INDEX

S.No.	Experiments	Date	Signature
1	Installing and understanding the KNIME workflow.		
2	Basic & Advanced Row and Column Filtering with KNIME.		
3	Interactive Univariate Visual Exploration with data exploration hub using KNIME.		
4	Interactive Bivariate Visual Exploration with a Scatter Plot using KNIME.		
5	Understanding and exploring composite views of datasets using KNIME.		
6	Introducing Exploratory Data Analysis (EDA) using SPARK.		
7	Using Spark SQL for basic data analysis e.g., Identifying Missing Data,computing basic statistics, Identifying data outliers etc.		
8	Visualizing data with Apache Zeppelin (web-based tool that supports interactive data analysis and visualization)		
9	VALUE ADDED EXPERIMENTS: Introduction to D3.js, basic principles. DEMO		
10	Basic Charting with D3: building bar chart, adding labels, building line chart, building scatter plots etc.		
11	Working with external data sources: CSV, JSON		
12	Working web data: Web APIs.		

Experiment- 1

AIM: Installing and Understanding the KNIME Workflow

THEORY:

To install KNIME and understand its workflow components and basic operations.

Steps:

1. Download KNIME Analytics Platform from the official website.

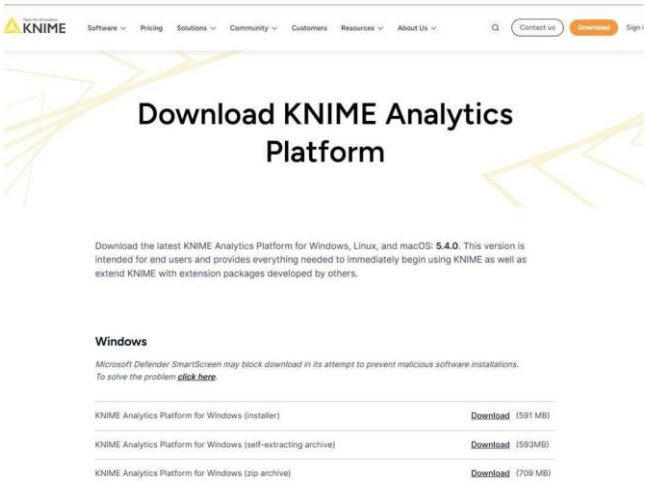


Figure 1: Download the Installation file (.exe)

2. Install the software following the on-screen instructions.

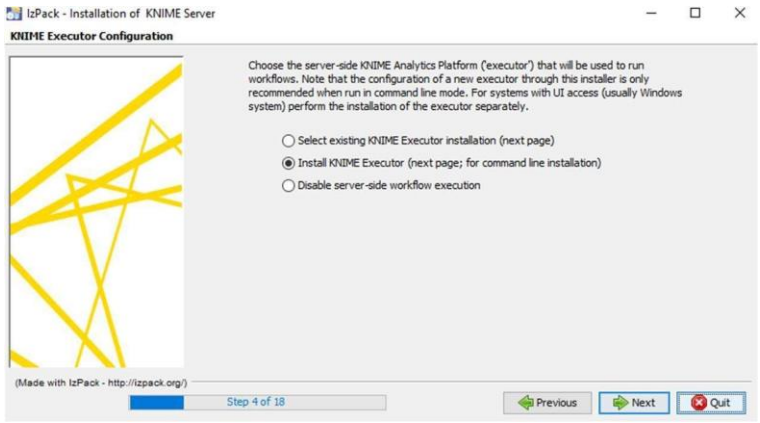


Figure 2: Steps while installing

3. Launch KNIME and explore the interface:
    a. KNIME Explorer: Manages workflows.
    b. Workflow Editor: Design and execute workflows.
    c. Node Repository: Contains various nodes for data processing.

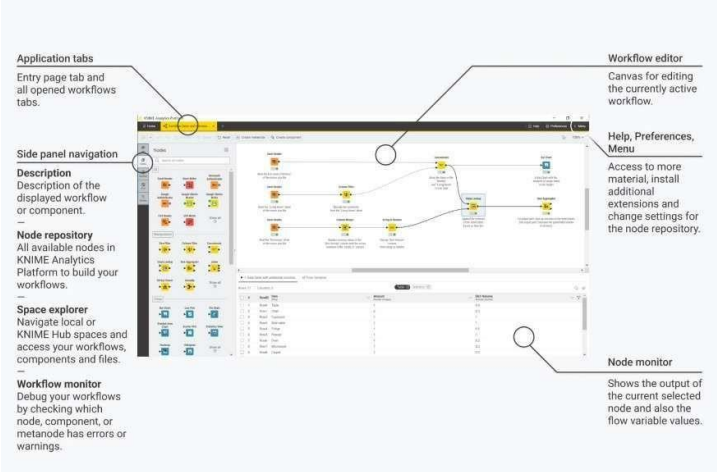


Figure 3: Summary of the working page of KNIME software

OUTPUT:

Hence, the KNIME software has been downloaded on the system.

Experiment- 2

AIM: Basic & Advanced Row and Column Filtering with KNIME

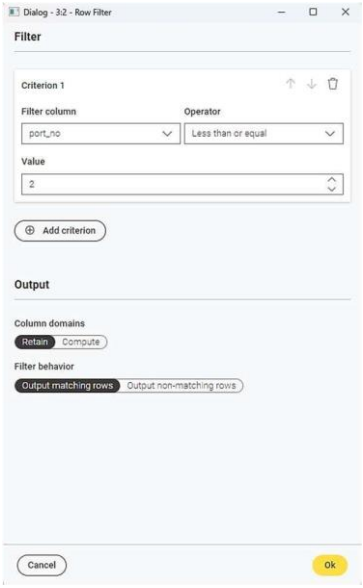
THEORY:

To perform row and column filtering using KNIME nodes.

