# Lab1 - Getting started on Watson Studio Hands-On

In this first set of hands-on Lab, we will start getting to grips with IBM Watson Studio artefacts:

- [0] Watson Studio and services setup

> Note: your IBM Watson Studio should be already configured as it is part of the pre-requisites. In that case jump directly to the next section A - Data Assets and Data Exploration.
> If your account is not configured, follow the Setup instructions rapidly.

- [A] Data Assets and Data Exploration
- [B] Data Transformation with Data Refinery
- [C] Jupyter Notebooks

[0]. Setup

Watson Studio is an IBM Cloud service, so in addition to the IBM Cloud account setup, you will need to create the Watson Studio instance. In addition, Watson Studio makes use of additional data and AI related services from the IBM Cloud platform, so we will create some artifacts for use within Watson Studio at runtime.

1. Create a Watson Studio service instance
2. Create a Watson Studio Project for the workshop.
3. Provision a set of additional services
4. Load data files into the project as Data Assets

[0.1]. Getting started with data exploration and notebooks

Once the Watson Studio project is completed, we can start our data related work

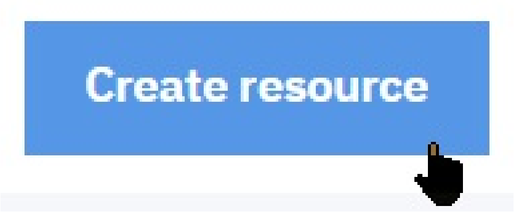
1. Quick assesment of the contents of a Data Asset
2. Work with Jupyter notebooks

The source material for the Workshop is held in a Box folder at URL
https://ibm.box.com/v/WatsonStudio-WS2019

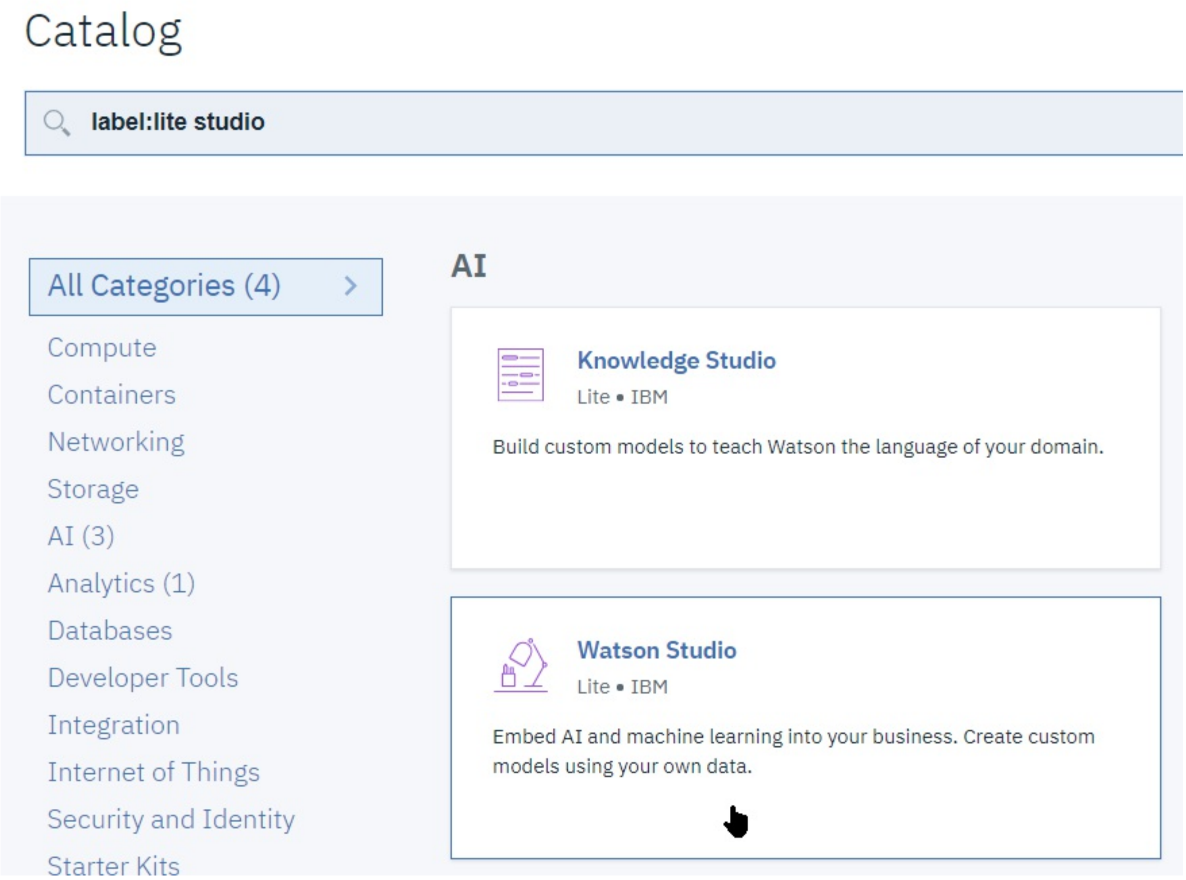
[0.2]. Creating a Watson Studio instance

