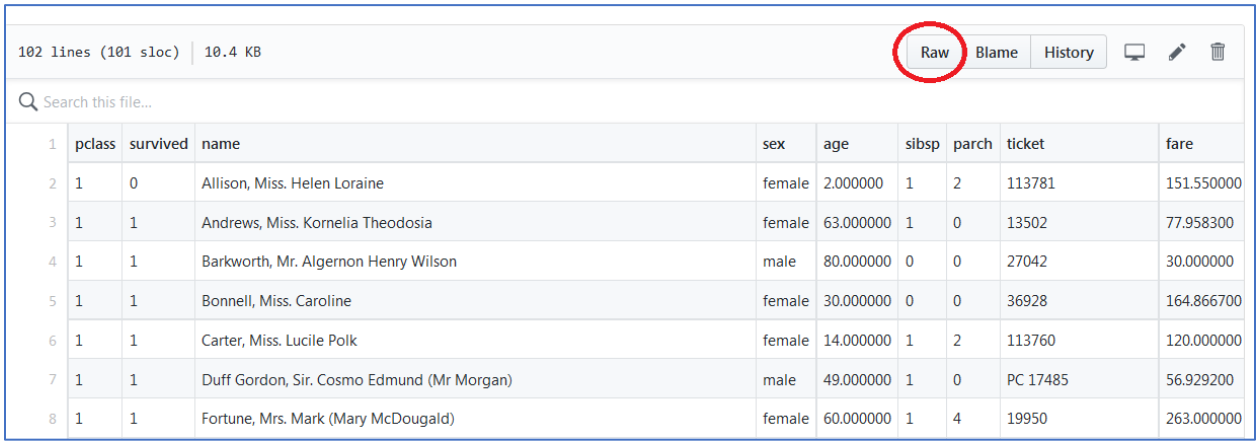
Data Refinery Lab

This lab will use the Titanic data set to demonstrate data profiling, data visualization, and data preparation capabilities of the Data Refinery tool. The lab consists of the following steps:

1. Use the Data Refinery Tool to:
    a. Profile the data to help determine missing values
    b. Visualize the data to gain a better understanding
    c. Prepare the data for modeling
    d. Run the sequence of data preparation operations on the entire data set.

Step 1: Adding a Data Asset to the DSX Local Labs project.

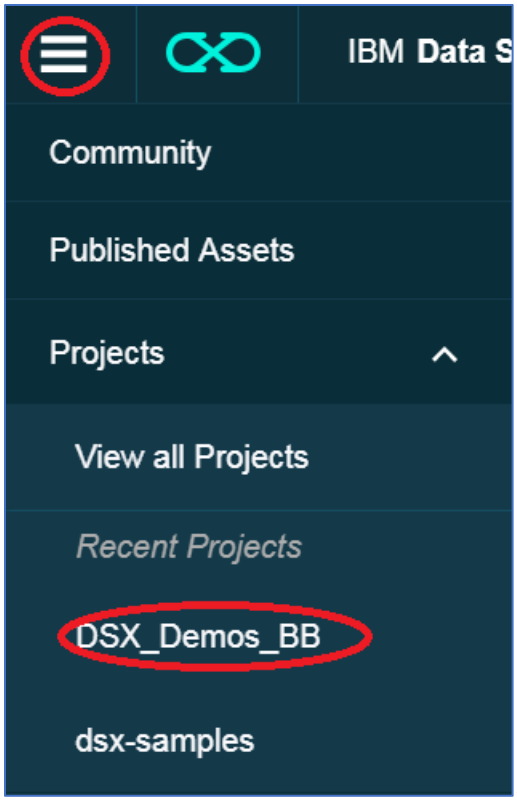
1. Download a file that contains a random sample of the Titanic data file (note the Data Refinery currently operates on 100 records of a dataset for interactive operation) by clicking on the link Titanic Random Data Set and following the instructions below.



	pclass	survived	name	sex	age	sibsp	parch	ticket	fare
2	1	0	Allison, Miss. Helen Loraine	female	2.000000	1	2	113781	151.550000
3	1	1	Andrews, Miss. Kornelia Theodosia	female	63.000000	1	0	13502	77.958300
4	1	1	Barkworth, Mr. Algernon Henry Wilson	male	80.000000	0	0	27042	30.000000
5	1	1	Bonnell, Miss. Caroline	female	30.000000	0	0	36928	164.866700
6	1	1	Carter, Miss. Lucile Polk	female	14.000000	1	2	113760	120.000000
7	1	1	Duff Gordon, Sir. Cosmo Edmund (Mr Morgan)	male	49.000000	1	0	PC 17485	56.929200
8	1	1	Fortune, Mrs. Mark (Mary McDougald)	female	60.000000	1	4	19950	263.000000

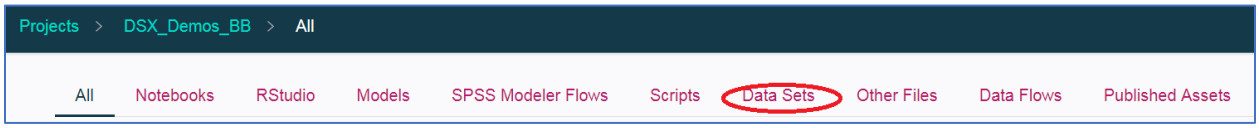
Right click on Raw and click on Save link as ….

2. Go to the DSX Local project. Click on the hamburger icon  , and then click on Projects and then the DSX_Demos_XX project.

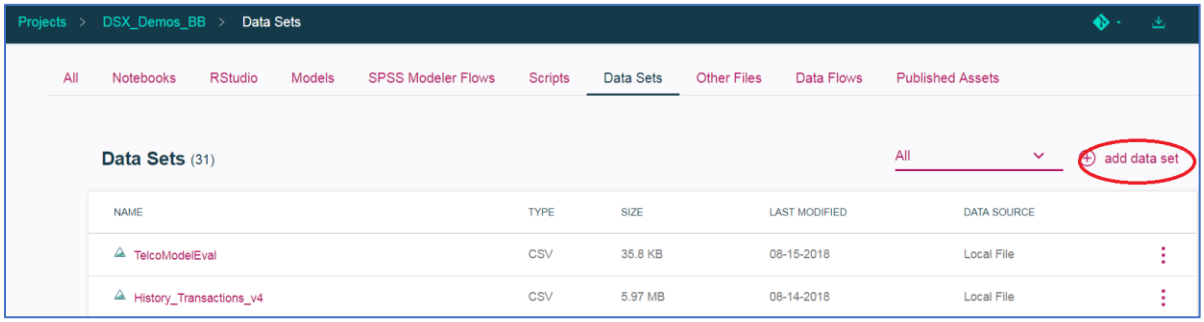


3. Click on **Assets.**

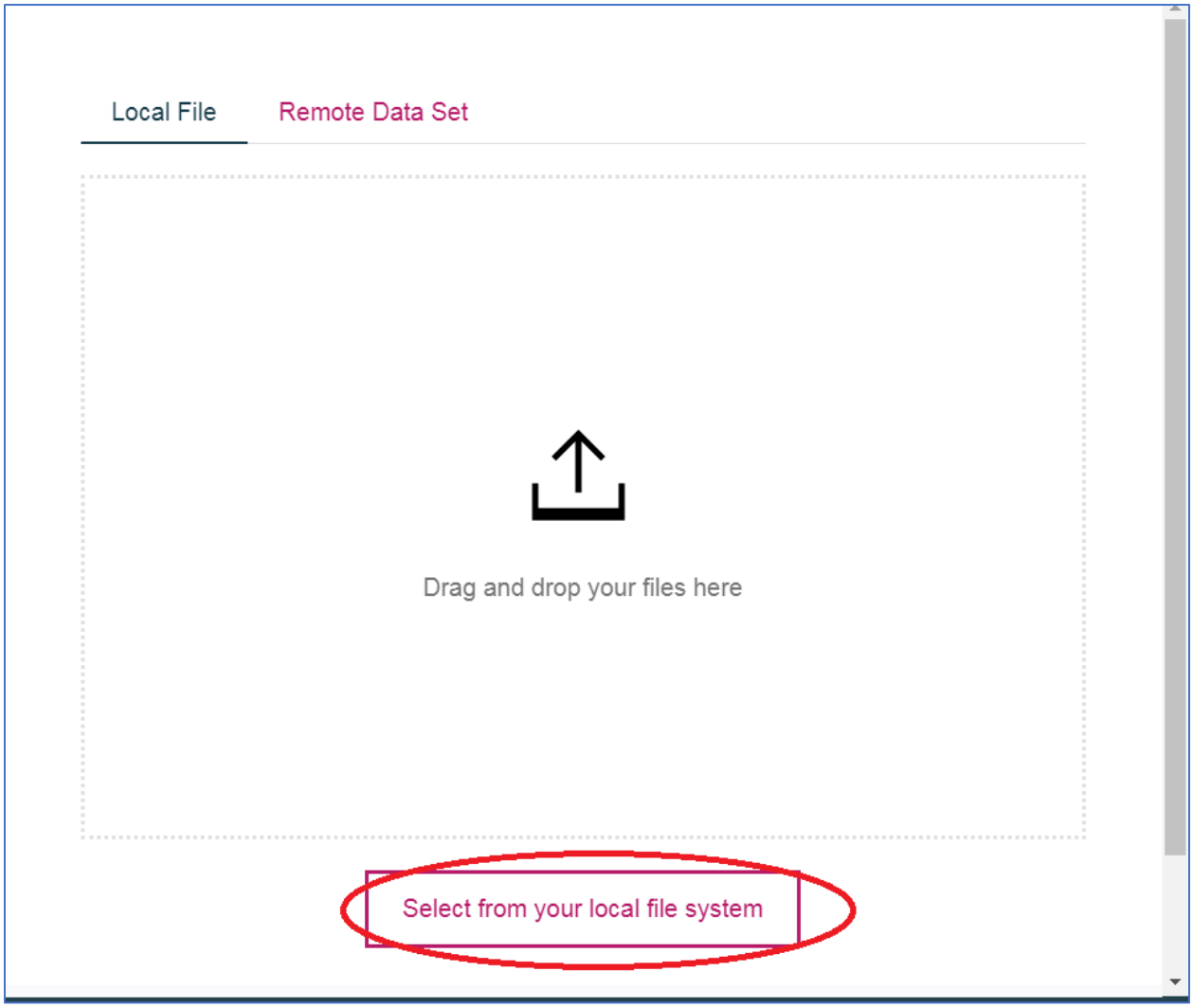
4. Click on **Data Sets**.



