Tutorial: Use Azure Machine Learning Workbench for advanced data preparation (Bike share data)

6/13/2018 • 24 minutes to read • Edit Online

Azure Machine Learning (preview) is an integrated, end-to-end data science and advanced analytics solution for professional data scientists to prepare data, develop experiments, and deploy models at cloud scale.

In this tutorial, you use Machine Learning (preview) to learn how to:

- Prepare data interactively with the Machine Learning data preparation tool.
- Import, transform, and create a test dataset.
- Generate a data preparation package.
- Run the data preparation package by using Python.
- Generate a training dataset by reusing the data preparation package for additional input files.
- Execute scripts in a local Azure CLI window.
- Execute scripts in a cloud Azure HDInsight environment.

If you don't have an Azure subscription, create a free account before you begin.

Prerequisites

- A local installation of Azure Machine Learning Workbench. For more information, follow the installation quickstart.
- If you don't have the Azure CLI installed, follow the instructions to install the latest Azure CLI version.
- An HDInsights Spark cluster created in Azure.
- An Azure storage account.
- Familiarity with how to create a new project in Workbench.
- Although it's not required, it's helpful to have Azure Storage Explorer installed so that you can upload, download, and view the blobs in your storage account.

Data acquisition

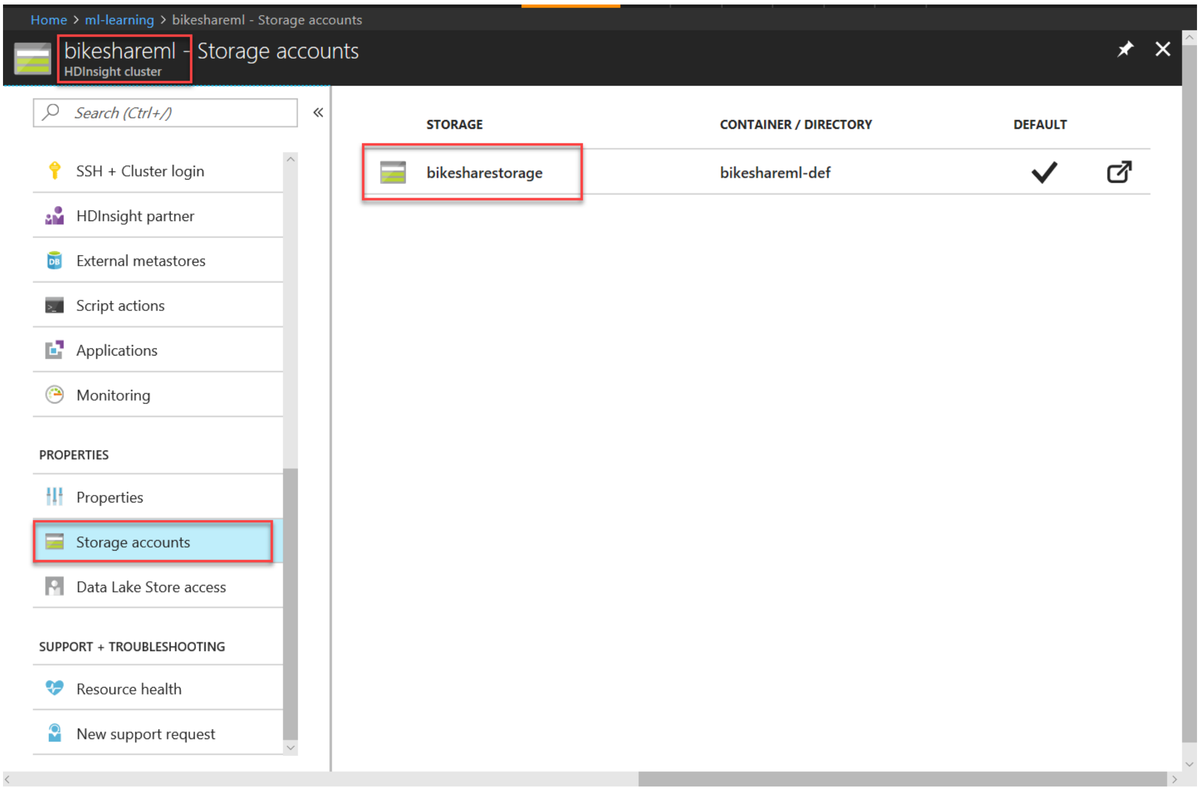
This tutorial uses the Boston hubway dataset and Boston weather data from NOAA.

1. Download the data files from the following links to your local development environment:
    - Boston weather data
    - Hubway trip data from the hubway website:
        - 201501-hubway-tripdata.zip
        - 201504-hubway-tripdata.zip
        - 201510-hubway-tripdata.zip
        - 201601-hubway-tripdata.zip
        - 201604-hubway-tripdata.zip
        - 201610-hubway-tripdata.zip
        - 201701-hubway-tripdata.zip
2. Unzip each .zip file after download.

Upload data files to Azure Blob storage

You can use Azure Blob storage to host your data files.

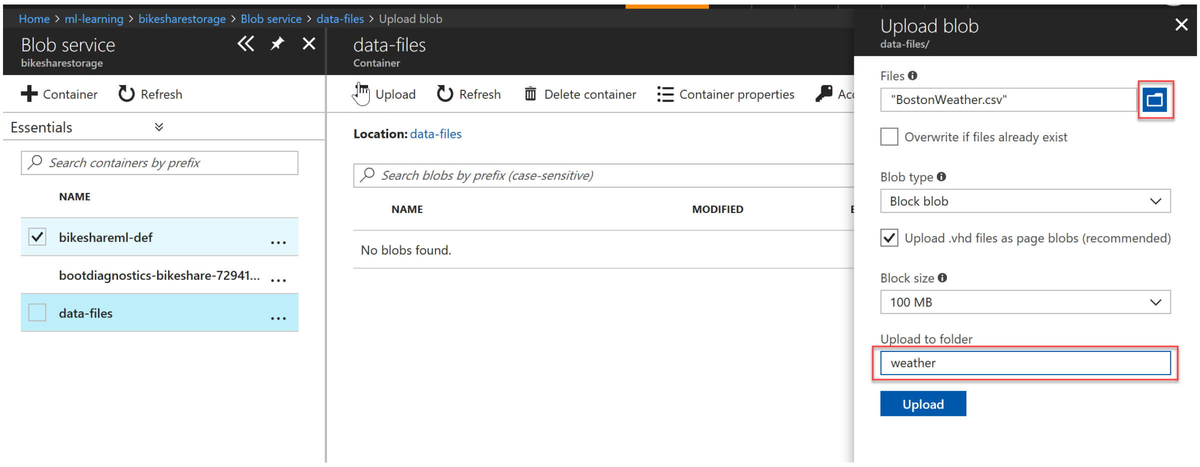
1. Use the same storage account that is used for the HDInsight cluster you use.



STORAGE	CONTAINER / DIRECTORY	DEFAULT
bikesharestorage	bikeshareml-def	✓

2. Create a new container named **data-files** to store the **BikeShare** data files.

3. Upload the data files. Upload `BostonWeather.csv` to a folder named `weather`. Upload the trip data files to a folder named `tripdata`.



Files: "BostonWeather.csv"

□ Overwrite if files already exist

Blob type: Block blob

☑ Upload .vhd files as page blobs (recommended)

Block size: 100 MB

Upload to folder: weather

> **TIP**
>
> You also can use Storage Explorer to upload blobs. Use this tool when you want to view the contents of any files generated in the tutorial, too.

Learn about the datasets

1. The **Boston weather** file contains the following weather-related fields, reported on an hourly basis:

- **DATE**

   - **REPORTTPYE**
   - **HOURLYDRYBULBTEMPF**
   - **HOURLYRelativeHumidity**
   - **HOURLYWindSpeed**

2. The **hubway** data is organized into files by year and month. For example, the file named `201501-hubway-tripdata.zip` contains a .csv file that contains data for January 2015. The data contains the following fields, with each row representing a bike trip:

   - **Trip Duration (in seconds)**
   - **Start Time and Date**
   - **Stop Time and Date**
   - **Start Station Name & ID**
   - **End Station Name & ID**
   - **Bike ID**
   - **User Type (Casual = 24-Hour or 72-Hour Pass user; Member = Annual or Monthly Member)**
   - **ZIP Code (if user is a member)**
   - **Gender (self-reported by member)**

Create a new project

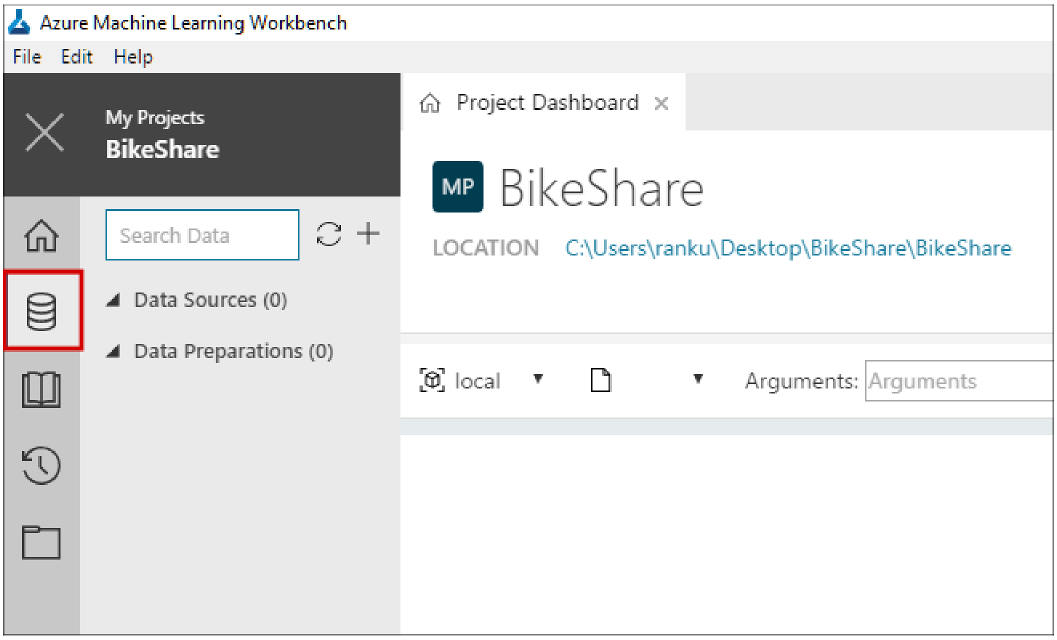
1. Start **Machine Learning Workbench** from your Start menu or launcher.

2. Create a new Machine Learning project. Select the **+** button on the **Projects** page, or select **File** > **New**.

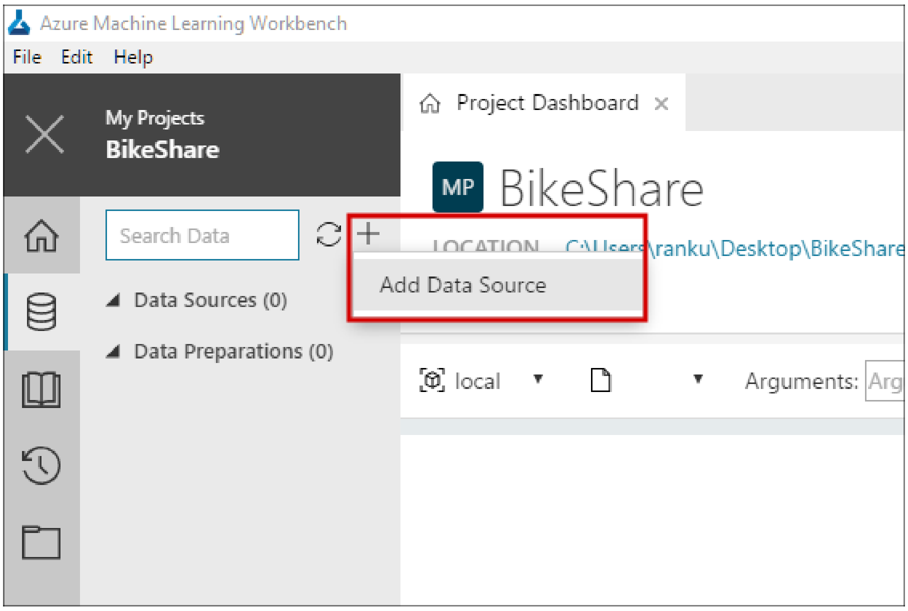
   - Use the **Bike Share** template.
   - Name your project **BikeShare**.

Create a new data source

1. Create a new data source. Select the **Data** button (cylinder icon) on the left toolbar to display the **Data** view.

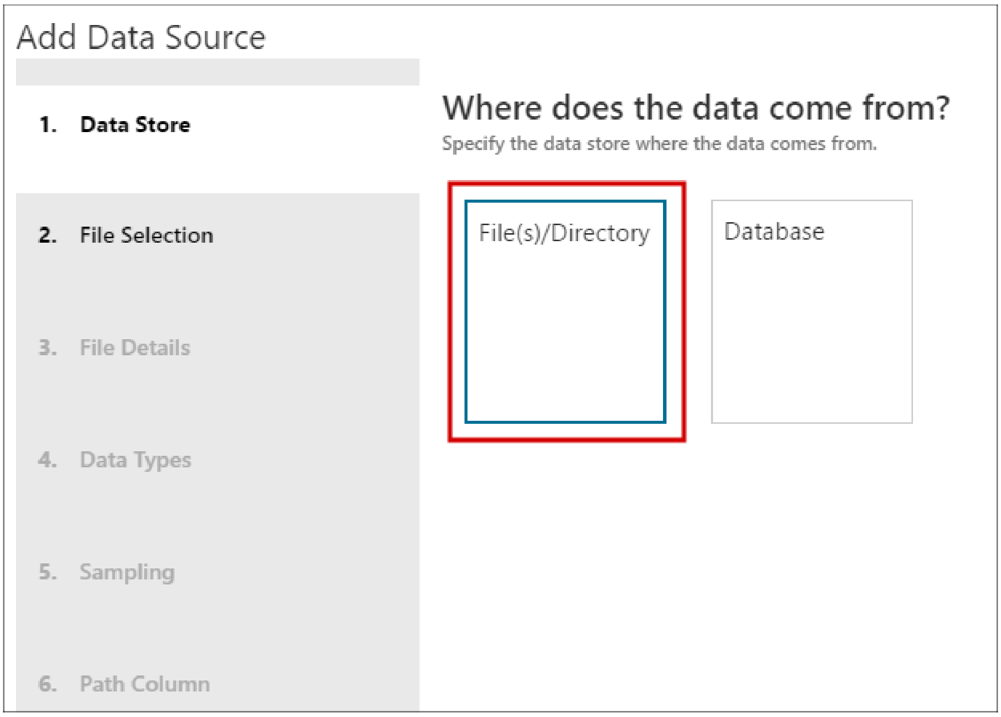


2. Add a data source. Select the **+** icon, and then select **Add Data Source**.



Add weather data

1. **Data Store**: Select the data store that contains the data. Because you're using files, select **File(s)/Directory**. Select **Next** to continue.



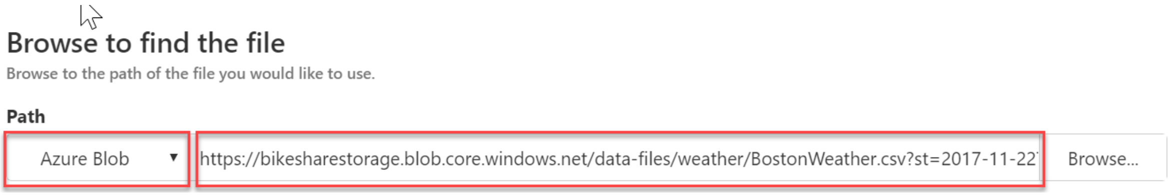
Add Data Source

1. Data Store
2. File Selection
3. File Details
4. Data Types
5. Sampling
6. Path Column

Where does the data come from?

Specify the data store where the data comes from.

2. **File Selection**: Add the weather data. Browse and select the `BostonWeather.csv` file that you uploaded to Blob Storage earlier. Select **Next**.

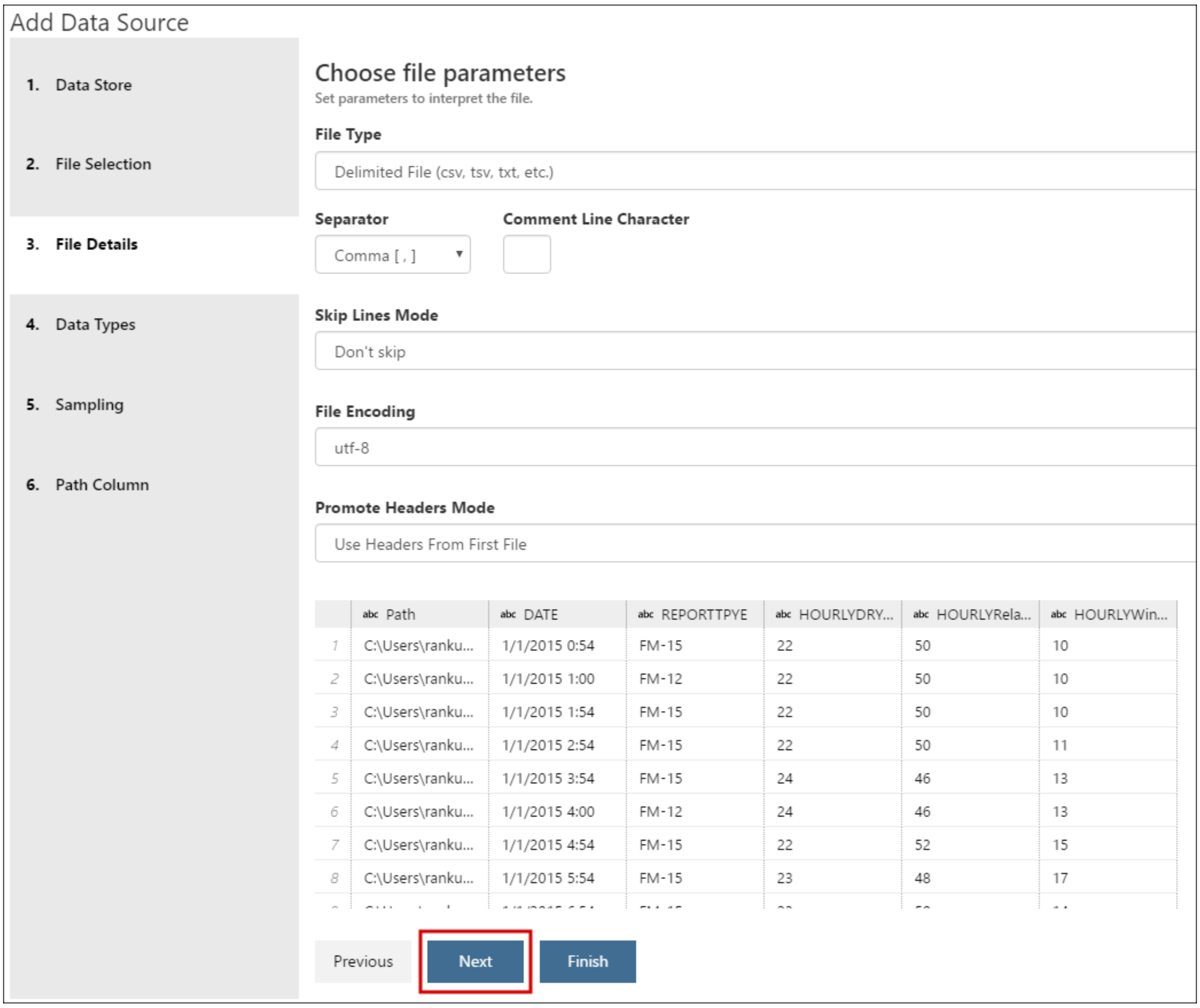


Browse to find the file

Browse to the path of the file you would like to use.

Path: Azure Blob ▾ https://bikesharestorage.blob.core.windows.net/data-files/weather/BostonWeather.csv?st=2017-11-22

3. **File Details**: Verify the file schema that is detected. Machine Learning Workbench analyzes the data in the file and infers the schema to use.



Add Data Source

1. Data Store
2. File Selection
3. **File Details**
4. Data Types
5. Sampling
6. Path Column

Choose file parameters

Set parameters to interpret the file.

File Type

Delimited File (csv, tsv, txt, etc.)

Separator

Comma [,]

Comment Line Character

Skip Lines Mode

Don't skip

File Encoding

utf-8

Promote Headers Mode

Use Headers From First File

	abc Path	abc DATE	abc REPORTTPYE	abc HOURLYDRY...	abc HOURLYRela...	abc HOURLYWin...
1	C:\Users\ranku...	1/1/2015 0:54	FM-15	22	50	10
2	C:\Users\ranku...	1/1/2015 1:00	FM-12	22	50	10
3	C:\Users\ranku...	1/1/2015 1:54	FM-15	22	50	10
4	C:\Users\ranku...	1/1/2015 2:54	FM-15	22	50	11
5	C:\Users\ranku...	1/1/2015 3:54	FM-15	24	46	13
6	C:\Users\ranku...	1/1/2015 4:00	FM-12	24	46	13
7	C:\Users\ranku...	1/1/2015 4:54	FM-15	22	52	15
8	C:\Users\ranku...	1/1/2015 5:54	FM-15	23	48	17

IMPORTANT

Workbench might not detect the correct schema in some cases. Always verify that the parameters are correct for your data set. For the weather data, verify that they are set to the following values:

- **File Type**: Delimited File (csv, tsv, txt, etc.)
- **Separator**: Comma [,]
- **Comment Line Character**: No value is set.
- **Skip Lines Mode**: Don't skip
- **File Encoding**: utf-8
- **Promote Headers Mode**: Use Headers From First File

The preview of the data should display the following columns:

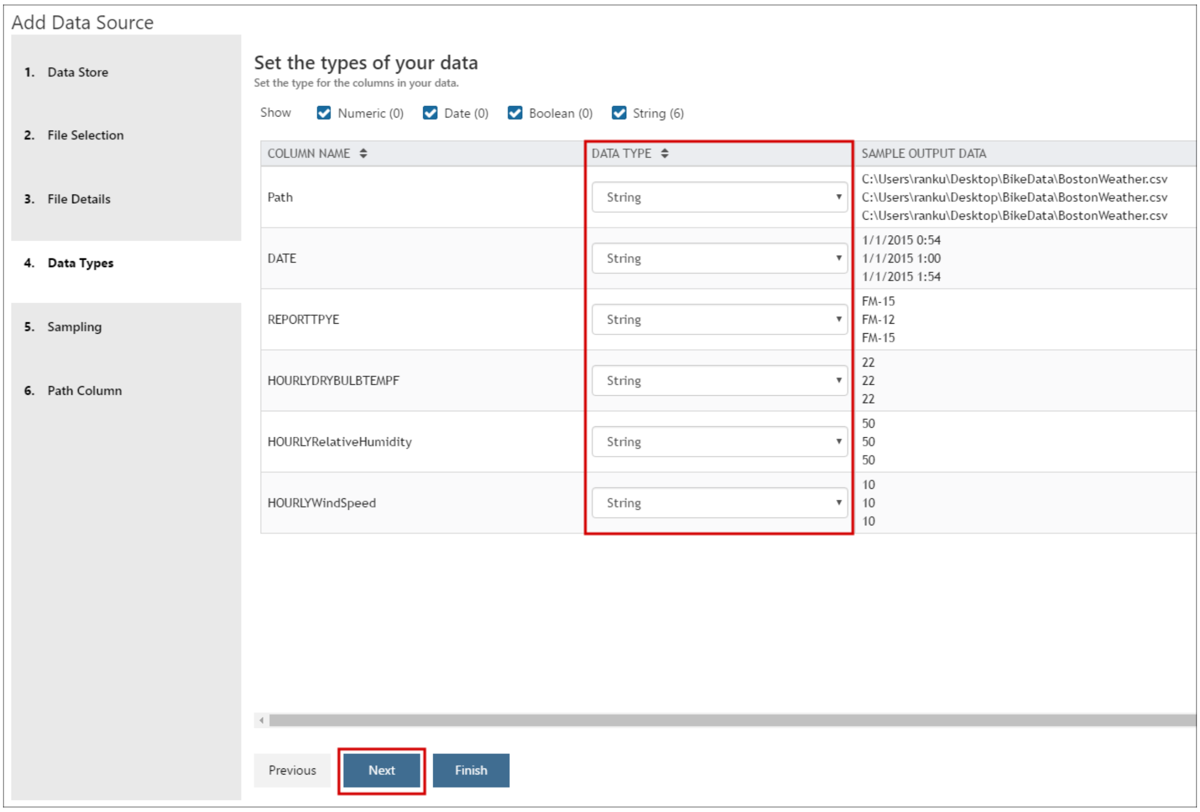
- **Path**
- **DATE**
- **REPORTTYPE**
- **HOURLYDRYBULBTEMPF**
- **HOURLYRelativeHumidity**
- **HOURLYWindSpeed**

To continue, select **Next**.