From IBM Cloud, we will instanciate a Watson Studio service, as the anchor for the toolset within IBM Cloud. Note that this is a one-time setup, only one instance of Watson Studio per region needs to be created.

1. Log-in to you IBM Cloud account's dashboard (https://cloud.ibm.com)
2. Click the `[Create Resource]` button at the top right



3. In the search filter field, add the single word `studio`. This should reveal the lite services having the `studio` word in their name.



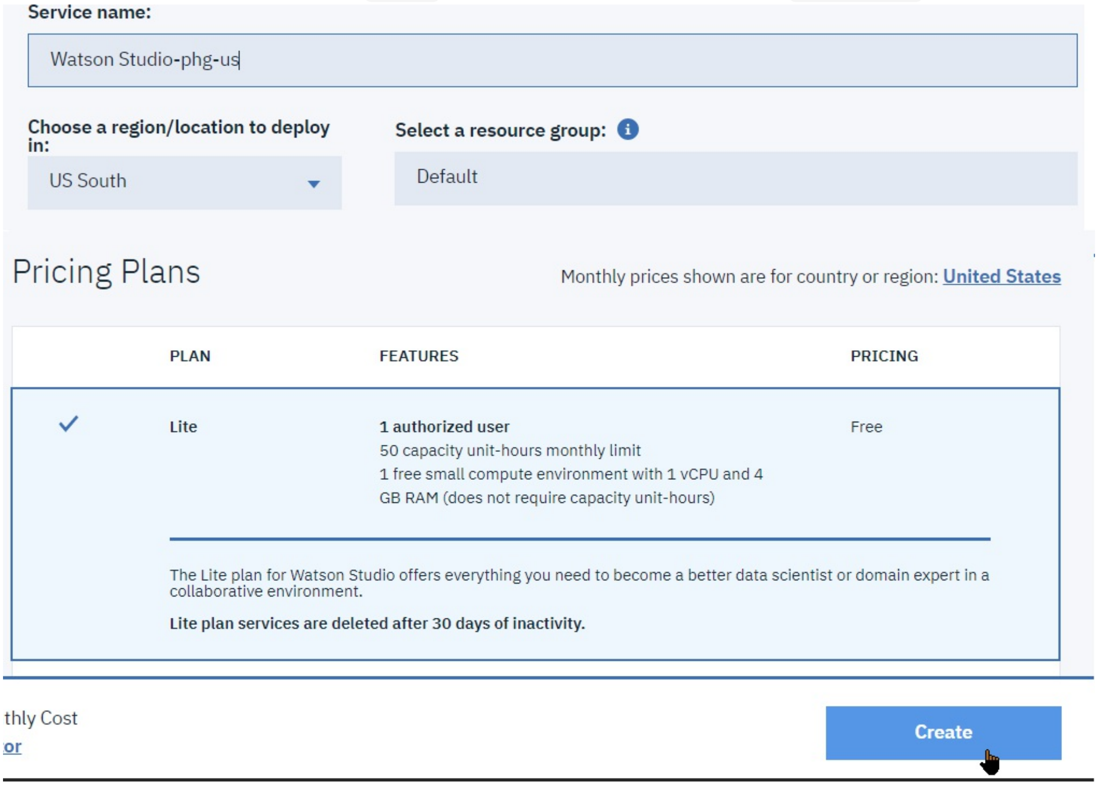
label:lite studio

Knowledge Studio
Lite • IBM
Build custom models to teach Watson the language of your domain.

Watson Studio
Lite • IBM
Embed AI and machine learning into your business. Create custom models using your own data.

and click the `watson studio` tile.