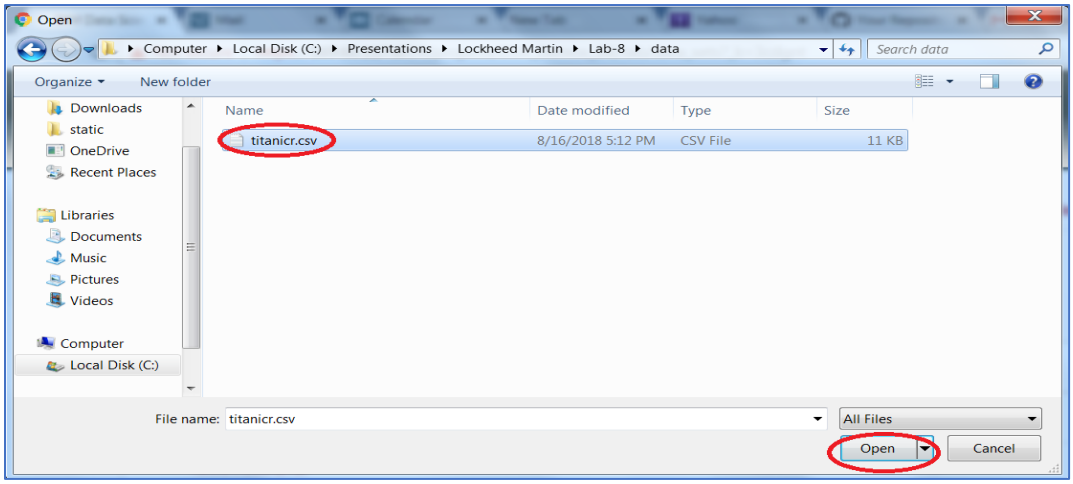
5. Click on **add data set**.



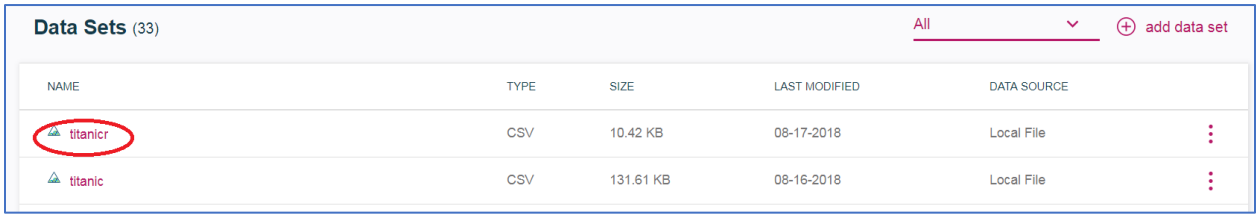
6. Click on **Select from your local file system**.



7. Navigate to the place where the **titanicr.csv** was downloaded. Select the **titanicr.csv** file and click **Open**.

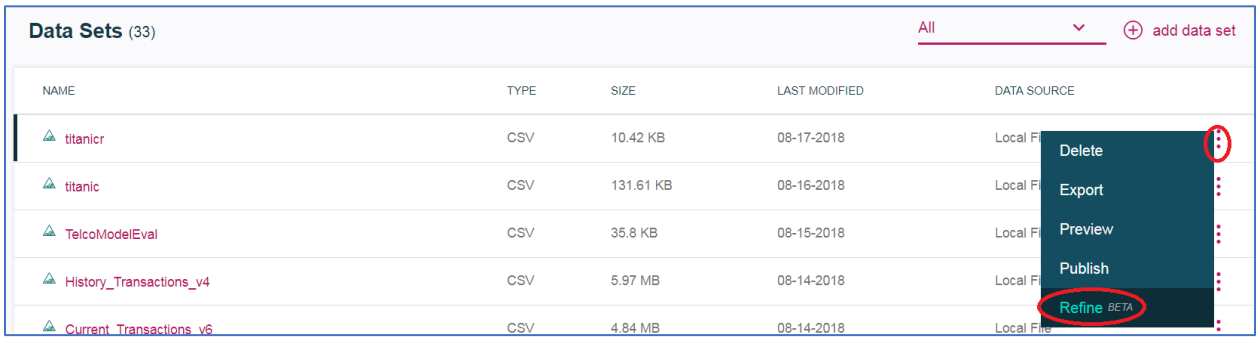


8. The titanicr.csv file is loaded into the project data sets.

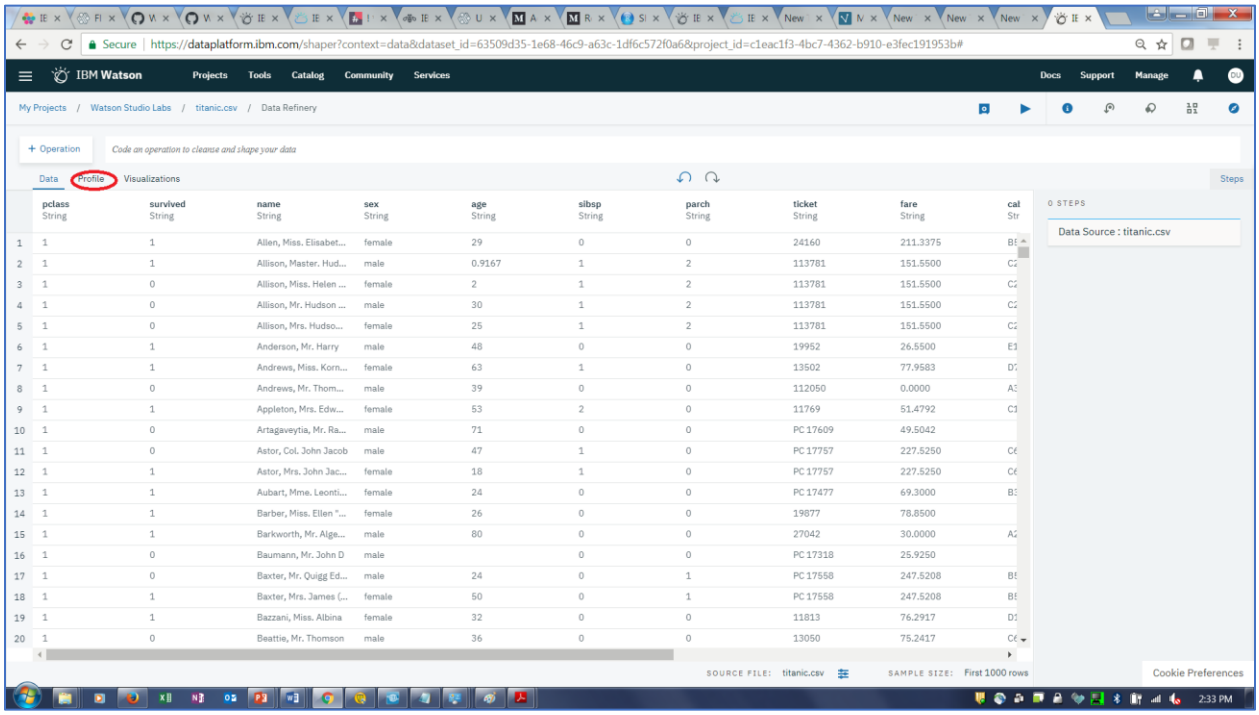


Step 2: Profile the data to help determine missing values.

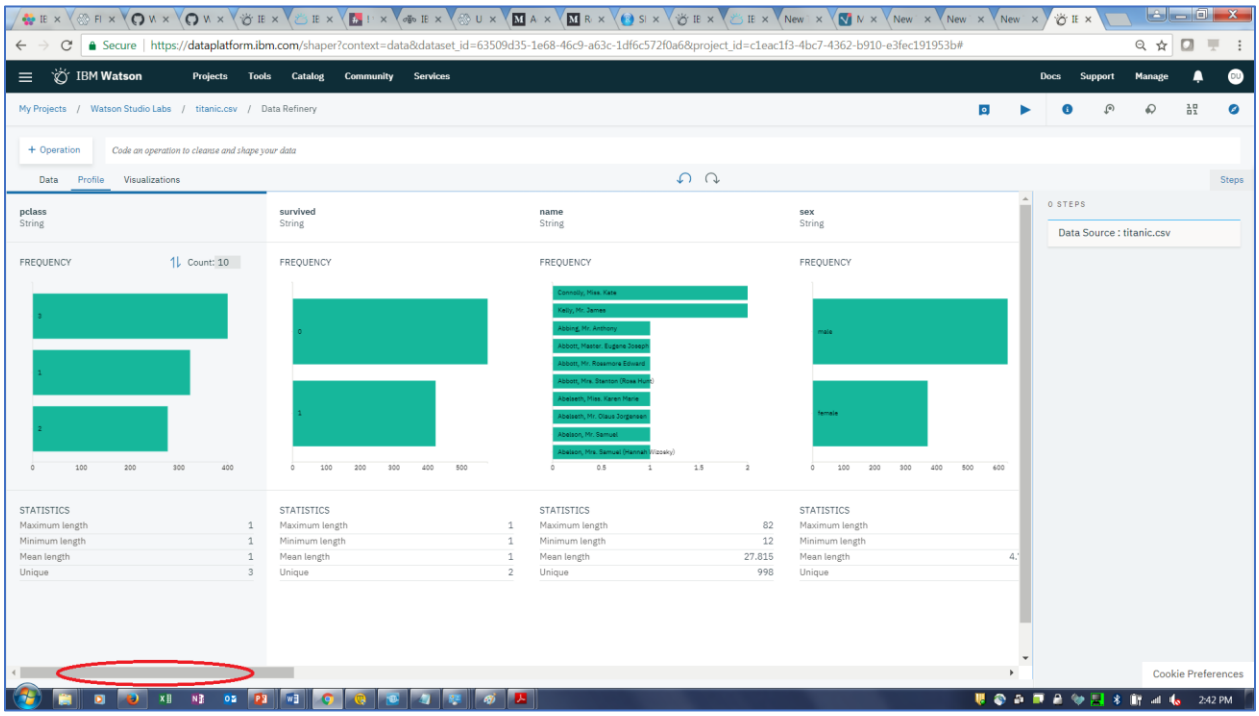
1. Add a Data Flow by clicking on the vertical ellipse ⋮ and then click on **Refine**.