4. **Data Types**: Review the data types that are detected automatically. Machine Learning Workbench analyzes the data in the file and infers the data types to use.

a. For this data, change **DATA TYPE** for all the columns to **String**.

> **NOTE**
>
> String is used to highlight the capabilities of Workbench later in this tutorial.



Add Data Source

1. Data Store
2. File Selection
3. File Details
4. **Data Types**
5. Sampling
6. Path Column

Set the types of your data

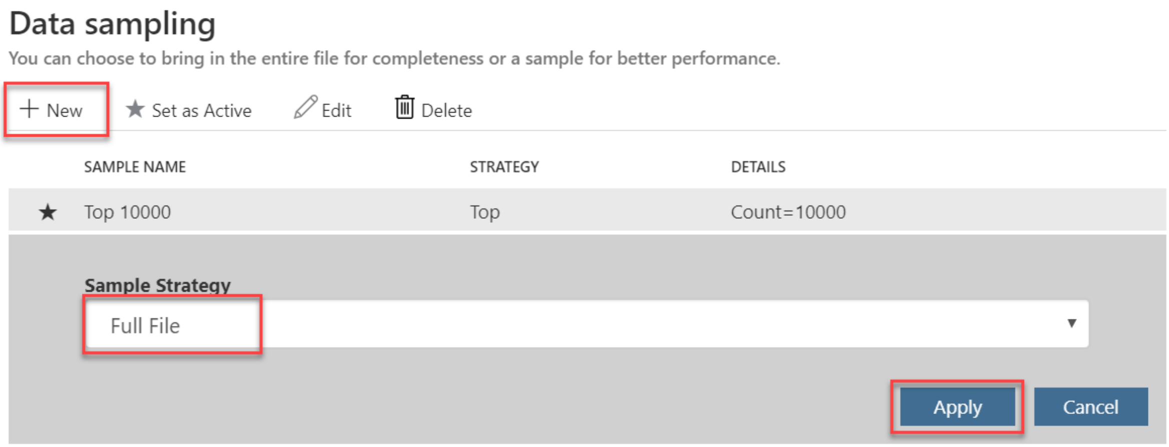
Set the type for the columns in your data.

Show ☑ Numeric (0) ☑ Date (0) ☑ Boolean (0) ☑ String (6)

COLUMN NAME	DATA TYPE	SAMPLE OUTPUT DATA
Path	String	C:\Users\ranku\Desktop\BikeData\BostonWeather.csv C:\Users\ranku\Desktop\BikeData\BostonWeather.csv C:\Users\ranku\Desktop\BikeData\BostonWeather.csv
DATE	String	1/1/2015 0:54 1/1/2015 1:00 1/1/2015 1:54
REPORTTPYE	String	FM-15 FM-12 FM-15
HOURLYDRYBULBTEMPF	String	22 22 22
HOURLYRelativeHumidity	String	50 50 50
HOURLYWindSpeed	String	10 10 10

b. To continue, select **Next**.

5. **Sampling**: To create a sampling scheme, select **Edit**. Select the new **Top 10000** row that is added, and then select **Edit**. Set **Sample Strategy** to **Full File**, and then select **Apply**.



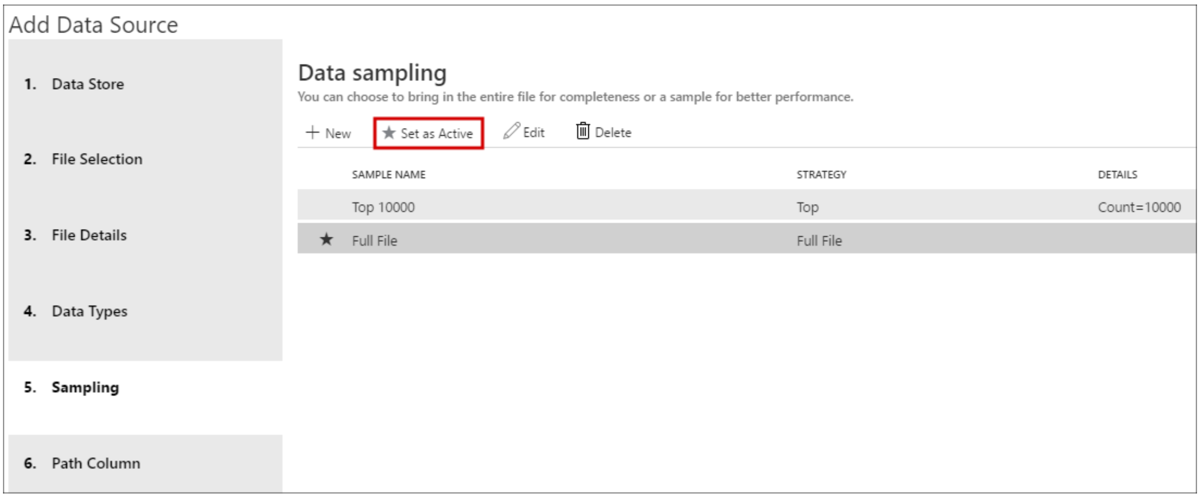
Data sampling

You can choose to bring in the entire file for completeness or a sample for better performance.

	SAMPLE NAME	STRATEGY	DETAILS
★	Top 10000	Top	Count=10000

Sample Strategy: Full File

To use the **Full File** strategy, select the **Full File** entry, and then select **Set as Active**. A star appears next to **Full File** to indicate that it's the active strategy.



Add Data Source

1. Data Store
2. File Selection
3. File Details
4. Data Types
5. Sampling
6. Path Column

Data sampling

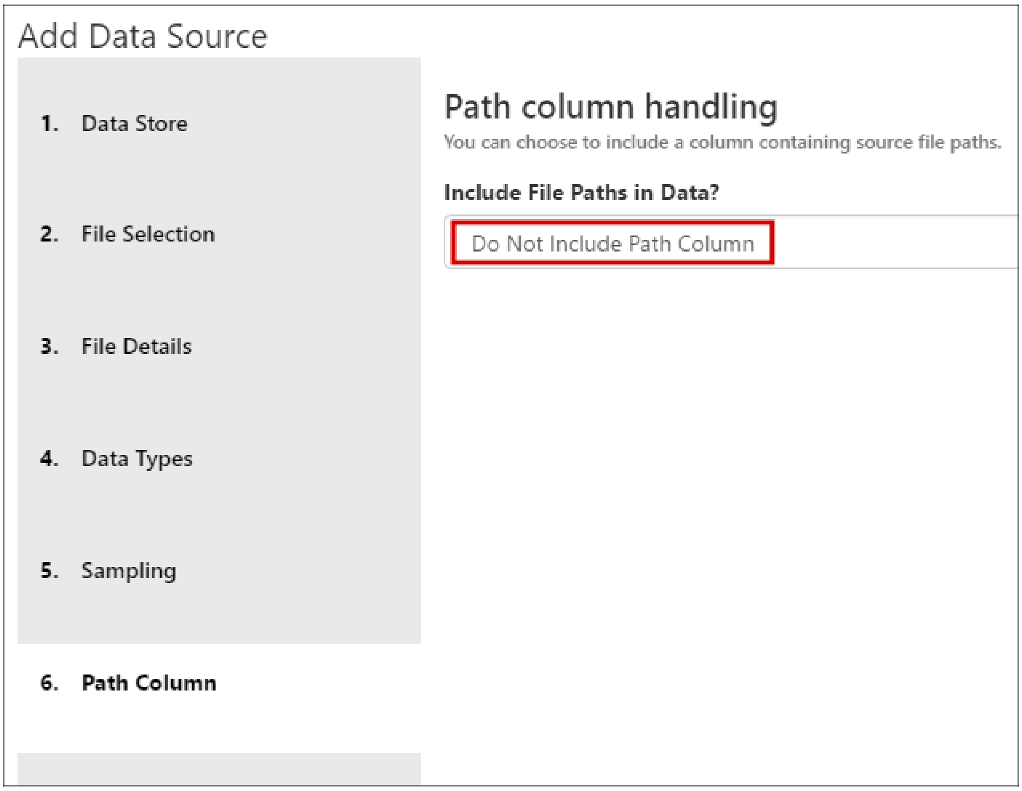
You can choose to bring in the entire file for completeness or a sample for better performance.

	SAMPLE NAME	STRATEGY	DETAILS
	Top 10000	Top	Count=10000
★	Full File	Full File	

To continue, select **Next**.

6. **Path Column**: Use the **Path Column** section to include the full file path as a column in the imported data. Select **Do Not Include Path Column**.

> **TIP**
>
> Including the path as a column is useful if you're importing a folder of many files with different file names. It's also useful if the file names contain information that you want to extract later.



Add Data Source

1. Data Store
2. File Selection
3. File Details
4. Data Types
5. Sampling
6. Path Column

Path column handling

You can choose to include a column containing source file paths.

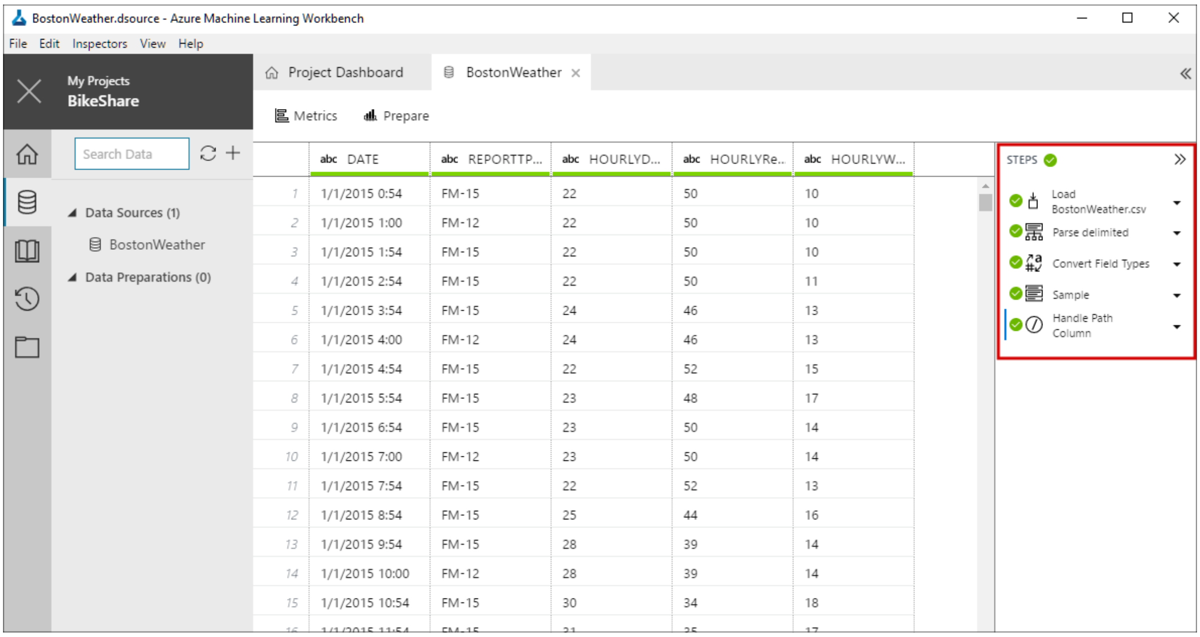
Include File Paths in Data?

Do Not Include Path Column

7. **Finish**: To finish creating the data source, select **Finish**.

A new data source tab named **BostonWeather** opens. A sample of the data is displayed in a grid view. The sample is based on the active sampling scheme specified previously.

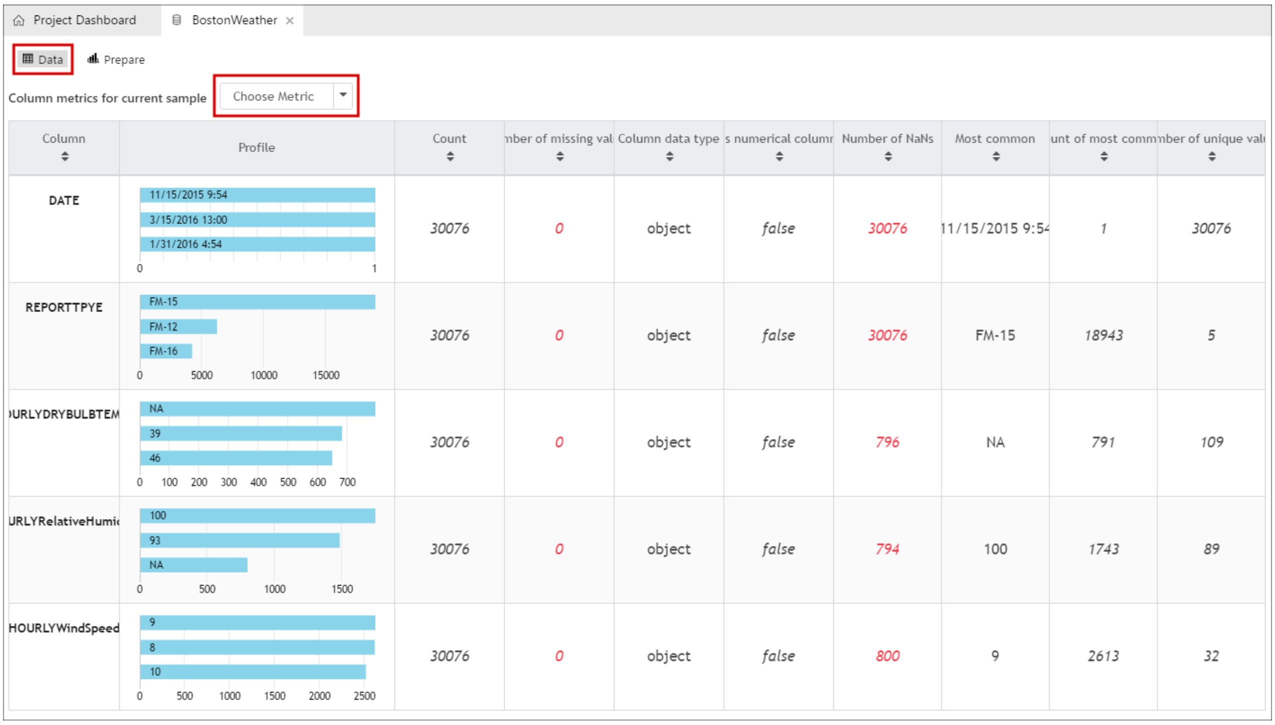
Notice that the **Steps** pane on the right side of the screen displays the individual actions taken while creating this data source.



	abc DATE	abc REPORTTP...	abc HOURLYD...	abc HOURLYRe...	abc HOURLYW...
1	1/1/2015 0:54	FM-15	22	50	10
2	1/1/2015 1:00	FM-12	22	50	10
3	1/1/2015 1:54	FM-15	22	50	10
4	1/1/2015 2:54	FM-15	22	50	11
5	1/1/2015 3:54	FM-15	24	46	13
6	1/1/2015 4:00	FM-12	24	46	13
7	1/1/2015 4:54	FM-15	22	52	15
8	1/1/2015 5:54	FM-15	23	48	17
9	1/1/2015 6:54	FM-15	23	50	14
10	1/1/2015 7:00	FM-12	23	50	14
11	1/1/2015 7:54	FM-15	22	52	13
12	1/1/2015 8:54	FM-15	25	44	16
13	1/1/2015 9:54	FM-15	28	39	14
14	1/1/2015 10:00	FM-12	28	39	14
15	1/1/2015 10:54	FM-15	30	34	18

View data source metrics

Select **Metrics** at the upper left of the tab's grid view. This view displays the distribution and other aggregated statistics of the sampled data.



Column	Profile	Count	nber of missing val	Column data type	s numerical columr	Number of NaNs	Most common	unt of most comm	nber of unique val
DATE	11/15/2015 9:54 3/15/2016 13:00 1/31/2016 4:54 0 1	30076	0	object	false	30076	11/15/2015 9:54	1	30076
REPORTTPYE	FM-15 FM-12 FM-16 0 5000 10000 15000	30076	0	object	false	30076	FM-15	18943	5
)URLYDRYBULBTEM	NA 39 46 0 100 200 300 400 500 600 700	30076	0	object	false	796	NA	791	109
URLYRelativeHumi	100 93 NA 0 500 1000 1500	30076	0	object	false	794	100	1743	89
HOURLYWindSpeed	9 8 10 0 500 1000 1500 2000 2500	30076	0	object	false	800	9	2613	32

> **NOTE**
>
> To configure the visibility of the statistics, use the **Choose Metric** drop-down list. Select and clear metrics there to change the grid view.

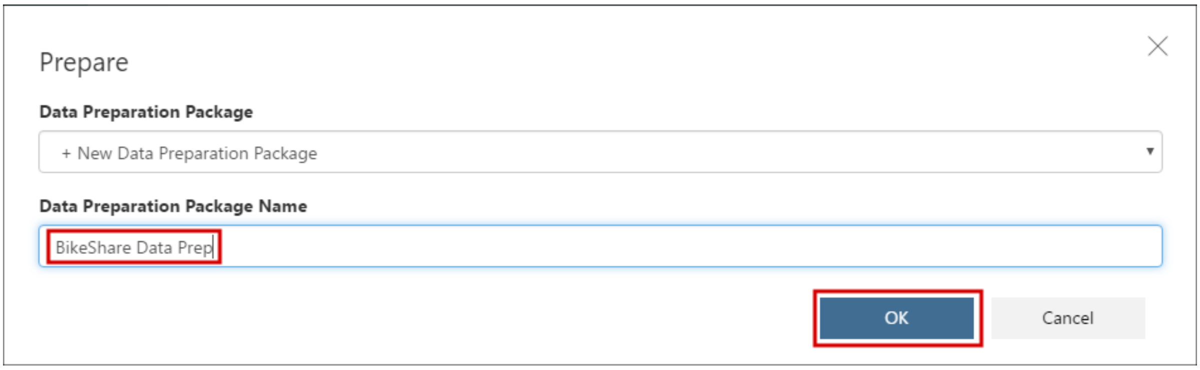
To return to the **Data** view, select **Data** in the upper left of the page.

Add a data source to the data preparation package

1. Select **Prepare** to begin preparing the data.

2. When prompted, enter a name for the data preparation package, such as **BikeShare Data Prep**.

3. Select **OK** to continue.



Prepare

Data Preparation Package: + New Data Preparation Package

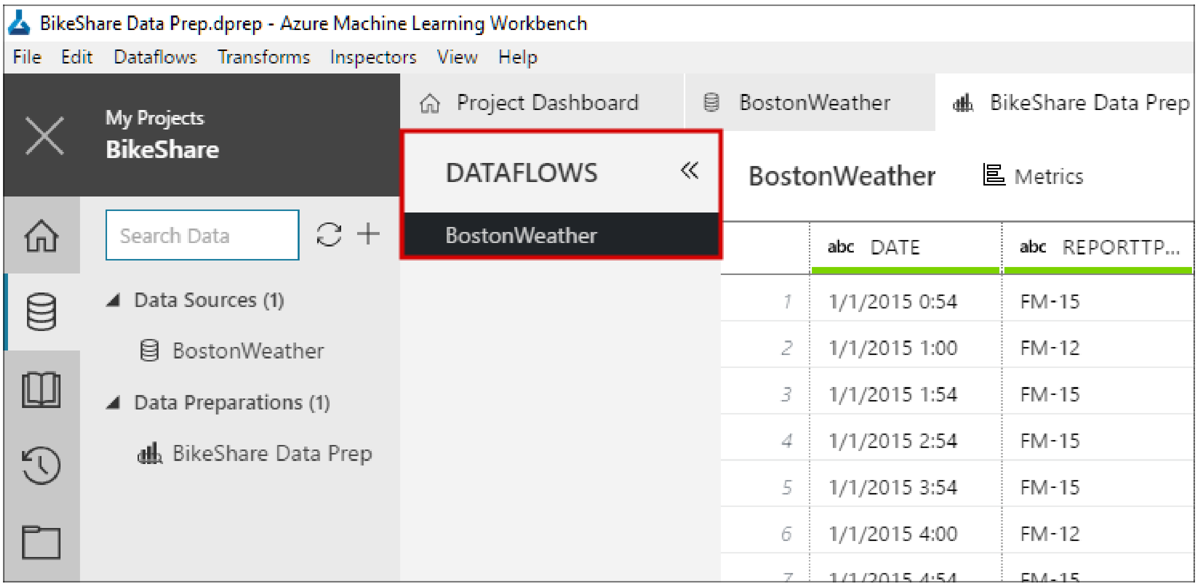
Data Preparation Package Name: BikeShare Data Prep

4. A new package named **BikeShare Data Prep** appears under the **Data Preparation** section of the **Data** tab.

   To display the package, select this entry.

5. Select the **>>** button to expand **Dataflows** and display the dataflows contained in the package. In this example, **BostonWeather** is the only dataflow.

> **IMPORTANT**
>
> A package can contain multiple dataflows.