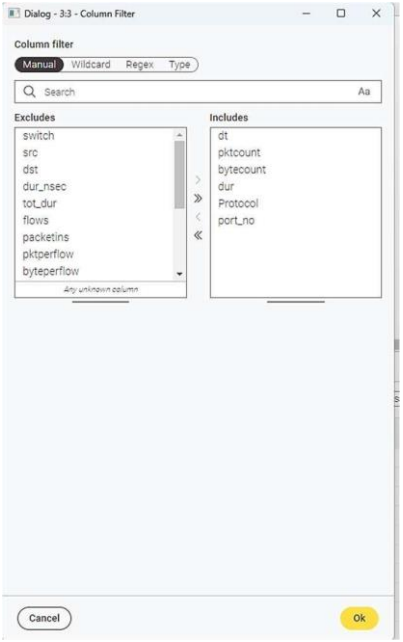
Steps:

Basic Filtering:

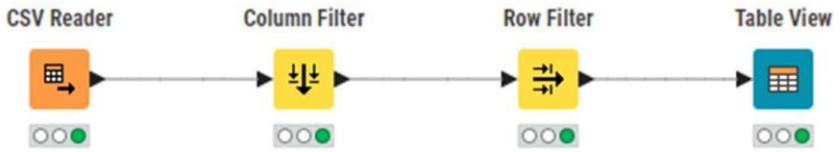
1. Load a dataset using the "CSV Reader" node.
2. Use the "Row Filter" node to filter rows based on a condition (e.g. port number less than or equal to 2).



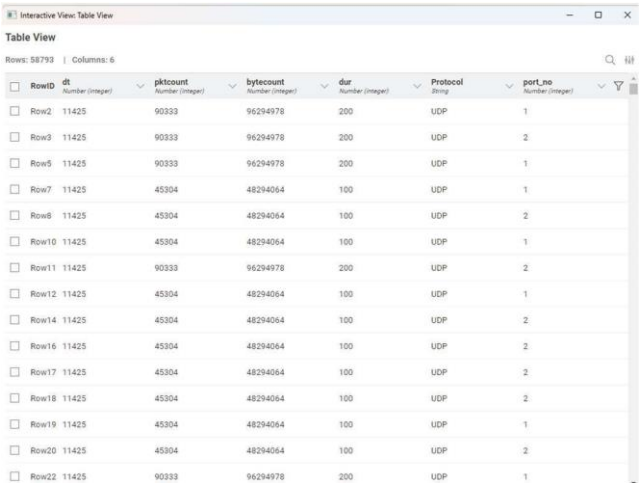
3. Use the "Column Filter" node to keep or remove specific columns.



4. Connect a "Table Viewer" node to visualize results.

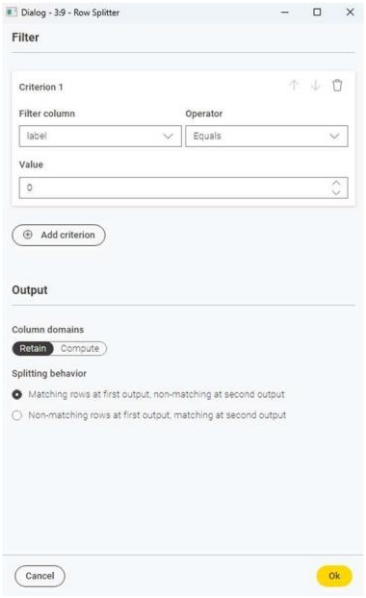


OUTPUT:



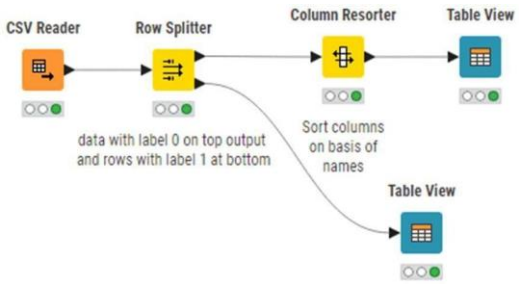
Advanced Filtering:

1. Use "Row Splitter" to divide data into subsets.



2. Use "Column Resorter" to rearrange columns.

3. Execute and compare the output with the original dataset.



OUTPUT:

Interactive View: Table View

Table View

Rows: 63561 | Columns: 23

RowID	bytecount	byteperfl...	dst	dt	dur	dur_nsec	flows	label	packetins
Row0	48294064	14428310	10.0.0.8	11425	100	716000000	3	0	1943
Row1	134737070	14424046	10.0.0.8	11605	280	734000000	2	0	1943
Row2	96294978	14427244	10.0.0.8	11425	200	744000000	3	0	1943
Row3	96294978	14427244	10.0.0.8	11425	200	744000000	3	0	1943
Row4	96294978	14427244	10.0.0.8	11425	200	744000000	3	0	1943
Row5	96294978	14427244	10.0.0.8	11425	200	744000000	3	0	1943
Row6	48294064	14428310	10.0.0.8	11425	100	716000000	3	0	1943
Row7	48294064	14428310	10.0.0.8	11425	100	716000000	3	0	1943
Row8	48294064	14428310	10.0.0.8	11425	100	716000000	3	0	1943
Row9	96294978	14427244	10.0.0.8	11425	200	744000000	3	0	1943
Row10	48294064	14428310	10.0.0.8	11425	100	716000000	3	0	1943
Row11	96294978	14427244	10.0.0.8	11425	200	744000000	3	0	1943
Row12	48294064	14428310	10.0.0.8	11425	100	716000000	3	0	1943
Row13	48294064	14428310	10.0.0.8	11425	100	716000000	3	0	1943