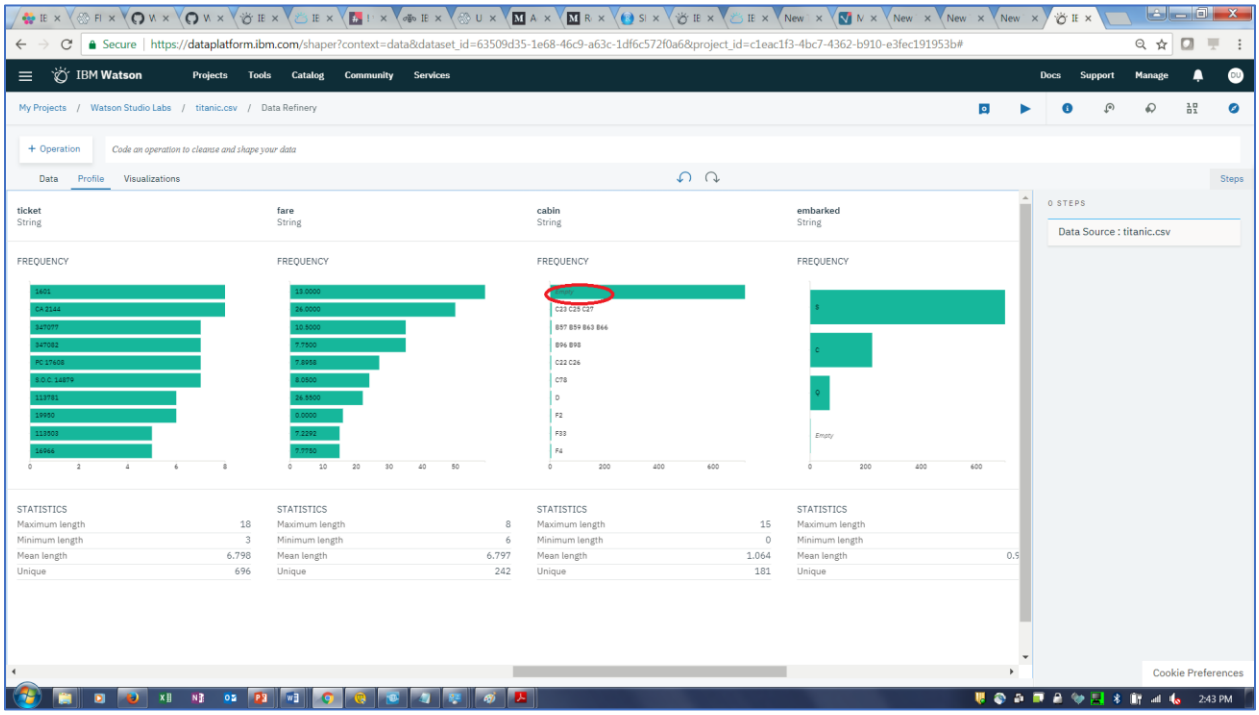
2. The Data Refinery panel will display the data set. Click on the **Profile** tab.



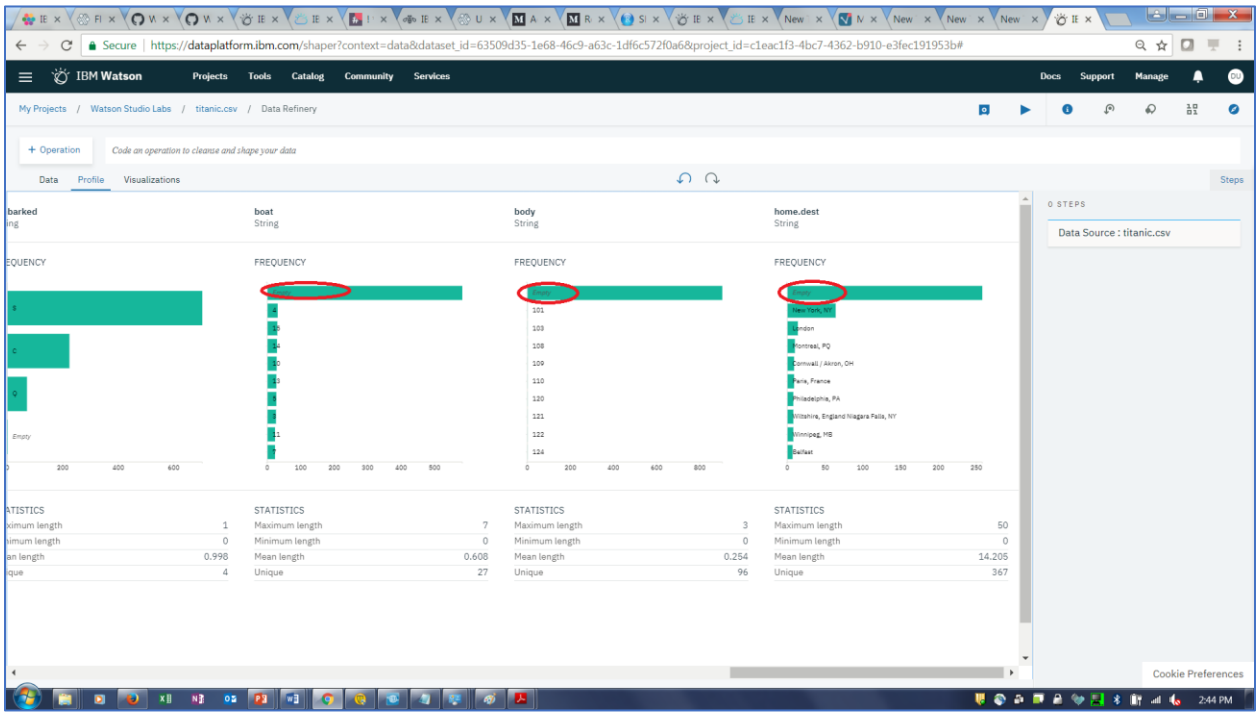
3. The Profile panel displays the counts of the top 10 count values for each column. Note that you can change 10 to another number if desired. You can also switch to the bottom 10 counts for a column. Scroll to the right to view the cabin column.



4. Note that the cabin column has many missing values and should be removed as part of the data preparation step.

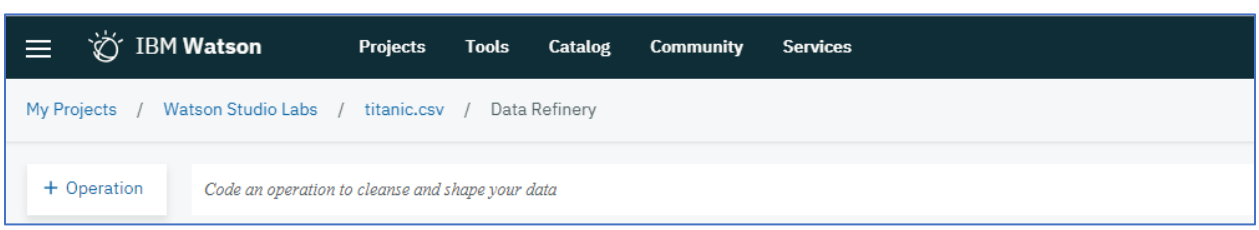


5. In a similar fashion, scroll to the right to examine the boat, body, and home.dest columns. These also have many missing values and should be removed as part of the data preparation step.



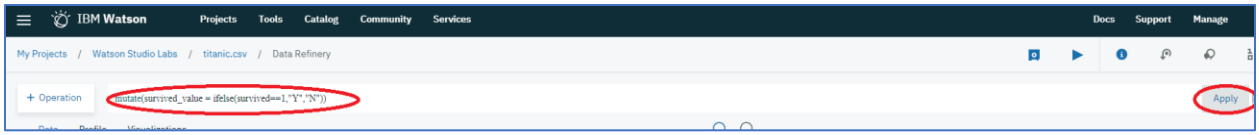
6. Age and Embarked also have missing values. Embarked has only 1 missing value. Age has over 10 missing values, but we will keep that column in the analysis. As part of data preparation, we will remove the rows that contain the missing age and embarked values.