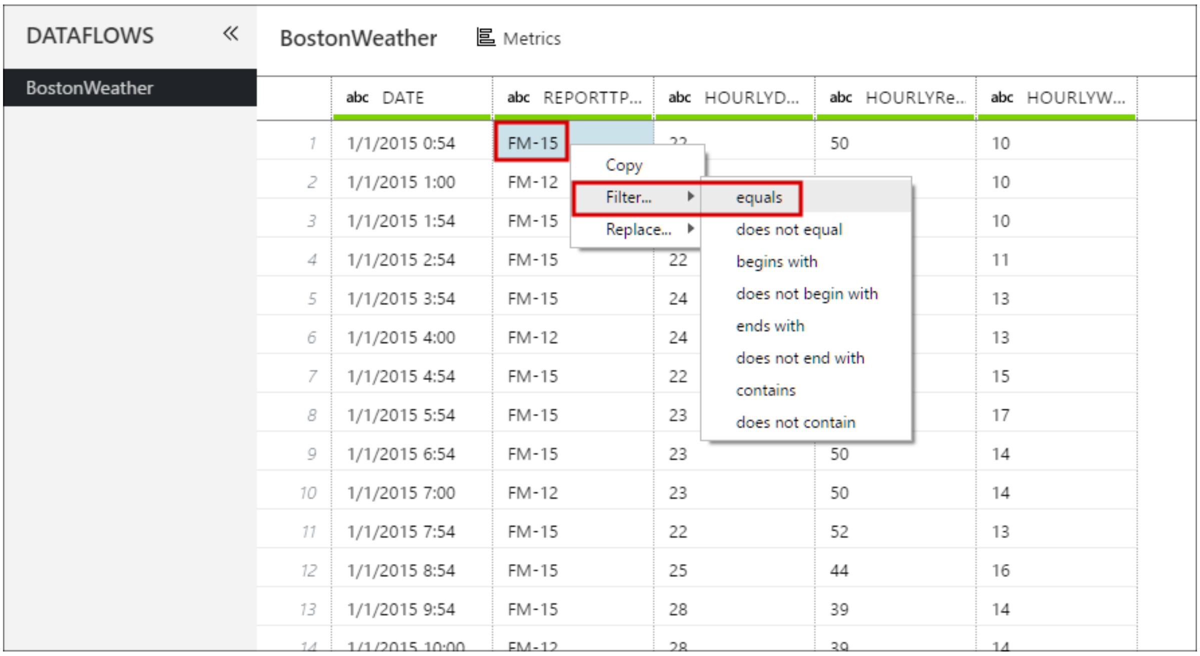
BikeShare Data Prep.dprep - Azure Machine Learning Workbench

	abc DATE	abc REPORTTP...
1	1/1/2015 0:54	FM-15
2	1/1/2015 1:00	FM-12
3	1/1/2015 1:54	FM-15
4	1/1/2015 2:54	FM-15
5	1/1/2015 3:54	FM-15
6	1/1/2015 4:00	FM-12
7	1/1/2015 4:54	FM-15

Filter data by value

1. To filter data, right-click a cell with a certain value, and select **Filter**. Then select the type of filter.

2. For this tutorial, select a cell that contains the value `FM-15`. Then set the filter to **equals**. Now the data is filtered to only return rows where the **REPORTTYPE** is `FM-15`.



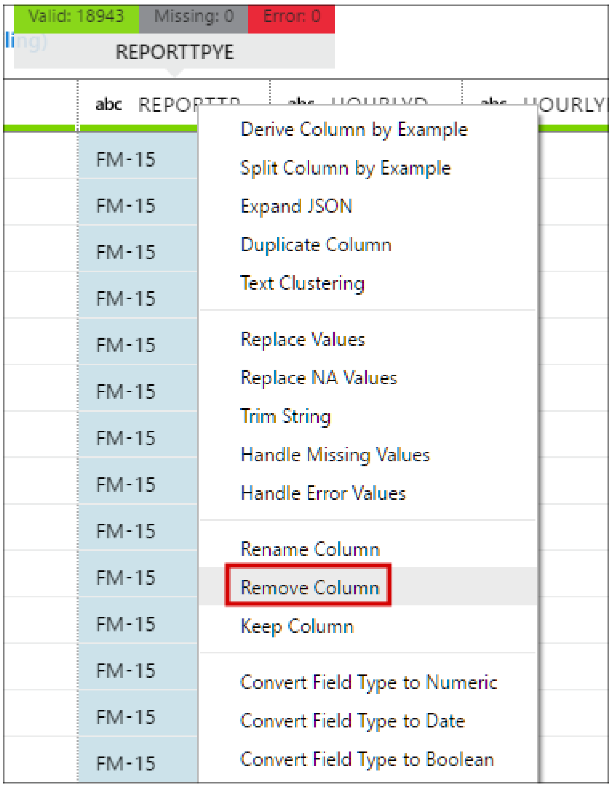
	abc DATE	abc REPORTTP...	abc HOURLYD...	abc HOURLYRe..	abc HOURLYW...
1	1/1/2015 0:54	FM-15	22	50	10
2	1/1/2015 1:00	FM-12			10
3	1/1/2015 1:54	FM-15			10
4	1/1/2015 2:54	FM-15	22		11
5	1/1/2015 3:54	FM-15	24		13
6	1/1/2015 4:00	FM-12	24		13
7	1/1/2015 4:54	FM-15	22		15
8	1/1/2015 5:54	FM-15	23		17
9	1/1/2015 6:54	FM-15	23	50	14
10	1/1/2015 7:00	FM-12	23	50	14
11	1/1/2015 7:54	FM-15	22	52	13
12	1/1/2015 8:54	FM-15	25	44	16
13	1/1/2015 9:54	FM-15	28	39	14
14	1/1/2015 10:00	FM-12	28	39	14

NOTE

FM-15 is a type of Meteorological Terminal Aviation Routine (METAR) weather report. The FM-15 reports are empirically observed to be the most complete, with little missing data.

Remove a column

You no longer need the **REPORTTYPE** column. Right-click the column header, and select **Remove Column**.



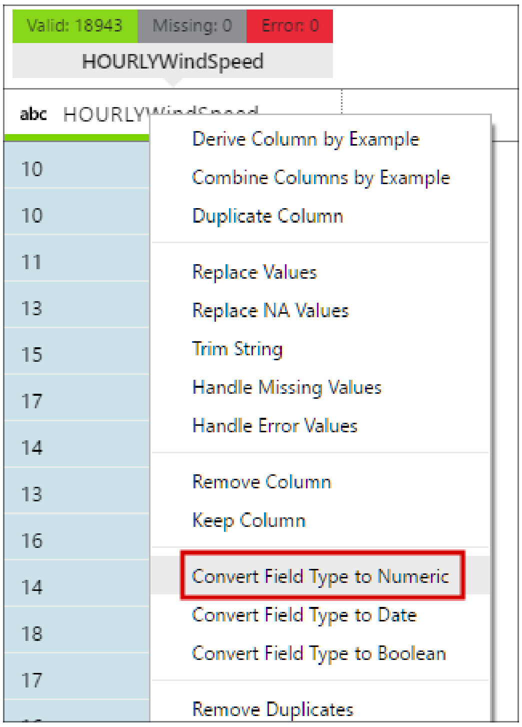
Change datatypes and remove errors

1. Select Ctrl (Command ⌘ on Mac) while you select column headers to select multiple columns at the same time. Use this technique to select the following column headers:

   - **HOURLYDRYBULBTEMPF**

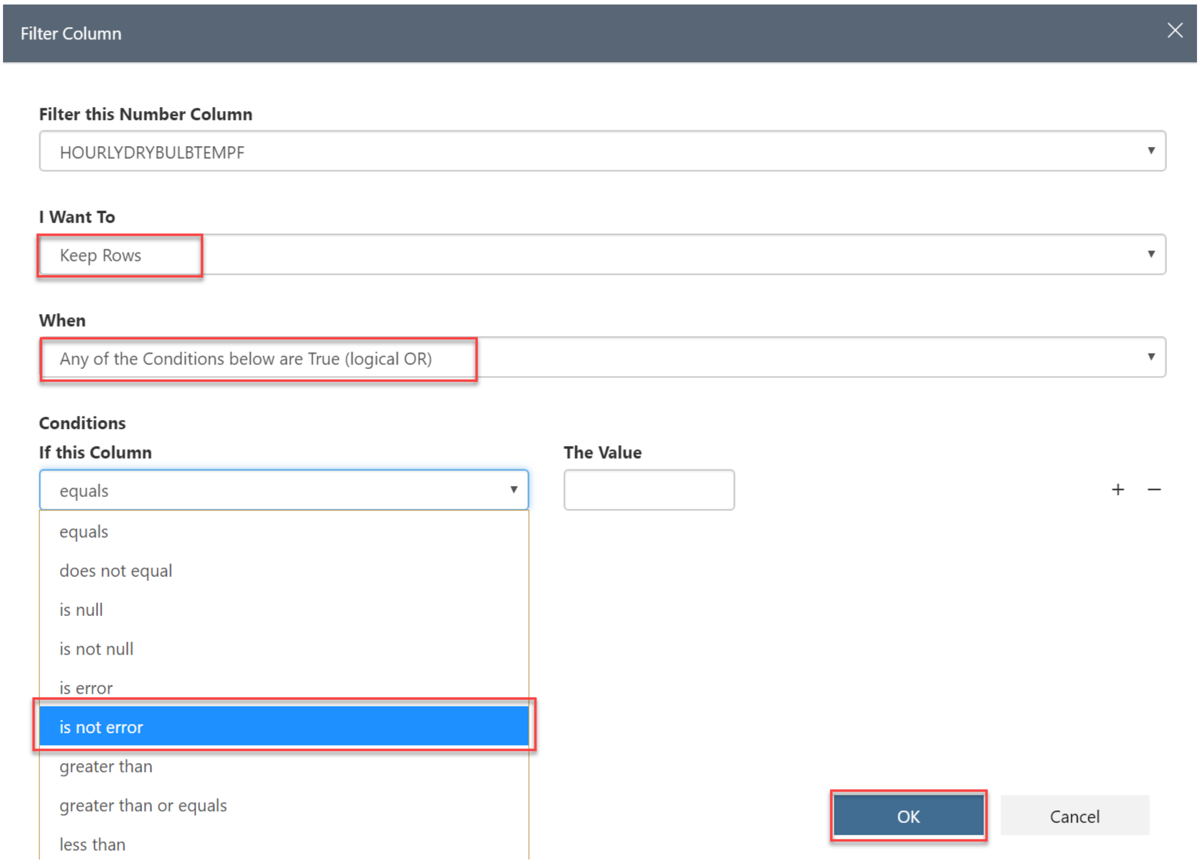
- **HOURLYRelativeHumidity**
- **HOURLYWindSpeed**

2. Right-click one of the selected column headers, and select **Convert Field Type to Numeric**. This option converts the data type for the columns to numeric.



3. Filter out the error values. Some columns have data type conversion problems. This problem is indicated by the red color in the **Data Quality Bar** for the column.

   To remove the rows that have errors, right-click the **HOURLYDRYBULBTEMPF** column heading. Select **Filter Column**. Use the default **I Want To** as **Keep Rows**. Change the **Conditions** drop-down list to select **is not error**. Select **OK** to apply the filter.


Filter Column

Filter this Number Column: HOURLYDRYBULBTEMPF

I Want To: Keep Rows

When: Any of the Conditions below are True (logical OR)

Conditions

If this Column: equals

The Value:

4. To eliminate the remaining error rows in the other columns, repeat this filter process for the **HOURLYRelativeHumidity** and **HOURLYWindSpeed** columns.

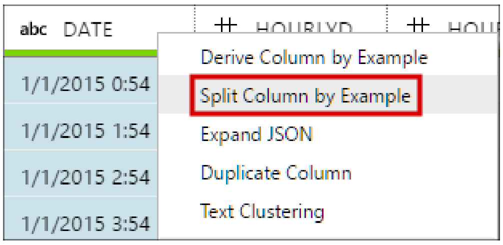
Use by example transformations

To use the data in a prediction for two-hour time blocks, you must compute the average weather conditions for two-hour periods. Use the following actions:

- Split the **DATE** column into separate **Date** and **Time** columns. See the following section for the detailed steps.
- Derive an **Hour_Range** column from the **Time** column. See the following section for the detailed steps.
- Derive a **Date_Hour_Range** column from the **DATE** and **Hour_Range** columns. See the following section for the detailed steps.

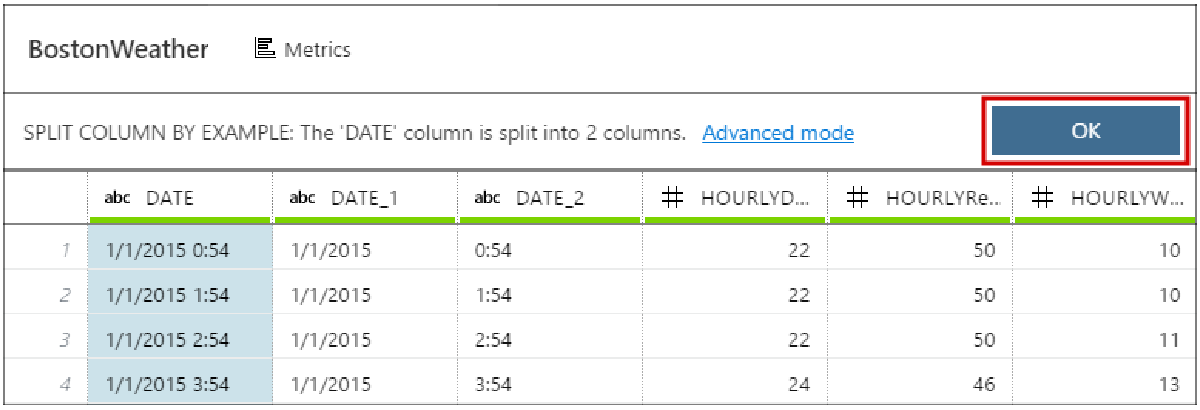
Split column by example

1. Split the **DATE** column into separate **Date** and **Time** columns. Right-click the **DATE** column header, and select **Split Column by Example**.



2. Machine Learning Workbench automatically identifies a meaningful delimiter and creates two columns by splitting the data into date and time values.

3. Select **OK** to accept the split operation results.

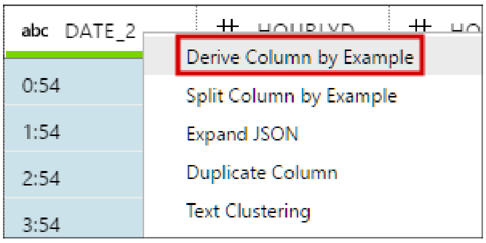


SPLIT COLUMN BY EXAMPLE: The 'DATE' column is split into 2 columns.

	abc DATE	abc DATE_1	abc DATE_2	# HOURLYD...	# HOURLYRe...	# HOURLYW...
1	1/1/2015 0:54	1/1/2015	0:54	22	50	10
2	1/1/2015 1:54	1/1/2015	1:54	22	50	10
3	1/1/2015 2:54	1/1/2015	2:54	22	50	11
4	1/1/2015 3:54	1/1/2015	3:54	24	46	13

Derive column by example

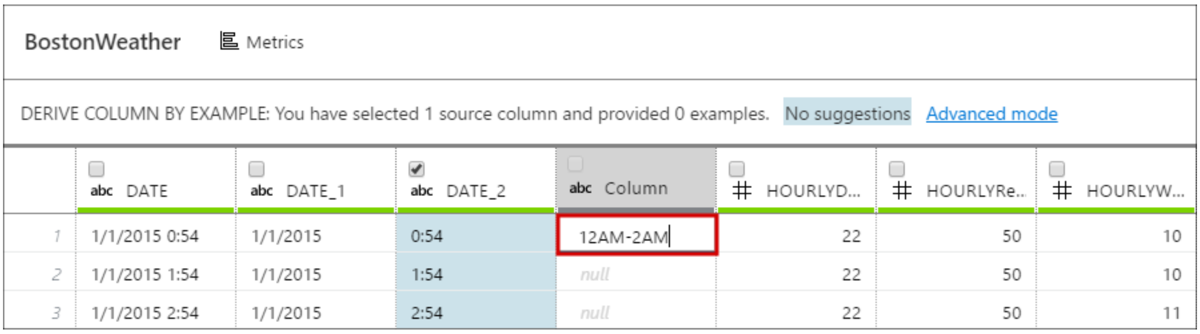
1. To derive a two-hour range, right-click the **DATE_2** column header, and select **Derive Column by Example**.



A new empty column is added with null values.

2. Select in the first empty cell in the new column. To provide an example of the time range desired, type

12AM-2AM in the new column, and then select Enter.



BostonWeather   Metrics

DERIVE COLUMN BY EXAMPLE: You have selected 1 source column and provided 0 examples. No suggestions Advanced mode

	□ abc DATE	□ abc DATE_1	☑ abc DATE_2	□ abc Column	□ # HOURLYD...	□ # HOURLYRe...	□ # HOURLYW...
1	1/1/2015 0:54	1/1/2015	0:54	12AM-2AM	22	50	10
2	1/1/2015 1:54	1/1/2015	1:54	null	22	50	10
3	1/1/2015 2:54	1/1/2015	2:54	null	22	50	11

NOTE

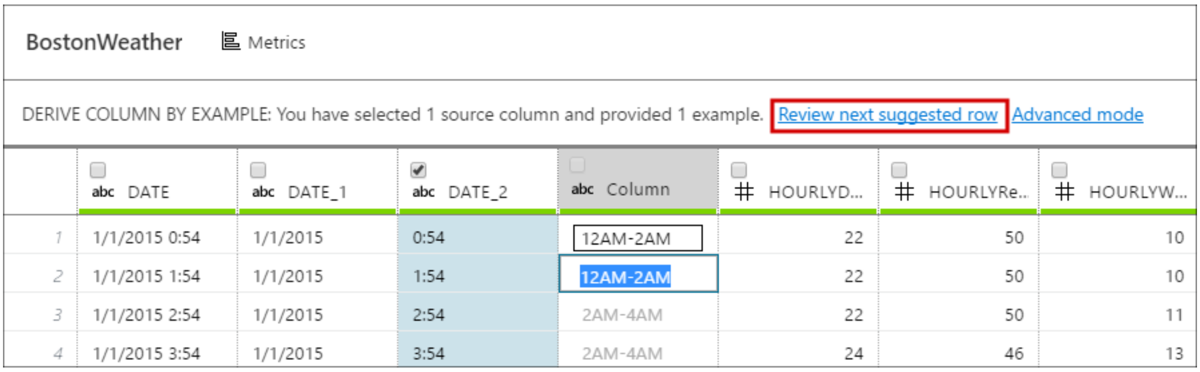
Machine Learning Workbench synthesizes a program based on the examples provided by you and applies the same program on remaining rows. All other rows are automatically populated based on the example you provided. Workbench also analyzes your data and tries to identify edge cases.

IMPORTANT

Identification of edge cases might not work on Mac in the current version of Workbench. Skip the following step 3 and step 4 on Mac. Instead, select **OK** after all the rows are populated with the derived values.

3. The text **Analyzing Data** above the grid indicates that Workbench is trying to detect edge cases. When finished, the status changes to **Review next suggested row** or **No suggestions**. In this example, **Review next suggested row** is returned.

4. To review the suggested changes, select **Review next suggested row**. The cell that you should review and correct, if needed, is highlighted on the display.



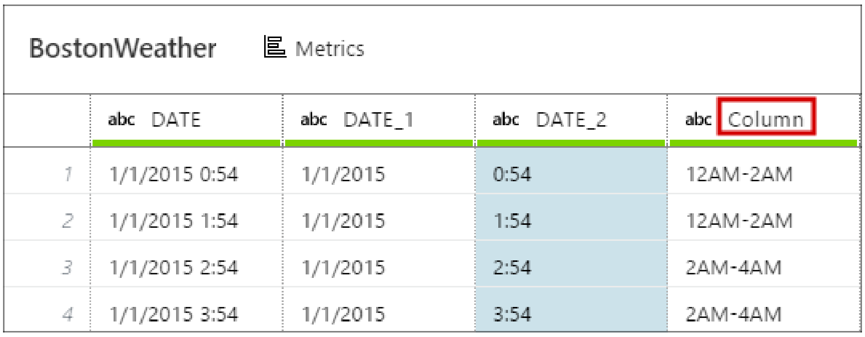
BostonWeather   Metrics

DERIVE COLUMN BY EXAMPLE: You have selected 1 source column and provided 1 example. Review next suggested row Advanced mode

	□ abc DATE	□ abc DATE_1	☑ abc DATE_2	□ abc Column	□ # HOURLYD...	□ # HOURLYRe...	□ # HOURLYW...
1	1/1/2015 0:54	1/1/2015	0:54	12AM-2AM	22	50	10
2	1/1/2015 1:54	1/1/2015	1:54	12AM-2AM	22	50	10
3	1/1/2015 2:54	1/1/2015	2:54	2AM-4AM	22	50	11
4	1/1/2015 3:54	1/1/2015	3:54	2AM-4AM	24	46	13

Select **OK** to accept the transformation.

5. You are returned to the grid view of data for **BostonWeather**. The grid now contains the three columns added previously.



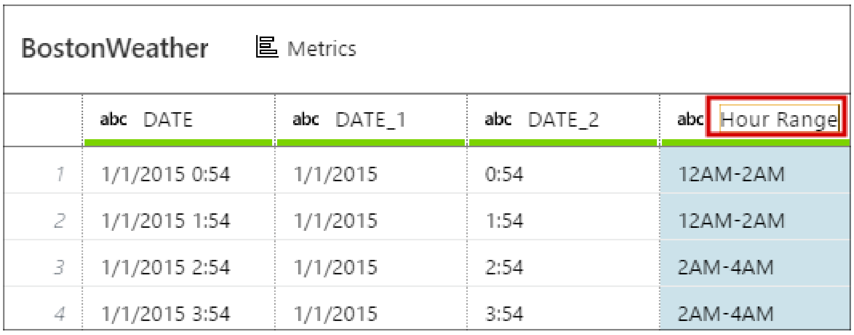
BostonWeather   Metrics

	abc DATE	abc DATE_1	abc DATE_2	abc Column
1	1/1/2015 0:54	1/1/2015	0:54	12AM-2AM
2	1/1/2015 1:54	1/1/2015	1:54	12AM-2AM
3	1/1/2015 2:54	1/1/2015	2:54	2AM-4AM
4	1/1/2015 3:54	1/1/2015	3:54	2AM-4AM

TIP

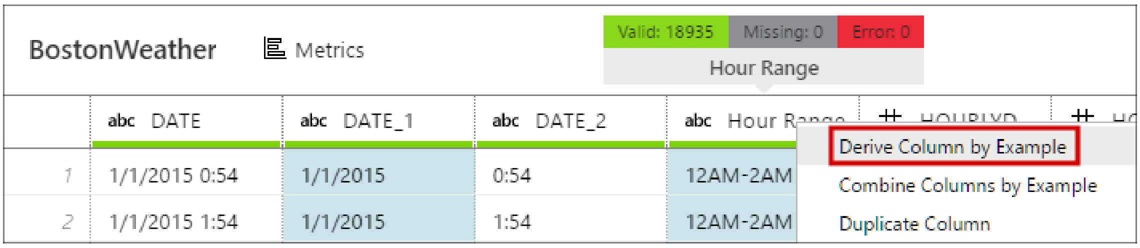
All the changes you made are preserved in the **Steps** pane. Go to the step that you created in the **Steps** pane, select the down arrow, and select **Edit**. The advanced window for **Derive Column by Example** is displayed. All your examples are preserved here. You also can add examples manually by double-clicking on a row in the following grid. Select **Cancel** to return to the main grid without applying changes. You also can access this view by selecting **Advanced mode** while you perform a **Derive Column by Example** transform.