Experiment- 3

AIM: Interactive Univariate Visual Exploration with Data Exploration Hub using KNIME

THEORY: To explore univariate data visually using KNIME's Data Exploration Hub. Steps:

1. Load a dataset using the "CSV Reader" node.
2. Use the "Missing Value" node to handle missing data.
3. Apply the "Duplicate Row Filter" node to remove duplicate entries.
4. Use the "Bar Chart" node to visualize categorical data distributions.
5. Use the "Histogram" node to visualize numerical data distributions.
6. Execute the workflow and analyze results.

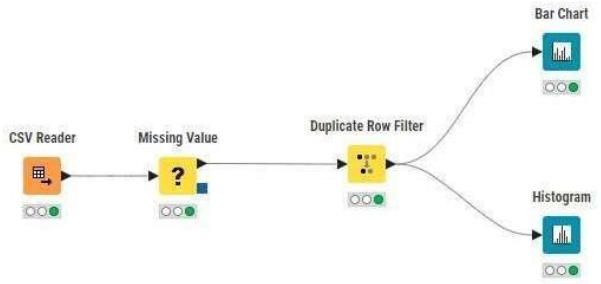


Figure 4: Workflow

OUTPUT:

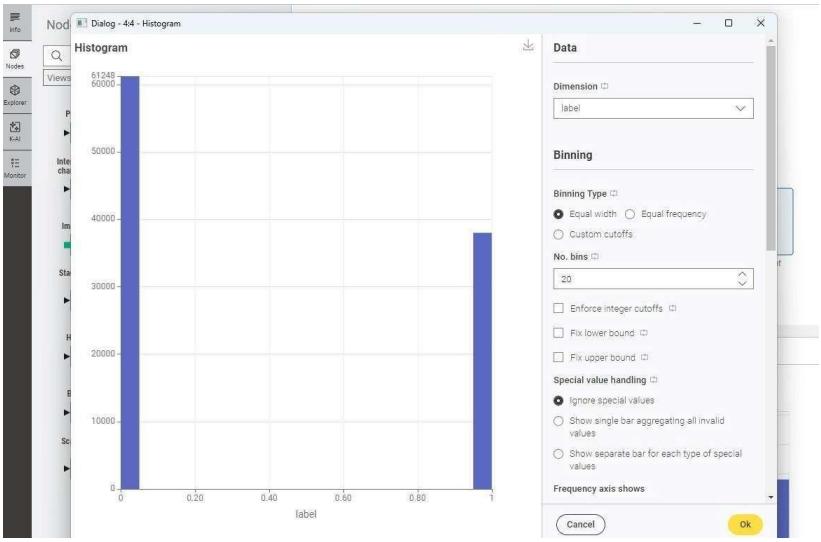


Figure 5: Bar Chart for binary classes

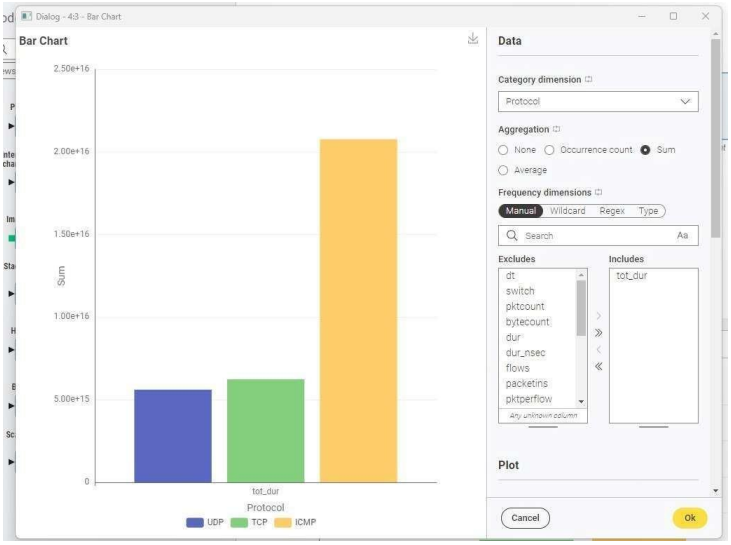


Figure 6: Histogram for Protocols

Experiment- 4

AIM: Interactive Bivariate Visual Exploration with Scatter Plot Hub using KNIME

THEORY:

To explore univariate data visually using KNIME's Data Exploration Hub.

Steps:

1. Load a dataset using the "CSV Reader" node.
2. Use the "Missing Value" node to handle missing data.
3. Apply the "Duplicate Row Filter" node to remove duplicate entries.
4. Use the "Scatter Plot" node to visualize ‘bytecount’ vs ‘pktcount’.
5. Use the "Scatter Plot" node to visualize ‘byteperflow’ vs ‘pktperflow’.
6. Execute the workflow and analyze results.

