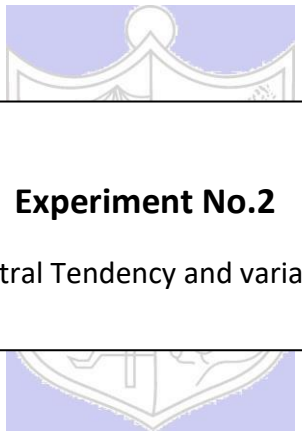

Experiment No.2

Title: Measuring Central Tendency and variability of the Data

Batch: 2 Roll No.: 16010420133 Experiment No. 2

Aim: To find measures of central tendency and variability of data of data using statistical analysis tool.

Resources needed: Any free and Open online statistical analysis tools

---

Results:

Code:

```
import pandas as pd

def heapify(column, n, i):
    largest = i
    l = 2 * i + 1
    r = 2 * i + 2

    if l < n and column[largest] < column[l]:
        largest = l

    if r < n and column[largest] < column[r]:
        largest = r

    if largest != i:
        column[i], column[largest] = column[largest], column[i]
        heapify(column, n, largest)

def heapSort(column):
    n = len(column)

    for i in range(n//2 - 1, -1, -1):
        heapify(column, n, i)

    for i in range(n-1, 0, -1):
        column[i], column[0] = column[0], column[i]
        heapify(column, i, 0)

    return column

column = []
csv_file = pd.read_csv('/Users/Soumen/Downloads/SEM_4
College_Stuff/HON/FDS/runs.csv')
csv_file_column = csv_file[csv_file['declared_weight'].notna()]
```

```
for i in range(0,len(csv_file_column)):
    column.append(int(csv_file_column['declared_weight'][i]))

column = heapSort(column)

column_sum = 0
for i in column:
    column_sum+=i

mean = column_sum/len(column)

print("The mean of all the declared weights of the horses is: " , mean)

multiplied_weights=0
for i in column:
    if i > 650 and i < 750:
        multiplied_weights += i*9
    elif i > 750 and i < 850:
        multiplied_weights += i*8
    elif i > 850 and i < 950:
        multiplied_weights += i*7
    elif i > 950 and i < 1050:
        multiplied_weights += i*6
    elif i > 1050 and i < 1150:
        multiplied_weights += i*5
    elif i > 1150 and i < 1250:
        multiplied_weights += i*4
    elif i > 1250 and i < 1350:
        multiplied_weights += i*3
    elif i > 1350:
        multiplied_weights += i*2

weighted_mean = multiplied_weights/44

print("The weighted mean of all the declared weights of the horses is: " , weighted_mean)

trim = int(0.05*(len(column)))
trimmed_column=[]
for i in range(len(column)):
    if i in range(0,trim) or i in range(len(column)- trim,len(column)):
        continue
    else:
        trimmed_column.append(column[i])

sum_trimmed=0
for i in trimmed_column:
```

```
    sum_trimmed += i

trimmed_mean = sum_trimmed/len(trimmed_column)

print("The trimmed mean of all the declared weights of the horses is: " , trimmed_mean)

grouped_data = {'650-750' : 0, '750-850' : 0, '850-950' : 0, '950-1050' : 0, '1050-1150' : 0,
'1150-1250' : 0, '1250-1350' : 0, '1350-1450' : 0}

for i in column:
    if i > 650 and i < 750:
        grouped_data['650-750'] += 1
    elif i > 750 and i < 850:
        grouped_data['750-850'] += 1
    elif i > 850 and i < 950:
        grouped_data['850-950'] += 1
    elif i > 950 and i < 1050:
        grouped_data['950-1050'] += 1
    elif i > 1050 and i < 1150:
        grouped_data['1050-1150'] += 1
    elif i > 1150 and i < 1250:
        grouped_data['1150-1250'] += 1
    elif i > 1250 and i < 1350:
        grouped_data['1250-1350'] += 1
    elif i > 1350 and i < 1450:
        grouped_data['1350-1450'] += 1

Lower_bound_median_interval = int((((column[0] - column[0] % 100 )+ 50) +
(column[len(column)-1] - (column[len(column)-1] % 100) + 50)) / 2) - 100

Number_values = 0
for i in grouped_data.values():
    Number_values += i

Number_values = Number_values/2
median_frequency = 0
if Lower_bound_median_interval < 750:
    median_frequency = grouped_data['650-750']
elif Lower_bound_median_interval > 750 and Lower_bound_median_interval < 850:
    median_frequency = grouped_data['750-850']
elif Lower_bound_median_interval > 850 and Lower_bound_median_interval < 950:
    median_frequency = grouped_data['850-950']
elif Lower_bound_median_interval > 950 and Lower_bound_median_interval < 1050:
    median_frequency = grouped_data['950-1050']
elif Lower_bound_median_interval > 1050 and Lower_bound_median_interval < 1150:
    median_frequency = grouped_data['1050-1150']
```

```
elif Lower_bound_median_interval > 1150 and Lower_bound_median_interval < 1250:
    median_frequency = grouped_data['1150-1250']
elif Lower_bound_median_interval > 1250 and Lower_bound_median_interval < 1350:
    median_frequency = grouped_data['1250-1350']
elif Lower_bound_median_interval > 1350 and Lower_bound_median_interval < 1450:
    median_frequency = grouped_data['1350-1450']

total_frequency = 0
for i,j in grouped_data.items():
    if j == median_frequency:
        class_range = i
        break
    total_frequency += j

class_range = class_range.split('-')
width = int(class_range[1])-int(class_range[0])

median = Lower_bound_median_interval + (((Number_values - total_frequency) /
median_frequency) * width)

print("The median of all the declared weights of the horses is: " , median)

minimum_frequency = 0
mode = 0
for i , j in grouped_data.items():
    if j > minimum_frequency:
        minimum_frequency = j
        mode = i

print("The mode of all the declared weights of the horses is: " , mode)

percent_25 = column[int(0.25 * len(column))]
percent_75 = column[int(0.75 * len(column))]

interquartile_range = percent_75 - percent_25

print("The interquartile range of all the declared weights of the horses is: " ,
interquartile_range)

total_sum = 0
for i in column:
    total_sum += (i - mean) ** 2

variance = total_sum / (len(column) - 1)

print("The variance of all the declared weights of the horses is: " , variance)
```

```
standard_deviation = variance ** 0.5

print("The standard deviation of all the declared weights of the horses is: " ,
standard_deviation)

absolute_sum = 0
for i in column:
   absolute_sum += abs(i - mean)

mean_deviation = absolute_sum / len(column)

print("The mean deviation of all the declared weights of the horses is: " , mean_deviation)

coefficient_variance = standard_deviation / mean

print("The coefficient of variance of all the declared weights of the horses is: " ,
coefficient_variance)

standard_error = standard_deviation / (len(column) ** 0.5)

print("The standard error of variance of all the declared weights of the horses is: " ,
standard_error)
```