6. To rename the column, double-click the column header, and type **Hour Range**. Select Enter to save the change.



	abc DATE	abc DATE_1	abc DATE_2	abc Hour Range
1	1/1/2015 0:54	1/1/2015	0:54	12AM-2AM
2	1/1/2015 1:54	1/1/2015	1:54	12AM-2AM
3	1/1/2015 2:54	1/1/2015	2:54	2AM-4AM
4	1/1/2015 3:54	1/1/2015	3:54	2AM-4AM

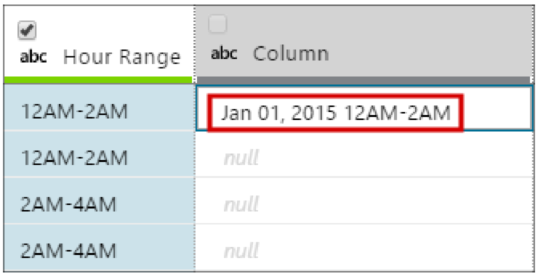
7. To derive the date and hour range, multi-select the **Date_1** and **Hour Range** columns, right-click, and then select **Derive Column by Example**.



	abc DATE	abc DATE_1	abc DATE_2	abc Hour Range
1	1/1/2015 0:54	1/1/2015	0:54	12AM-2AM
2	1/1/2015 1:54	1/1/2015	1:54	12AM-2AM

Type **Jan 01, 2015 12AM-2AM** as the example against the first row, and select Enter.

Workbench determines the transformation based on the example you provide. In this example, the result is that the date format is changed and concatenated with the two-hour window.



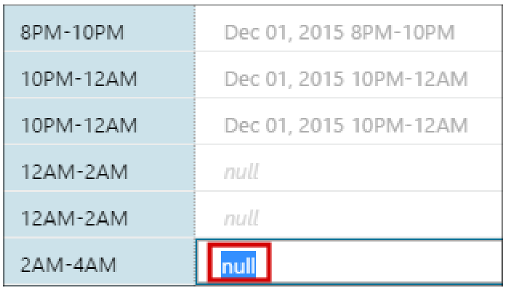
☑ abc Hour Range	□ abc Column
12AM-2AM	Jan 01, 2015 12AM-2AM
12AM-2AM	*null*
2AM-4AM	*null*
2AM-4AM	*null*

IMPORTANT

On a Mac, take the following step instead of step 8:

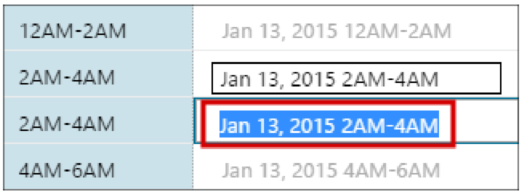
- Go to the first cell that contains **Feb 01, 2015 12AM-2AM**. It should be row 15. Correct the value to **Jan 02, 2015 12AM-2AM**, and select Enter.

8. Wait for the status to change from **Analyzing Data** to **Review next suggested row**. This change might take several seconds. Select the status link to go to the suggested row.



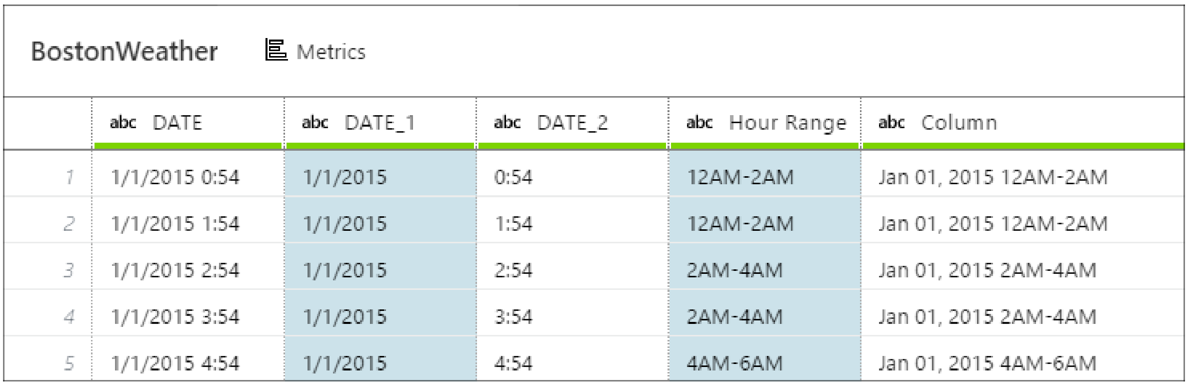
8PM-10PM	Dec 01, 2015 8PM-10PM
10PM-12AM	Dec 01, 2015 10PM-12AM
10PM-12AM	Dec 01, 2015 10PM-12AM
12AM-2AM	null
12AM-2AM	null
2AM-4AM	null

The row has a null value because the source date value can be for either dd/mm/yyyy or mm/dd/yyyy. Type the correct value of **Jan 13, 2015 2AM-4AM**, and select Enter. Workbench uses the two examples to improve the derivation for the remaining rows.



12AM-2AM	Jan 13, 2015 12AM-2AM
2AM-4AM	Jan 13, 2015 2AM-4AM
2AM-4AM	Jan 13, 2015 2AM-4AM
4AM-6AM	Jan 13, 2015 4AM-6AM

9. Select **OK** to accept the transform.

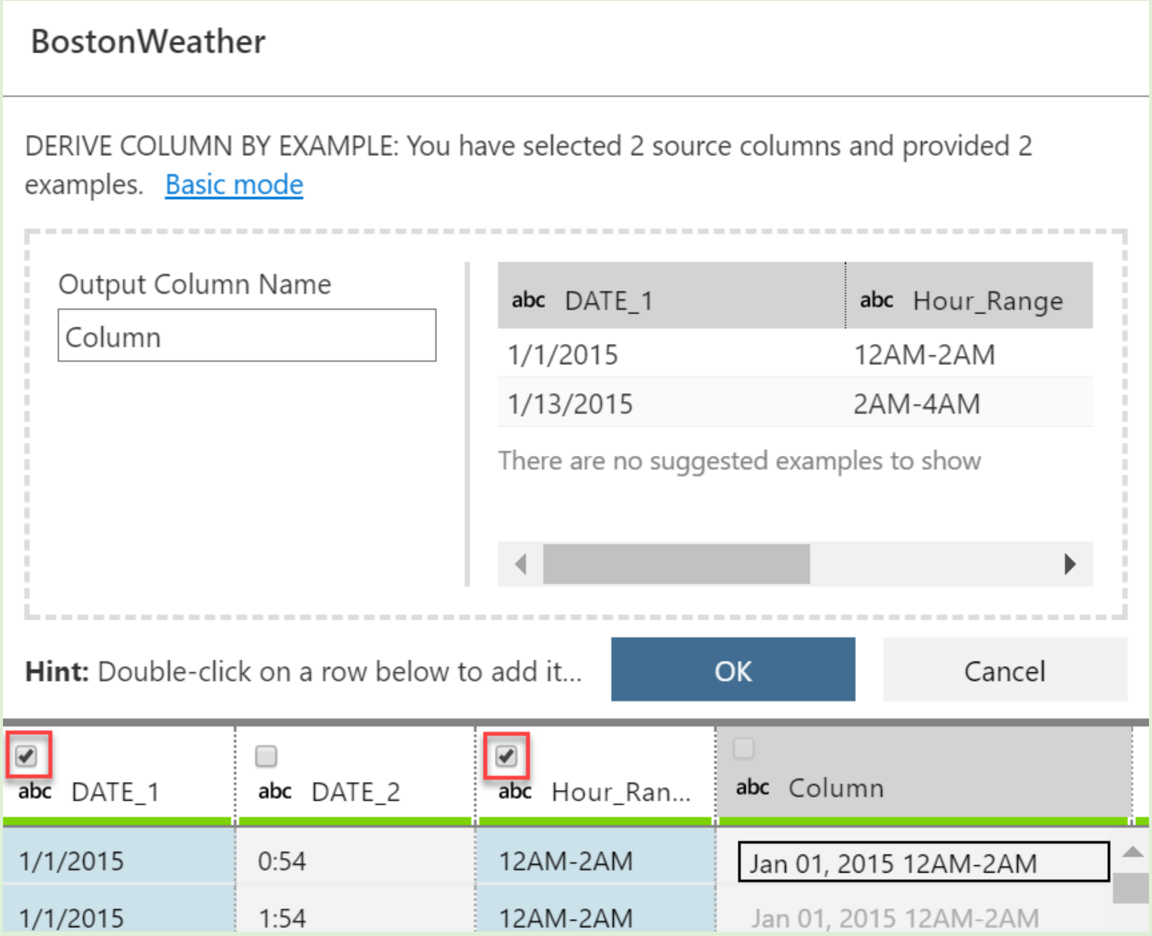


BostonWeather

	abc DATE	abc DATE_1	abc DATE_2	abc Hour Range	abc Column
1	1/1/2015 0:54	1/1/2015	0:54	12AM-2AM	Jan 01, 2015 12AM-2AM
2	1/1/2015 1:54	1/1/2015	1:54	12AM-2AM	Jan 01, 2015 12AM-2AM
3	1/1/2015 2:54	1/1/2015	2:54	2AM-4AM	Jan 01, 2015 2AM-4AM
4	1/1/2015 3:54	1/1/2015	3:54	2AM-4AM	Jan 01, 2015 2AM-4AM
5	1/1/2015 4:54	1/1/2015	4:54	4AM-6AM	Jan 01, 2015 4AM-6AM

TIP

To use the **Advanced mode** of **Derive Column by Example** for this step. select the down arrow in the **Steps** pane. In the data grid, there are check boxes next to the **DATE_1** and **Hour Range** columns. Clear the check box next to the **Hour Range** column to see how the output changes. In the absence of the **Hour Range** column as input, **12AM-2AM** is treated as a constant and is appended to the derived values. Select **Cancel** to return to the main grid without applying your changes.



BostonWeather

DERIVE COLUMN BY EXAMPLE: You have selected 2 source columns and provided 2 examples. Basic mode

Output Column Name: Column

abc DATE_1	abc Hour_Range
1/1/2015	12AM-2AM
1/13/2015	2AM-4AM

There are no suggested examples to show

Hint: Double-click on a row below to add it...

☑ abc DATE_1	□ abc DATE_2	☑ abc Hour_Ran...	□ abc Column
1/1/2015	0:54	12AM-2AM	Jan 01, 2015 12AM-2AM
1/1/2015	1:54	12AM-2AM	Jan 01, 2015 12AM-2AM

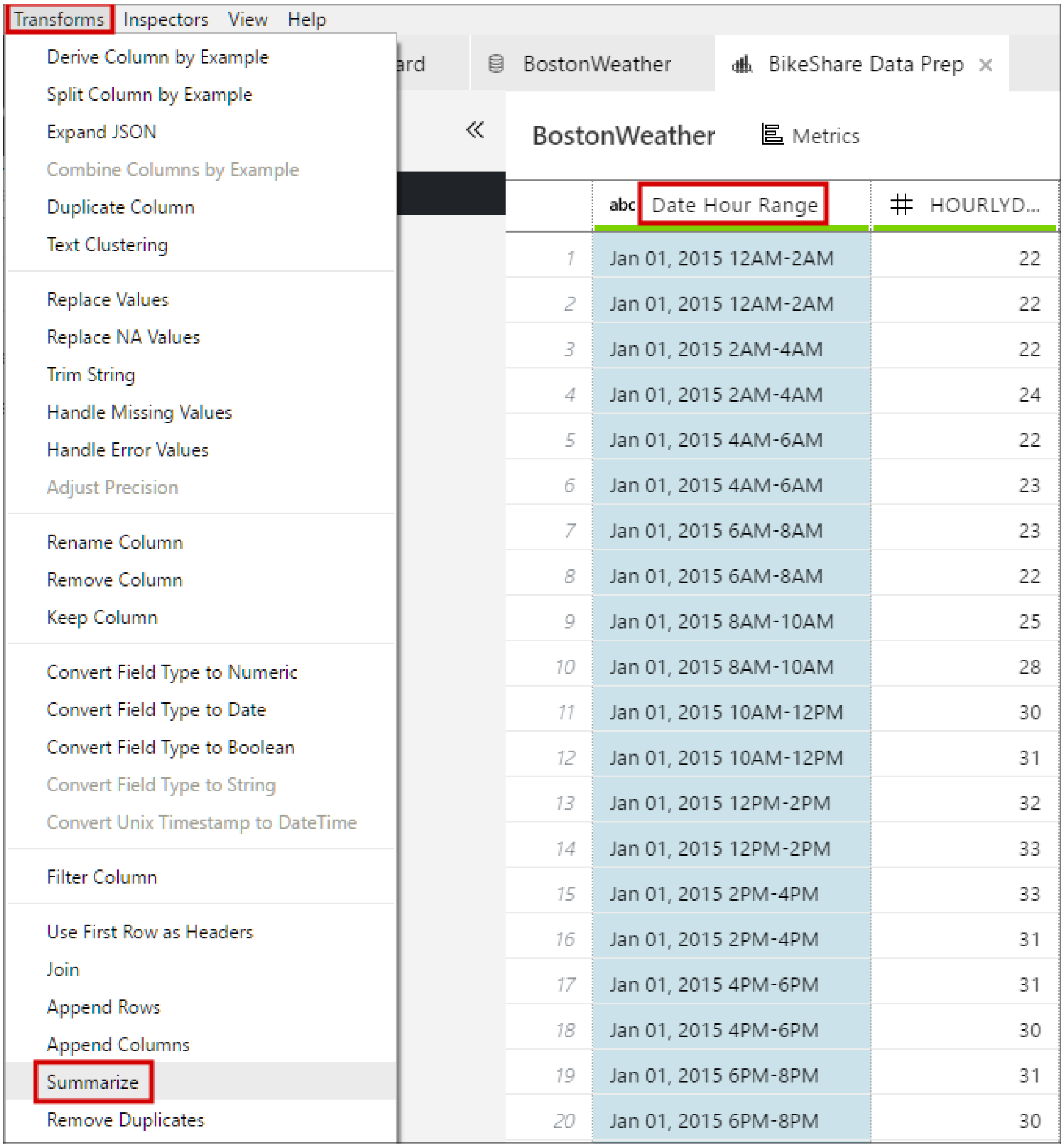
10. To rename the column, double-click the header. Change the name to **Date Hour Range**, and then select Enter.

11. Multi-select the **DATE**, **DATE_1**, **DATE_2**, and **Hour Range** columns. Right-click, and then select **Remove column**.

Summarize data (mean)

The next step is to summarize the weather conditions by taking the mean of the values, grouped by hour range.

1. Select the **Date Hour Range** column, and then on the **Transforms** menu, select **Summarize**.

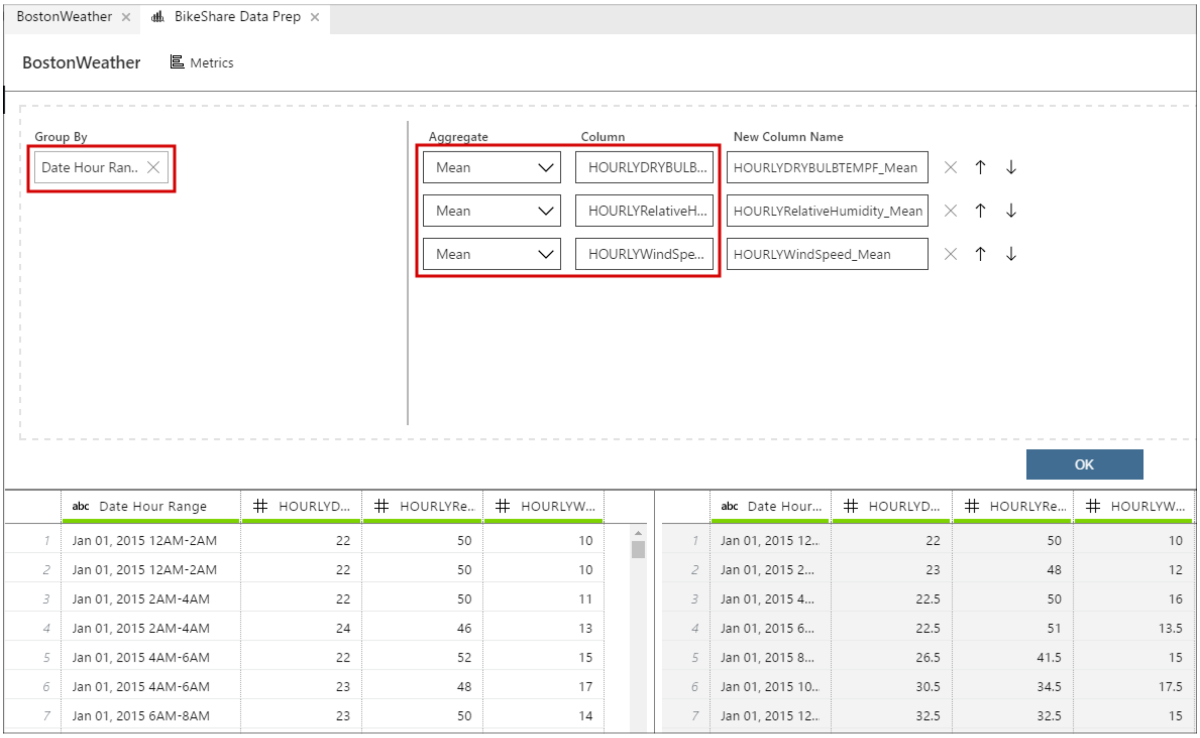


	abc Date Hour Range	# HOURLYD...
1	Jan 01, 2015 12AM-2AM	22
2	Jan 01, 2015 12AM-2AM	22
3	Jan 01, 2015 2AM-4AM	22
4	Jan 01, 2015 2AM-4AM	24
5	Jan 01, 2015 4AM-6AM	22
6	Jan 01, 2015 4AM-6AM	23
7	Jan 01, 2015 6AM-8AM	23
8	Jan 01, 2015 6AM-8AM	22
9	Jan 01, 2015 8AM-10AM	25
10	Jan 01, 2015 8AM-10AM	28
11	Jan 01, 2015 10AM-12PM	30
12	Jan 01, 2015 10AM-12PM	31
13	Jan 01, 2015 12PM-2PM	32
14	Jan 01, 2015 12PM-2PM	33
15	Jan 01, 2015 2PM-4PM	33
16	Jan 01, 2015 2PM-4PM	31
17	Jan 01, 2015 4PM-6PM	31
18	Jan 01, 2015 4PM-6PM	30
19	Jan 01, 2015 6PM-8PM	31
20	Jan 01, 2015 6PM-8PM	30

2. To summarize the data, drag columns from the grid at the bottom of the page to the left and right panes at the top. The left pane contains the text **Drag columns here to group data**. The right pane contains the text **Drag columns here to summarize data**.

   a. Drag the **Date Hour Range** column from the grid at the bottom to the left pane. Drag **HOURLYDRYBULBTEMPF**, **HOURLYRelativeHumidity**, and **HOURLYWindSpeed** to the right pane.

   b. In the right pane, select **Mean** as the **Aggregate** measure for each column. Select **OK** to finish the summarization.



Group By: Date Hour Ran..

Aggregate	Column	New Column Name
Mean	HOURLYDRYBULB...	HOURLYDRYBULBTEMPF_Mean
Mean	HOURLYRelativeH...	HOURLYRelativeHumidity_Mean
Mean	HOURLYWindSpe...	HOURLYWindSpeed_Mean

	Date Hour Range	HOURLYD...	HOURLYRe..	HOURLYW...
1	Jan 01, 2015 12AM-2AM	22	50	10
2	Jan 01, 2015 12AM-2AM	22	50	10
3	Jan 01, 2015 2AM-4AM	22	50	11
4	Jan 01, 2015 2AM-4AM	24	46	13
5	Jan 01, 2015 4AM-6AM	22	52	15
6	Jan 01, 2015 4AM-6AM	23	48	17
7	Jan 01, 2015 6AM-8AM	23	50	14

	Date Hour...	HOURLYD...	HOURLYRe...	HOURLYW...
1	Jan 01, 2015 12...	22	50	10
2	Jan 01, 2015 2...	23	48	12
3	Jan 01, 2015 4...	22.5	50	16
4	Jan 01, 2015 6...	22.5	51	13.5
5	Jan 01, 2015 8...	26.5	41.5	15
6	Jan 01, 2015 10...	30.5	34.5	17.5
7	Jan 01, 2015 12...	32.5	32.5	15

Transform dataflow by using script

Changing the data in the numeric columns to a range of 0 to 1 allows some models to converge quickly. Currently, there is no built-in transformation to generically do this transformation. Use a Python script to perform this operation.

1. On the **Transform** menu, select **Transform Dataflow (Script)**.

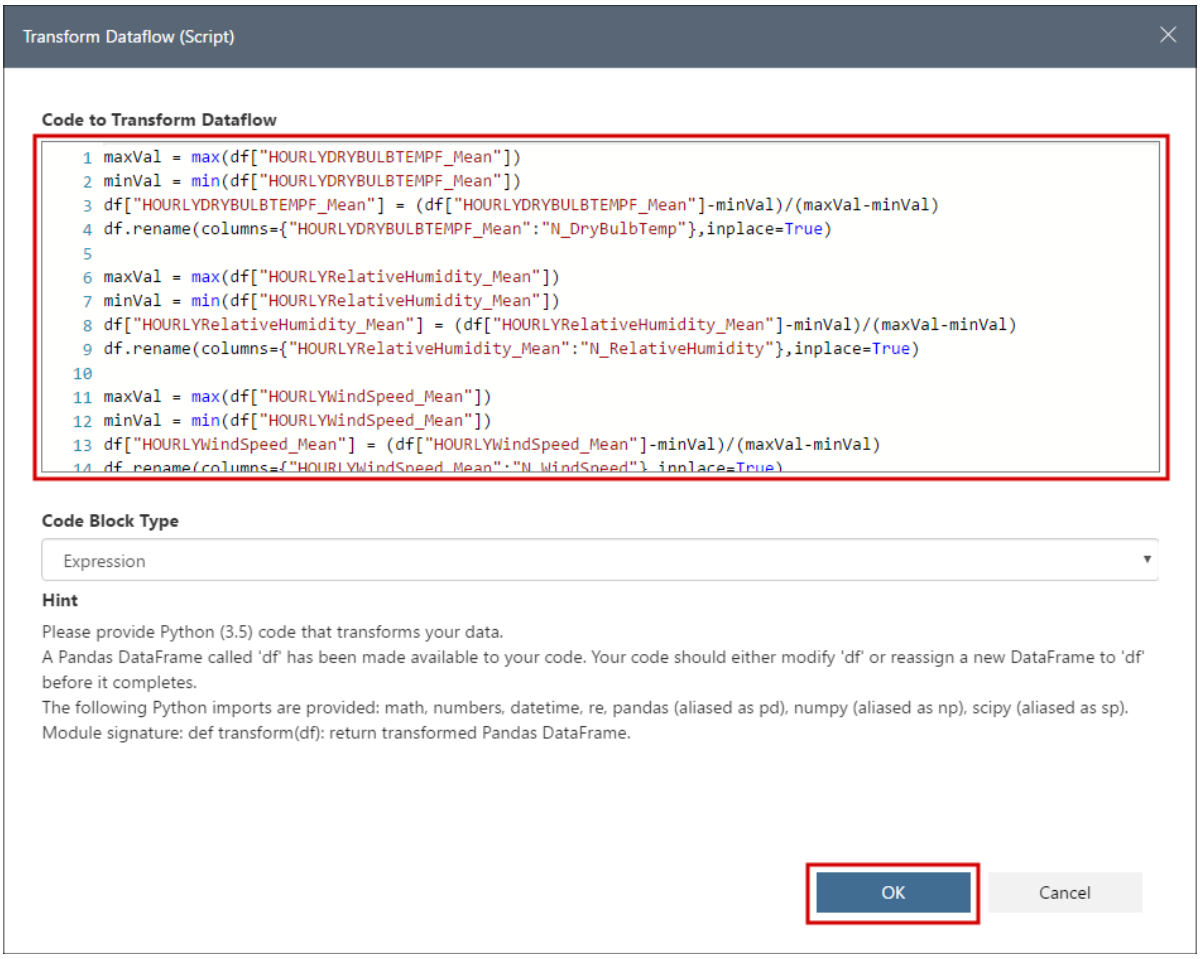
2. Enter the following code in the text box that appears. If you used the column names, the code should work without modification. You are writing a simple min-max normalization logic in Python.

