



**K.R. MANGALAM UNIVERSITY**  
THE COMPLETE WORLD OF EDUCATION

# **Data Analysis with Power BI & KNIME**

## **(ETSEDA115)**

**MCA (AI & ML)- Sem 1**

## **Assignment 2**

**Roll No:-**

**2501940072**

**Submitted by:-**

**KULSUM BANO**

**Submitted to:-**

**MR. MOHAMMAD AIJAZ**

- 1) Read the adult.csv file available in the **data** folder on the KNIME Hub. The data are provided by the **UCI Machine Learning Repository**.
- 2) Calculate the average age and count for each one of the 4 groups defined by sex and income values
- 3) Join the two aggregated values to the original table

## 1) Read the adult.csv file

The screenshot shows the KNIME workspace with a 'CSV Reader' node connected to a 'Joiner' node. The 'CSV Reader' node is configured to read the 'adult.csv' file. The 'Joiner' node is configured to join the output of the 'CSV Reader' node with itself. The resulting data table is displayed below the nodes.

**CSV Reader**

Reads CSV files. To auto-guess the structure of the file click the Autodetect format button. If you encounter problems with incorrect guessed data types disable the Limit data rows scanned option in the Advanced Settings tab. If the input file structure changes between different invocations, enable the Support changing file schemas option in the Advanced Settings tab. For further details see the KNIME File Handling Guide [File Handling Guide](#).

**Note:** If you find that this node can't read your file, try the **File Reader** node. It offers more options for reading complex files.

This node can access a variety of different **file systems**. More information about file handling in KNIME can be found in the official [File Handling Guide](#).

**Parallel reading:** Individual files can be read in parallel if

- They are located on the machine that is running this node.
- They don't contain any quotes that contain row delimiters.
- They are not gzip compressed.
- No lines or rows are limited or skipped.
- The file index is not prepended to the RowID.
- They are not encoded with UTF-16 (UTF-16LE and UTF-16BE are fine).

**Ports** **Options** **Views**

**Output ports**

**1: File Table** **Flow Variables**

Rows: 32561 | Columns: 15

#	RowID	age	workclass	fnlwgt	education	education...	marital-st...	occupation	relations...	race	sex
1	Row0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male
2	Row1	50	Self-emp-not-in	83311	Bachelors	13	Married-civ-spo	Exec-managerial	Husband	White	Male
3	Row2	38	Private	215646	HS-grad	9	Divorced	Handlers-clean	Not-in-family	White	Male
4	Row3	53	Private	234721	11th	7	Married-civ-spo	Handlers-clean	Husband	Black	Male
5	Row4	28	Private	338409	Bachelors	13	Married-civ-spo	Prof-specialty	Wife	Black	Female
6	Row5	37	Private	284582	Masters	14	Married-civ-spo	Exec-managerial	Wife	White	Female
7	Row6	49	Private	160187	9th	5	Married-spouse	Other-service	Not-in-family	Black	Female
8	Row7	52	Self-emp-not-in	209642	HS-grad	9	Married-civ-spo	Exec-managerial	Husband	White	Male
9	Row8	31	Private	45781	Masters	14	Never-married	Prof-specialty	Not-in-family	White	Female
10	Row9	42	Private	159449	Bachelors	13	Married-civ-spo	Exec-managerial	Husband	White	Male

## 2) Calculate the average age and count for each one of the 4 groups defined by sex and income values

The screenshot shows the KNIME workspace with a 'GroupBy' node connected to a 'Joiner' node. The 'GroupBy' node is configured to calculate the average age and count for each group defined by sex and income. The resulting aggregated table is displayed below the nodes.

**GroupBy**

Groups the rows of a table by the unique values in the selected group columns. A row is created for each unique set of values of the selected group column. The remaining columns are aggregated based on the specified aggregation settings. The output table contains one row for each unique value combination of the selected group columns.

The columns to aggregate can be either defined by selecting the columns directly, by name based on a search pattern or based on the data type. Input columns are handled in this order and only considered once e.g. columns that are added directly on the 'Manual Aggregation' tab are ignored even if their name matches a search pattern on the 'Pattern Based Aggregation' tab or their type matches a defined type on the 'Type Based Aggregation' tab. The same holds for columns that are added based on a search pattern. They are ignored even if they match a criterion that has been defined in the 'Type Based Aggregation' tab.

The 'Manual Aggregation' tab allows you to change the aggregation method of more than one column. In order to do so select the columns to change, open the context menu with a right mouse click and select the aggregation method to use.

In the 'Pattern Based Aggregation' tab you can assign aggregation methods to columns based on a search pattern. The pattern can be either a string with wildcards or a **regular expression**. Columns where the name matches the pattern but where the data type is not compatible with the selected aggregation method are ignored. Only columns that have not been selected as group column or that have not been selected as aggregation column on the 'Manual Aggregation' tab are considered.

**CSV Reader**

Reads CSV files. To auto-guess the structure of the file click the Autodetect format button. If you encounter problems with incorrect guessed data types disable the Limit data rows scanned option in the Advanced Settings tab. If the input file structure changes between different invocations, enable the Support changing file schemas option in the Advanced Settings tab. For further details see the KNIME File Handling Guide [File Handling Guide](#).

**Note:** If you find that this node can't read your file, try the **File Reader** node. It offers more options for reading complex files.

This node can access a variety of different **file systems**. More information about file handling in KNIME can be found in the official [File Handling Guide](#).

**Parallel reading:** Individual files can be read in parallel if

- They are located on the machine that is running this node.
- They don't contain any quotes that contain row delimiters.
- They are not gzip compressed.
- No lines or rows are limited or skipped.
- The file index is not prepended to the RowID.
- They are not encoded with UTF-16 (UTF-16LE and UTF-16BE are fine).

**Ports** **Options** **Views**

**Output ports**

**1: Group table** **Flow Variables**

Rows: 4 | Columns: 4

#	RowID	sex	income	Mean(age)	Count(age)
1	Row0	Female	<=50K	36.211	9592
2	Row1	Female	>50K	42.126	1179
3	Row2	Male	<=50K	37.147	15128
4	Row3	Male	>50K	44.626	6662

### 3) Join the two aggregated values to the original value

The screenshot shows the KNIME software interface with the Joiner node configured. The workflow includes a CSV Reader, a GroupBy node, and the Joiner node. The Joiner node is set to 'All of the following' matching criteria. The output table displays the result of the join operation.

**Joiner Configuration:**

- Matching Criteria: All of the following
- Compare values in join columns by: Value and type

**Output Table:**

sex	capital-g...	capital-lo...	hours-per...	native-co...	Income	sex (Right)	income (...)	Mean(age)	Count*(a...	
T: String	Number (L...	Number (L...	Number (L...	T: String	T: String	T: String	T: String	Number (L...	Number (L...	
te	Male	2174	0	40	United-States	<=50K	Female	<=50K	36.211	9592
te	Male	0	0	13	United-States	<=50K	Female	>50K	42.126	1179
te	Male	0	0	40	United-States	<=50K	Male	<=50K	37.147	15128
sk	Male	0	0	40	United-States	<=50K	Male	>50K	44.626	6662