7. Click on the **Data** tab. We will add columns that contain more readable values for the survived and pclass columns. The column survived_value will contain a "Y" or "N". The pclass_value column will contain "first", "second", or "third". We will use the mutate (R dpylr function) and ifelse functions to do the conversion. Click on the **Code an operation to cleanse and shape your data.**

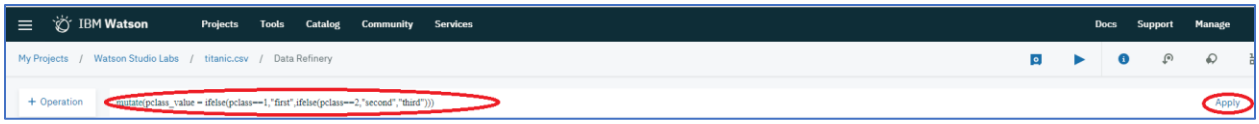


8. Type the following:
   mutate(survived_value=ifelse(survived==1, "Y", "N"))

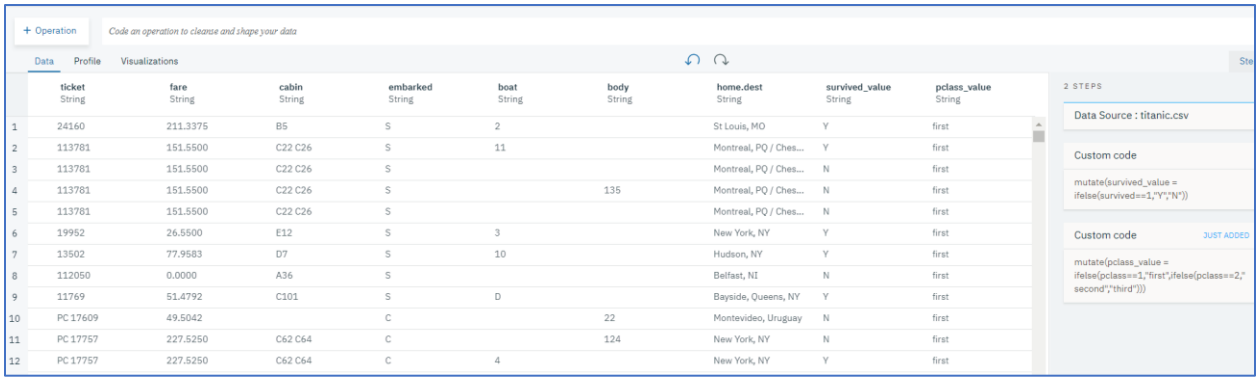
   and then click Apply. If you scroll to the right you should see the new column "survived_value".



9. Type the following to create pclass_value, mutate(pclass_value=ifelse(pclass==1,"first",ifelse(pclass==2,"second","third")))

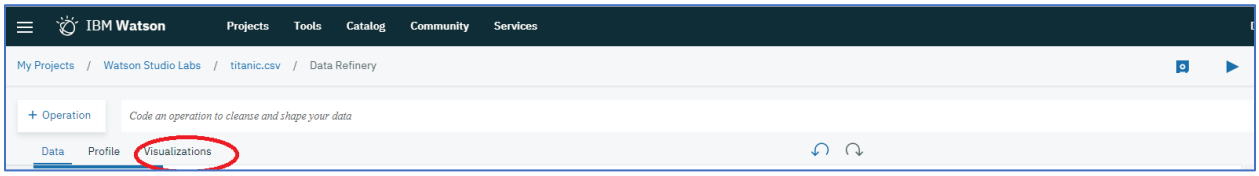


10. The result is shown below. Notice that the right panel will contain a running list of the transformations.

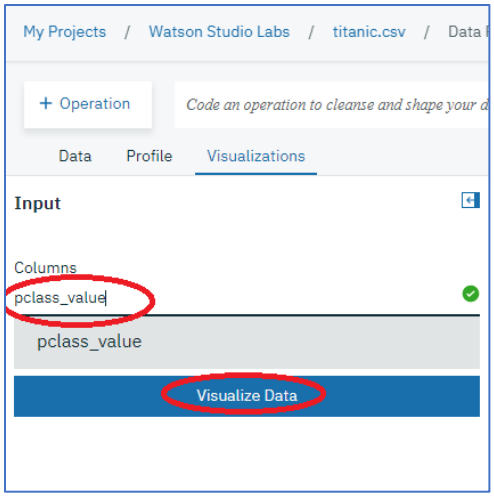


Step 3: Visualize the data to get a better understanding

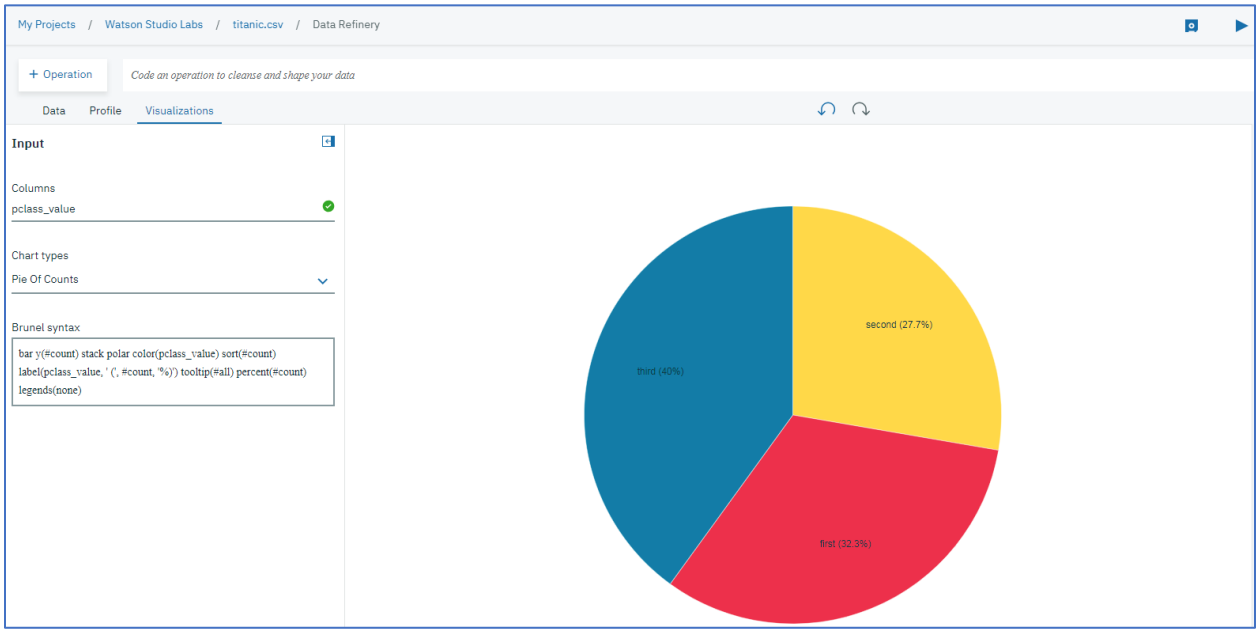
1. Click on the **Visualizations** tab.



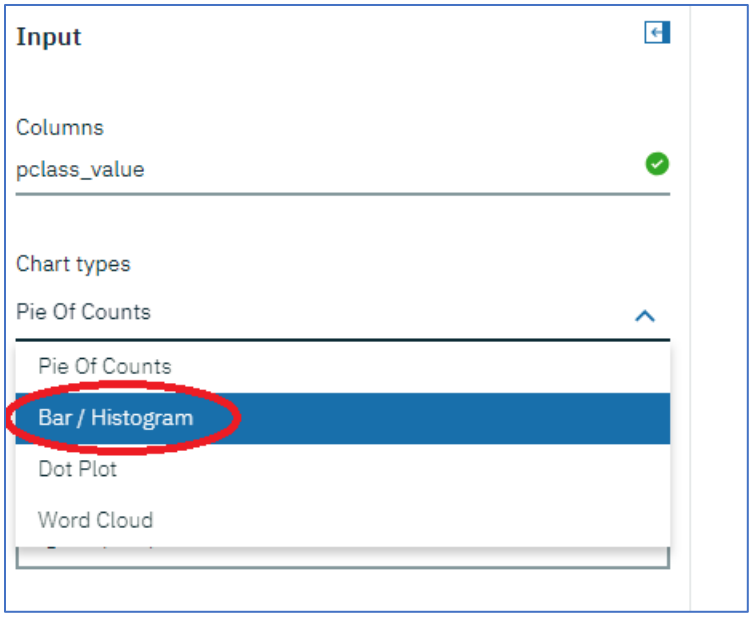
2. Let's take a look at the breakdown of passengers by passenger class. We will use our new pclass_value field. Enter or select pclass_value and then click **Visualize Data**



