Statistics and Probability /MAT19234E

PROJECT REPORT

Name Surname	Merve Çınar
Number	1821221025
Dataset Name	Iris Species
Column Name	SepalLengthCm
Dataset Link	**https://www.kaggle.com/uciml/iris**

My Dataset and Column;

My dataset is Iris species. What is Iris species? It's a kinda type of flower. These datasets included Length of Sepal, Width of Sepal, Width of Petal, and name of Species. All of length and width in cm. So, experts are grouped according to this information by dividing irises into types. I choose this dataset because, I understand what is the mean of Iris and columns, and this column how it works to dividing irises into types. And I choose SepalLengthCm columns. İt is included cm of Sepal lengths of Irises. I choose this column because this column's value is clear. Another fact that to choose this dataset, it has SQLite file and this dataset's value is very clear and small numbers. So, understanding and calculating the values is not very hard.

Here are my SepalLengthCm values;

[5.1, 4.9, 4.7, 4.6, 5.0, 5.4, 4.6, 5.0, 4.4, 4.9, 5.4, 4.8, 4.8, 4.3, 5.8, 5.7, 5.4, 5.1, 5.7, 5.1, 5.4, 5.1, 4.6, 5.1, 4.8, 5.0, 5.0, 5.2, 5.2, 4.7, 4.8, 5.4, 5.2, 5.5, 4.9, 5.0, 5.5, 4.9, 4.4, 5.1, 5.0, 4.5, 4.4, 5.0, 5.1, 4.8, 5.1, 4.6, 5.3, 5.0, 7.0, 6.4, 6.9, 5.5, 6.5, 5.7, 6.3, 4.9, 6.6, 5.2, 5.0, 5.9, 6.0, 6.1, 5.6, 6.7, 5.6, 5.8, 6.2, 5.6, 5.9, 6.1, 6.3, 6.1, 6.4, 6.6, 6.8, 6.7, 6.0, 5.7, 5.5, 5.5, 5.8, 6.0, 5.4, 6.0, 6.7, 6.3, 5.6, 5.5, 5.5, 6.1, 5.8, 5.0, 5.6, 5.7, 5.7, 6.2, 5.1, 5.7, 6.3, 5.8, 7.1, 6.3, 6.5, 7.6, 4.9, 7.3, 6.7, 7.2, 6.5, 6.4, 6.8, 5.7, 5.8, 6.4, 6.5, 7.7, 7.7, 6.0, 6.9, 5.6, 7.7, 6.3, 6.7, 7.2, 6.2, 6.1, 6.4, 7.2, 7.4, 7.9, 6.4, 6.3, 6.1, 7.7, 6.3, 6.4, 6.0, 6.9, 6.7, 6.9, 5.8, 6.8, 6.7, 6.7, 6.3, 6.5, 6.2, 5.9]

Firstly I order from smallest to largest;

[4.3, 4.4, 4.4, 4.4, 4.5, 4.6, 4.6, 4.6, 4.6, 4.7, 4.7, 4.8, 4.8, 4.8, 4.8, 4.8, 4.9, 4.9, 4.9, 4.9, 4.9, 4.9, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0, 5.1, 5.1, 5.1, 5.1, 5.1, 5.1, 5.1, 5.1, 5.1, 5.2, 5.2, 5.2, 5.2, 5.3, 5.4, 5.4, 5.4, 5.4, 5.4, 5.4, 5.5, 5.5, 5.5, 5.5, 5.5, 5.5, 5.5, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.7, 5.7, 5.7, 5.7, 5.7, 5.7, 5.7, 5.7, 5.8, 5.8, 5.8, 5.8, 5.8, 5.8, 5.8, 5.9, 5.9, 5.9, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.2, 6.2, 6.2, 6.2, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.5, 6.5, 6.5, 6.5, 6.5, 6.6, 6.6, 6.7, 6.7, 6.7, 6.7, 6.7, 6.7, 6.7, 6.7, 6.8, 6.8, 6.8, 6.9, 6.9, 6.9, 6.9, 7.0, 7.1, 7.2, 7.2, 7.2, 7.3, 7.4, 7.6, 7.7, 7.7, 7.7, 7.7, 7.9] ,

Size of SepalLengthCm values=n=150

1) Mean :

Mean is very important data for my homework because I use mean in other calculations. So, usually, I calculate the mean at first. Mean's other name is average.it calculates as sum all of my data element and then divided by a number of data size. This gives me a population means. The Mean formula is that;