> **WARNING**
>
> The script expects the column names used previously in this tutorial. If you have different column names, you must change the names in the script.

```
maxVal = max(df["HOURLYDRYBULBTEMPF_Mean"])
minVal = min(df["HOURLYDRYBULBTEMPF_Mean"])
df["HOURLYDRYBULBTEMPF_Mean"] = (df["HOURLYDRYBULBTEMPF_Mean"]-minVal)/(maxVal-minVal)
df.rename(columns={"HOURLYDRYBULBTEMPF_Mean":"N_DryBulbTemp"},inplace=True)

maxVal = max(df["HOURLYRelativeHumidity_Mean"])
minVal = min(df["HOURLYRelativeHumidity_Mean"])
df["HOURLYRelativeHumidity_Mean"] = (df["HOURLYRelativeHumidity_Mean"]-minVal)/(maxVal-minVal)
df.rename(columns={"HOURLYRelativeHumidity_Mean":"N_RelativeHumidity"},inplace=True)

maxVal = max(df["HOURLYWindSpeed_Mean"])
minVal = min(df["HOURLYWindSpeed_Mean"])
df["HOURLYWindSpeed_Mean"] = (df["HOURLYWindSpeed_Mean"]-minVal)/(maxVal-minVal)
df.rename(columns={"HOURLYWindSpeed_Mean":"N_WindSpeed"},inplace=True)

df
```

TIP

The Python script must return `df` at the end. This value is used to populate the grid.



Code to Transform Dataflow

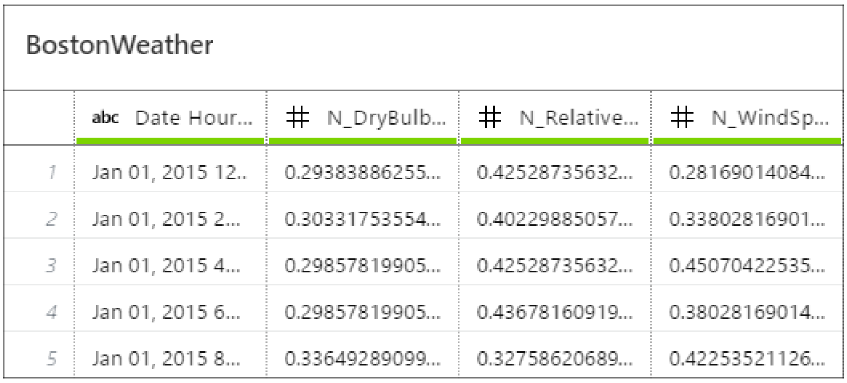
```
maxVal = max(df["HOURLYDRYBULBTEMPF_Mean"])
minVal = min(df["HOURLYDRYBULBTEMPF_Mean"])
df["HOURLYDRYBULBTEMPF_Mean"] = (df["HOURLYDRYBULBTEMPF_Mean"]-minVal)/(maxVal-minVal)
df.rename(columns={"HOURLYDRYBULBTEMPF_Mean":"N_DryBulbTemp"},inplace=True)

maxVal = max(df["HOURLYRelativeHumidity_Mean"])
minVal = min(df["HOURLYRelativeHumidity_Mean"])
df["HOURLYRelativeHumidity_Mean"] = (df["HOURLYRelativeHumidity_Mean"]-minVal)/(maxVal-minVal)
df.rename(columns={"HOURLYRelativeHumidity_Mean":"N_RelativeHumidity"},inplace=True)

maxVal = max(df["HOURLYWindSpeed_Mean"])
minVal = min(df["HOURLYWindSpeed_Mean"])
df["HOURLYWindSpeed_Mean"] = (df["HOURLYWindSpeed_Mean"]-minVal)/(maxVal-minVal)
df.rename(columns={"HOURLYWindSpeed_Mean":"N_WindSpeed"},inplace=True)
```

Code Block Type: Expression

Hint

Please provide Python (3.5) code that transforms your data.
A Pandas DataFrame called 'df' has been made available to your code. Your code should either modify 'df' or reassign a new DataFrame to 'df' before it completes.
The following Python imports are provided: math, numbers, datetime, re, pandas (aliased as pd), numpy (aliased as np), scipy (aliased as sp).
Module signature: def transform(df): return transformed Pandas DataFrame.

3. Select **OK** to use the script. The numeric columns in the grid now contain values in the range of 0 to 1.



BostonWeather

	abc Date Hour...	# N_DryBulb...	# N_Relative...	# N_WindSp...
1	Jan 01, 2015 12..	0.29383886255...	0.42528735632...	0.28169014084...
2	Jan 01, 2015 2...	0.30331753554...	0.40229885057...	0.33802816901...
3	Jan 01, 2015 4...	0.29857819905...	0.42528735632...	0.45070422535...
4	Jan 01, 2015 6...	0.29857819905...	0.43678160919...	0.38028169014...
5	Jan 01, 2015 8...	0.33649289099...	0.32758620689...	0.42253521126...

You have finished preparing the weather data. Next, prepare the trip data.

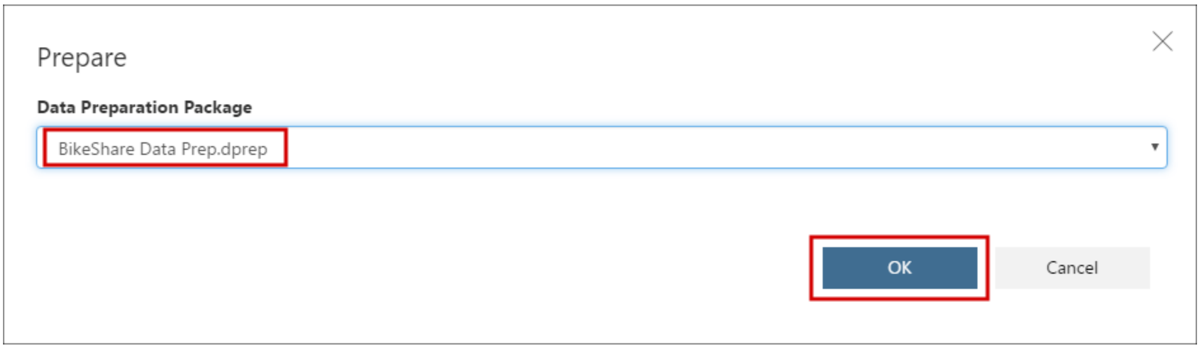
Load trip data

1. To import the `201701-hubway-tripdata.csv` file, use the steps in the Create a new data source section. Use the following options during the import process:

   - **File Selection**: Select **Azure Blob** when you browse to select the file.
   - **Sampling Scheme**: Select **Full File** sampling scheme, and make the sample active.
   - **Data Type**: Accept the defaults.

2. After you import the data, select **Prepare** to begin preparing the data. Select the existing **BikeShare Data Prep.dprep** package, and then select **OK**.

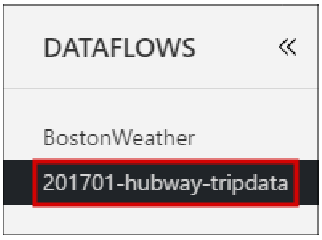
   This process adds a **Dataflow** to the existing **Data Preparation** file rather than creating a new one.

   
   Prepare

   Data Preparation Package

   BikeShare Data Prep.dprep

3. After the grid has loaded, expand **DATAFLOWS**. There are now two dataflows: **BostonWeather** and **201701-hubway-tripdata**. Select the **201701-hubway-tripdata** entry.

   
   DATAFLOWS

   BostonWeather

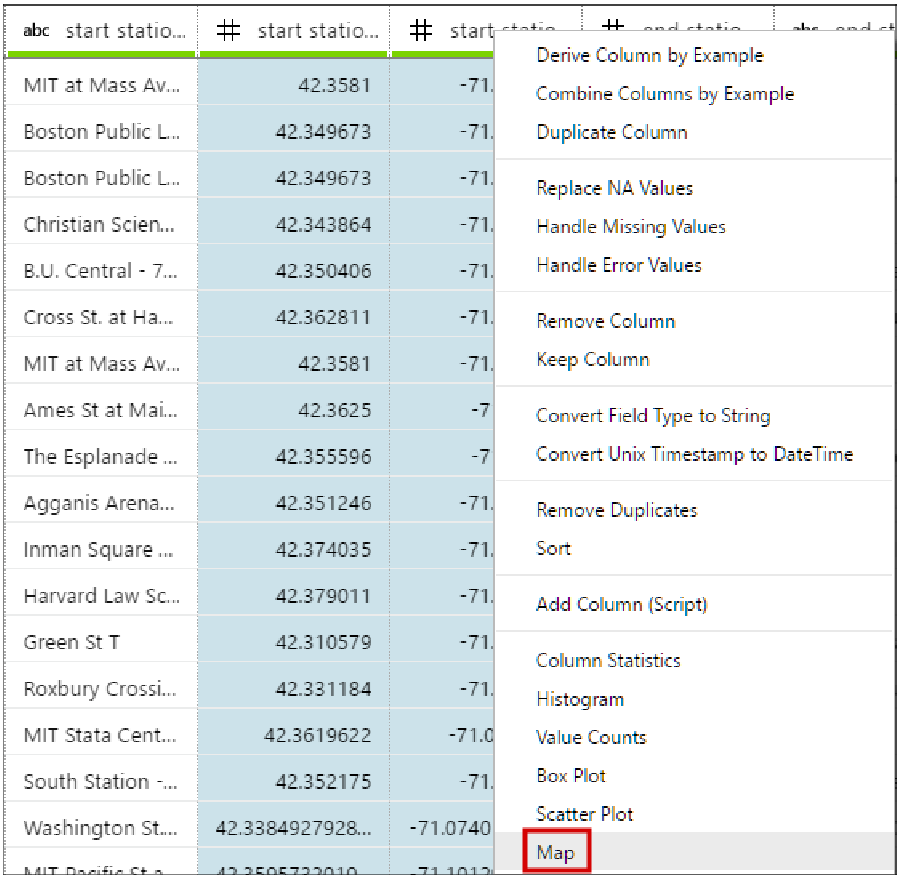
   201701-hubway-tripdata

Use the map inspector

For data preparation, useful visualizations called inspectors are available for string, numeric, and geographical data. They help you to understand the data better and identify outliers. Follow these steps to use the map inspector.

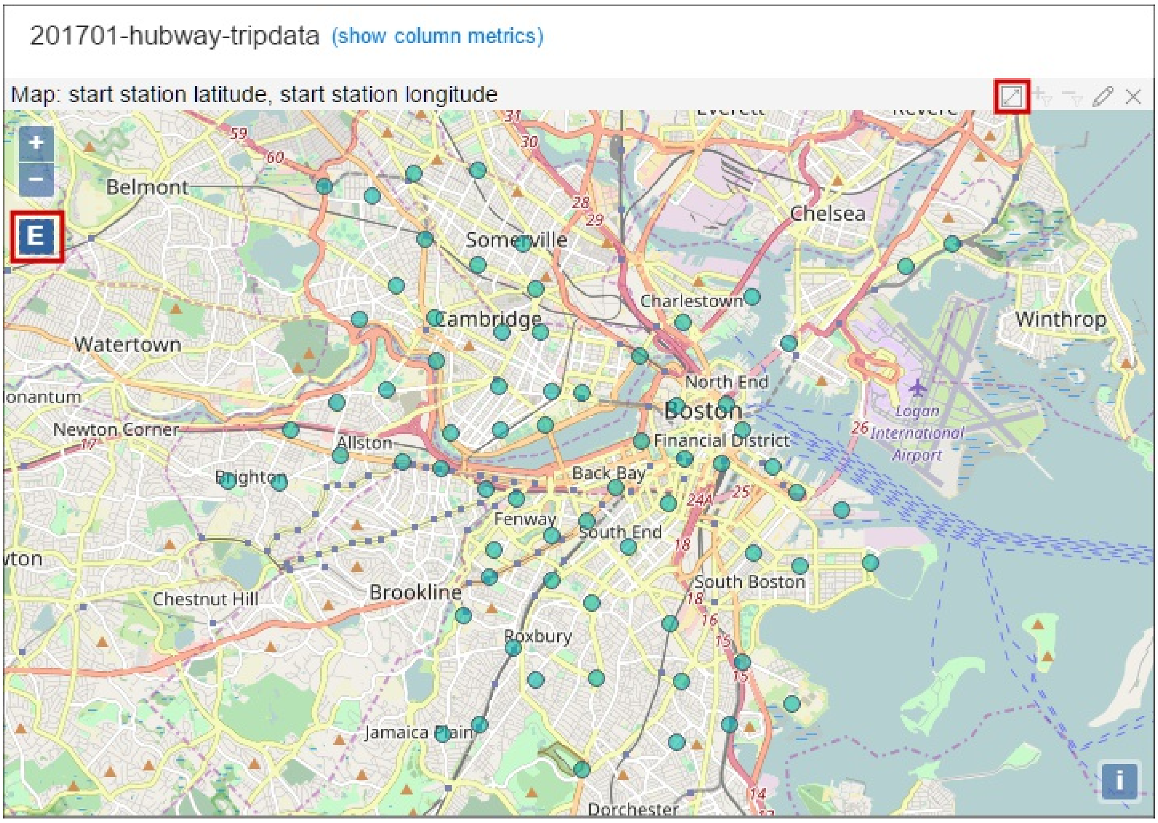
1. Multi-select the **start station latitude** and **start station longitude** columns. Right-click one of the columns, and then select **Map**.

   > **TIP**
   >
   > To enable multi-select, hold down the Ctrl key (Command ⌘ on Mac), and select the header for each column.



abc start statio...	# start statio...	# start
MIT at Mass Av...	42.3581	-71.
Boston Public L...	42.349673	-71.
Boston Public L...	42.349673	-71.
Christian Scien...	42.343864	-71.
B.U. Central - 7...	42.350406	-71.
Cross St. at Ha...	42.362811	-71.
MIT at Mass Av...	42.3581	-71.
Ames St at Mai...	42.3625	-7
The Esplanade ...	42.355596	-7
Agganis Arena...	42.351246	-71.
Inman Square ...	42.374035	-71.
Harvard Law Sc...	42.379011	-71.
Green St T	42.310579	-71.
Roxbury Crossi...	42.331184	-71.
MIT Stata Cent...	42.3619622	-71.0
South Station -...	42.352175	-71.
Washington St...	42.3384927928...	-71.0740

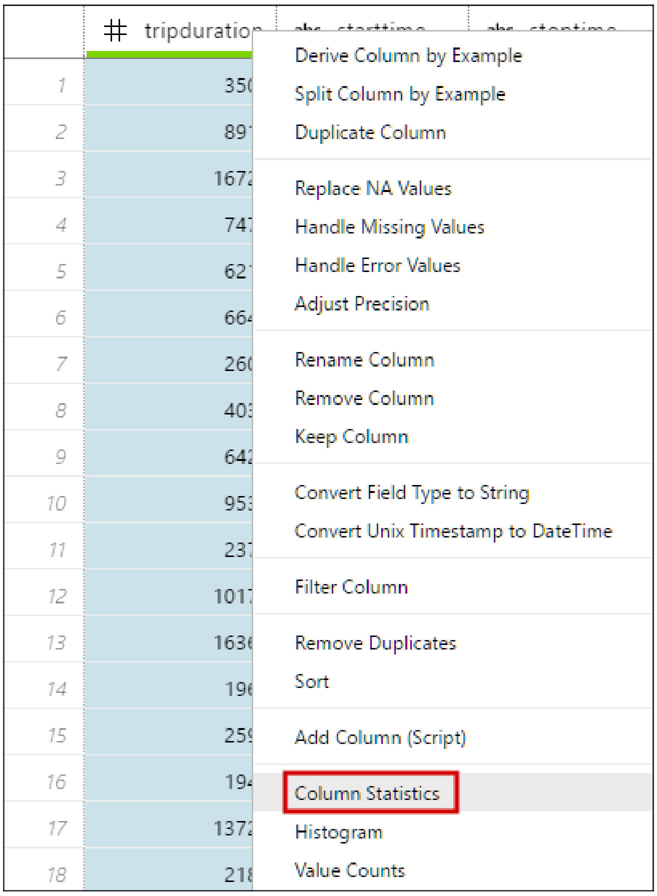
2. To maximize the map visualization, select the **Maximize** icon. To fit the map to the window, select the **E** icon on the upper-left side of the visualization.



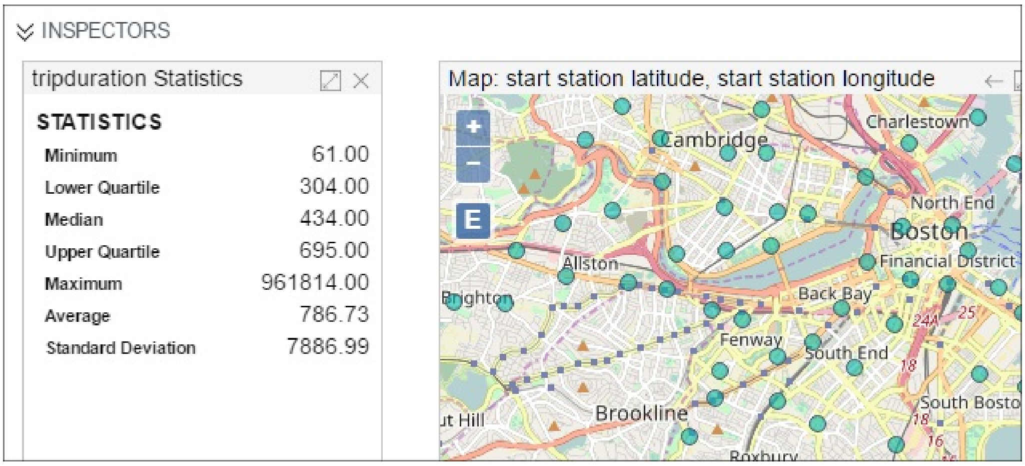
3. Select the **Minimize** button to return to the grid view.

Use the column statistics inspector

To use the column statistics inspector, right-click the **tripduration** column, and select **Column Statistics**.



This process adds a new visualization titled **tripduration Statistics** in the **INSPECTORS** pane.



tripduration Statistics

STATISTICS

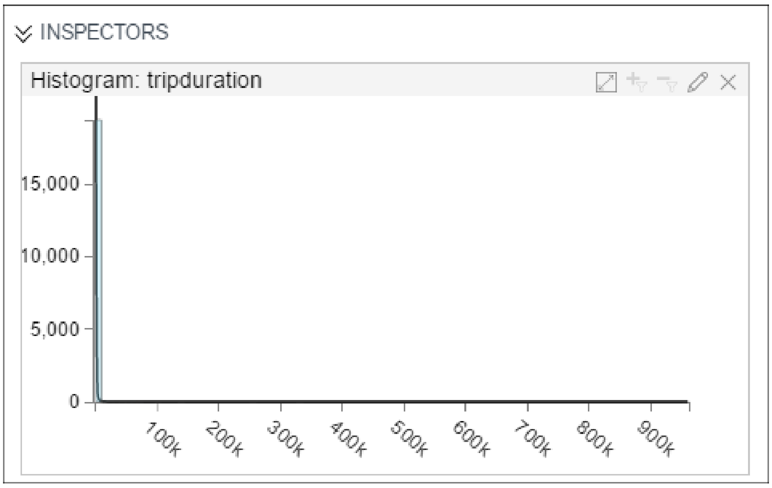
Minimum	61.00
Lower Quartile	304.00
Median	434.00
Upper Quartile	695.00
Maximum	961814.00
Average	786.73
Standard Deviation	7886.99

IMPORTANT

The maximum value of the trip duration is 961,814 minutes, which is about two years. It seems there are some outliers in the dataset.

Use the histogram inspector

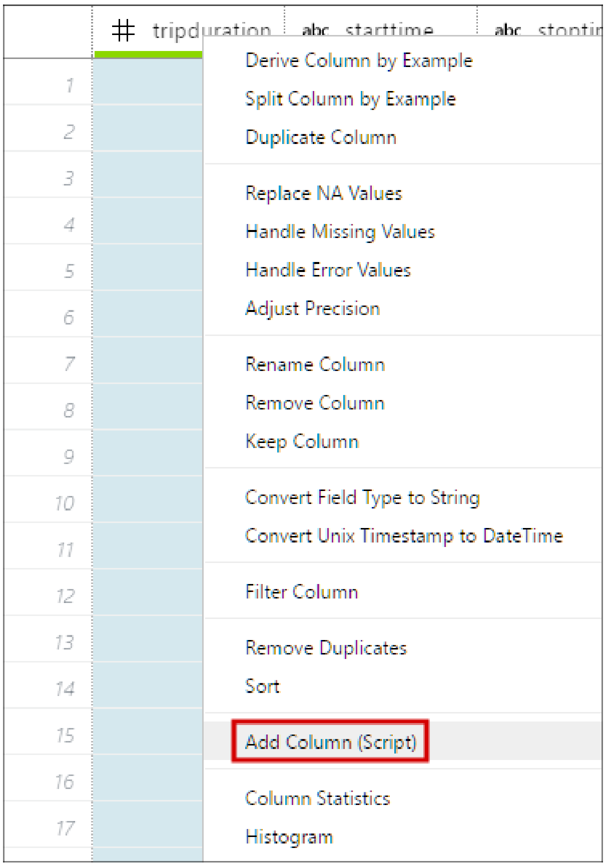
To attempt to identify outliers, right-click the **tripduration** column, and select **Histogram**.



The histogram isn't helpful because the outliers skew the graph.