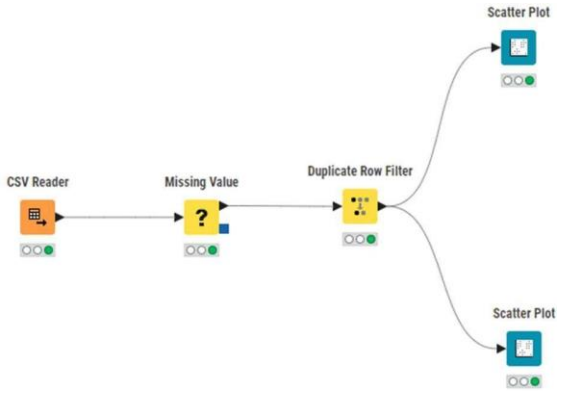


Figure1: Workflow

OUTPUT:

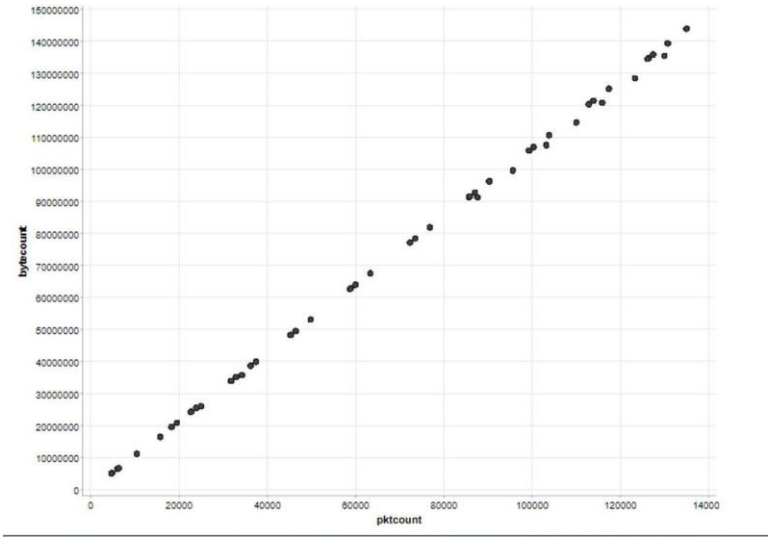


Figure 2: bytecount vs pktcount

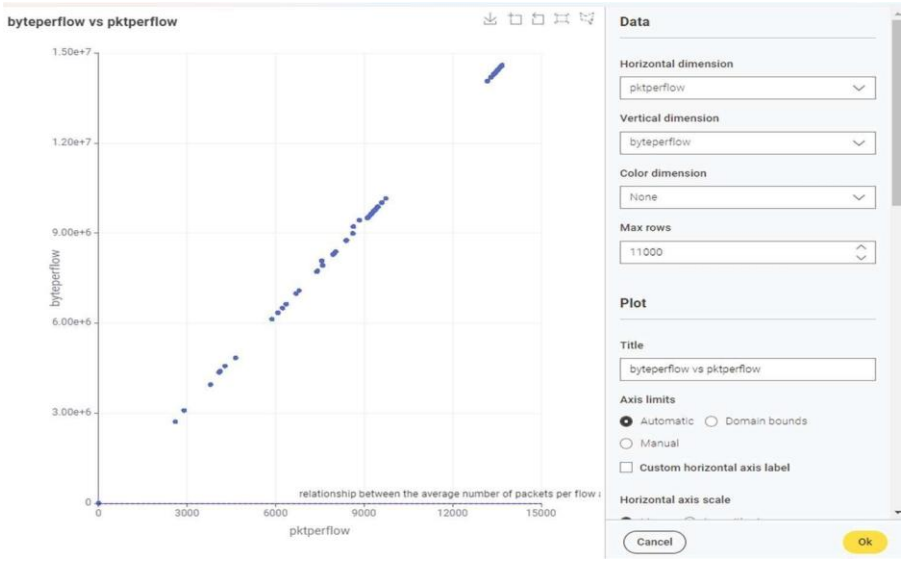


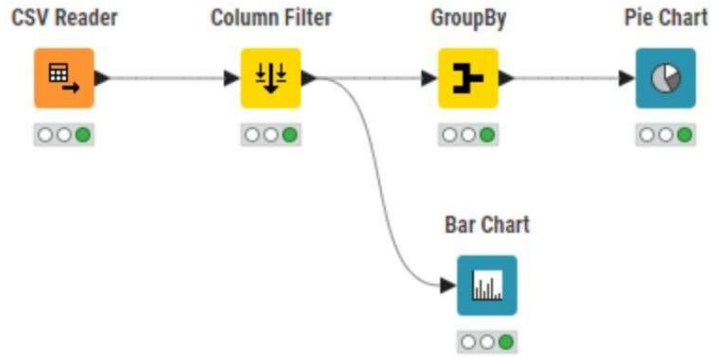
Figure 3: byterperflow vs pktperflow