$\sum x_i$ = Sum of all data = 5.1 + 4.9 + 4.7 + 4.6 + 5.0 + 5.4 + 4.6 + 5.0 + 4.4 + 4.9 + 5.4 + 4.8 + 4.8 + 4.3 + 5.8 + 5.7 + 5.4 + 5.1 + 5.7 + 5.1 + 5.4 + 5.1 + 4.6 + 5.1 + 4.8 + 5.0 + 5.0 + 5.2 + 5.2 + 4.7 + 4.8 + 5.4 + 5.2 + 5.5 + 4.9 + 5.0 + 5.5 + 4.9 + 4.4 + 5.1 + 5.0 + 4.5 + 4.4 + 5.0 + 5.1 + 4.8 + 5.1 + 4.6 + 5.3 + 5.0 + 7.0 + 6.4 + 6.9 + 5.5 + 6.5 + 5.7 + 6.3 + 4.9 + 6.6 + 5.2 + 5.0 + 5.9 + 6.0 + 6.1 + 5.6 + 6.7 + 5.6 + 5.8 + 6.2 + 5.6 + 5.9 + 6.1 + 6.3 + 6.1 + 6.4 + 6.6 + 6.8 + 6.7 + 6.0 + 5.7 + 5.5 + 5.5 + 5.8 + 6.0 + 5.4 + 6.0 + 6.7 + 6.3 + 5.6 + 5.5 + 5.5 + 6.1 + 5.8 + 5.0 + 5.6 + 5.7 + 5.7 + 6.2 + 5.1 + 5.7 + 6.3 + 5.8 + 7.1 + 6.3 + 6.5 + 7.6 + 4.9 + 7.3 + 6.7 + 7.2 + 6.5 + 6.4 + 6.8 + 5.7 + 5.8 + 6.4 + 6.5 + 7.7 + 7.7 + 6.0 + 6.9 + 5.6 + 7.7 + 6.3 + 6.7 + 7.2 + 6.2 + 6.1 + 6.4 + 7.2 + 7.4 + 7.9 + 6.4 + 6.3 + 6.1 + 7.7 + 6.3 + 6.4 + 6.0 + 6.9 + 6.7 + 6.9 + 5.8 + 6.8 + 6.7 + 6.7 + 6.3 + 6.5 + 6.2 + 5.9= 876,5000000000002 ,

n = size of data=150

Mean = $\sum x_i$ /n= μ = 876,5000000000002/150= 5,843

2) Median :

When the elements are ordered from smallest to largest, the middle element is call median in the data. There is no difficult to the find median in little dataset but ,very hard to find median in large dataset. The formula to make it easier to find the median is;

-When the dataset size is odd the formula to find median's index;

Median = M = $[(n+1)/2]^{th}$ term

-When the dataset size is even the formula to find median's index;

Median = $[(n/2)^{th}$ term + $\{(n/2)+1\}^{th}]/2$

Mine dataset size is even , so I used second formula;

Size of data=n=150

Median = $[(n/2)^{th}$ term + $\{(n/2)+1\}^{th}]/2$

Median = $[(150/2)^{th}$ term + $\{(150/2)+1\}^{th}]/2$

Median = $[(75)^{th}$ term + $(76)^{th}]/2$

Median=[(5,8+5,8)/2]=5,8

//COMMENT: Also, I look at the ordered dataset I saw most repetitive element ("MODE") is 5.0.

3) Standard Deviation :

A standard deviation say that, on average ,how far each value lies from the mean. I calculate the standard deviation is the square root of variance. Also, it is calculated as square root of variance by determining each data point's deviation relative to the mean. If the data points are far from the mean, there is a higher deviation within the data thus, the more far from the mean out the data, the higher the standard deviation. The formula is that;

μ=5,843

$\sum(x-\mu)^2 = (5.1 - 5.843)^2+(4.9 - 5.843)^2 +(4.7 - 5.843)^2 +(4.6 - 5.843)^2 +(5.0 - 5.843)^2 +(5.4 - 5.843)^2 +(4.6 - 5.843)^2 +(5.0 - 5.843)^2 +(4.4 - 5.843)^2 +(4.9 - 5.843)^2 +.....+(6.9 - 5.843)^2 +(5.8 - 5.843)^2 +(6.8 - 5.843)^2 +(6.7 - 5.843)^2 +(6.7 - 5.843)^2 +(6.3 - 5.843)^2 +(6.5 - 5.843)^2 +(6.2 - 5.843)^2 +(5.9 - 5.843)^2 = 102.1683$

(n-1)=150-1=149

$\sigma = \sqrt{\sum(x-\mu)^2/(n-1)} = \sqrt{(102,1683/149)} = \sqrt{(0,6856)} = 0,828$

//COMMENT: Standard Deviation is has not big value because elements are close the mean value.