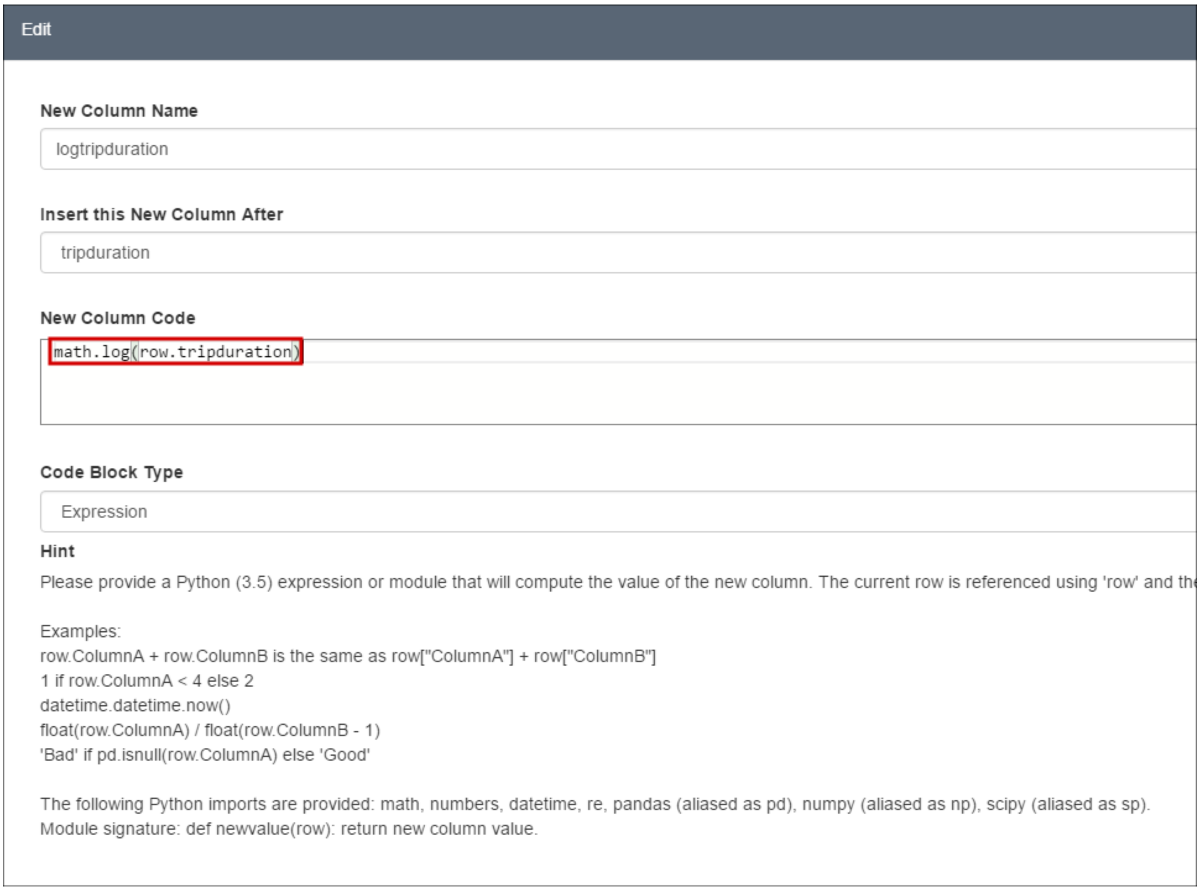
Add a column by using script

1. Right-click the **tripduration** column, and select **Add Column (Script)**.



2. In the **Add Column (Script)** dialog box, use the following values:

   - **New Column Name**: logtripduration
   - **Insert this New Column After**: tripduration
   - **New Column Code**: `math.log(row.tripduration)`
   - **Code Block Type**: Expression



Edit

New Column Name

logtripduration

Insert this New Column After

tripduration

New Column Code

```
math.log(row.tripduration)
```

Code Block Type

Expression

Hint

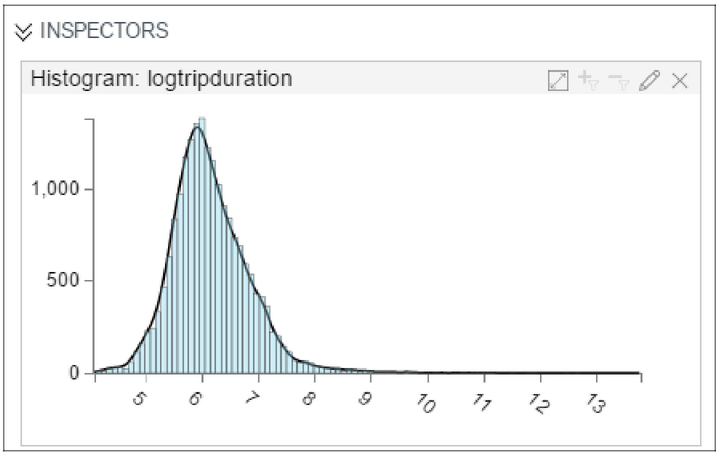
Please provide a Python (3.5) expression or module that will compute the value of the new column. The current row is referenced using 'row' and the

Examples:
row.ColumnA + row.ColumnB is the same as row["ColumnA"] + row["ColumnB"]
1 if row.ColumnA < 4 else 2
datetime.datetime.now()
float(row.ColumnA) / float(row.ColumnB - 1)
'Bad' if pd.isnull(row.ColumnA) else 'Good'

The following Python imports are provided: math, numbers, datetime, re, pandas (aliased as pd), numpy (aliased as np), scipy (aliased as sp).
Module signature: def newvalue(row): return new column value.

3. Select **OK** to add the **logtripduration** column.

4. Right-click the column, and select **Histogram**.



Visually, this histogram seems like a normal distribution with an abnormal tail.

Use an advanced filter

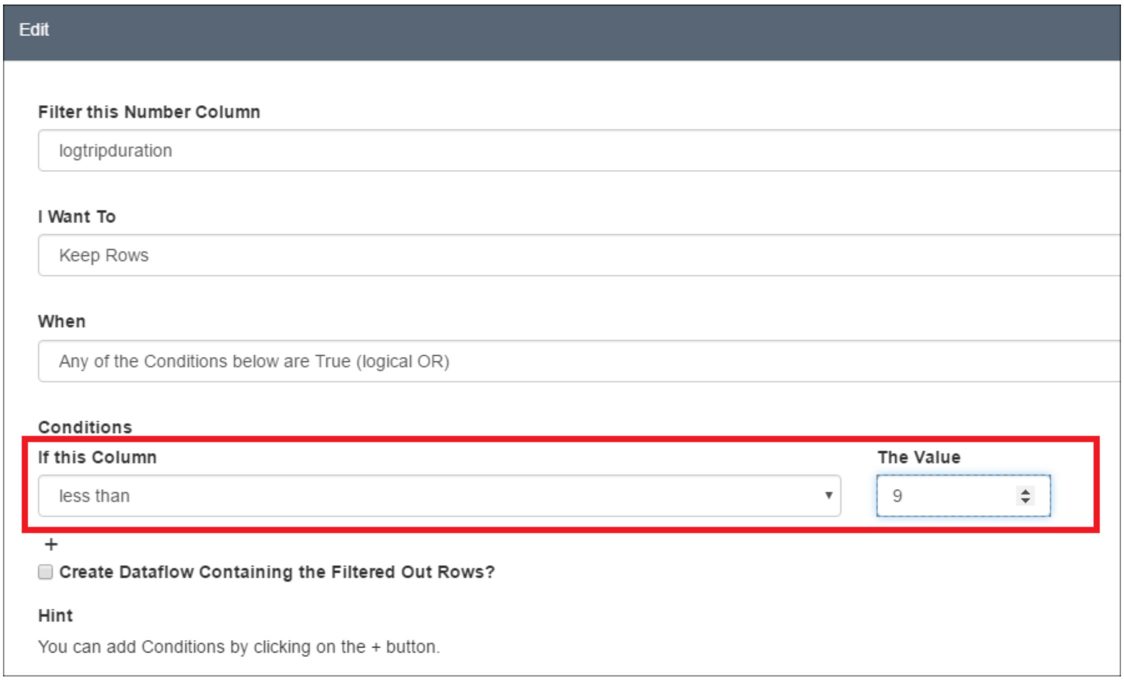
Using a filter on the data updates the inspectors with the new distribution.

1. Right-click the **logtripduration** column, and select **Filter Column**.

2. In the **Edit** dialog box, use the following values:

   - **Filter this Number Column**: logtripduration
   - **I Want To**: Keep Rows
   - **When**: Any of the Conditions below are True (logical OR)
   - **If this Column**: less than

- **The Value**: 9



Edit

Filter this Number Column

logtripduration

I Want To

Keep Rows

When

Any of the Conditions below are True (logical OR)

Conditions

If this Column: less than

The Value: 9

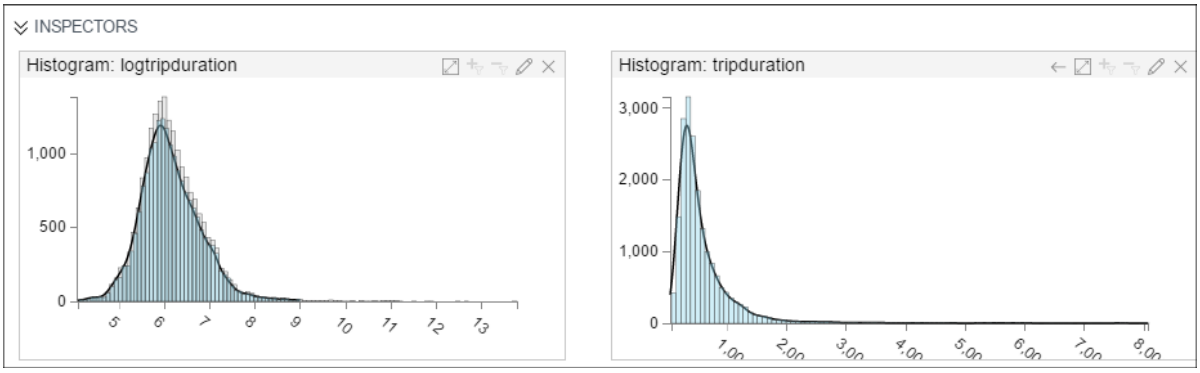
+

□ Create Dataflow Containing the Filtered Out Rows?

Hint

You can add Conditions by clicking on the + button.

3. Select **OK** to apply the filter.



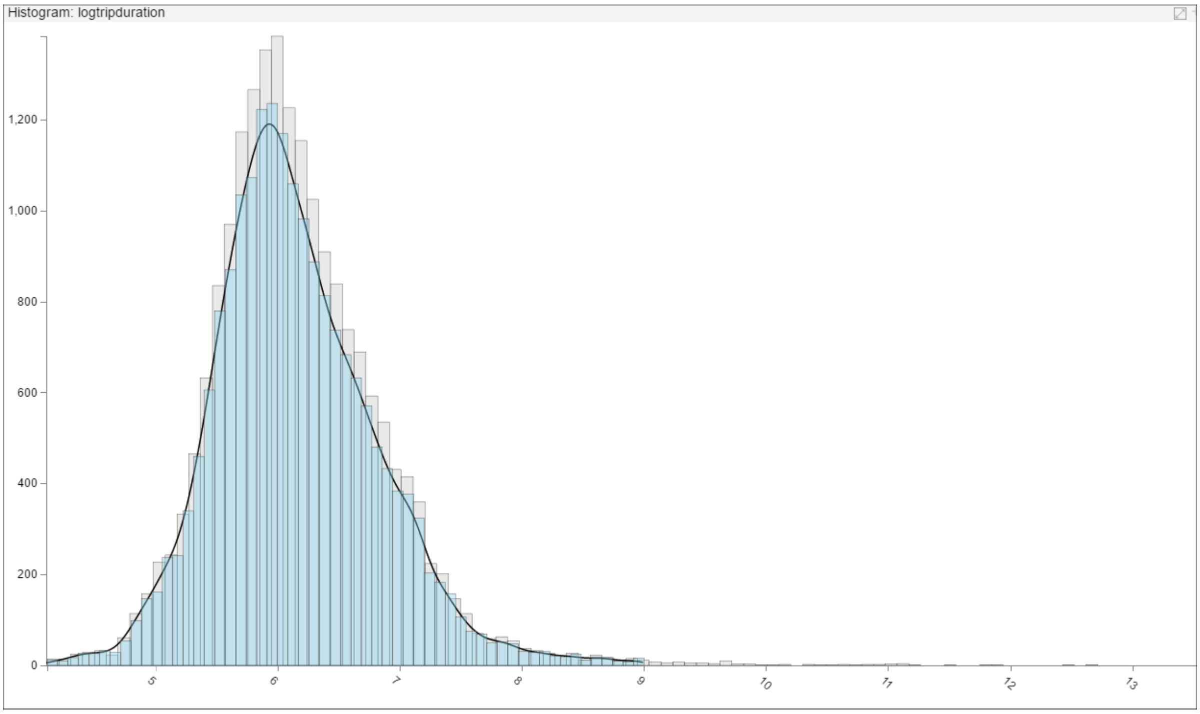
The halo effect

1. Maximize the **logtripduration** histogram. A blue histogram is overlaid on a gray histogram. This display is called the **Halo Effect**:

   - The gray histogram represents the distribution before the operation (in this case, the filtering operation).

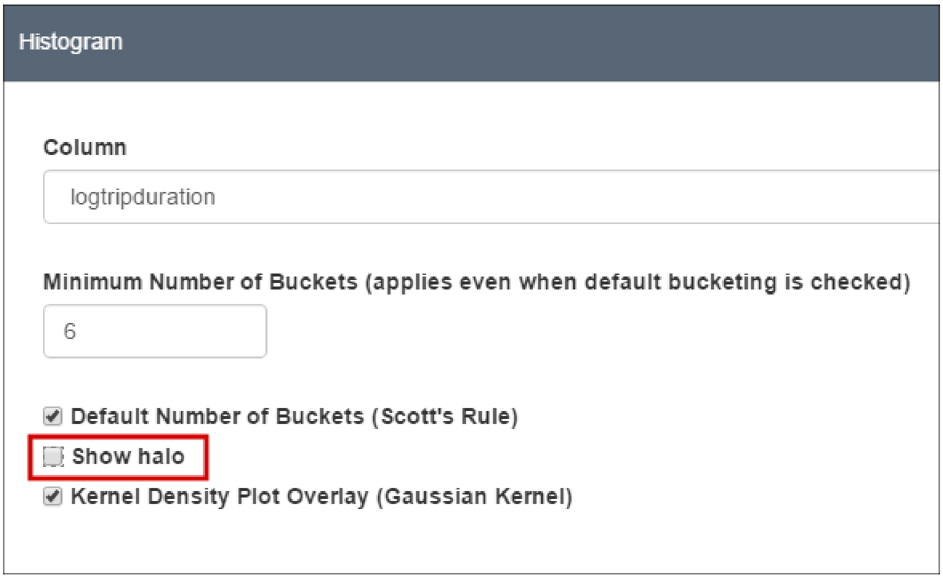
   - The blue histogram represents the histogram after the operation.

   The halo effect helps with visualizing the effect of an operation on the data.



> **NOTE**
>
> The blue histogram appears shorter compared to the previous one. This difference is due to automatic re-bucketing of data in the new range.

2. To remove the halo, select **Edit** and clear **Show halo**.



Histogram

Column

logtripduration

Minimum Number of Buckets (applies even when default bucketing is checked)

6

☑ Default Number of Buckets (Scott's Rule)

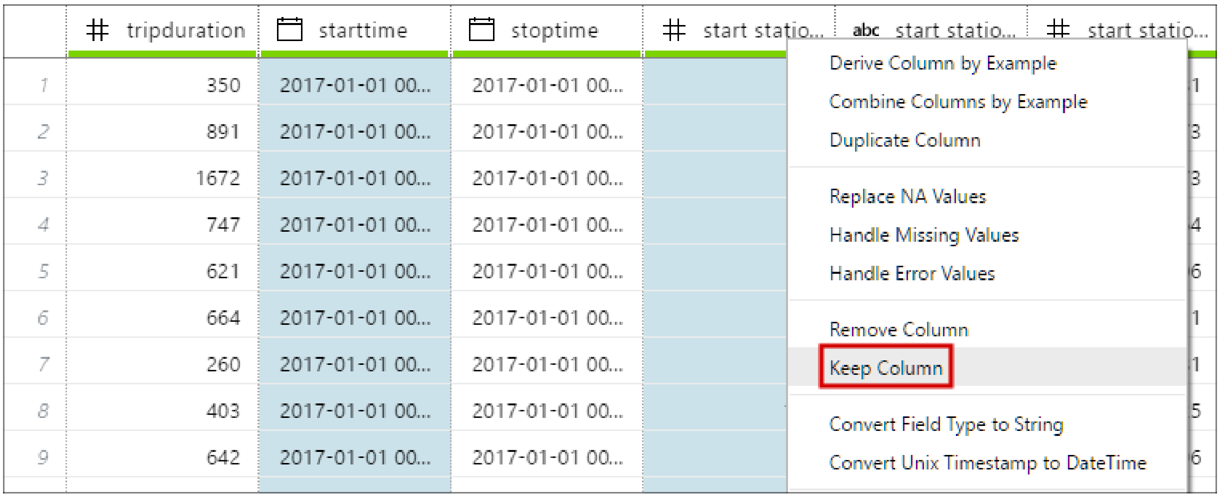
☐ Show halo

☑ Kernel Density Plot Overlay (Gaussian Kernel)

3. Select **OK** to disable the halo effect. Then minimize the histogram.

Remove columns

In the trip data, each row represents a bike pickup event. For this tutorial, you need only the **starttime** and **start station id** columns. To remove the other columns, multi-select these two columns, right-click the column header, and then select **Keep Column**. Other columns are removed.

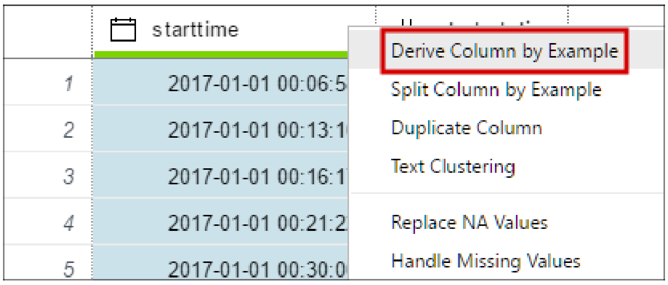


	tripduration	starttime	stoptime	start statio...	start statio...	start statio...
1	350	2017-01-01 00...	2017-01-01 00...			1
2	891	2017-01-01 00...	2017-01-01 00...			3
3	1672	2017-01-01 00...	2017-01-01 00...			3
4	747	2017-01-01 00...	2017-01-01 00...			4
5	621	2017-01-01 00...	2017-01-01 00...			6
6	664	2017-01-01 00...	2017-01-01 00...			1
7	260	2017-01-01 00...	2017-01-01 00...			1
8	403	2017-01-01 00...	2017-01-01 00...			5
9	642	2017-01-01 00...	2017-01-01 00...			6

Summarize data (count)

To summarize bike demand for a two-hour period, use derived columns.

1. Right-click the **starttime** column, and select **Derive Column by Example**.



	starttime
1	2017-01-01 00:06:5
2	2017-01-01 00:13:1
3	2017-01-01 00:16:1
4	2017-01-01 00:21:2
5	2017-01-01 00:30:0

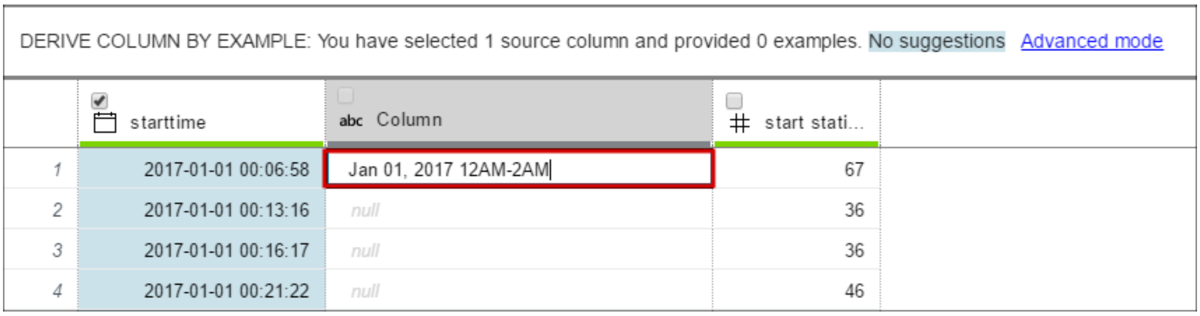
2. For the example, enter a value of **Jan 01, 2017 12AM-2AM** for the first row.

IMPORTANT

In the previous example of deriving columns, you used multiple steps to derive a column that contained the date and time period. In this example, you can see that this operation can be performed as a single step by providing an example of the final output.

NOTE

You can give an example against any of the rows. For this example, the value of **Jan 01, 2017 12AM-2AM** is valid for the first row of data.



DERIVE COLUMN BY EXAMPLE: You have selected 1 source column and provided 0 examples. No suggestions Advanced mode

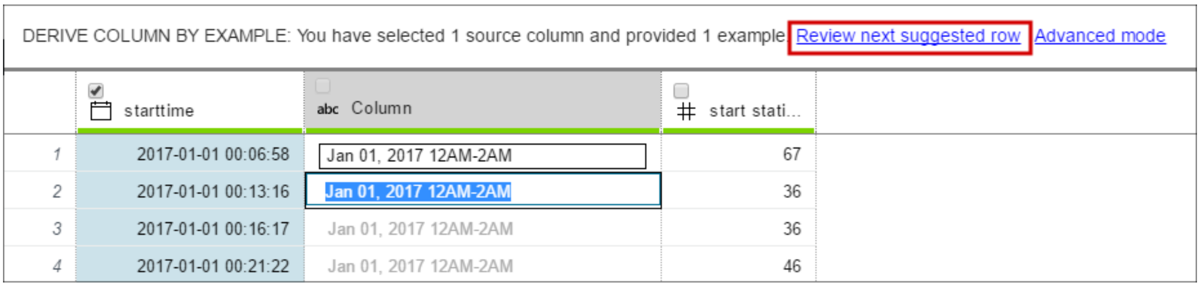
	☑ starttime	□ Column	□ start stati...
1	2017-01-01 00:06:58	Jan 01, 2017 12AM-2AM	67
2	2017-01-01 00:13:16	null	36
3	2017-01-01 00:16:17	null	36
4	2017-01-01 00:21:22	null	46

IMPORTANT

On a Mac, follow this step instead of step 3:

- Go to the first cell that contains **Jan 01, 2017 1AM-2AM**. It should be row 14. Correct the value to **Jan 01, 2017 12AM-2AM**, and select Enter.

3. Wait until the application computes the values against all the rows. The process might take several seconds. After the analysis is finished, use the **Review next suggested row** link to review data.