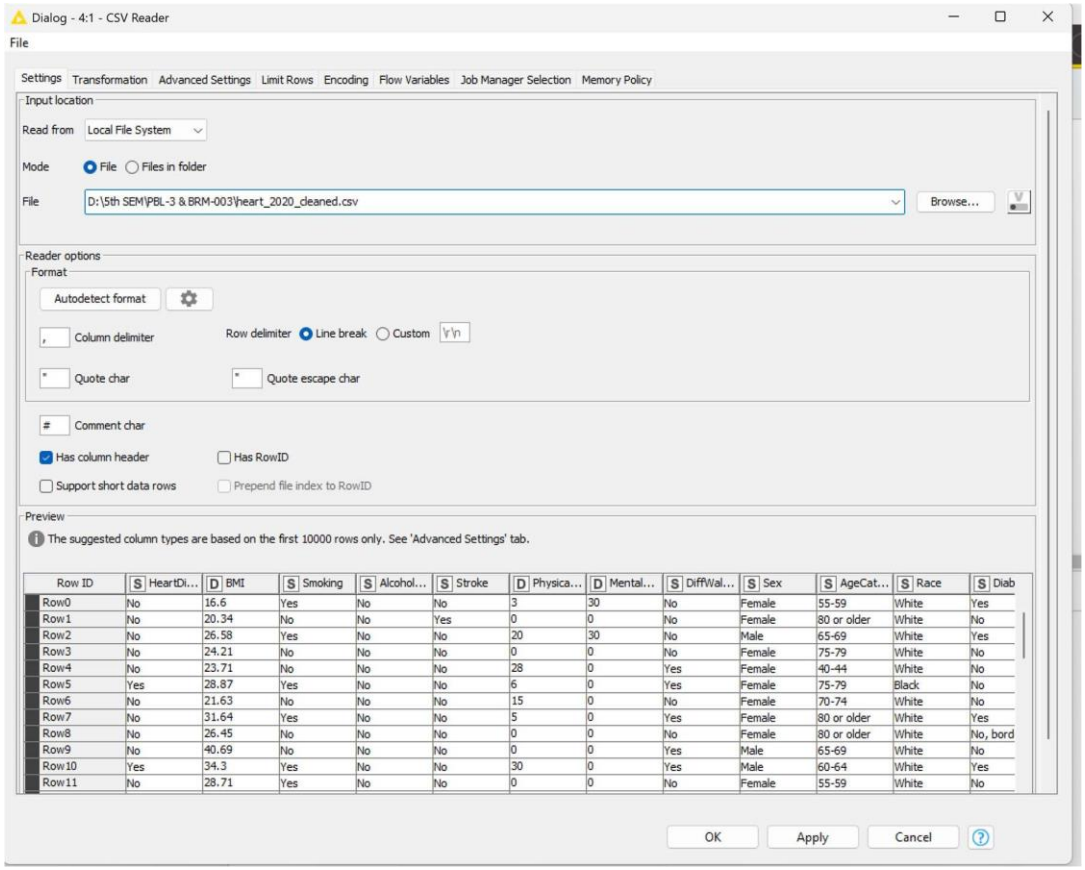
Experiment- 5

AIM: To understand and explore composite views of datasets using KNIME, focusing on different visualization and transformation techniques.

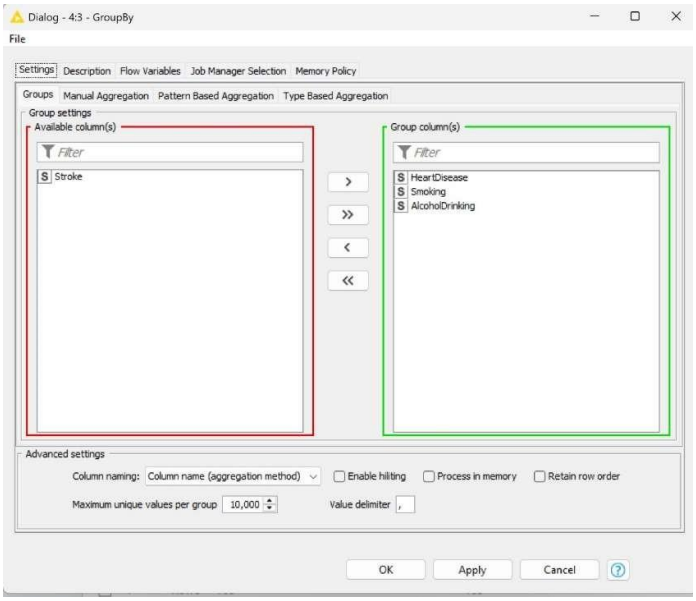
THEORY:

1. Load Dataset in KNIME
   - Open KNIME and create a new workflow.
   - Drag 'File Reader' node to import a dataset.



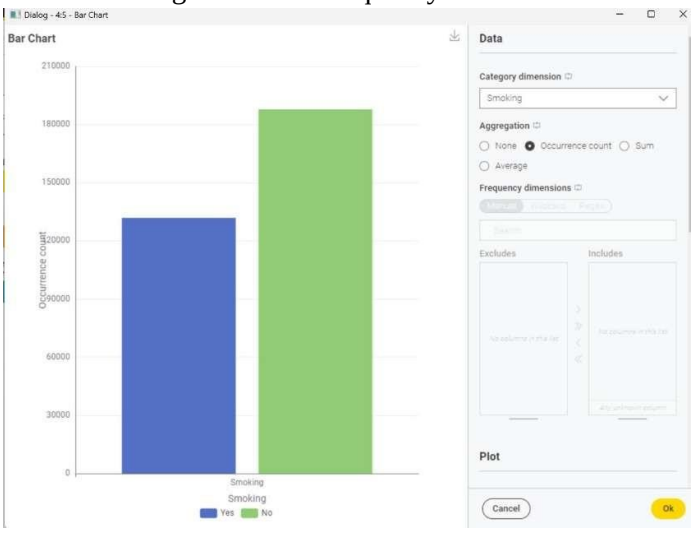


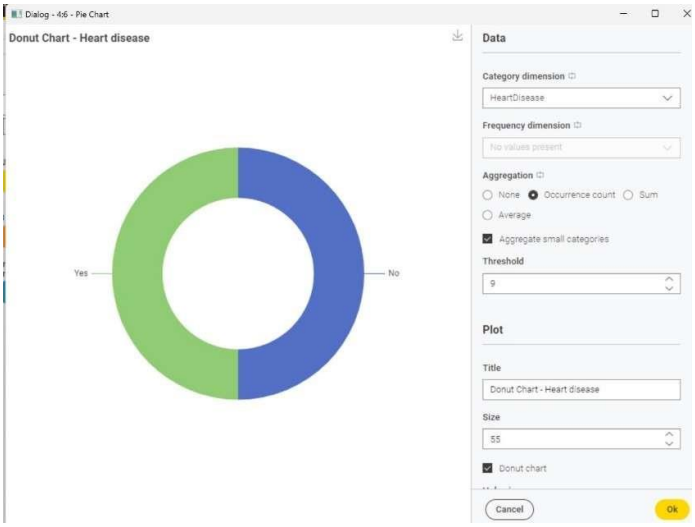
   - Connect the output to a 'Data Explorer' node to view basic statistics.
2. Exploring Data with Composite Views
   - Use 'Column Filter' to select specific columns.
   - Apply 'GroupBy' to aggregate and summarize data.



- Drag 'Pivoting' to transform the dataset into different views.

3. Visualizing the Data
   - Use 'Histogram' to see frequency distributions.





- 'Scatter Plot' to analyze relationships between variables.

- 'Bar Chart' for categorical data analysis.

4. Exporting Processed Data
   - Connect 'CSV Writer' to save transformed data.
   - Run and verify the output.

Experiment- 6

AIM: To perform Exploratory Data Analysis (EDA) using Apache Spark to summarize, visualize, and clean data.

THEORY:

1. Setting Up Spark

```
[1]  from pyspark.sql import SparkSession
     spark = SparkSession.builder.appName("EDA_Experiment").getOrCreate()
```

2. Loading Dataset into Spark

```
[8]  df = spark.read.csv("/content/Titanic-Dataset.csv", header=True, inferSchema=True)
     df.show(5)  # Display first 5 rows
```

```
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+-----+--------+
|PassengerId|Survived|Pclass|                Name|   Sex| Age|SibSp|Parch|          Ticket|   Fare|Cabin|Embarked|
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+-----+--------+
|          1|       0|     3|Braund, Mr. Owen ...|  male|22.0|    1|    0|       A/5 21171|   7.25| NULL|       S|
|          2|       1|     1|Cumings, Mrs. Joh...|female|38.0|    1|    0|        PC 17599|71.2833|  C85|       C|
|          3|       1|     3|Heikkinen, Miss. ...|female|26.0|    0|    0|STON/O2. 3101282|  7.925| NULL|       S|
|          4|       1|     1|Futrelle, Mrs. Ja...|female|35.0|    1|    0|          113803|   53.1| C123|       S|
|          5|       0|     3|Allen, Mr. Willia...|  male|35.0|    0|    0|          373450|   8.05| NULL|       S|
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+-----+--------+
only showing top 5 rows
```

3. Understanding Dataset Structure

```
[9] df.printSchema()  # Display column data types
    df.describe().show()  # Summary statistics
```

```
root
 |-- PassengerId: integer (nullable = true)
 |-- Survived: integer (nullable = true)
 |-- Pclass: integer (nullable = true)
 |-- Name: string (nullable = true)
 |-- Sex: string (nullable = true)
 |-- Age: double (nullable = true)
 |-- SibSp: integer (nullable = true)
 |-- Parch: integer (nullable = true)
 |-- Ticket: string (nullable = true)
 |-- Fare: double (nullable = true)
 |-- Cabin: string (nullable = true)
 |-- Embarked: string (nullable = true)

+-------+------------------+--------------------+------------------+--------------------+------+------------------+------------------+--------------------+-----------------+-----------------+-----+--------+
|summary|       PassengerId|            Survived|            Pclass|                Name|   Sex|               Age|             SibSp|               Parch|           Ticket|             Fare|Cabin|Embarked|
+-------+------------------+--------------------+------------------+--------------------+------+------------------+------------------+--------------------+-----------------+-----------------+-----+--------+
|  count|               891|                 891|               891|                 891|   891|               714|               891|                 891|              891|              891|  204|     889|
|   mean|             446.0|  0.3838383838383838|2.3086419753086642|                NULL|  NULL|29.699117647058832|0.5230078563411896|0.38159371492704824|260318.54916792738|32.204207968574604| NULL|    NULL|
| stddev|257.35384201523014|  0.4865924542648575|0.8360712409770491|                NULL|  NULL|14.526497332334035|1.1027434322934315| 0.8060572211299488|471609.26868834975|49.693428597180899| NULL|    NULL|
|    min|                 1|                   0|                 1|"Andersson, Mr. A...|female|              0.42|                 0|                   0|           110152|              0.0|  A10|       C|
|    max|               891|                   1|                 3|van Melkebeke, Mr...|  male|              80.0|                 8|                   6|           WE/P 5735|         512.3292|    T|       S|
+-------+------------------+--------------------+------------------+--------------------+------+------------------+------------------+--------------------+-----------------+-----------------+-----+--------+
```