4) Variance :

The variance refers to a statistical calculate of the far between numbers in a data set. It is used analysts and traders to determine volatility and market security. The square root of the variance is the standard deviation. The formula like that;

μ=5,843

$\sum(x-\mu)^2 = (5.1 - 5.843)^2+(4.9 - 5.843)^2 +(4.7 - 5.843)^2 +(4.6 - 5.843)^2 +(5.0 - 5.843)^2 +(5.4 - 5.843)^2 +(4.6 - 5.843)^2 +(5.0 -5.843)^2 +(4.4-5.843)^2+(4.9-5.843)^2 + ... +(6.9 - 5.843)^2 +(5.8 - 5.843)^2 +(6.8 - 5.843)^2 +(6.7 - 5.843)^2 +(6.7 - 5.843)^2 +(6.3 - 5.843)^2 +(6.5 - 5.843)^2 +(6.2 - 5.843)^2 +(5.9 - 5.843)^2 = 102.1683$

(n-1)=150-1=149

$\sigma^2 = \sum(x - \mu)^2/(n-1) = 102.1683/149 = 0,6856$

5) Standard Error :

the standard error of a statistic's mean is the close standard deviation of a statistical sample population. The standard error is a statistical term that calculates the reality with which a sample distribution represents a population by using standard deviation.

$\sigma = \sqrt{\sum(x - \mu)^2/(n-1)} = \sqrt{102.1683/149} = 0,828$,n=size=150

$SE=\sigma/\sqrt{n}=0,828/(\sqrt{150})= 0,0676$

6) Shape of Distribution:

M(Median)= μ(mean) shape of distribution is "SYMMETRIC DISTRIBUTION"

M(Median)>μ(mean) shape of distribution is "LEFT-SKEWED DISTRIBUTION"

M(Median)<μ(mean) shape of distribution is "RIGHT-SKEWED DISTRIBUTION"

Mean = $\sum x_i /n$= μ=876,5000000000002/150= 5,84

Median=M=5.8

5,8<5,843 so , M(Median)<μ(mean) shape of distribution is "RIGHT-SKEWED DISTRIBUTION"

//COMMENT: My median is 5.8 and My mean is 5.84 these values very close to each other when the graph drown it can looking "SYMMETRIC DISTRIBUTION".

7)

-QUARTİLES: Quartiles are divided your data into four pieces. It helps to understand which part has which value. It turns the whole data into a small group.

There are three quartiles;

N=150

- Q1(%25)= $[(n+1)/4]^{th}$ term , Q2(%50)= $[(n+1)/2]^{th}$ term , Q3(%75)= $[3(n+1)/4]^{th}$ term

Q1=$[(150+1)/4]^{th}$ term=Q1$(37,75)^{th}$ term=5,1

Q2=$[(150+1)/2]^{th}$ term=Q1$(75,50)^{th}$ term=5,8

Q3=$[3(150+1)/4]^{th}$ Term=Q1$(113,25)^{th}$ term=6,4

-IQR, İNTERQUARTİLE RANGE : it is calculate the distance between first quartile and last quartile.

IQR=Q3-Q1=6,4-5,1=1,3

-LOWER AND UPPER LİMİTS : it calculate the area of data set. I use IQR value to find area.

Start=Q1-1,5*(IQR) =5,1-1,5*(1,3) =3,15, End=Q3+1,5*(IQR) =6,4-1,5*(1,3) =8,35

Result =[3,15 , 8,35]

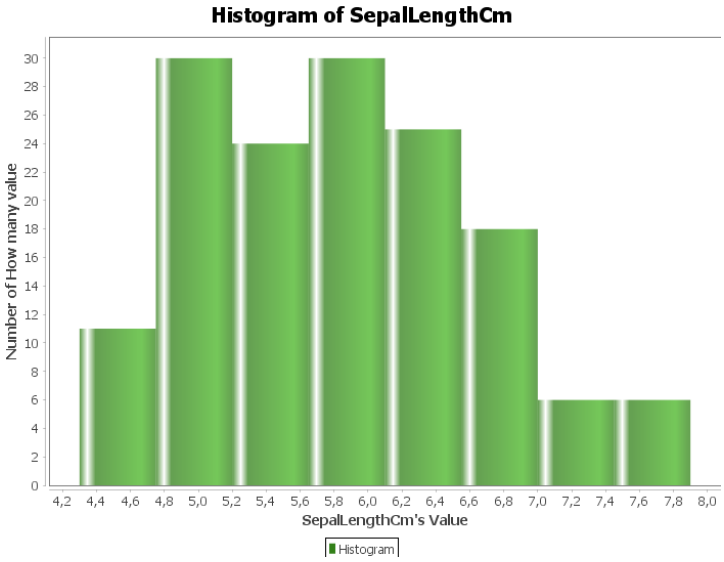
-MAXİMUM AND MİNİMUM VALUE: When the order data from smallest to largest the first element is Minimum data , last element is maximum data; Max=7.9 , Min=4.3

Also,I can find the Range to use max and min value.Because, Range is the difference between Max and Min value . So, Range =7.9-4.3=3,6.

8) HISTOGRAM :

Histogram is bar graph of shows the how many values inside the group of numbers. You have to find bins and you can group by number of columns by using bins.

N=150 , Bin=$\log_2{}^{n}+1= \log_2{}^{150}+1=8$ (so there are 8 bars.)