3. The result is shown below.

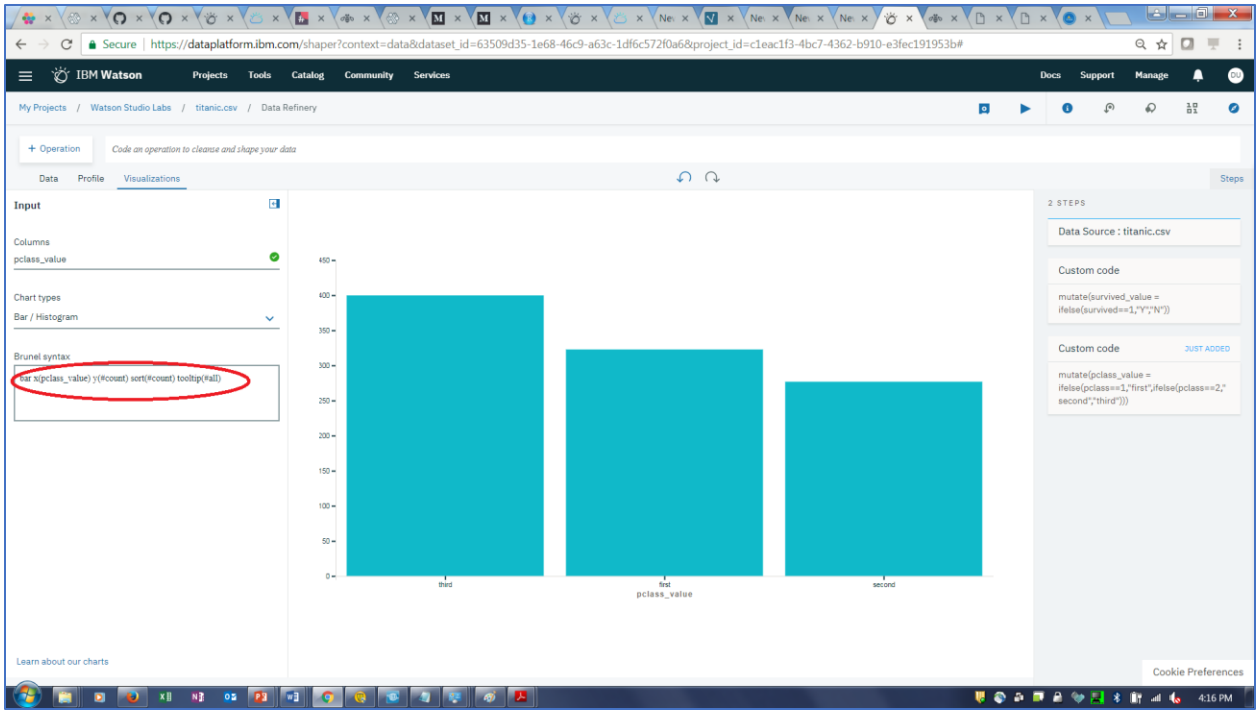


4. We can switch this to a bar chart, by switching the Chart type.

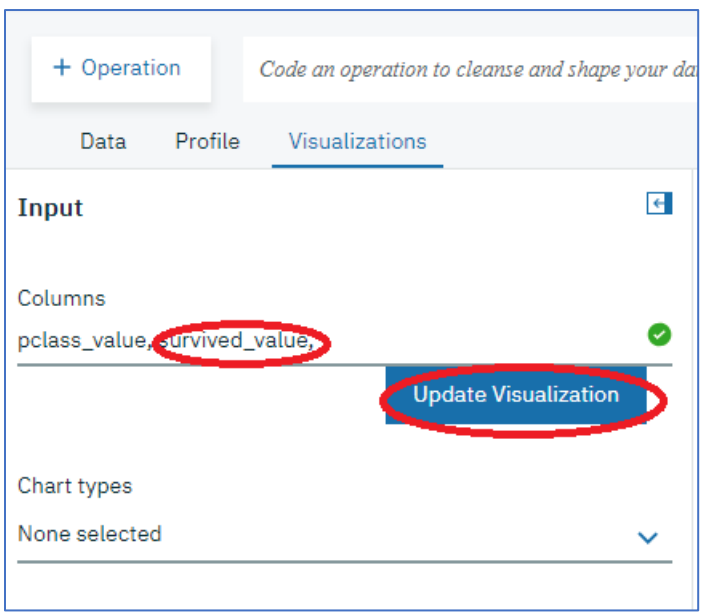


5. The result is shown below. Note the Brunel coding syntax. According to the Brunel github repo, *Brunel defines a highly succinct and novel language that defines interactive data visualizations based on tabular data. The language is well suited for both data scientists and more aggressive business users. The system interprets the language and produces visualizations using the user's choice of existing lower-level visualization technologies typically used by application engineers such as RAVE or D3. It can operate stand-alone and integrated into Jupyter (IPython) notebooks with further integrations as well as other low-level rendering support depending on the desires of the community.*

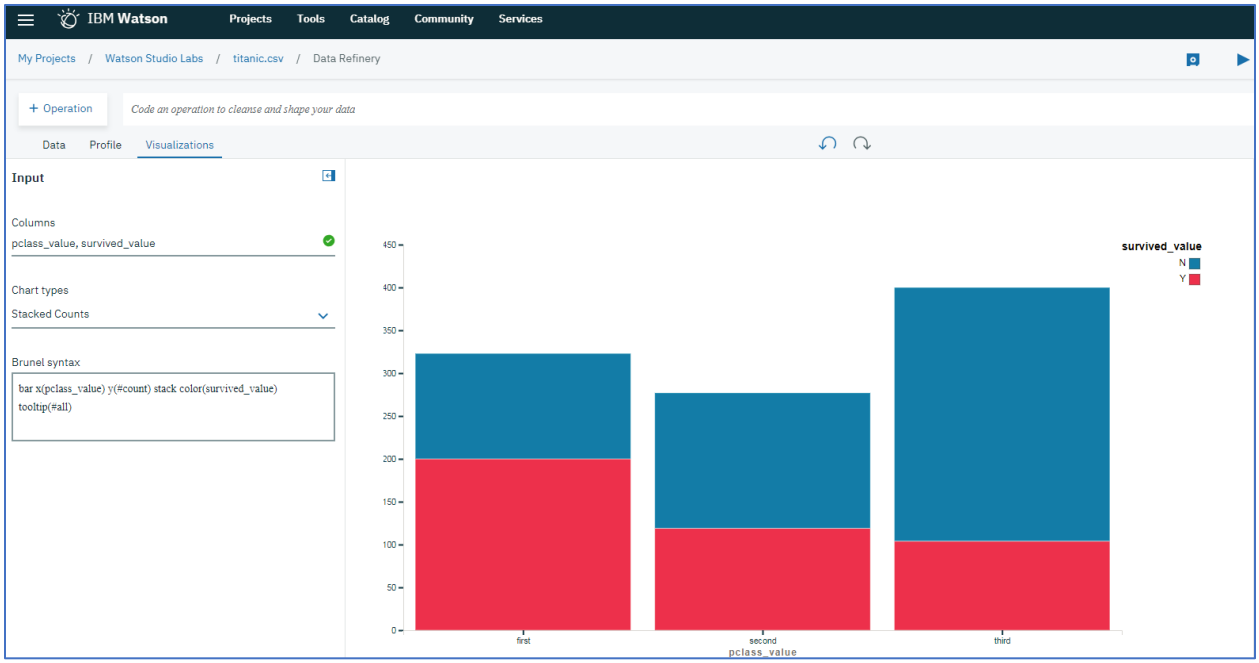
   If you understand the syntax, you can make changes and update the visualization.



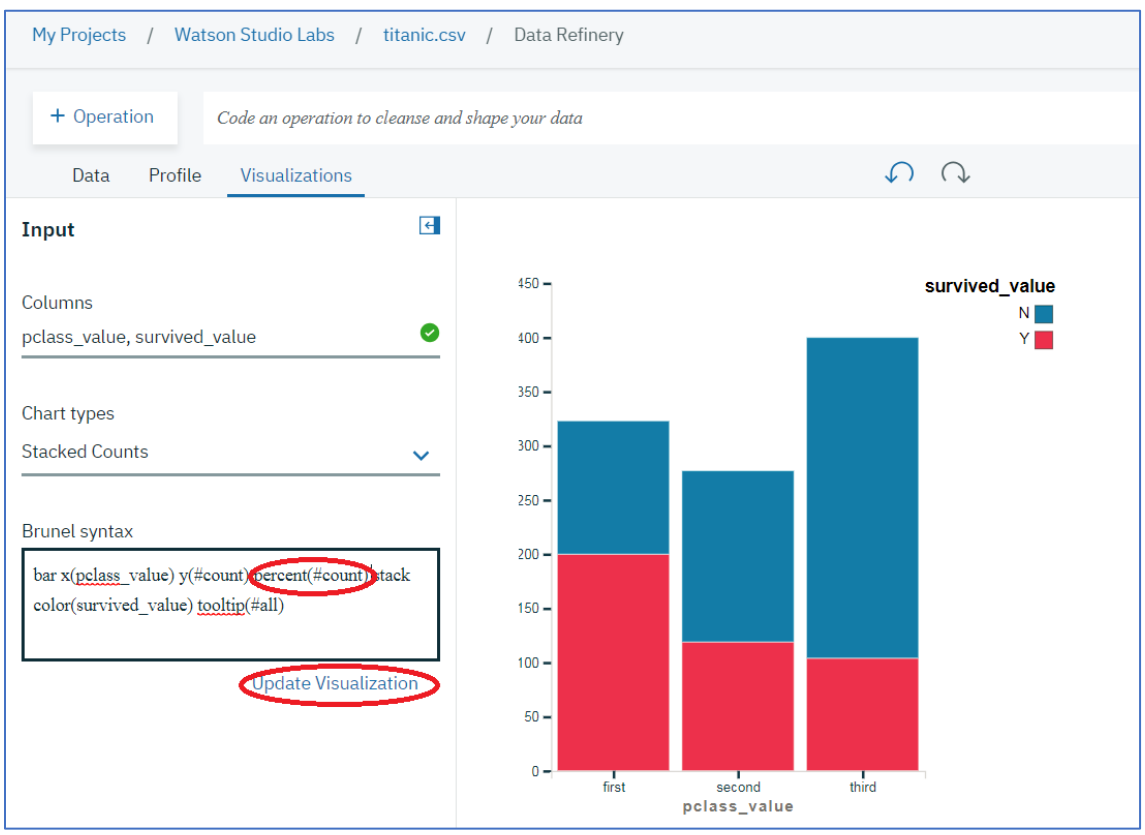
6. Let's examine the relationship between survival and the passenger class. We will add the survived_value and click **Update Visualizataion**.