> Note: Make sure to use `Watson Studio`, and *not* `Knowledge Studio`
>
> 4. You are taken to the service creation page. Although it is possible to create an instance of Watson Studio in either `US South` or `United Kingdom` regions, it is recommended to use `US South` because this is where services, including new

> beta ones are updated first. You can change the service name suffix or keep the suggested name. Keep the `Lite` service plan and click the `[Create]` button.
>
> 
>
> Service name: Watson Studio-phg-us
>
> Choose a region/location to deploy in: US South
>
> Select a resource group: Default
>
> Pricing Plans
>
> Monthly prices shown are for country or region: United States
>
> | | PLAN | FEATURES | PRICING |
> |---|---|---|---|
> | ✓ | Lite | 1 authorized user
50 capacity unit-hours monthly limit
1 free small compute environment with 1 vCPU and 4 GB RAM (does not require capacity unit-hours) | Free |
>
> The Lite plan for Watson Studio offers everything you need to become a better data scientist or domain expert in a collaborative environment.
>
> Lite plan services are deleted after 30 days of inactivity.
>
> thly Cost
>
> or
>
>

> NOTE: In the rest of the labs, if you created your Waston Studio instance in the `US-South` region, you will need to use the plain URLs without prefix, e.g. `dataplatform.ibm.com`, but if you created in the `United Kingdom` region, you will need to use the `eu-gb` URLs, e.g. `eu-gb.dataplatform.ibm.com`.

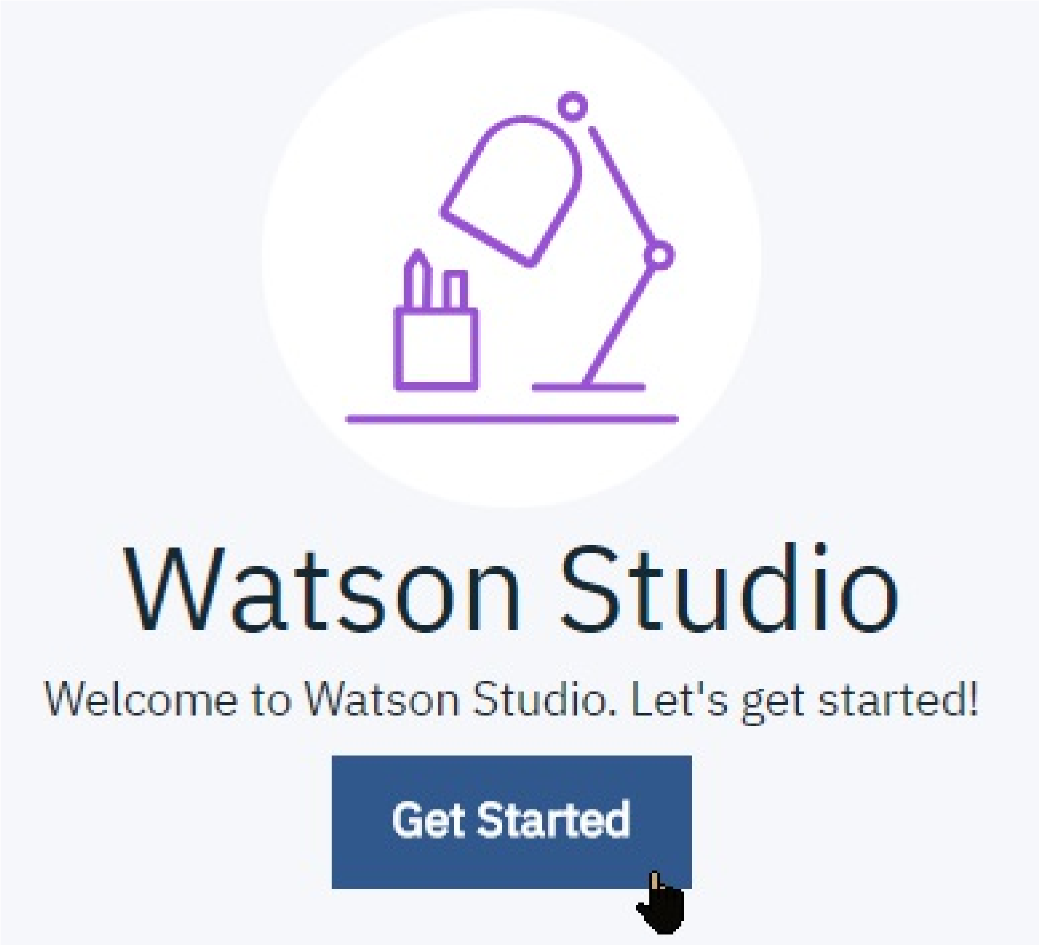
[A] Getting started with Watson Studio

[A.1]. Creating a Watson Studio project

> Note: as a reminder, the URL to access IBM Watson Studio is http://dataplatform.ibm.com

Now that we have put in place the infrastructure to work with Data & AI, we can start creating a project for a specific data handling project.