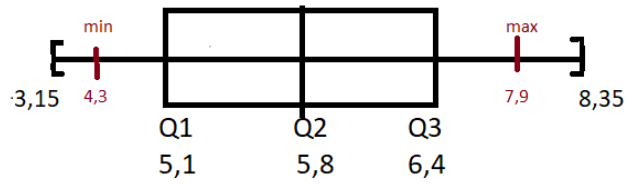
//COMMENT : When I look at the Histogram, There are too many values around (4,8-5,2)and (5.8-6,1). So, ı can say that Most of the elements in the dataset are between 4.8 and 6.1. Also, last of the elements in the dataset are between 7.1 and 7.9. I can calculate the sample size from y axis. So, I saw the size of dataset is 150.

9) BOXPLOT

:Boxplot, gives a lot of information about the dataset. I have to find Q1,Q2,Q3,IQR,MAX,MİN and LIMITS.

My dataset hasn't outliers because the maximum data is 7.9 and minimum data is 4.3 does not exceed the limits [3,15,8,35]. So these value inside the are.

Q1=5.1, Q2=5.8, Q3=6,4, IQR=1,3 , RİGHT DİSTRİBUTİON. Also, limits are [3,15,8,35]



//COMMENT: I think the boxplot is a very nice thing, I can find many values in this image. I can see the shape of the distribution, my mean and median is very close,(almost equal). For this reason and this image, ı understand, the shape of the distribution is "SYMMETRIC DISTRIBUTION". Another thing that, Interquartile range is not very large, Here the values are generally close to each other I realized it was. Lastly, I saw there are no outliers because the max value is 7.9 and the min value is 4,3, so I know the limits [3,15, 8,35] don't out of bounds.

10) CONFIDENCE INTERVAL :

Confidence intervals calculate the degree of unreliability or reality in a sampling method. They can take any number of probability limits, 90% or 95% confidence level. Confidence intervals are such a test to measures reliability.

I create a sample value to find Confidence intervals.

Sample value = [6.2, 6.5, 5.5, 6.5, 5.4, 5.1, 6.7, 4.6, 5.7, 5.7, 4.7, 7.7, 7.4, 4.5, 5.6, 5.0, 7.4, 5.4, 5.7, 6.3, 6.1, 6.3, 5.6, 5.1, 7.9, 4.4, 6.0, 7.7, 6.1, 6.1, 4.6, 4.9, 5.2, 7.7, 6.3, 5.8, 5.6, 6.3, 6.6, 6.2, 6.3] (I choose randomly)

Sample size =41

A) %95 Confidence Interval for a mean;

- ❖ Known population standard deviation.
- ❖ N>30 large.
- ❖ Known sample mean

Sum of All Sample= $\sum x_i$ =6.2 + 6.5 + 5.5 + 6.5 + 5.4 + 5.1 + 6.7 + 4.6 + 5.7 + 5.7 + 4.7 + 7.7 + 7.4 + 4.5 + 5.6 + 5.0 + 7.4 + 5.4 + 5.7 + 6.3 + 6.1 + 6.3 + 5.6 + 5.1 + 7.9 + 4.4 + 6.0 + 7.7 + 6.1 + 6.1 + 4.6 + 4.9 + 5.2 + 7.7 + 6.3 + 5.8 + 5.6 + 6.3 + 6.6 + 6.2 + 6.3 = 244.4

Sample size =n=41 , Sample mean = $\bar{x} = (\Sigma x_i) / n$=244,7/41= 5.96.

$\sigma = \sqrt{\sum(x - \mu)^2/(n-1)}$= √102.1683/149= 0,828

1-α=%95 , α=%5=0,5 , $Z_{\alpha/2} = Z_{0,025}$=1,96(From Z table)

$CI=(\bar{x} - Z_{\alpha/2} * \sigma/\sqrt{n}, \ \bar{x} + Z_{\alpha/2} * \sigma/\sqrt{n},)$

CI1=5,96(center)-(1.96)(0,828 √41)(margin), CI2=5,96 (center)+(1.96)(0,828) (margin)

CI1=5,707 and CI2=6,213, SO CI= 5,707 < μ < 6,213

B) %95 Confidence Interval for a variance;

- ❖ Known sample standard deviation.
- ❖ Known sample mean

N=41 , Sample mean = $\bar{x} = (\Sigma x_i) / n$=244,7/41= 5.96

$\sum(x - \bar{x})^2 = (6.2 - 5.96)^2+(6.5 - 5.96)^2 +(5.5 - 5.96)^2 +(6.5 - 5.96)^2 +(5.4 - 5.96)^2$ +.....+ $(6.6 - 5.96)^2 +(6.2 - 5.96)^2 +(6.3 - 5.96)^2$ = 34.6976