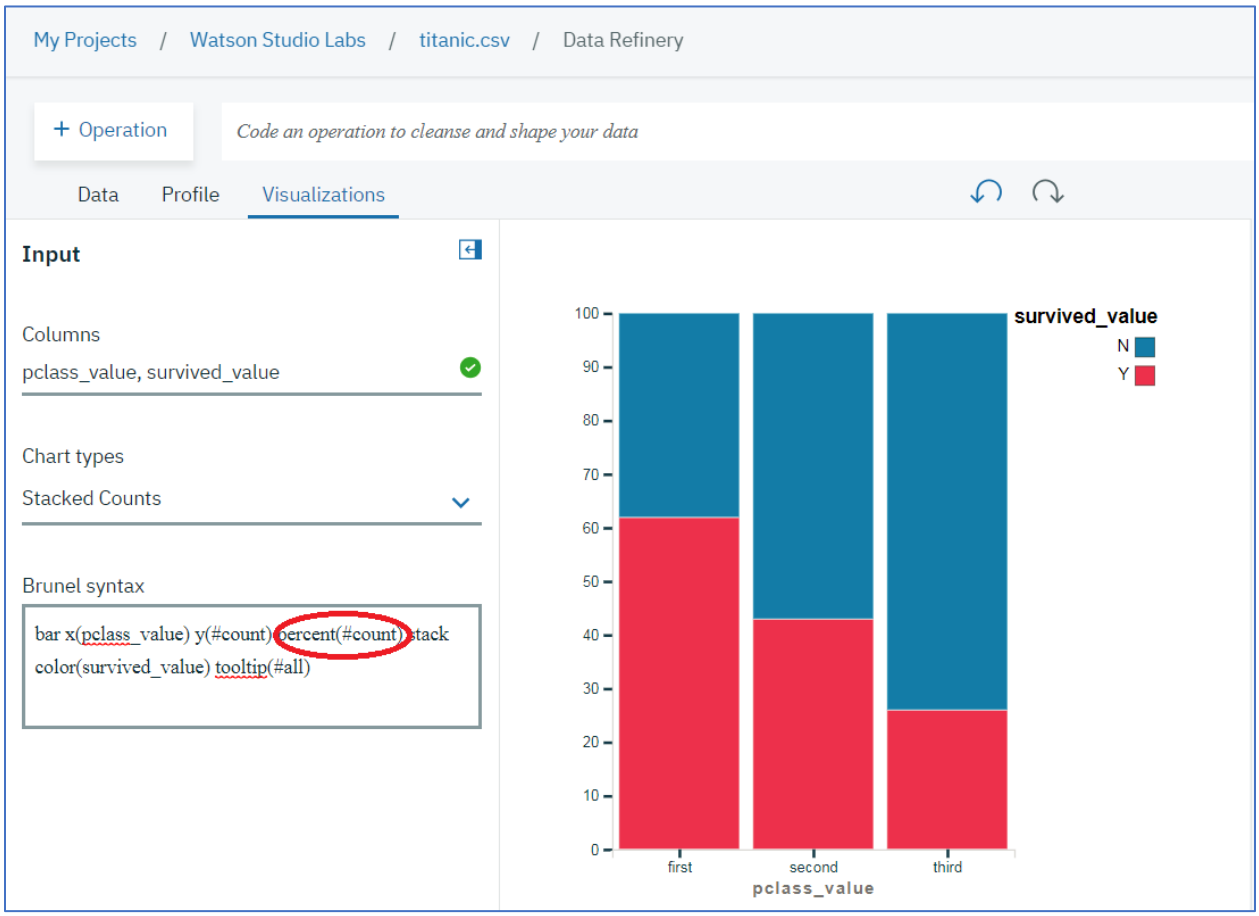
7. The result is shown below. We can see that survival probability for first class customers is significantly better.



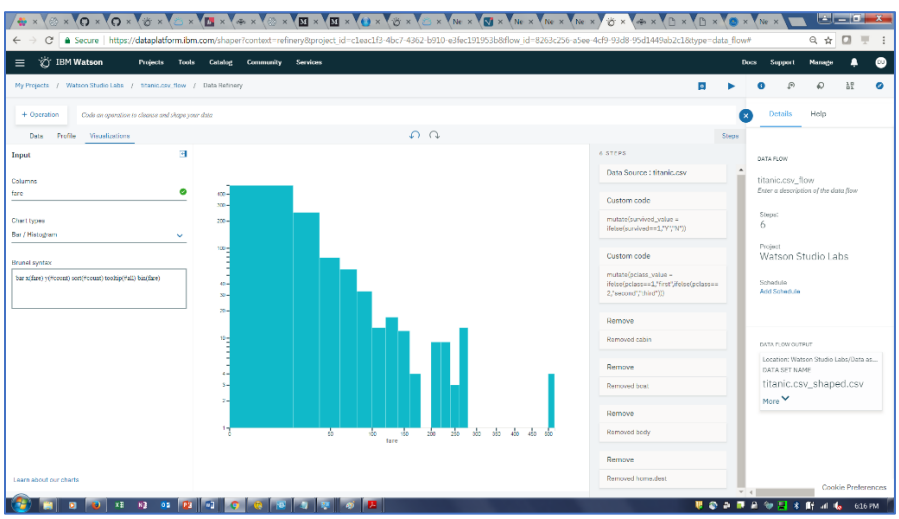
8. If we want to normalize the results so that each column is shown as a percentage to allow comparisons more easily, then add **percent(#count)** to the Brunel syntax and click on **Update Visualization**.



9. The result is shown below. We can see that the percentage of survival is greatest for first class and lowest for third class.



10. Plot the fare values. The result is shown below. Note that it is highly skewed which affects the performance of some machine learning algorithms. One way to deal with this is to apply a logarithmic transformation. We will do that as part of data preparation.



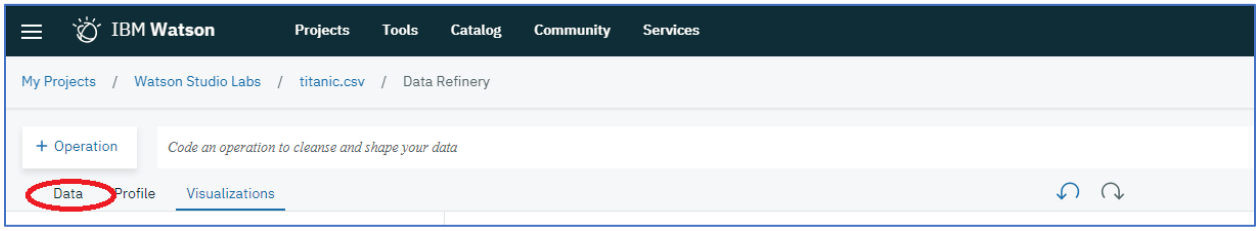
Step 4: Prepare the data for modeling

Based on the data analysis, we need to do the following to prepare the data for modeling.

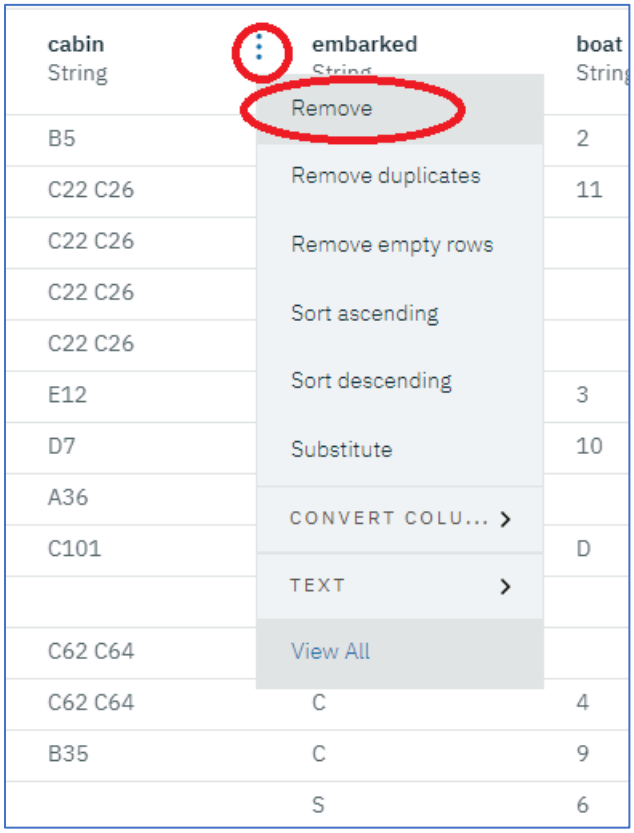
1. Remove columns cabin, boat, body, home.dest
2. Remove rows with missing values of age, and embarked.
3. Create a new column(log_fare) that is the logarithm of the fare column

We will also bin the age, and log_fare fields.

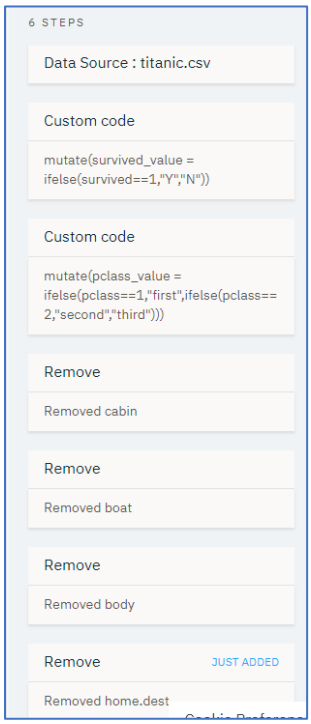
1. Return to the Data panel by clicking on the **Data** tab



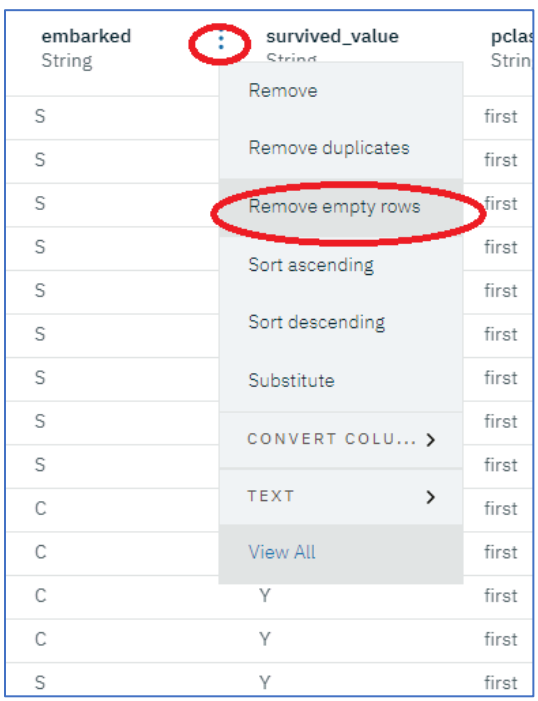
2. Remove the cabin column by selecting on the vertical ellipse and then clicking on **Remove**.