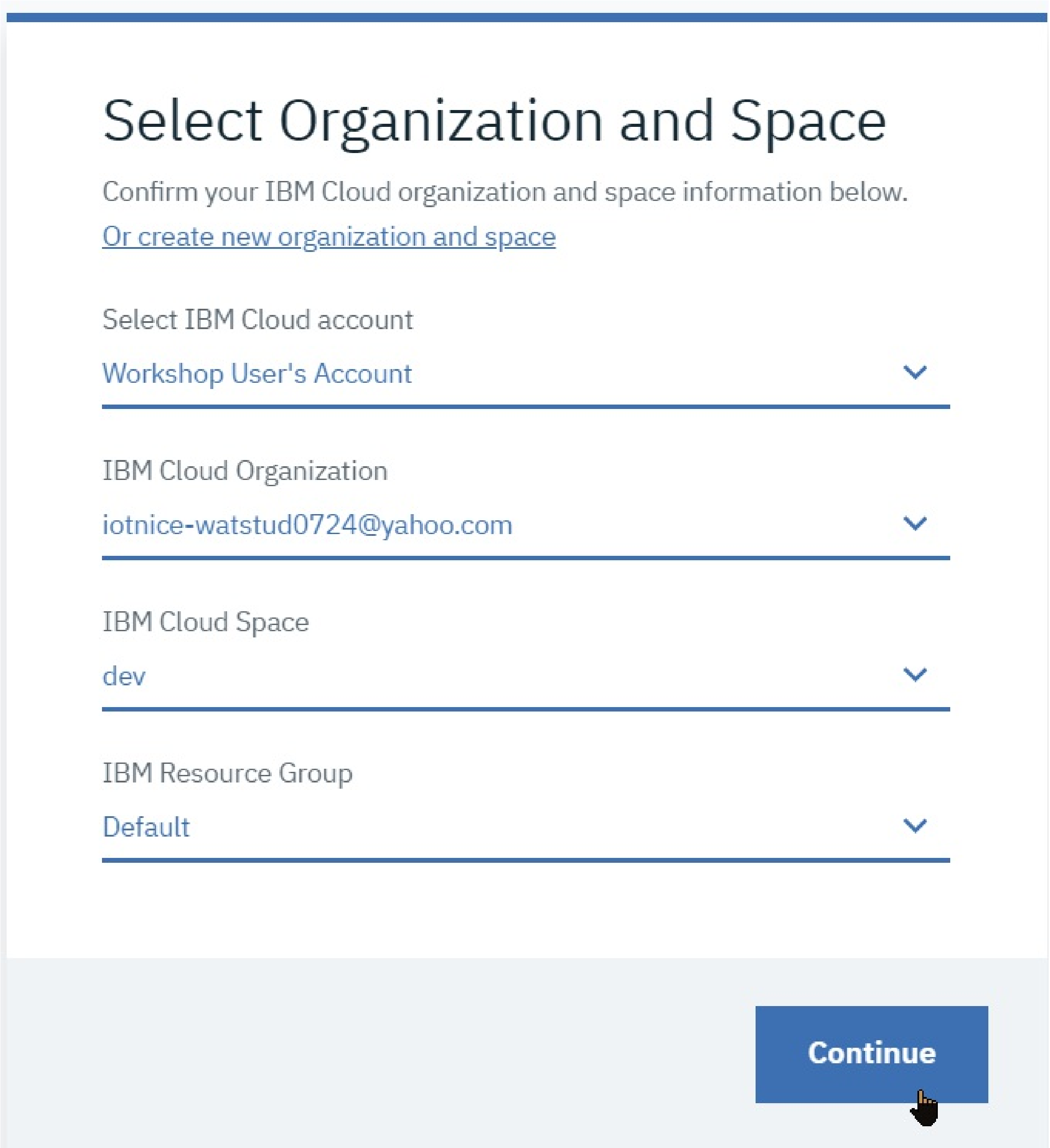
1. If not already signed-in, login to your Watson Studio environment within IBM Data Platform. For this, go back to the IBM Cloud dashboard, select the `Watson Studio` service instance, and click the '[Get Started]' button


Watson Studio

Welcome to Watson Studio. Let's get started!