S= standard deviation= $\sqrt{\sum(x - \bar{x})^2/(n)}$= √(34,6976/41)=0,919

S^2 = sample variance= 0,846

1-α=%95, α=%5=0,5

$X_{\alpha/2,40} = X_{0,025,40}$= 59.3417

$X_{\alpha/2,40} = X_{(1-0,025),40} = 24.4331$

$CI=[(n-1)\ S^2 \ / \ \chi^2_{\alpha/2\ ,\ n-1}\ ,\ (n-1)\ S^2 / \chi^2_{1-\alpha/2,\ n-109}]$

$CI1=(40*(0,846))/59,3417= 0.57$, $CI1=(40*(0,846))/24,4331= 1.385$

$CI= 0.57 \leq \sigma^2 \leq 1.385$

//COMMENT:I choose the sample data randomly and I think that My selection is very good. Because, When I calculate the sample mean is very close to the population mean. Another things is , $5,707 < \mu < 6,213$ is %95 Confidence Interval for mean. And population mean value is 5,83. And I understand the %95 Confidence Interval for mean reliability. For %95 Confidence Interval variance, the sample variance and mean very close to the population values. $0.57 \leq \sigma^2 \leq 1.385$ is %95 Confidence Interval for variance. And population variance is 0.68, so it is reliable.

10) How large a sample for your data should be collected to estimate the population mean with a margin at most 0.1 units with confidence 90%?

$1-\alpha=\%90$,$\alpha=\%10=0,1$

$Z_{\alpha/2} = Z_{0,05}=1,645$(From Z table)

$\sigma = \sqrt{\sum(x-\mu)^2/(n-1)}= \sqrt{102.1683/149}= 0,828$

$\Delta=0.1$units

$N>=((Z^{\alpha/2} * \ \sigma)/\Delta)^2$

$N>= ((1,645)*(0,828))^2/0,1,$ $N>=184$

//COMMENT: If I want to margin at most 0.1 units with confidence 90% I should select at least 184 data.