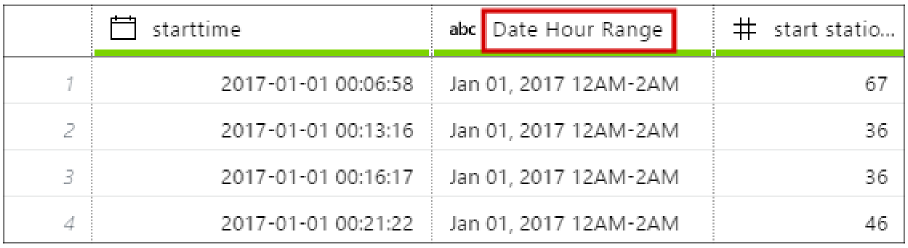


DERIVE COLUMN BY EXAMPLE: You have selected 1 source column and provided 1 example Review next suggested row Advanced mode

	☑ starttime	□ abc Column	□ start stati...
1	2017-01-01 00:06:58	Jan 01, 2017 12AM-2AM	67
2	2017-01-01 00:13:16	Jan 01, 2017 12AM-2AM	36
3	2017-01-01 00:16:17	Jan 01, 2017 12AM-2AM	36
4	2017-01-01 00:21:22	Jan 01, 2017 12AM-2AM	46

Ensure that the computed values are correct. If not, update the value with the expected value, and select Enter. Then wait for the analysis to finish. Complete the **Review next suggested row** process until you see **No suggestions**. **No suggestions** means the application looked at the edge cases and is satisfied with the synthesized program. It's a best practice to perform a visual inspection of the transformed data before you accept the transformation.

4. Select **OK** to accept the transform. Rename the newly created column to **Date Hour Range**.

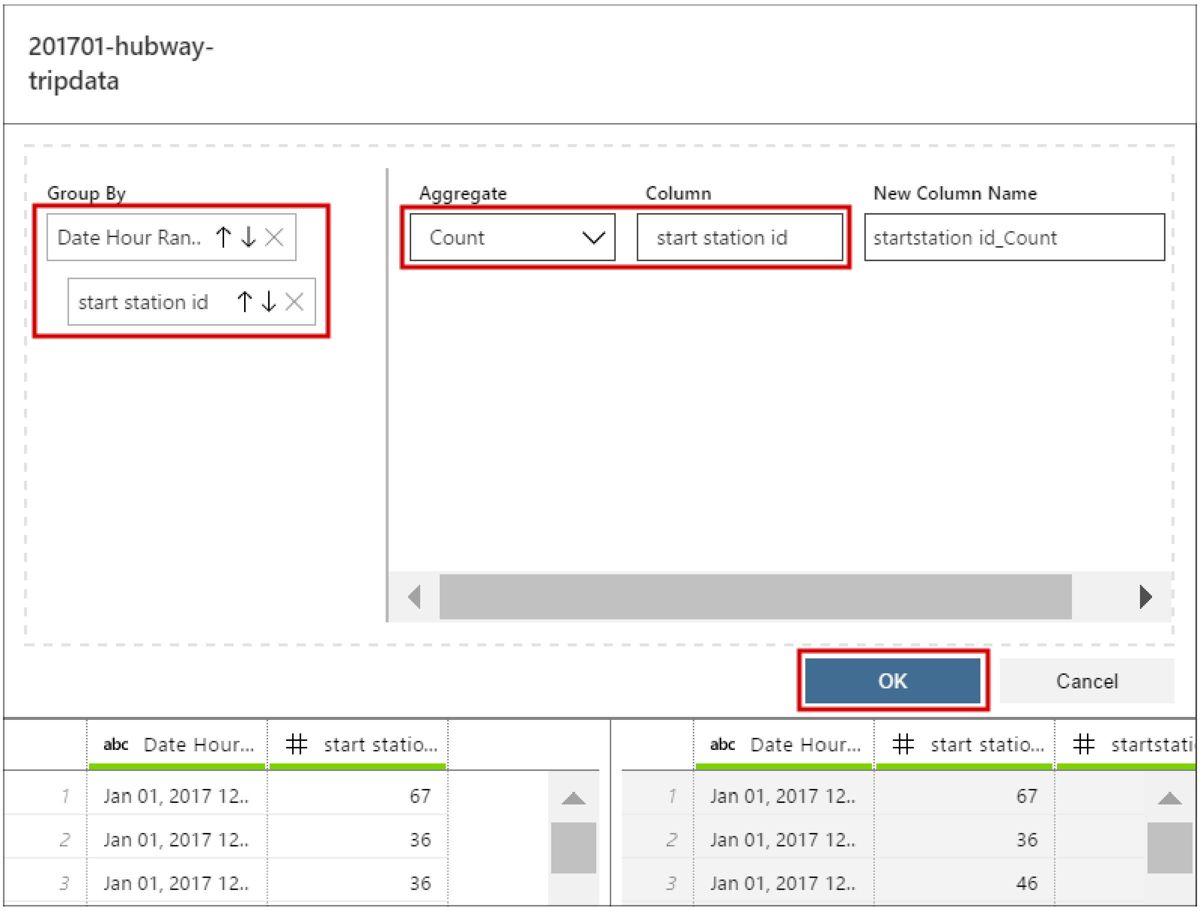


	starttime	abc Date Hour Range	start statio...
1	2017-01-01 00:06:58	Jan 01, 2017 12AM-2AM	67
2	2017-01-01 00:13:16	Jan 01, 2017 12AM-2AM	36
3	2017-01-01 00:16:17	Jan 01, 2017 12AM-2AM	36
4	2017-01-01 00:21:22	Jan 01, 2017 12AM-2AM	46

5. Right-click the **starttime** column header, and select **Remove column**.

6. To summarize the data, on the **Transform** menu, select **Summarize**. To create the transformation, use the following steps:

   - Drag **Date Hour Range** and **start station id** to the **Group By** pane on the left.
   - Drag **start station id** to the **summarize data** pane on the right.


201701-hubway-tripdata

Group By:
- Date Hour Ran..
- start station id

Aggregate	Column	New Column Name
Count	start station id	startstation id_Count

	abc Date Hour...	# start statio...
1	Jan 01, 2017 12..	67
2	Jan 01, 2017 12..	36
3	Jan 01, 2017 12..	36

	abc Date Hour...	# start statio...	# startstati
1	Jan 01, 2017 12..	67	
2	Jan 01, 2017 12..	36	
3	Jan 01, 2017 12..	46	

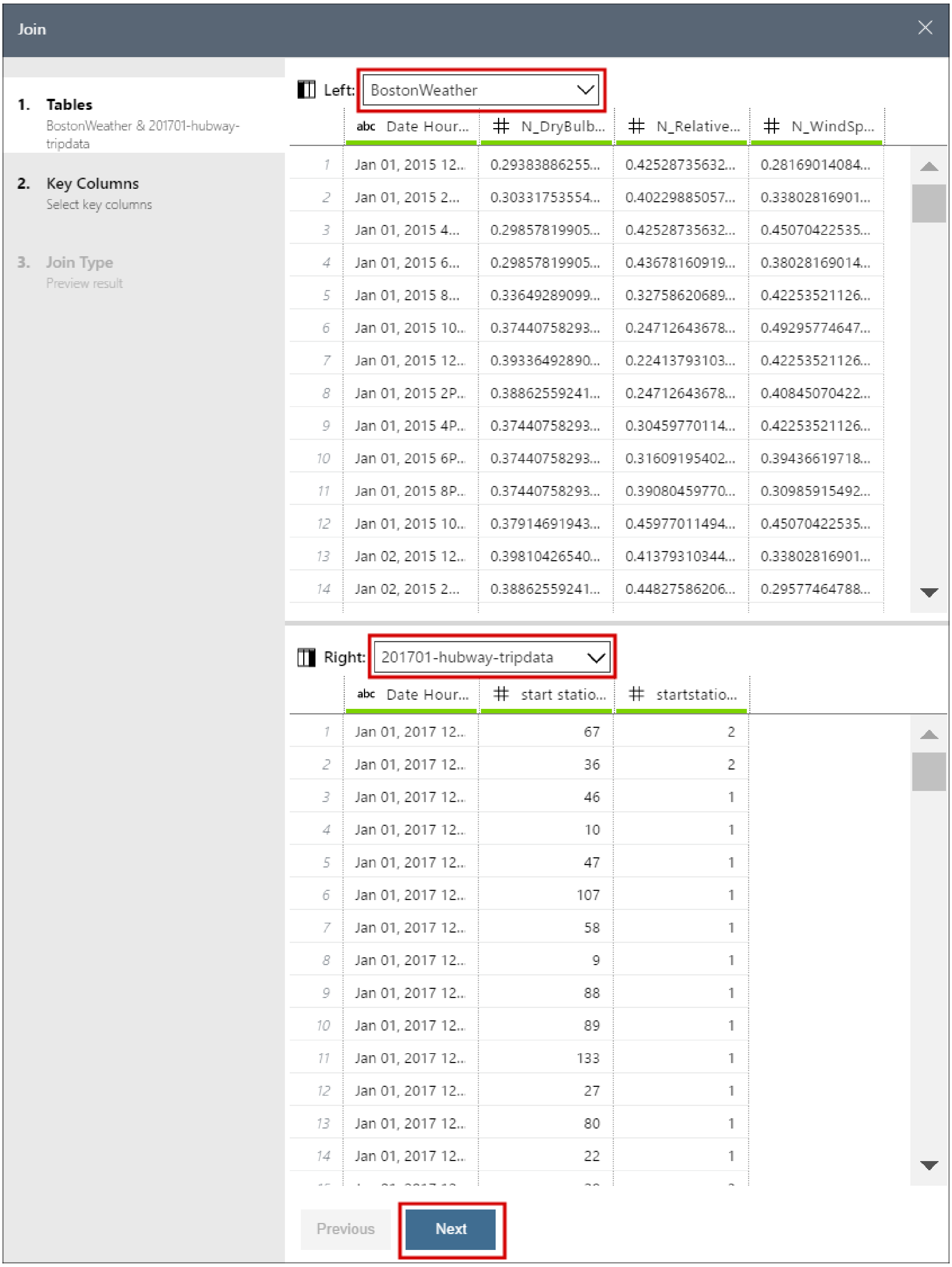
7. Select **OK** to accept the summary result.

Join dataflows

To join the weather data with the trip data, use the following steps:

1. On the **Transforms** menu, select **Join**.

2. **Tables**: Select **BostonWeather** as the **Left** dataflow and **201701-hubway-tripdata** as the **Right** dataflow. To continue, select **Next**.



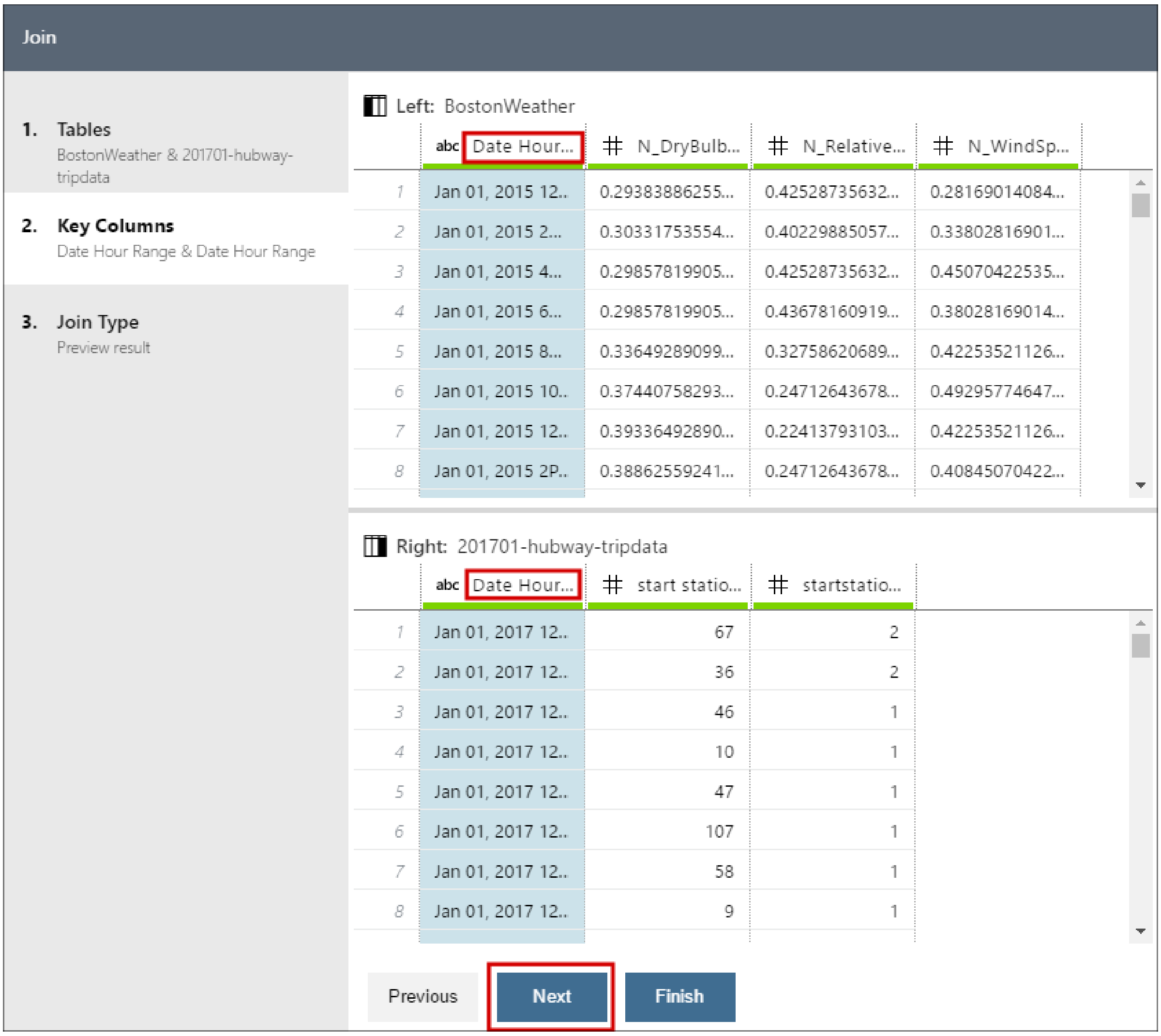
Left: BostonWeather

	abc Date Hour...	# N_DryBulb...	# N_Relative...	# N_WindSp...
1	Jan 01, 2015 12...	0.29383886255...	0.42528735632...	0.28169014084...
2	Jan 01, 2015 2...	0.30331753554...	0.40229885057...	0.33802816901...
3	Jan 01, 2015 4...	0.29857819905...	0.42528735632...	0.45070422535...
4	Jan 01, 2015 6...	0.29857819905...	0.43678160919...	0.38028169014...
5	Jan 01, 2015 8...	0.33649289099...	0.32758620689...	0.42253521126...
6	Jan 01, 2015 10...	0.37440758293...	0.24712643678...	0.49295774647...
7	Jan 01, 2015 12...	0.39336492890...	0.22413793103...	0.42253521126...
8	Jan 01, 2015 2P...	0.38862559241...	0.24712643678...	0.40845070422...
9	Jan 01, 2015 4P...	0.37440758293...	0.30459770114...	0.42253521126...
10	Jan 01, 2015 6P...	0.37440758293...	0.31609195402...	0.39436619718...
11	Jan 01, 2015 8P...	0.37440758293...	0.39080459770...	0.30985915492...
12	Jan 01, 2015 10...	0.37914691943...	0.45977011494...	0.45070422535...
13	Jan 02, 2015 12...	0.39810426540...	0.41379310344...	0.33802816901...
14	Jan 02, 2015 2...	0.38862559241...	0.44827586206...	0.29577464788...

Right: 201701-hubway-tripdata

	abc Date Hour...	# start statio...	# startstatio...
1	Jan 01, 2017 12...	67	2
2	Jan 01, 2017 12...	36	2
3	Jan 01, 2017 12...	46	1
4	Jan 01, 2017 12...	10	1
5	Jan 01, 2017 12...	47	1
6	Jan 01, 2017 12...	107	1
7	Jan 01, 2017 12...	58	1
8	Jan 01, 2017 12...	9	1
9	Jan 01, 2017 12...	88	1
10	Jan 01, 2017 12...	89	1
11	Jan 01, 2017 12...	133	1
12	Jan 01, 2017 12...	27	1
13	Jan 01, 2017 12...	80	1
14	Jan 01, 2017 12...	22	1

3. **Key Columns**: Select the **Date Hour Range** column in both the tables, and then select **Next**.



1. **Tables**
   BostonWeather & 201701-hubway-tripdata
2. **Key Columns**
   Date Hour Range & Date Hour Range
3. **Join Type**
   Preview result

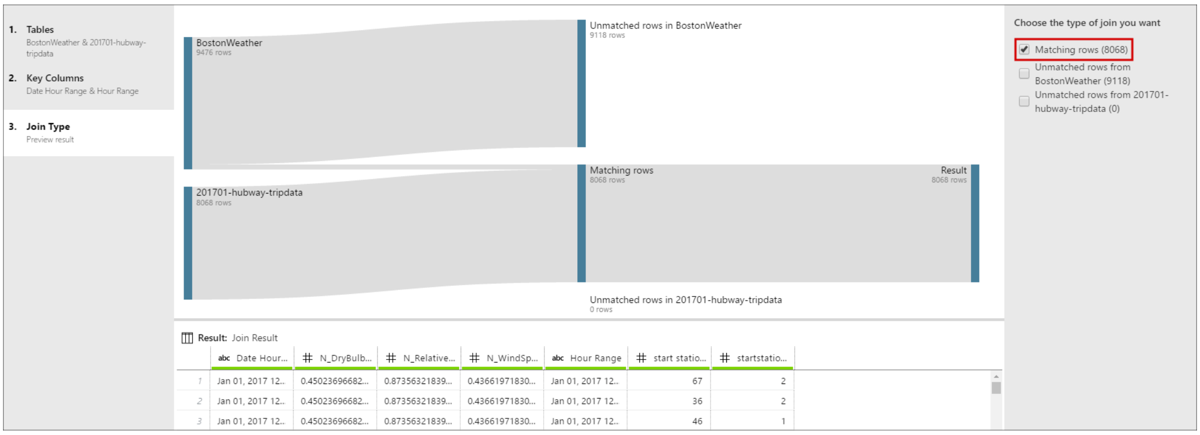
Left: BostonWeather

	abc Date Hour...	# N_DryBulb...	# N_Relative...	# N_WindSp...
1	Jan 01, 2015 12..	0.29383886255...	0.42528735632...	0.28169014084...
2	Jan 01, 2015 2...	0.30331753554...	0.40229885057...	0.33802816901...
3	Jan 01, 2015 4...	0.29857819905...	0.42528735632...	0.45070422535...
4	Jan 01, 2015 6...	0.29857819905...	0.43678160919...	0.38028169014...
5	Jan 01, 2015 8...	0.33649289099...	0.32758620689...	0.42253521126...
6	Jan 01, 2015 10..	0.37440758293...	0.24712643678...	0.49295774647...
7	Jan 01, 2015 12..	0.39336492890...	0.22413793103...	0.42253521126...
8	Jan 01, 2015 2P..	0.38862559241...	0.24712643678...	0.40845070422...

Right: 201701-hubway-tripdata

	abc Date Hour...	# start statio...	# startstatio...
1	Jan 01, 2017 12..	67	2
2	Jan 01, 2017 12..	36	2
3	Jan 01, 2017 12..	46	1
4	Jan 01, 2017 12..	10	1
5	Jan 01, 2017 12..	47	1
6	Jan 01, 2017 12..	107	1
7	Jan 01, 2017 12..	58	1
8	Jan 01, 2017 12..	9	1

4. **Join Type**: Select **Matching rows** as the join type, and then select **Finish**.



1. **Tables**
   BostonWeather & 201701-hubway-tripdata
2. **Key Columns**
   Date Hour Range & Hour Range
3. **Join Type**
   Preview result

Choose the type of join you want

☑ Matching rows (8068)
☐ Unmatched rows from BostonWeather (9118)
☐ Unmatched rows from 201701-hubway-tripdata (0)

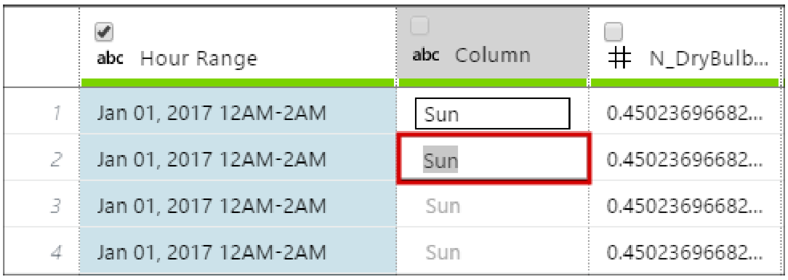
Result: Join Result

	abc Date Hour...	# N_DryBulb...	# N_Relative...	# N_WindSp...	abc Hour Range	# start statio...	# startstatio...
1	Jan 01, 2017 12..	0.45023696682...	0.87356321839...	0.43661971830...	Jan 01, 2017 12..	67	2
2	Jan 01, 2017 12..	0.45023696682...	0.87356321839...	0.43661971830...	Jan 01, 2017 12..	36	2
3	Jan 01, 2017 12..	0.45023696682...	0.87356321839...	0.43661971830...	Jan 01, 2017 12..	46	1

This process creates a new dataflow named **Join Result**.

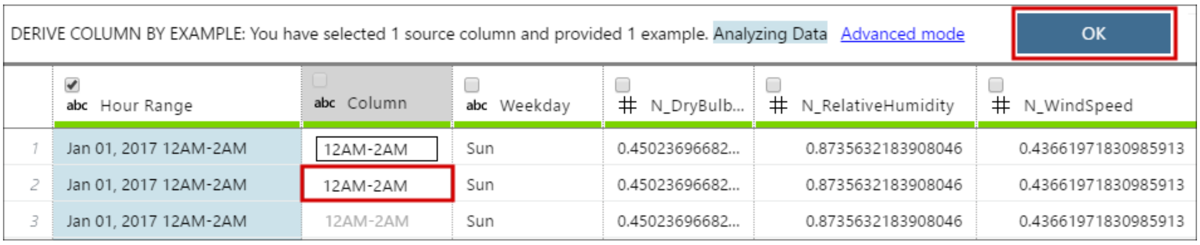
Create additional features

1. To create a column that contains the day of the week, right-click the **Date Hour Range** column and select **Derive Column by Example**. Use a value of **Sun** for a date that occurred on a Sunday. An example is **Jan 01, 2017 12AM-2AM**. Select Enter, and then select **OK**. Rename this column to **Weekday**.



	☑ abc Hour Range	□ abc Column	□ # N_DryBulb...
1	Jan 01, 2017 12AM-2AM	Sun	0.45023696682...
2	Jan 01, 2017 12AM-2AM	Sun	0.45023696682...
3	Jan 01, 2017 12AM-2AM	Sun	0.45023696682...
4	Jan 01, 2017 12AM-2AM	Sun	0.45023696682...

2. To create a column that contains the time period for a row, right-click the **Date Hour Range** column, and select **Derive Column by example**. Use a value of **12AM-2AM** for the row that contains **Jan 01, 2017 12AM-2AM**. Select Enter, and then select **OK**. Rename this column to **Period**.



