

Name :- Suhel S. Maktum

Class :- SY

Div :- C

R.N. :- 73

Assignment

Unit 1: Frequency Distribution and Measure of Central Tendency

Q.1. Write C-program to find the range and coefficient of range of the raw data.

```
#include<stdio.h>

int main()
{
    int array[100],n,i;
    float coeffrange,range,min,max;
    printf("Enter the number of element you want to insert");
    scanf("%d",&n);
    printf("Enter the element of array");
    for(i=0;i<n;i++)
    {
        scanf("%d",&array[i]);
    }
    min=max=array[0];
    for(i=0;i<n;i++)
    {
        if(array[i]>=max)
        {
            max=array[i];
        }
        else if(array[i]<=min)
        {
            min=array[i];
        }
        else
        {
        }
    }

    range= max-min;
```

```

printf("\n range = %f",range);

coeffrange = (max-min)/(max+min);

printf("\n coefficient of range = %f",coeffrange);
return 0;
}

```

Q.2. Write C-program to find mean, median and mode of the raw data.

Mean

```

#include<stdio.h>
int main()
{
    int array[100],n,i;
    float sum=0;
    printf("Enter the number of element you want to insert in array");
    scanf("%d",&n);

    for(i=0;i<n;i++)
    {
        scanf("%d",&array[i]);
        sum=sum+array[i];
    }
    float mean=sum/n;
    printf("Mean =%f",mean);

    return 0;
}

```

Mode

```

#include <stdio.h>

int main() {
    int n, i, j, mode, count, maxCount = 0;
    int arr[100];

    printf("Enter the number of elements: ");
    scanf("%d", &n);

    printf("Enter the elements: ");
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
}

```

```

for (i = 0; i < n; i++) {
    count = 0;
    for (j = 0; j < n; j++) {
        if (arr[j] == arr[i]) {
            count++;
        }
    }
    if (count > maxCount) {
        maxCount = count;
        mode = arr[i];
    }
}

printf("Mode = %d\n", mode);

return 0;
}

```

Median

```

#include<stdio.h>
int main()
{
    int n,i,result;
    int median;
    printf("Enter the number of element you want to insert in array");
    scanf("%d",&n);
    int array[100];
    printf("Enter the array :");
    for(i=0;i<n;i++)
    {
        scanf("%d",&array[i]);
    }
    if(n%2==0)
    {
        result= ((n/2)+((n+1)/2))/2;
        median= array[result];
    }
    else
    {
        result=(n+1)/2;
        median= array[result];
    }
    printf("Median = %d",median);

    return 0;
}

```

```
}
```

Q.3. Write C-program to find quartile deviation.

```
#include<stdio.h>
int main()
{
    int n,array[100],i;
    float Q1,Q3,QD;//result1,result2;
    printf("Enter the number of element you want to insert in array");
    scanf("%d",&n);
    printf("Enter element of an array ");
    for(i=0;i<n;i++)
    {
        scanf("%d",&array[i]);
    }
    // result1=(n+1)/4;
    //result2=3*((n+1)/4);
    Q1=array[(n+1)/4];
    Q3=array[3*((n+1)/4)];
    QD= (Q3-Q1)/2;

    printf(" QD= %f",QD);
    return 0;
}
```

Unit 3: Correlation and regression

Q.4. Write C-program to find correlation coefficient for bi-variate data.

```
#include <stdio.h>

// Function to calculate the mean of an array
float mean(float arr[], int n) {
    float sum = 0;
    for (int i = 0; i < n; i++) {
        sum += arr[i];
    }
    return sum / n;
}

// Function to calculate the standard deviation of an array
float stddev(float arr[], int n, float m) {
    float sum = 0;
    for (int i = 0; i < n; i++) {
        sum += (arr[i] - m) * (arr[i] - m);
    }
}
```

```

        return sqrt(sum / (n - 1));
    }

// Function to calculate the covariance between two arrays
float covariance(float arr1[], float arr2[], int n, float m1, float m2) {
    float sum = 0;
    for (int i = 0; i < n; i++) {
        sum += (arr1[i] - m1) * (arr2[i] - m2);
    }
    return sum / (n - 1);
}

int main() {
    int n;
    printf("Enter the number of data points: ");
    scanf("%d", &n);

    float x[n], y[n];
    printf("Enter the data points for x and y:\n");
    for (int i = 0; i < n; i++) {
        scanf("%f %f", &x[i], &y[i]);
    }

    // Calculate the mean of x and y
    float mean_x = mean(x, n);
    float mean_y = mean(y, n);

    // Calculate the standard deviation of x and y
    float stddev_x = stddev(x, n, mean_x);
    float stddev_y = stddev(y, n, mean_y);

    // Calculate the covariance between x and y
    float cov = covariance(x, y, n, mean_x, mean_y);

    // Calculate the correlation coefficient
    float corrcoef = cov / (stddev_x * stddev_y);

    printf("Correlation coefficient = %.2f\n", corrcoef);

    return 0;
}

```