FINALLY:THE CODES (I USE JAVA SWİNG , JAVA JFREECHART AND JAVA DB)

```
public class main extends javax.swing.JFrame {
    ArrayList<Double> db = new ArrayList<>();
    int counter = 0;
    double mean = 0.0;
    double median = 0.0;
    double stdDeviation = 0.0;
    double variance = 0.0;
    static DefaultTableModel table = new DefaultTableModel();
    public main() {
        initComponents();
        table.setColumnIdentifiers(new Object[]{"ID", "SepalLengthCm"});
        jTable1.setModel(table);
    }
    @SuppressWarnings("unchecked")
```

```
private void b_viewdataActionPerformed(java.awt.event.ActionEvent evt) {
    //ORDER THE VALUES FROM SMALLEST TO LARGEST  ON TABLE
    int i = 0;
    if (counter == 0) {
        try {
            Connection con = DriverManager.getConnection("jdbc:derby://localhost:1527/
            java.sql.Statement st = con.createStatement();
            String sql = ("SELECT * FROM  IRIS order by SepalLengthCm ASC");

            ResultSet rs = st.executeQuery(sql);
            while (rs.next()) {
                double SepalLengthCm = rs.getDouble("SepalLengthCm");
                i = i + 1;
                table.addRow(new Object[]{i, SepalLengthCm});
            }
        } catch (SQLException e) {
            System.out.println(e);
        }
        counter++;
    }
```

```
private void b_medianActionPerformed(java.awt.event.ActionEvent evt) {
    // MEDIAN HESABI
    try {
        Connection cnn = DriverManager.getConnection("jdbc:derby://localhost
        Statement stmt = cnn.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT * FROM IRIS");
        while (rs.next()) {
            if (db.size() < 150) {
                db.add(rs.getDouble("SepalLengthCm"));
            }
        }
    } catch (SQLException e) {
        System.out.println(e);
    }
    int size = db.size();
    double[] sArr = new double[size];
    for (int i = 0; i < size; i++) {
        sArr[i] = db.get(i);
    }
    Arrays.sort(sArr);
    if (size % 2 == 0) {
        median = ((double) sArr[size / 2] + (double) sArr[size / 2 - 1]) / 2;
    } else if (size % 2 == 1) {
        median = (double) sArr[size / 2];
    }
    jTextPane2.setText(String.valueOf(median));
    System.out.println("MEDİAN = " + median);
}
```

```
private void b_meanActionPerformed(java.awt.event.ActionEvent evt) {
    //MEAN
    try {
        Connection cnn = DriverManager.getConnection("jdbc:derby://localhost
        Statement stmt = cnn.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT * FROM IRIS");
        while (rs.next()) {
            if (db.size() < 150) {
                db.add(rs.getDouble("SepalLengthCm"));
            }
        }
    } catch (SQLException e) {
        System.out.println(e);
    }
    int size = db.size();
    double[] sArr = new double[size];
    for (int i = 0; i < size; i++) {
        sArr[i] = db.get(i);
    }
    Arrays.toString(sArr);
    double sum = 0;
    for (int i = 0; i < sArr.length; i++) {
        sum = sum + sArr[i];
    }
    mean = sum / size;
    jTextPane1.setText(String.valueOf(mean));
    System.out.println("MEAN = " + mean);
```

```
private void b_shapeofdisActionPerformed(java.awt.event.ActionEvent evt) {
    //SHAPE OF DISTRIBUTION
    if (mean == 0 || median == 0) {
        JOptionPane.showMessageDialog(null, "Firstly,you have to calculate me
    } else {
        if (mean == median) {
            jTextPane3.setText("SYMMETRIC DISTRIBUTION");
            System.out.println("SYMMETRIC DISTRIBUTION");
        } else if (mean > median) {
            jTextPane3.setText("RIGHT-SKEWED DISTRIBUTION");
            System.out.println("RIGHT-SKEWED DISTRIBUTION");

        } else if (mean < median) {
            jTextPane3.setText("LEFT-SKEWED DISTRIBUTION");
            System.out.println("LEFT-SKEWED DISTRIBUTION");
        }
    }
```

```
private void b_stdActionPerformed(java.awt.event.ActionEvent evt) {
//STANDARD DAVITION
    try {
        Connection cnn = DriverManager.getConnection("jdbc:derby://localhc
        Statement stmt = cnn.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT * FROM IRIS");
        while (rs.next()) {
            if (db.size() < 150) {
                db.add(rs.getDouble("SepalLengthCm"));
            }}
            } catch (SQLException e) {
        System.out.println(e); }
    int size = db.size();
    double[] sArr = new double[size];
    for (int i = 0; i < size; i++) {
        sArr[i] = db.get(i);
    }
    if (stdDeviation == 0) {
        double sum = 0;
        for (int i = 0; i < size; i++) {
            sum = sum + sArr[i];
        }
        mean = sum / size;
        for (int i = 0; i < size; i++) {
            sArr[i] = db.get(i);
        }
        double k = 0.0;
        for (int i = 0; i < size; i++) {
            k += Math.pow(sArr[i] - mean, 2);
        }
        stdDeviation = Math.sqrt(k / size);
    }
    jTextPane4.setText(String.valueOf(stdDeviation));
    System.out.println("STANDART DAVIATION = " + stdDeviation);
}
```

```
private void b_varianceActionPerformed(java.awt.event.ActionEvent evt) {
//VARIANCE HESABI
    try {
        Connection cnn = DriverManager.getConnection("jdbc:derby://localhc
        Statement stmt = cnn.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT * FROM IRIS");
        while (rs.next()) {
            if (db.size() <= 150) {
                db.add(rs.getDouble("SepalLengthCm"));
            }}
    } catch (SQLException e) {
        System.out.println(e);
    }
