

# 1 Power BI Assignment 1

- 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 count and average age of women with income >50K
- 3) Calculate the averages of all numerical columns for each one of the 4 groups defined by sex and income values
- 4) Calculate
  - the number of missing values in the occupation column
  - the number of non-missing rows in the occupation column
  - the number of rows in the occupation column
  - the number of rows in the marital-status column

Notice that the last two aggregations should provide the same numbers!

## 1) Read the adult.csv file

The screenshot shows a KNIME workflow titled "1: File Table". The workflow consists of the following steps:

- A "CSV Reader" node is connected to a "Row Filter" node.
- The output of the "Row Filter" node is connected to three "GroupBy" nodes.
- The outputs of the three "GroupBy" nodes are connected to a fourth "GroupBy" node.

On the right side of the interface, there is a panel for the "CSV Reader" node. It displays the message: "This node dialog is not supported here." and has a "Open dialog" button.

Below the workflow, a table titled "1: File Table" is displayed. It contains 32561 rows and 15 columns. The columns are: #, RowID, age, workclass, fnwgt, education, educationn, marital-st., occupation, relations..., race, sex, capital-g, capital-lo, hours-per..., and capital-wk. The data includes various demographic and socioeconomic information for individuals.

## 2) A) Filter Female and Income >50k using Row Filter

## 2 Power BI Assignment 1

The screenshot shows the KNIME Data Wrangler interface. A workflow is built with the following steps:

- CSV Reader**: Loads data from a CSV file.
- Row Filter**: Filters rows where the sex column is "Female".
- GroupBy**: Groups the filtered data by the income column (`capital-gain`).
- GroupBy**: Groups the data by the education level column (`education`).
- GroupBy**: Groups the data by the occupation column (`occupation`).

The resulting table has 1179 rows and 15 columns. The data includes columns like RowID, age, workclass, fnlwgt, education, marital-status, occupation, relations, race, sex, capital-gain, capital-loss, and hours-per-week.

- 2) B) Calculate the Count and Average age of women with income >50k

The screenshot shows the KNIME Data Wrangler interface. A workflow is built with the following steps:

- CSV Reader**: Loads data from a CSV file.
- Row Filter**: Filters rows where the sex column is "Female".
- GroupBy**: Groups the filtered data by the income column (`capital-gain`).
- Add comment GroupBy**: Adds a comment to the GroupBy node.
- GroupBy**: Groups the data by the education level column (`education`).

A message box on the right says: "This node dialog is not supported here." with a "Open dialog" button.

The resulting table has 1 row and 2 columns. The data includes columns RowID and Count\*(age). The mean age is 42.126.

- 3) Calculate the averages of all numerical columns for each one of the 4 groups defined by sex and income value

### 3 Power BI Assignment 1

The screenshot shows the KNIME interface with a workflow titled "Local - tutorial". The workflow consists of a "CSV Reader" node connected to three "GroupBy" nodes. The first "GroupBy" node has a "Row Filter" node preceding it. The second and third "GroupBy" nodes also have "Row Filter" nodes preceding them. The output of the first "GroupBy" node is a table with 4 rows and 7 columns, showing mean values for age, education, capital-gain, capital-loss, hours-per-week, and hours-per-month across four categories (RowID 1-4). The second and third "GroupBy" nodes produce empty tables with 1 row and 3 columns each, labeled "Missing value count(occupation)" and "Count(marital-status)" respectively.

#	RowID	sex	income	Mean(age)	Mean(education)	Mean(capital-gain)	Mean(capital-loss)	Mean(hours-per-week)
1	Row0	Female	<=50K	36.211	9.82	121.986	47.364	35.917
2	Row1	Female	>50K	42.126	11.787	4,200.389	173.649	40.427
3	Row2	Male	<=50K	37.147	9.452	165.724	56.807	40.694
4	Row3	Male	>50K	44.626	11.581	3,971.766	198.78	46.366

#	RowID	Missing value count(occupation)	Count(occupation)	Count(marital-status)
1	Row0	0	32561	32561

#### 4) Calculate:

- the number of **missing values** in the *occupation* column
- the number of **non-missing rows** in the *occupation* column
- the **number of rows** in the *occupation* column
- the **number of rows** in the *marital-status* column

This screenshot shows the same KNIME workflow as the previous one, but with different data in the tables. The first "GroupBy" node now produces a table with 1 row and 3 columns, showing the count of missing values in the occupation column. The second "GroupBy" node produces a table with 1 row and 2 columns, showing the count of rows in the occupation column. The third "GroupBy" node produces a table with 1 row and 2 columns, showing the count of rows in the marital-status column.

#	RowID	Missing value count(occupation)	Count(occupation)	Count(marital-status)
1	Row0	0	32561	32561