4. Handling Missing Values

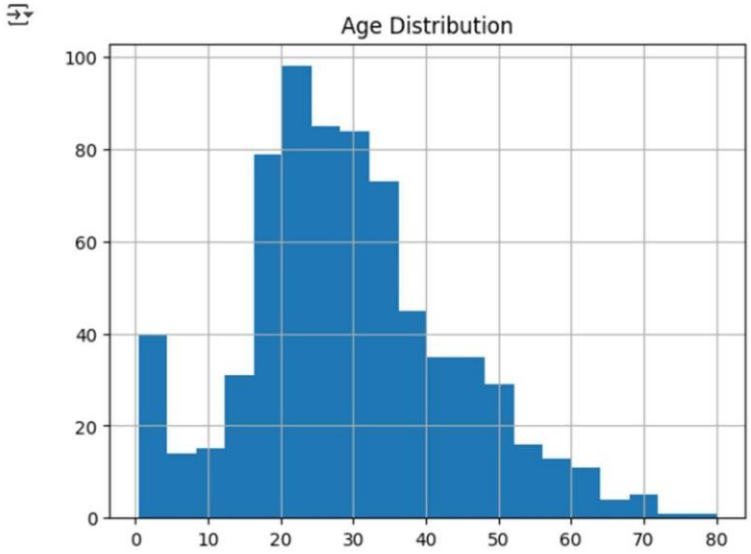
```
[10] df.na.drop().show()  # Dropping missing values
```

```
+-----------+--------+------+--------------------+------+----+-----+-----+----------+-------+-----------+--------+
|PassengerId|Survived|Pclass|                Name|   Sex| Age|SibSp|Parch|    Ticket|   Fare|      Cabin|Embarked|
+-----------+--------+------+--------------------+------+----+-----+-----+----------+-------+-----------+--------+
|          2|       1|     1|Cumings, Mrs. Joh...|female|38.0|    1|    0|  PC 17599|71.2833|        C85|       C|
|          4|       1|     1|Futrelle, Mrs. Ja...|female|35.0|    1|    0|    113803|   53.1|       C123|       S|
|          7|       0|     1|McCarthy, Mr. Tim...|  male|54.0|    0|    0|     17463|51.8625|        E46|       S|
|         11|       1|     3|Sandstrom, Miss. ...|female| 4.0|    1|    1|   PP 9549|   16.7|         G6|       S|
|         12|       1|     1|Bonnell, Miss. El...|female|58.0|    0|    0|    113783|  26.55|       C103|       S|
|         22|       1|     2|Beesley, Mr. Lawr...|  male|34.0|    0|    0|    248698|   13.0|        D56|       S|
|         24|       1|     1|Sloper, Mr. Willi...|  male|28.0|    0|    0|    113788|   35.5|         A6|       S|
|         28|       0|     1|Fortune, Mr. Char...|  male|19.0|    3|    2|     19950|263.0|C23 C25 C27|       S|
|         53|       1|     1|Harper, Mrs. Henr...|female|49.0|    1|    0|  PC 17572|76.7292|        D33|       C|
|         55|       0|     1|Ostby, Mr. Engelh...|  male|65.0|    0|    1|    113509|61.9792|        B30|       C|
|         63|       0|     1|Harris, Mr. Henry...|  male|45.0|    1|    0|     36973|83.475|        C83|       S|
|         67|       1|     2|Nye, Mrs. (Elizab...|female|29.0|    0|    0| C.A. 29395|   10.5|        F33|       S|
|         76|       0|     3|Moen, Mr. Sigurd ...|  male|25.0|    0|    0|    348123|   7.65|      F G73|       S|
|         89|       1|     1|Fortune, Miss. Ma...|female|23.0|    3|    2|     19950|263.0|C23 C25 C27|       S|
|         93|       0|     1|Chaffee, Mr. Herb...|  male|46.0|    1|    0|W.E.P. 5734|61.175|        E31|       S|
|         97|       0|     1|Goldschmidt, Mr. ...|  male|71.0|    0|    0|  PC 17754|34.6542|         A5|       C|
|         98|       1|     1|Greenfield, Mr. W...|  male|23.0|    0|    1|  PC 17759|63.3583|    D10 D12|       C|
|        103|       0|     1|White, Mr. Richar...|  male|21.0|    0|    1|     35281|77.2875|        D26|       S|
|        111|       0|     1|Porter, Mr. Walte...|  male|47.0|    0|    0|    110465|   52.0|       C110|       S|
|        119|       0|     1|Baxter, Mr. Quigg...|  male|24.0|    0|    1|  PC 17558|247.5208|    B58 B60|       C|
+-----------+--------+------+--------------------+------+----+-----+-----+----------+-------+-----------+--------+
only showing top 20 rows
```

5. Performing Basic Visualization (Using Pandas & Matplotlib)

```
[11] import pandas as pd
     import matplotlib.pyplot as plt

     pdf = df.toPandas()  # Convert Spark DataFrame to Pandas DataFrame
     pdf['Age'].hist(bins=20)
     plt.title("Age Distribution")
     plt.show()
```