Output:

```
The mean of all the declared weights of the horses is:  1104.9535413546137
The weighted mean of all the declared weights of the horses is:  9698295.159090908
The trimmed mean of all the declared weights of the horses is:  1104.1320084471981
The median of all the declared weights of the horses is:  12783.08157099698
The mode of all the declared weights of the horses is:  1050-1150
The interquartile range of all the declared weights of the horses is:  84
The variance of all the declared weights of the horses is:  3887.2235168685165
The standard deviation of all the declared weights of the horses is:  62.34760233456068
The mean deviation of all the declared weights of the horses is:  49.65081922234295
The coefficient of variance of all the declared weights of the horses is:  0.05642554189031864
The standard error of variance of all the declared weights of the horses is:  0.22119790302384207
```

Questions:

1. What are the various applications of central tendency and variability of data?

   The arithmetic mean is considered a deal average. It is frequently used in all the aspects of business i.e. number of items produced per day on a large assembly line, number of orders received per month for a firm. further In economic analysis arithmetic mean is used extensively to calculate average production, Median is positional measures of central tendency. The median salary gives a value close to the average salary commonly paid, without taking the extreme values into consideration. Geometric Mean is used in the construction of index number.

2. What are the outlier's data? What are the different ways to find out it? Give suitable example with its effect on central tendency and variability of data?

   Outlier is a data point that differs significantly from other observations, Given mu and sigma, a simple way to identify outliers is to compute a z-score for every xi, which is defined as the number of standard deviations away xi is from the mean […] Data values that have a z-score sigma greater than a threshold, It will move towards the outlier.

Outcomes:

CO2: Comprehend descriptive and proximity measures of data

Conclusion:

We were able to understand the concept of measuring data and we have thus calculated the mean (mean , weighted mean , trimmed mean) , median , mode , variance , standard deviation , mean deviation , interquartile range , coefficient of variance and standard error for a column from our data set.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

Books/ Journals/ Websites:

1. Han, Kamber, "Data Mining Concepts and Techniques", Morgan Kaufmann 3nd Edition
2. S.C. Gupta , V. K. Kapoor Fundamentals of mathematical statistics Sultan Chand and Sons 2014