    double sum = 0;
    int size = db.size();
    double[] sArr = new double[size];
    for (int i = 0; i < size; i++) {
        sArr[i] = db.get(i);
    }
    Arrays.sort(sArr);
    if (mean == 0.0) {
        for (int i = 0; i < sArr.length; i++) {
            sum = sum + sArr[i];
        }
        mean = sum / size;
    }
    double k = 0.0;
    for (int i = 0; i < size; i++) {
        k += Math.pow(sArr[i] - mean, 2);
    }
    variance = k / size;
    System.out.println("VARIANCE= " + variance);
    jTextPane5.setText(String.valueOf(variance));
}
```

```
private void b_sterorActionPerformed(java.awt.event.ActionEvent evt) {
    //STANDART EROR
    if (stdDeviation == 0.0) {
        try {
            Connection cnn = DriverManager.getConnection("jdbc:derby://local
            Statement st = cnn.createStatement();
            ResultSet rs = st.executeQuery("SELECT * FROM IRIS");
            while (rs.next()) {
                if (db.size() < 150) {
                    db.add(rs.getDouble("SepalLengthCm"));
                } } } catch (SQLException ex) {
            System.out.println(ex); }
        int size = db.size();
        double[] sArr = new double[size];
        for (int i = 0; i < size; i++) {
            sArr[i] = db.get(i); }
        Arrays.sort(sArr);
        if (stdDeviation == 0) {
            double total = 0;
            for (int i = 0; i < sArr.length; i++) {
               total = total + sArr[i];}
            mean = total / size;
            for (int i = 0; i < size; i++) {
                sArr[i] = db.get(i);}
            Arrays.sort(sArr);
            double c = 0.0;
            for (int i = 0; i < size; i++) {
                c += (sArr[i] - mean)* (sArr[i] - mean);}
            stdDeviation = Math.sqrt(c / size);    } }
    double size = db.size();
    double sizesqr = Math.sqrt(size);
    double SE = stdDeviation / sizesqr;
    jTextPane6.setText(String.valueOf(SE));
    System.out.println("STANDART EROR =" + SE);
```

```
private void b_histogramActionPerformed(java.awt.event.ActionEvent evt) {
    try { //HİSTOGRAM
        Connection cnn = DriverManager.getConnection("jdbc:derby://localhost:1527/MER
        Statement stmt = cnn.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT * FROM IRIS");
        while (rs.next()) {
            if (db.size() < 150) {
                db.add(rs.getDouble("SepalLengthCm"));}}
    } catch (SQLException e) {
        System.out.println(e);}
    int size = db.size();
    double[] sArr = new double[size];
    for (int i = 0; i < size; i++) {
        sArr[i] = db.get(i);    }
    int bin = (int) ((Math.log(size) / Math.log(2)) + 1);
    int number = bin;
    HistogramDataset dataset = new HistogramDataset();
    dataset.setType(HistogramType.FREQUENCY);
    dataset.addSeries("Histogram", sArr, number);
    String plotTitle = "Histogram of SepalLengthCm";
    String xeks = "SepalLengthCm's Value";
    String yeks = "Number of How many value";
    PlotOrientation orientation = PlotOrientation.VERTICAL;
    boolean show,urls  = false;
    boolean toolTips = true;
    int width,height = 100;
    JFreeChart chart = ChartFactory.createHistogram(plotTitle, xeks, yeks,dataset, or
    XYPlot plot = chart.getXYPlot();
    plot.getRenderer().setSeriesPaint(0, new Color(50, 128, 24));
    plot.setRangeCrosshairPaint(Color.BLACK);
    plot.setBackgroundPaint(Color.WHITE);
    ChartFrame chartfrm = new ChartFrame(" ", chart, true);
    chartfrm.setVisible(true);
    chartfrm.setSize(750, 600);
```

```
private void b_boxplotActionPerformed(java.awt.event.ActionEvent evt) {
    //QUALTİLS(Q1,Q2,Q3), MAX,MİN IQR , LİMİTS,
    ArrayList<Double> qualtines = new ArrayList<>();
    double q1,q2,q3 = 0.0;
    try {
        Connection cnn = DriverManager.getConnection("jdbc:derby://localhost
        Statement stmt = cnn.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT * FROM IRIS");
        while (rs.next()) {
            if (db.size() < 150) {
                db.add(rs.getDouble("SepalLengthCm"));} }
    } catch (SQLException e) {
        System.out.println(e);
    }
    int size = db.size();
    double[] sArr = new double[size];
    for (int i = 0; i < size; i++) {
        sArr[i] = db.get(i);
    }
    for (int i = 1; i <= 3; i++) {
        int sizel = sArr.length + 1;
        double q;
        double sizex = (sizel * ((double) (i) * 25 / 100)) - 1;
        Arrays.sort(sArr);
        if (sizex % 2 == 1) {
            q = sArr[(int) (sizex)];
        } else {
            int sizey = (int) (sizex);
            q = (sArr[sizey] + sArr[sizey + 1]) / 2;
        }
        qualtines.add(q);        }
    int size5 = qualtines.size();
```

```
int size5 = qualtines.size();
double[] s = new double[size5];
for (int i = 0; i < size5; i++) {
    s[i] = qualtines.get(i);}
q1 = s[0];
q2 = s[1];
q3 = s[2];
Arrays.sort(sArr);
int a = sArr.length;
double minData = sArr[0];
double maxData = sArr[a - 1];
double IQR = q3 - q1;
double altsinir = q1 - ((1.5) * (IQR));
double ustsinir = q3 + ((1.5) * (IQR));
System.out.println("altsinir=" + altsinir + "üstsınır" + ustsinir);
int syac = 0;
for (int i = 0; i < a; i++) {
    if (sArr[i] < altsinir || sArr[i] > ustsinir) {
        System.out.println("outliers: " + sArr[i]);
        syac++;
    } }
if (syac == 0) {
    System.out.println("there is no outliers ");
}
System.out.println("q1: " + q1);
System.out.println("q2: " + q1);
System.out.println("q3: " + q1);
System.out.println("IQR : " + IQR);