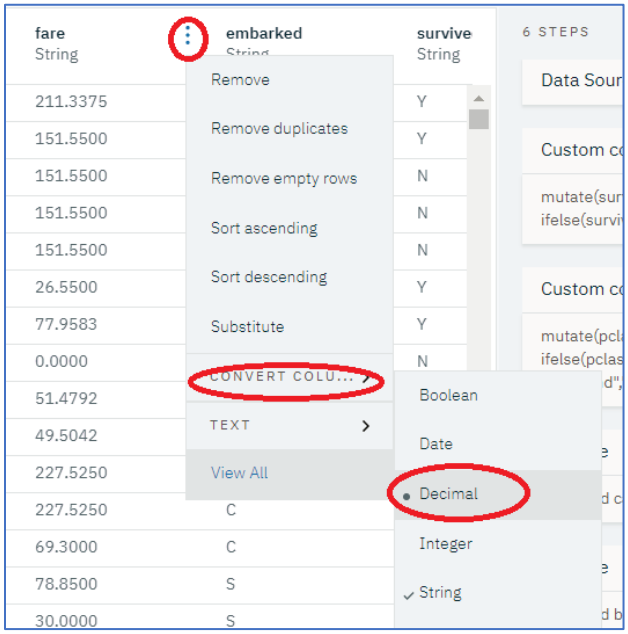
3. Remove the boat, body, and home.dest columns in a similar manner by selecting on the vertical ellipse adjacent to the column and clicking on **Remove**. Notice the STEPS panel on the right-hand side that provides a running list of the data operations.



4. For the age and embarked columns, click on the vertical ellipse adjacent to the columns, and click on **Remove empty rows**.



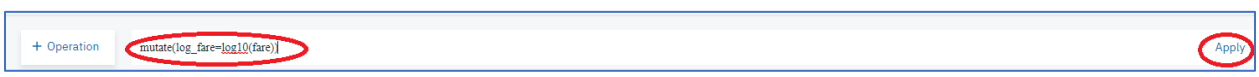
5. Convert the fare column from a String to a Decimal by clicking on the vertical ellipse adjacent to the column, click on **Convert Column**, and then click on **Decimal**.



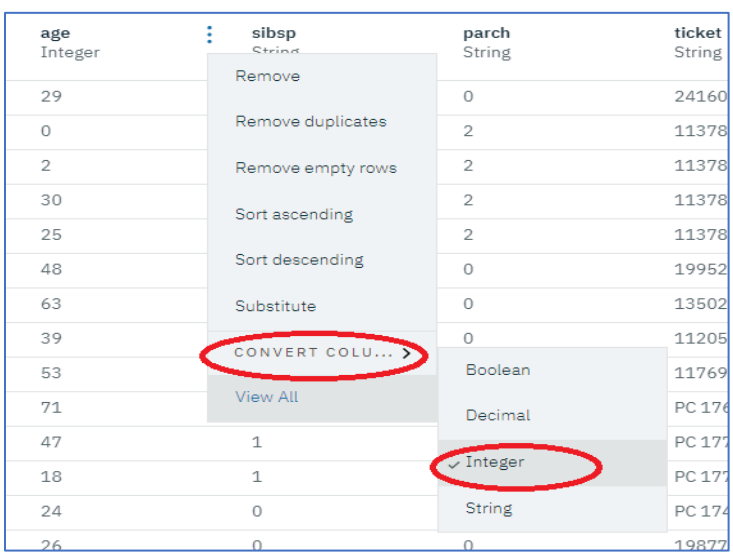
6. Create a new column that is the log to the base 10 of the fare by clicking into the **Code an operation to cleanse and shape your data,** and entering

   mutate(log_fare=log10(fare))

   then click **Apply**.



7. Convert the age from String to Integer by clicking on the vertical ellipse adjacent to the age column, clicking on **Convert Column**, and clicking on **Integer**.

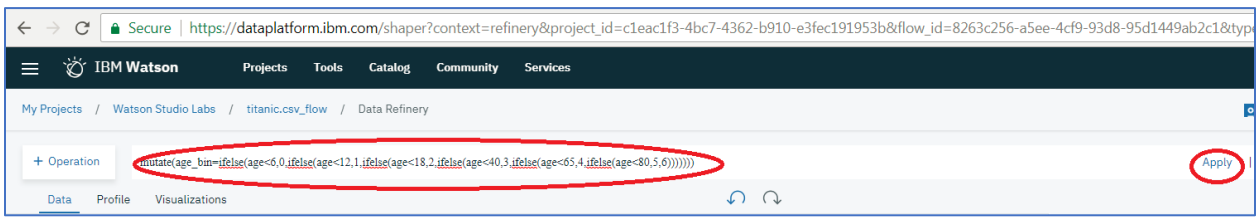


8. Bin the age column into the following bins by clicking into the **Code an operation to cleanse and shape your data,** and entering

   mutate(age_bin=ifelse(age<6,0,ifelse(age<12,1,ifelse(age<18,2,ifelse(age<40,3,ifelse(age<65,4,ifelse(age<80,5,6)))))))

   and then click **Apply**.

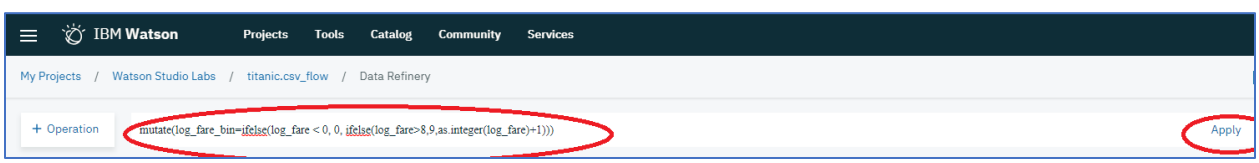
Bin	Age Range
0	0-5
1	6-11
2	12-17
3	18-39
4	40-64
5	65-79
6	Over 79



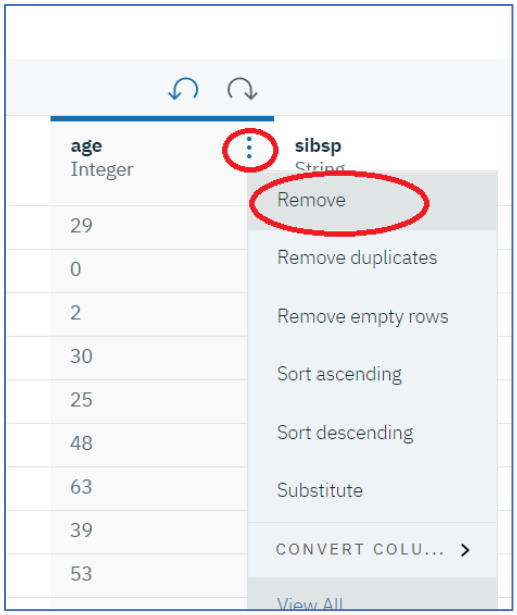
9. Bin the log_fare column, by clicking into the **Code an operation to cleanse and shape your data,** and entering

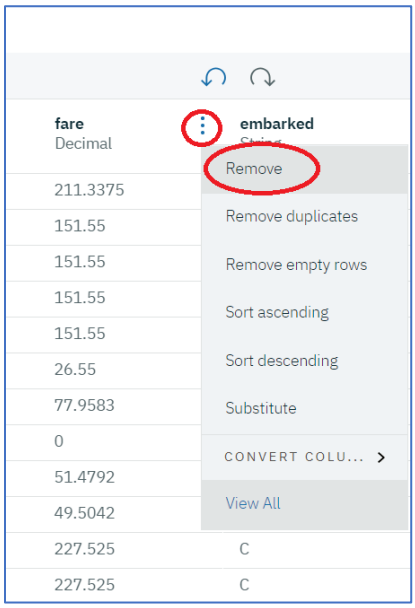
   mutate(log_fare_bin=ifelse(log_fare<0,0,ifelse(log_fare>8,9,as.integer(log_fare)+1)))

   and then clicking **Apply**

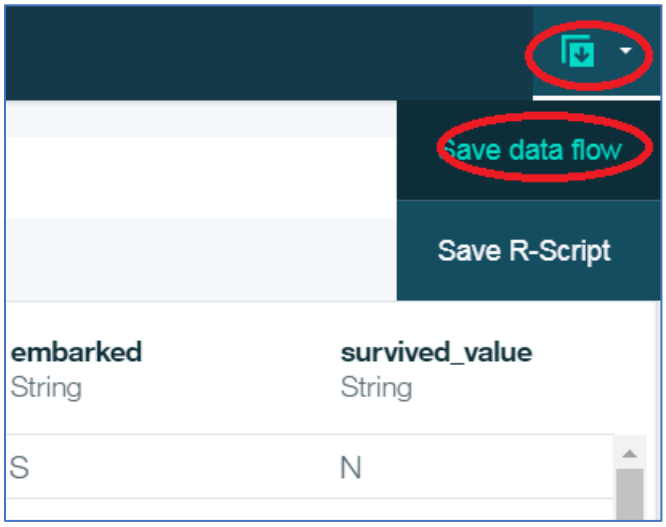


10. Now we will drop the age, fare, and log_fare columns as they are no longer needed for modeling purposes. Select the vertical ellipse adjacent to the column and click on **Remove** as shown below.


age
Integer
sibsp
29
0
2
30
25
48
63
39
53
Remove
Remove duplicates
Remove empty rows
Sort ascending
Sort descending
Substitute
CONVERT COLU...


fare
Decimal
embarked
211.3375
151.55
151.55
151.55
151.55
26.55
77.9583
0
51.4792
49.5042
227.525
227.525
C
C
Remove
Remove duplicates
Remove empty rows
Sort ascending
Sort descending
Substitute
CONVERT COLU...
View All

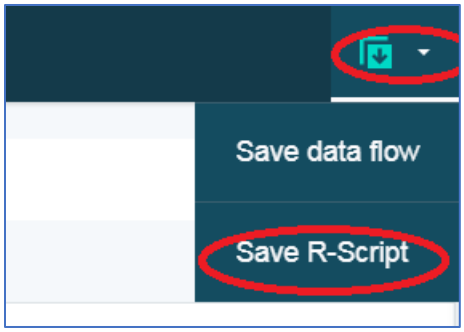
11. Save the Data Flow by clicking on the Save Data Flow icon .



