard deviation of Group A = S1=0.8

standard deviation of Group B =S2=0.6

sample size of Group A =n1= 30

sample size of Group B ==n2= 30

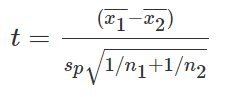
Pooled standard deviation formula



sp=sqrt(((30-1)\*(0.8)\*\*2+(30-1)\*(0.6)\*\*2)/(30+30-2))

sp=0.707

t-static



t=(2.5-2.2)/(0.707\*sqrt(1/30+1/30))

t=1.643

DOF=30+30-2

DOF=58

The *t* value with α = 0.05 and 58 degrees of freedom is 2.001

Since 1.64<2.001 , the t-statistic does not exceed the critical value

b. Since t-static(1.64)< critical value(2.001) we failed to reject null hypothesis

**Machine learning**

Answer. 1 Machine learning -> <https://github.com/Lokeshrajkp/Placement-Assignment_Lokeshraj-K-P/blob/main/Machine%20learning/Machine_learning_1.ipynb>

Answer. 2 Machine learning -> <https://github.com/Lokeshrajkp/Placement-Assignment_Lokeshraj-K-P/blob/main/Machine%20learning/Machine_learning_answer_2.ipynb>

Answer. 3 Machine learning ->

<https://github.com/Lokeshrajkp/Placement-Assignment_Lokeshraj-K-P/blob/main/Machine%20learning/Machine_learning_3.ipynb>

Answer. 4 Machine learning ->

<https://github.com/Lokeshrajkp/Placement-Assignment_Lokeshraj-K-P/blob/main/Machine%20learning/Machine_learning_4.ipynb>

Answer. 5 Machine learning -> <https://github.com/Lokeshrajkp/Placement-Assignment_Lokeshraj-K-P/blob/main/Machine%20learning/Machine%20learning%205.ipynb>

Answer. 6 Machine learning -> <https://github.com/Lokeshrajkp/loan_prediction.git>