System.out.println("Sınırlar : [" + altsinir + "," + ustsinir + "]");
System.out.println("MaxData = " + maxData);
System.out.println("MinData = " + minData);
jTextPane7.setText(String.valueOf("IQR : " + IQR));
jTextPane8.setText(String.valueOf("[" + altsinir + "," + ustsinir + "]"));
jTextPane9.setText(String.valueOf("MaxData = " + maxData + "MinData = " + minData
```

```
public class cv extends javax.swing.JFrame {
    ArrayList<Double> db2 = new ArrayList<>();
    ArrayList<Double> db1 = new ArrayList<>();
    ArrayList<Double> sample = new ArrayList<>();
    DefaultListModel dlm = new DefaultListModel();
    int counter = 0;
    double SampleMean;
    double samplesd;
    double variance;
    static DefaultTableModel a = new DefaultTableModel();

    public cv() {
        initComponents();
        a.setColumnIdentifiers(new Object[]{"ID", "SAMPLE"});
        jTable1.setModel(a);
        jList1.setModel(dlm);

    }
```

```
private void confide_meanActionPerformed(java.awt.event.ActionEvent evt) {
    double std, meann = 0.0;//CONFİDE INTERVAL FOR MEAN
    int size2 = sample.size();
    if (size2 == 0) {JOptionPane.showMessageDialog(null, "Please, click the VIEW
    } else { int size = db2.size();
        double[] sArr = new double[size];
        for (int i = 0; i < size; i++) {
            sArr[i] = db2.get(i); }
        for (int i = 0; i < size; i++) {
            sArr[i] = db2.get(i);}
        double sum = 0;
        for (int i = 0; i < size; i++) {
            sum = sum + sArr[i];}
        meann = sum / size;
        double k = 0.0;
        for (int i = 0; i < size; i++) {
            k += Math.pow(sArr[i] - meann, 2);}std = Math.sqrt(k / size);
        System.out.println("STANDART DAVIATION = " + std);
        if (samplesd == 0) {
            samplesd = std; }
        double[] sampleArr = new double[size2];
        for (int i = 0; i < size2; i++) {
            sampleArr[i] = sample.get(i);}
        Arrays.sort(sampleArr);
        double sumSample = 0;
        for (int i = 0; i < sampleArr.length; i++) {
            sumSample = sumSample + sampleArr[i];}
        SampleMean = sumSample / size2;
        System.out.println("Sample Mean = " + SampleMean);
        double sizex = Math.sqrt(sampleArr.length);
        double cvu = SampleMean + (1.96) * (samplesd / sizex);
        double cva = SampleMean - (1.96) * (samplesd / sizex);
        dlm.removeAllElements();
        dlm.addElement(cva + "< μ <" + cvu); }
}
```

```
    private void confide_varianceActionPerformed(java.awt.event.ActionEvent evt) {
//CONFİDE INTERVAL FOR VARİANCE
        int sizeofsample = sample.size();
        if (sizeofsample == 0) {
            JOptionPane.showMessageDialog(null, "Please, click the VIEW DATA to create
        } else {
            double[] sampleArr = new double[sizeofsample];
            for (int i = 0; i < sizeofsample; i++) {
                sampleArr[i] = sample.get(i);}
            Arrays.sort(sampleArr);
            double totalSum = 0;
            for (int i = 0; i < sampleArr.length; i++) {
                totalSum = totalSum + sampleArr[i];            }
            SampleMean = totalSum / sizeofsample;
            System.out.println("mean : " + SampleMean);
            double samplesd = 0.0;
            double c = 0.0;
            for (int i = 0; i < sampleArr.length; i++) {
                c += (sampleArr[i] - SampleMean) * (sampleArr[i] - SampleMean);}
            samplesd = Math.sqrt(c / (sampleArr.length - 1));
            double samplevar = Math.pow(samplesd, 2);
            System.out.println("SAMPLE STD : " + samplesd);
            double sizex = Math.sqrt(sampleArr.length);
            double cvav = ((sampleArr.length - 1) * samplevar) / (59.3417);
            double cvuv = ((sampleArr.length - 1) * samplevar) / (24.4331);
            System.out.println("Sample Variance " + samplevar);
            System.out.println(cvav + " < σ2 < " + cvuv);
            dlm.removeAllElements();
            dlm.addElement(cvav + "< σ2 <" + cvuv);
        }
    }
```

```
    private void b_sizeActionPerformed(java.awt.event.ActionEvent evt) {
//How large size unit 0.1
        double std = 0.0 , meann = 0.0;
        try {
            Connection cnn = DriverManager.getConnection("jdbc:derby://localhost:1527/MERVE"
            Statement stmt = cnn.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT * FROM IRIS");
            while (rs.next()) {
                if (db.size() < 150) {
                    db.add(rs.getDouble("SepalLengthCm"));  } }
        } catch (SQLException e) {
            System.out.println(e);}
        int size = db.size();
        double[] sArr = new double[size];
        for (int i = 0; i < size; i++) {
            sArr[i] = db.get(i); }
        if (std == 0) {
            double sum = 0;
            for (int i = 0; i < size; i++) {
                sum = sum + sArr[i];    }
            meann = sum / size;
            for (int i = 0; i < size; i++) {
                sArr[i] = db.get(i);
            }
            double k = 0.0;
            for (int i = 0; i < size; i++) {
                k += Math.pow(sArr[i] - meann, 2); }
            std = Math.sqrt(k / size);   }
        System.out.println("STANDART DAVIATION = " + std);
        double EROR = 0.1;
        double Zdis = 1.645;
        int x = (int) (Math.pow(((Zdis * std) / EROR), 2));
        int a = x + 1;
        System.out.println("size=" + a);
        jTextPane10.setText(String.valueOf("it should be minimum = " + a));
```