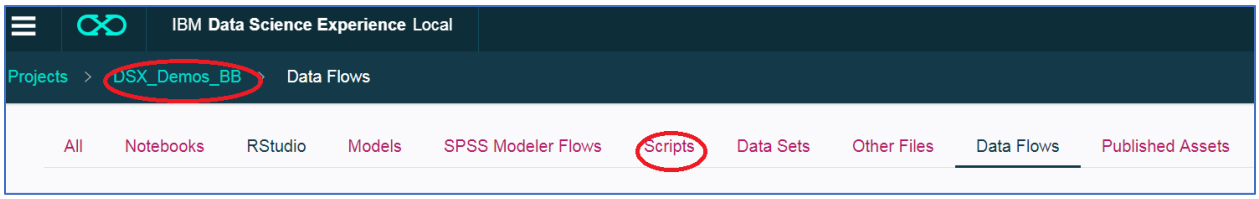
Step 5: Run the sequence of Data Flow operations on the entire data set.

When users are interacting with the Data Refinery tool, the operations are applied to a subset of the data set to facilitate faster response times. To run the data operations on the entire data set (titanic.csv), the user will generate an **R-Script** and run a **Job**.

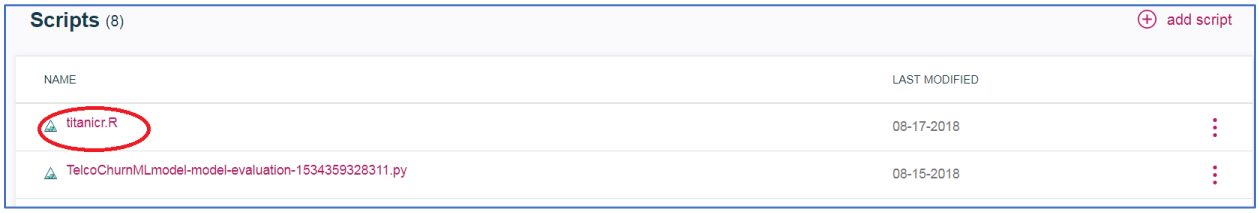
1. Click on the  icon to **Save R-Script**.



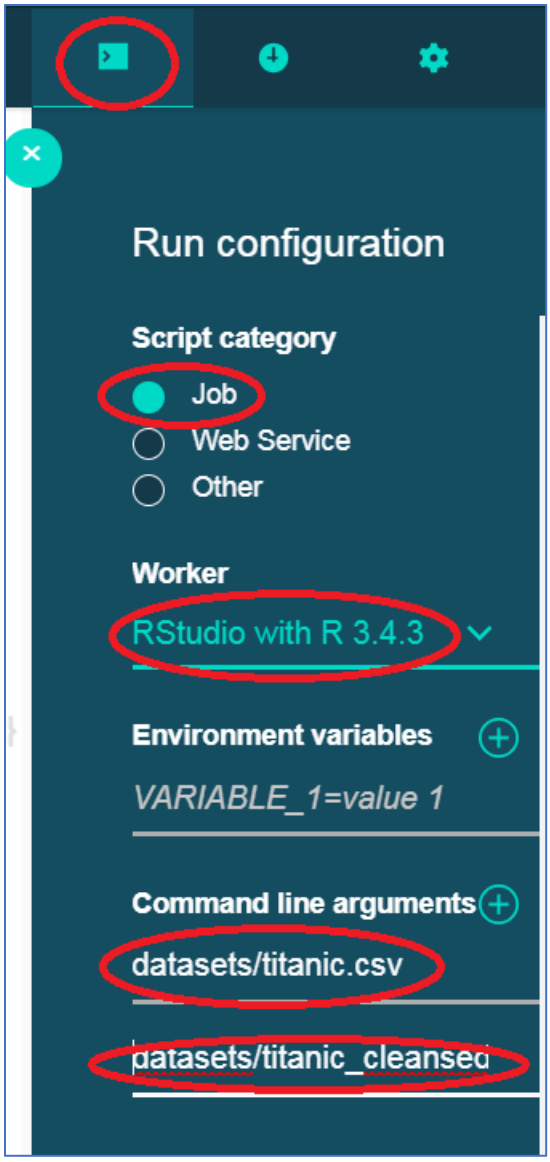
2. Navigate to the project **Scripts** by clicking on the **DSX_Demos_XX** link and then clicking on **Scripts**.

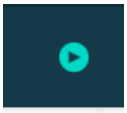


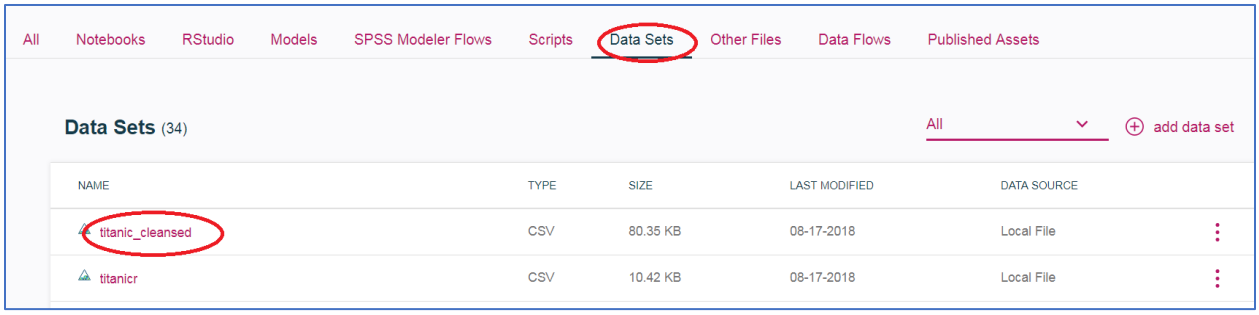
3. Click on the **titanic.R** script.



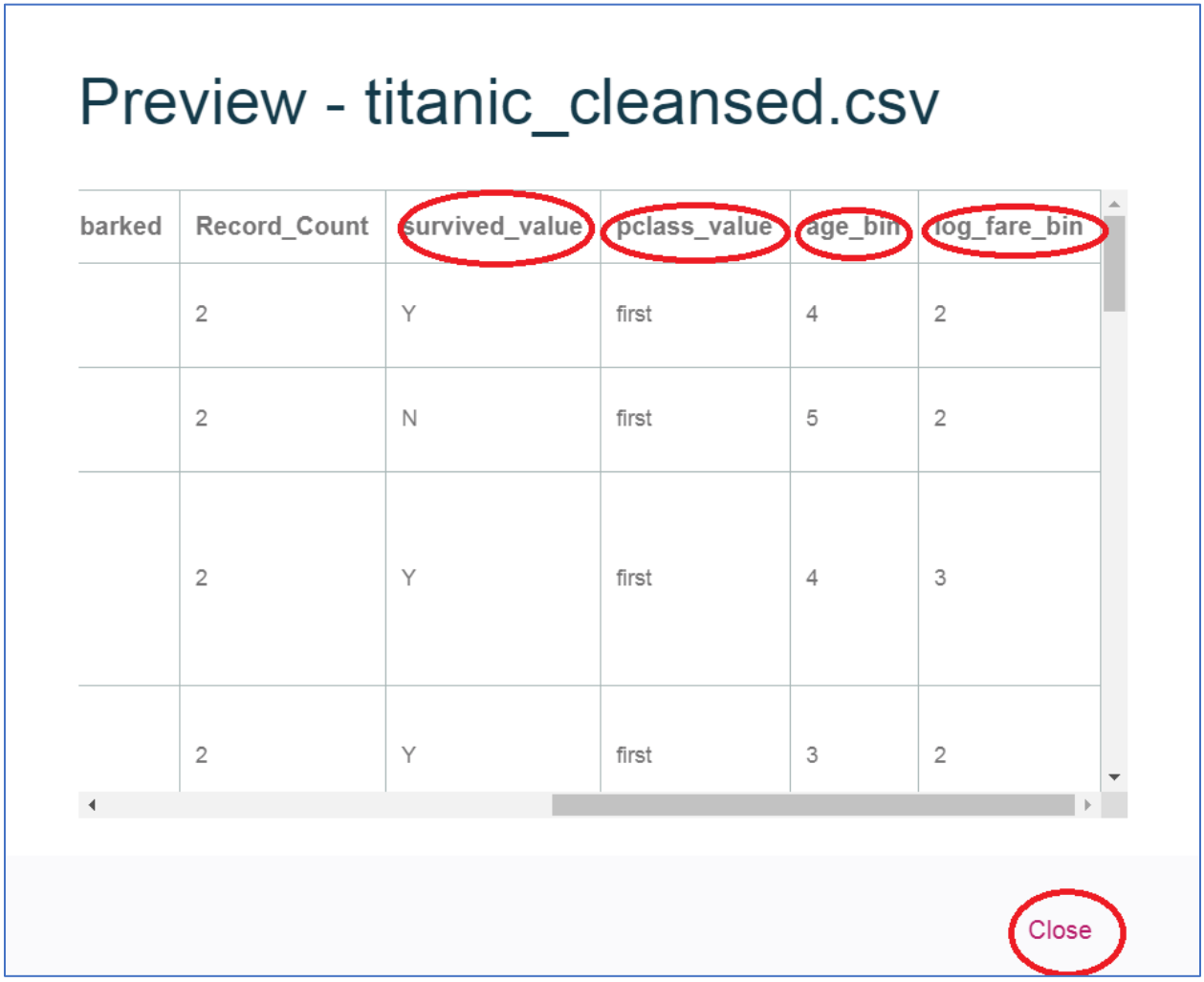
4. Click on the Run Configuration icon, click on **Job** for the **Script category,** select **R Studio with R 3.4.3** for the **Worker,** enter two command line arguments, one for the input file (**datasets/titanic.csv**), and the second for the output file (**datasets/titanic_cleansed.csv**).



5. Click on the run icon  .for a one-time run. Note, we could also set up a job schedule.
6. When the script completes navigate to Data Sets by clicking on the **DSX_Local_XX** project link and then click on **Data Sets**. Notice that the **titanic_cleansed.csv** data set has been created.



7. Click on the **titanic_cleansed.csv** to preview the data set. We can see that the new fields defined in the Data Refinery flow have been created. Click on **Close** to remove the Preview panel.



barked	Record_Count	survived_value	pclass_value	age_bin	log_fare_bin
	2	Y	first	4	2
	2	N	first	5	2
	2	Y	first	4	3
	2	Y	first	3	2