6. Finding Correlations

```
[12] from pyspark.sql.functions import col
     df.select([col(column).cast("float") for column in df.columns]).summary().show()
```

```
+-------+-----------+------------------+------------------+----+----+------------------+------------------+------------------+------------------+-----------------+----+--------+
|summary|PassengerId|          Survived|            Pclass|Name| Sex|               Age|             SibSp|             Parch|            Ticket|             Fare|Cabin|Embarked|
+-------+-----------+------------------+------------------+----+----+------------------+------------------+------------------+------------------+-----------------+----+--------+
|  count|        891|               891|               891|   0|   0|               714|               891|               891|               661|              891|   0|       0|
|   mean|      446.0|0.3838383838383838| 2.308641975308642|NULL|NULL| 29.69911764704046|0.5230078563411896|0.3815937149270482|260318.54916792738|32.20420804114722|NULL|    NULL|
| stddev|257.3538420152301|0.4865924542648575|0.8360712409770491|NULL|NULL|14.526497332370992|1.1027434322934315|0.8060572211299488|471609.26868834975|49.69342916316158|NULL|    NULL|
|    min|        1.0|               0.0|               1.0|NULL|NULL|              0.42|               0.0|               0.0|             693.0|              0.0|NULL|    NULL|
|    25%|      223.0|               0.0|               2.0|NULL|NULL|              20.0|               0.0|               0.0|           19996.0|           7.8958|NULL|    NULL|
|    50%|      446.0|               0.0|               3.0|NULL|NULL|              28.0|               0.0|               0.0|          236171.0|          14.4542|NULL|    NULL|
|    75%|      669.0|               1.0|               3.0|NULL|NULL|              38.0|               1.0|               0.0|          347743.0|             31.0|NULL|    NULL|
|    max|      891.0|               1.0|               3.0|NULL|NULL|              80.0|               8.0|               6.0|         3101298.0|         512.3292|NULL|    NULL|
+-------+-----------+------------------+------------------+----+----+------------------+------------------+------------------+------------------+-----------------+----+--------+
```

Experiment 7

Objective: Using spark SQL for basic data analysis e.g. identifying missing data, computing basic statistics, identifying data outliers etc.

1. Initialize Spark Session: Set up the Spark environment to run SQL queries.

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, mean, stddev, count, when

# Initialize Spark Session
spark = SparkSession.builder.appName("SparkSQLAnalysis").getOrCreate()
```

2. Load Data: Read the Titanic dataset as a Spark DataFrame.

```
df = spark.read.csv("/content/Titanic-Dataset.csv", header=True, inferSchema=True)
df.show(5)
```

Python

```
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+-----+----
|PassengerId|Survived|Pclass|                Name|   Sex| Age|SibSp|Parch|          Ticket|   Fare|Cabin|Emba
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+-----+----
|          1|       0|     3|Braund, Mr. Owen ...|  male|22.0|    1|    0|       A/5 21171|   7.25| NULL|
|          2|       1|     1|Cumings, Mrs. Joh...|female|38.0|    1|    0|        PC 17599|71.2833|  C85|
|          3|       1|     3|Heikkinen, Miss. ...|female|26.0|    0|    0|STON/O2. 3101282|  7.925| NULL|
|          4|       1|     1|Futrelle, Mrs. Ja...|female|35.0|    1|    0|          113803|   53.1| C123|
|          5|       0|     3|Allen, Mr. Willia...|  male|35.0|    0|    0|          373450|   8.05| NULL|
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+-----+----
only showing top 5 rows
```

3. Missing Data: Check for missing values in each column.

```
df.select([count(when(col(c).isNull(), c)).alias(c) for c in df.columns]).show()
```

Python

```
+-----------+--------+------+----+---+---+-----+-----+------+----+-----+--------+
|PassengerId|Survived|Pclass|Name|Sex|Age|SibSp|Parch|Ticket|Fare|Cabin|Embarked|
+-----------+--------+------+----+---+---+-----+-----+------+----+-----+--------+
|          0|       0|     0|   0|  0|177|    0|    0|     0|   0|  687|       2|
+-----------+--------+------+----+---+---+-----+-----+------+----+-----+--------+
```

4. Basic Statistics: Generate summary statistics like count, mean, standard deviation, etc.

```
df.describe().show()
```

Python

```
+-------+-----------------+-------------------+------------------+--------------------+------+----------------
|summary|      PassengerId|           Survived|            Pclass|                Name|   Sex|               A
+-------+-----------------+-------------------+------------------+--------------------+------+----------------
|  count|              891|                891|               891|                 891|   891|               7
|   mean|            446.0| 0.3838383838383838| 2.308641975308642|                NULL|  NULL| 29.699117647058
| stddev|257.3538420152301|0.48659245426485753|0.8360712409770491|                NULL|  NULL|14.526497332334
|    min|                1|                  0|                 1|"Andersson, Mr. A...|female|              0.
|    max|              891|                  1|                 3|van Melkebeke, Mr...|  male|              8
+-------+-----------------+-------------------+------------------+--------------------+------+----------------
```

5. Outlier Detection: Identify outliers in the Age column using the Z-score method (values beyond 3 standard deviations from the mean).

```
stats_df = df.select("Age").summary("mean", "stddev")
stats_df.show()

# Convert mean and stddev values from string to float
mean_value = float(stats_df.collect()[0][1])
stddev_value = float(stats_df.collect()[1][1])

outliers_df = df.filter((col("Age") > mean_value + 3 * stddev_value) | (col("Age") < mean_value - 3 * stddev_value))
outliers_df.show()
```

Python

```
+-------+------------------+
|summary|               Age|
+-------+------------------+
|   mean| 29.69911764705882|
| stddev|14.526497332334035|
+-------+------------------+

+-----------+--------+------+--------------------+----+----+-----+-----+------+-----+-----+--------+
|PassengerId|Survived|Pclass|                Name| Sex| Age|SibSp|Parch|Ticket|  Fare|Cabin|Embarked|
+-----------+--------+------+--------------------+----+----+-----+-----+------+-----+-----+--------+
|        631|       1|     1|Barkworth, Mr. Al...|male|80.0|    0|    0| 27042| 30.0|  A23|       S|
|        852|       0|     3| Svensson, Mr. Johan|male|74.0|    0|    0|347060|7.775| NULL|       S|
+-----------+--------+------+--------------------+----+----+-----+-----+------+-----+-----+--------+
```