DERIVE COLUMN BY EXAMPLE: You have selected 1 source column and provided 1 example. Analyzing Data Advanced mode

	☑ abc Hour Range	□ abc Column	□ abc Weekday	□ # N_DryBulb...	□ # N_RelativeHumidity	□ # N_WindSpeed
1	Jan 01, 2017 12AM-2AM	12AM-2AM	Sun	0.45023696682...	0.8735632183908046	0.43661971830985913
2	Jan 01, 2017 12AM-2AM	12AM-2AM	Sun	0.45023696682...	0.8735632183908046	0.43661971830985913
3	Jan 01, 2017 12AM-2AM	12AM-2AM	Sun	0.45023696682...	0.8735632183908046	0.43661971830985913

3. To remove the **Date Hour Range** and **r_Date Hour Range** columns, select Ctrl (Command ⌘ on Mac), and then select each column header. Right-click, and select **Remove Column**.

Read data from Python

You can run a data preparation package from Python or PySpark and retrieve the result as a **Data Frame**.

To generate an example Python script, right-click **BikeShare Data Prep**, and select **Generate Data Access Code File**. The example Python file is created in your **Project Folder** and is also loaded in a tab within Workbench. The following Python script is an example of the code that is generated:

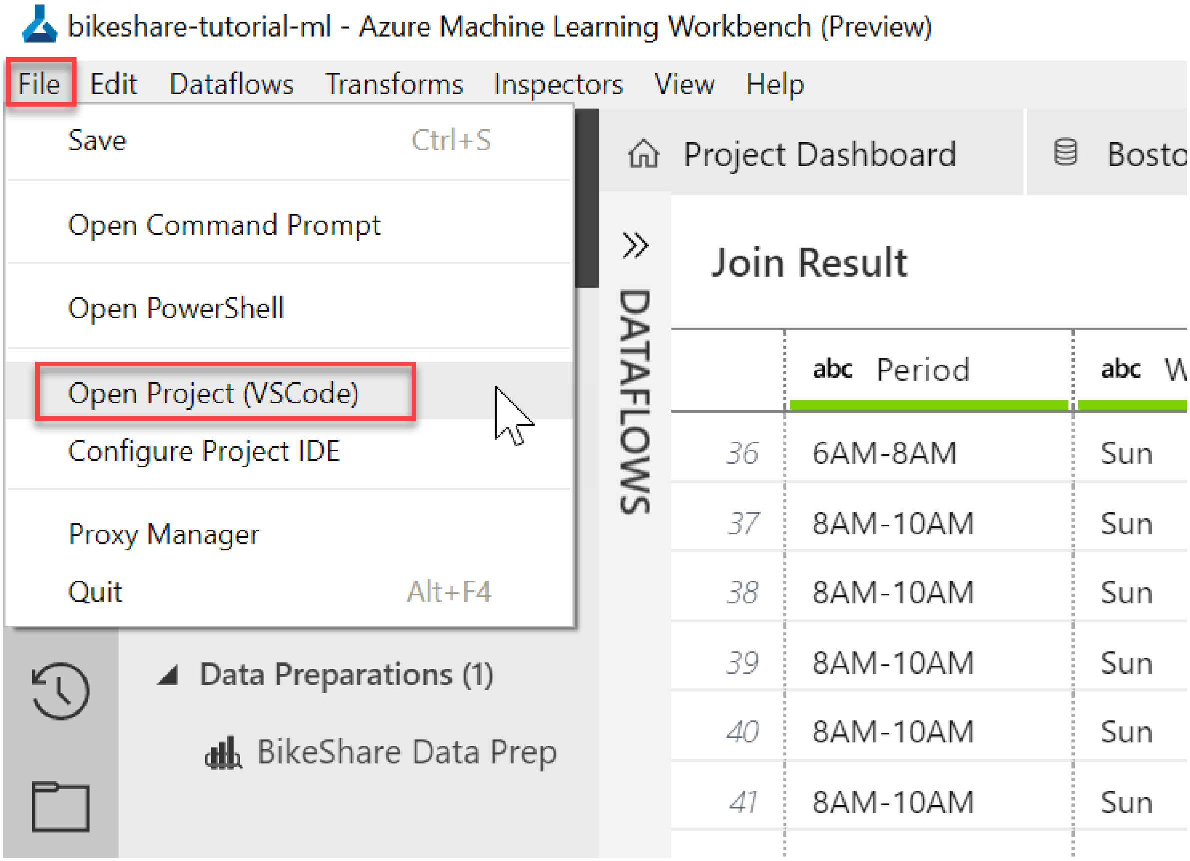
```
# Use the Azure Machine Learning data preparation package
from azureml.dataprep import package

# Use the Azure Machine Learning data collector to log various metrics
from azureml.logging import get_azureml_logger
logger = get_azureml_logger()

# This call will load the referenced package and return a DataFrame.
# If run in a PySpark environment, this call returns a
# Spark DataFrame. If not, it will return a Pandas DataFrame.
df = package.run('BikeShare Data Prep.dprep', dataflow_idx=0)

# Remove this line and add code that uses the DataFrame
df.head(10)
```

For this tutorial, the name of the file is `BikeShare Data Prep.py`. This file is used later in the tutorial.

Save test data as a CSV file

To save the **Join Result** dataflow to a .csv file, you must change the `BikeShare Data Prep.py` script.

1. Open the project for editing in Visual Studio Code.



	Period	W
36	6AM-8AM	Sun
37	8AM-10AM	Sun
38	8AM-10AM	Sun
39	8AM-10AM	Sun
40	8AM-10AM	Sun
41	8AM-10AM	Sun

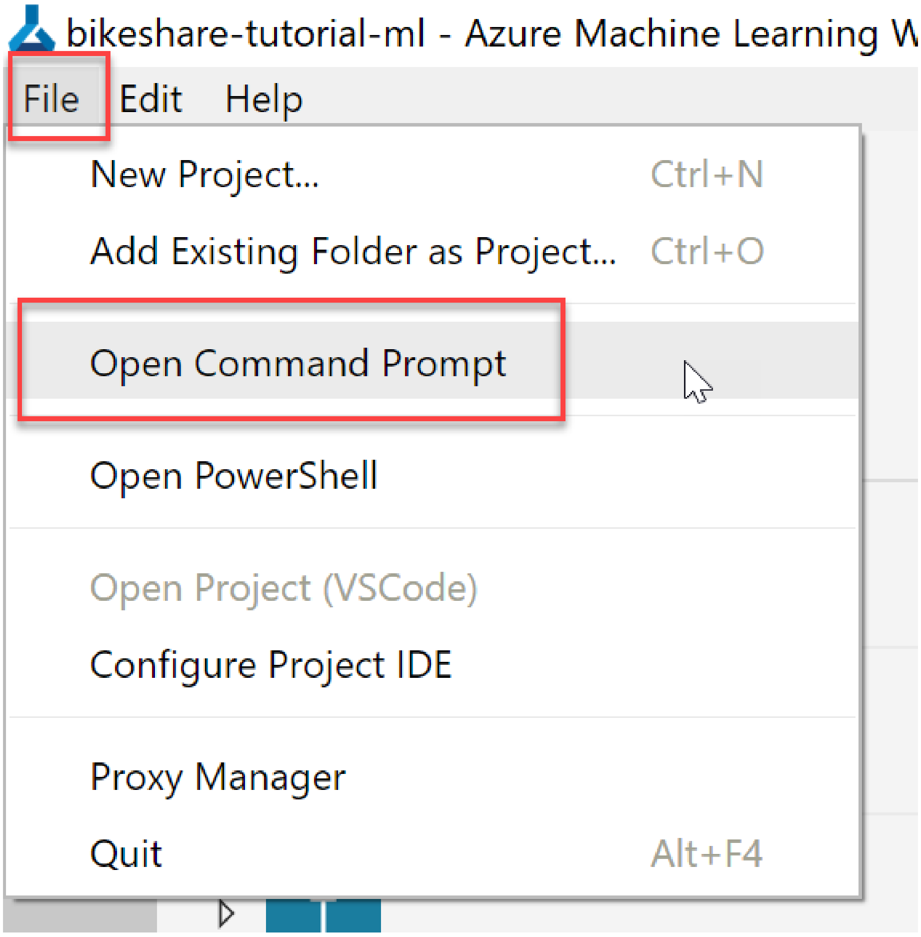
2. Update the Python script in the `BikeShare Data Prep.py` file by using the following code:

```
import pyspark

from azureml.dataprep.package import run
from pyspark.sql.functions import *

# start Spark session
spark = pyspark.sql.SparkSession.builder.appName('BikeShare').getOrCreate()

# dataflow_idx=2 sets the dataflow to the 3rd dataflow (the index starts at 0), the Join Result.
df = run('BikeShare Data Prep.dprep', dataflow_idx=2)
df.show(n=10)
row_count_first = df.count()

# Example file name: 'wasb://data-files@bikesharestorage.blob.core.windows.net/testata'
# 'wasb://<your container name>@<your azure storage name>.blob.core.windows.net/<csv folder name>
blobfolder = 'Your Azure Storage blob path'

df.write.csv(blobfolder, mode='overwrite')

# retrieve csv file parts into one data frame
csvfiles = "<Your Azure Storage blob path>/*.csv"
df = spark.read.option("header", "false").csv(csvfiles)
row_count_result = df.count()
print(row_count_result)
if (row_count_first == row_count_result):
    print('counts match')
else:
    print('counts do not match')
print('done')
```

3. Replace `Your Azure Storage blob path` with the path to the output file to be created. Replace for both the `blobfolder` and `csvfiles` variables.

Create an HDInsight run configuration

1. In Machine Learning Workbench, open the command-line window, select the **File** menu, and then select **Open Command Prompt**. Your command prompt starts in the project folder with the prompt `C:\Projects\BikeShare>`.



> **IMPORTANT**
>
> You must use the command-line window (opened from Workbench) to accomplish the steps that follow.

2. Use the command prompt to sign in to Azure.

   The Workbench app and CLI use independent credential caches when you authenticate against Azure resources. You need to do this only once until the cached token expires. The `az account list` command returns the list of subscriptions available to your login. If there is more than one, use the ID value from the desired subscription. Set that subscription as the default account to use with the `az account set -s` command, and then provide the subscription ID value. Then confirm the setting by using the account `show` command.

```
REM login by using the aka.ms/devicelogin site
az login

REM lists all Azure subscriptions you have access to
az account list -o table

REM sets the current Azure subscription to the one you want to use
az account set -s <subscriptionId>

REM verifies that your current subscription is set correctly
az account show
```

3. Create the HDInsight run config. You need the name of your cluster and the `sshuser` password.

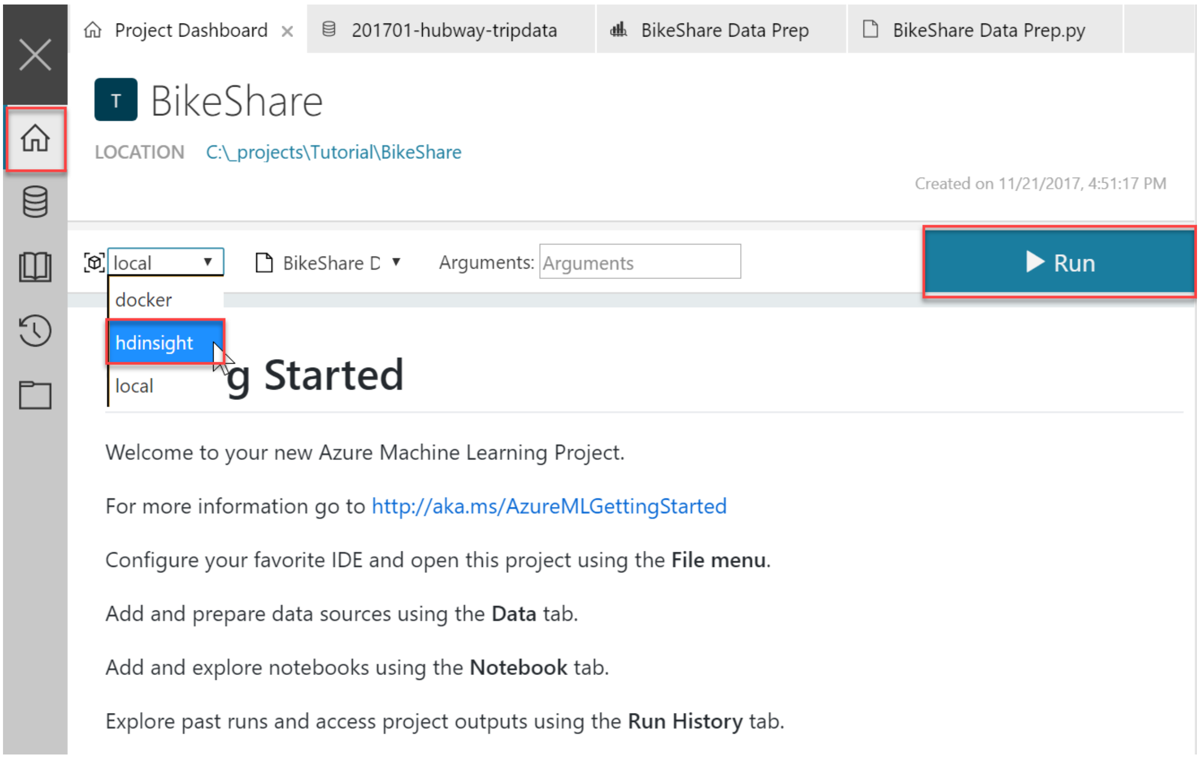
```
az ml computetarget attach cluster --name hdinsight --address <yourclustername>.azurehdinsight.net --
username sshuser --password <your password>
az ml experiment prepare -c hdinsight
```

> **NOTE**
>
> When a blank project is created, the default run configurations are **local** and **docker**. This step creates a new run configuration that is available in Workbench when you run your scripts.

Run in an HDInsight cluster

Return to the Machine Learning Workbench application to run your script in the HDInsight cluster.

1. Return to the home screen of your project by selecting the **Home** icon on the left.
2. Select **hdinsight** from the drop-down list to run your script in the HDInsight cluster.
3. Select **Run**. The script is submitted as a job. The job status changes to **Completed** after the file is written to the specified location in your storage container.



g Started

Welcome to your new Azure Machine Learning Project.

For more information go to http://aka.ms/AzureMLGettingStarted

Configure your favorite IDE and open this project using the **File menu**.

Add and prepare data sources using the **Data** tab.

Add and explore notebooks using the **Notebook** tab.

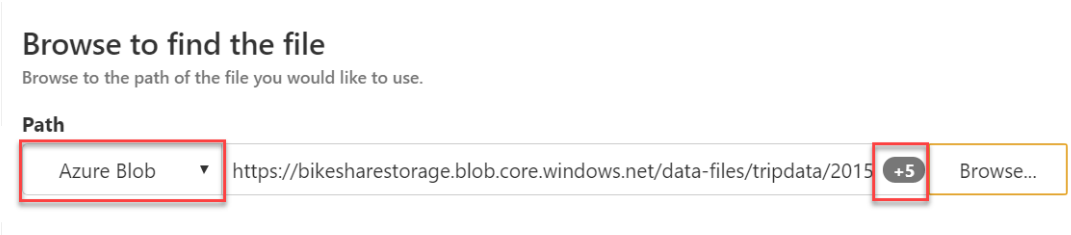
Explore past runs and access project outputs using the **Run History** tab.

Substitute data sources

In the previous steps, you used the `201701-hubway-tripdata.csv` and `BostonWeather.csv` data sources to prepare the test data. To use the package with the other trip data files, use the following steps:

1. Create a new data source by using the steps given previously, with the following changes to the process:

   - **File Selection**: When you select a file, multi-select the six remaining trip tripdata .csv files.

     Browse to find the file

     Browse to the path of the file you would like to use.

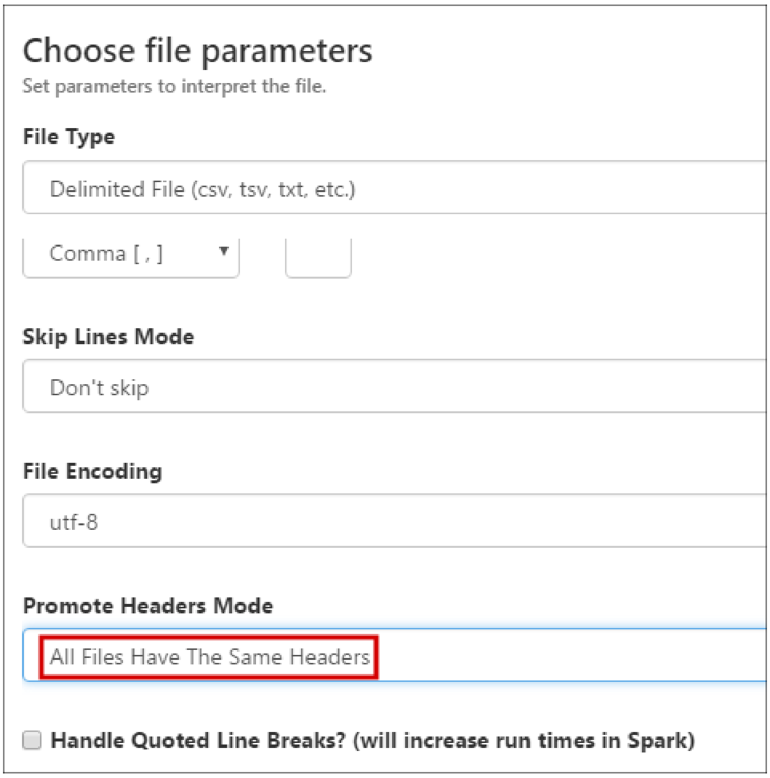
     Path

     Azure Blob ▾ https://bikesharestorage.blob.core.windows.net/data-files/tripdata/2015 +5

     NOTE

     The **+5** entry indicates that there are five additional files beyond the one that is listed.

   - **File Details**: Set **Promote Headers Mode** to **All Files Have The Same Headers**. This value indicates that each of the files contains the same header.

     Choose file parameters

     Set parameters to interpret the file.

     File Type: Delimited File (csv, tsv, txt, etc.)

     Comma [,] ▾

     Skip Lines Mode: Don't skip

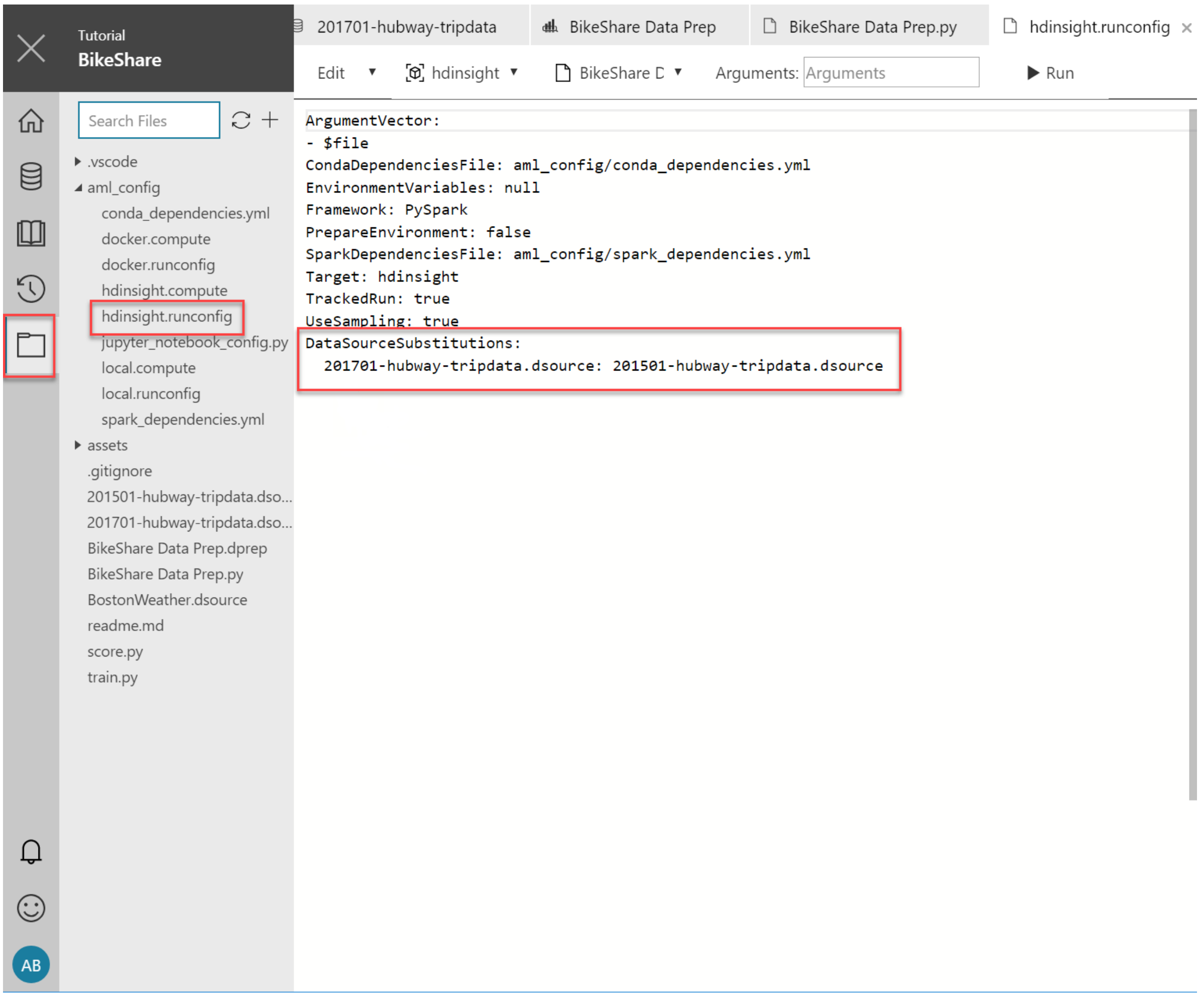
     File Encoding: utf-8

     Promote Headers Mode: All Files Have The Same Headers

     □ Handle Quoted Line Breaks? (will increase run times in Spark)

   Save the name of this data source because it's used in later steps.

2. Select the folder icon to view the files in your project. Expand the **aml_config** directory, and then select the `hdinsight.runconfig` file.



```
ArgumentVector:
- $file
CondaDependenciesFile: aml_config/conda_dependencies.yml
EnvironmentVariables: null
Framework: PySpark
PrepareEnvironment: false
SparkDependenciesFile: aml_config/spark_dependencies.yml
Target: hdinsight
TrackedRun: true
UseSampling: true
DataSourceSubstitutions:
  201701-hubway-tripdata.dsource: 201501-hubway-tripdata.dsource
```

3. Select the **Edit** button to open the file in Visual Studio Code.

4. Add the following lines at the end of the `hdinsight.runconfig` file, and then select the disk icon to save the file.

   ```
   DataSourceSubstitutions:
     201701-hubway-tripdata.dsource: 201501-hubway-tripdata.dsource
   ```

   This change replaces the original data source with the one that contains the six trip data files.

Save training data as a CSV file

1. Browse to the Python file `BikeShare Data Prep.py` that you edited previously. Provide a different file path to save the training data.