.

The first time you start the Watson Studio UI, you will be asked to confirm some details, click the `[Continue]` button:



Select Organization and Space

Confirm your IBM Cloud organization and space information below.

Or create new organization and space

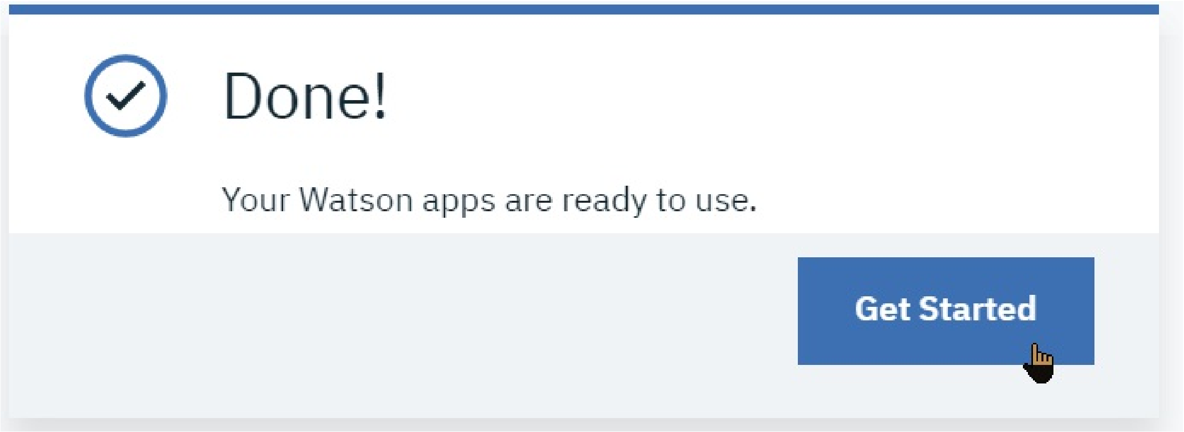
Select IBM Cloud account: Workshop User's Account

IBM Cloud Organization: iotnice-watstud0724@yahoo.com

IBM Cloud Space: dev

IBM Resource Group: Default

, and then validate

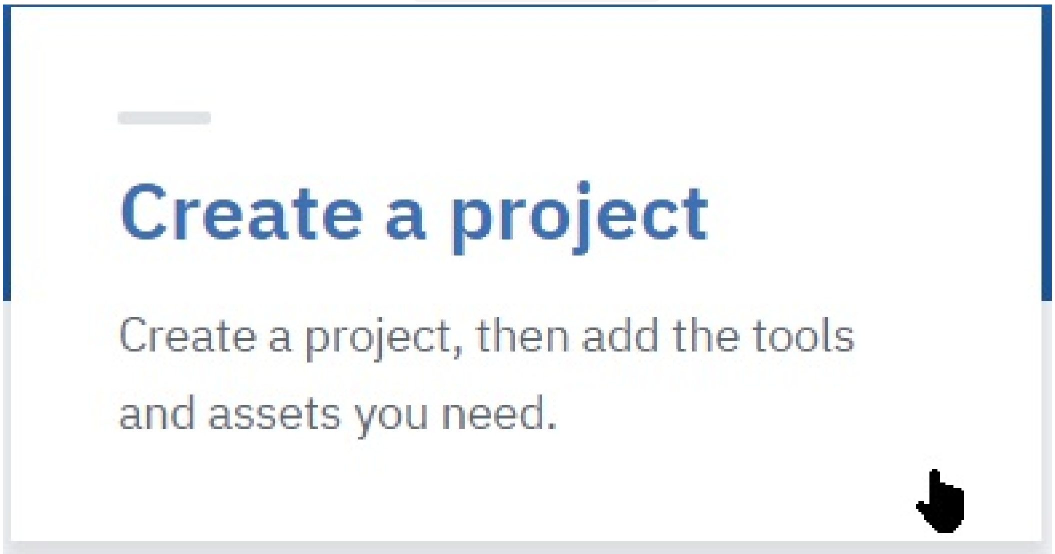


Done!

Your Watson apps are ready to use.

Note that you can also go directly to the service's Cloud Web UI using the URL for the region where the service has been created, either https://dataplatform.ibm.com/projects?context=analytics for 'US-South' or https://eu-gb.dataplatform.ibm.com/projects?

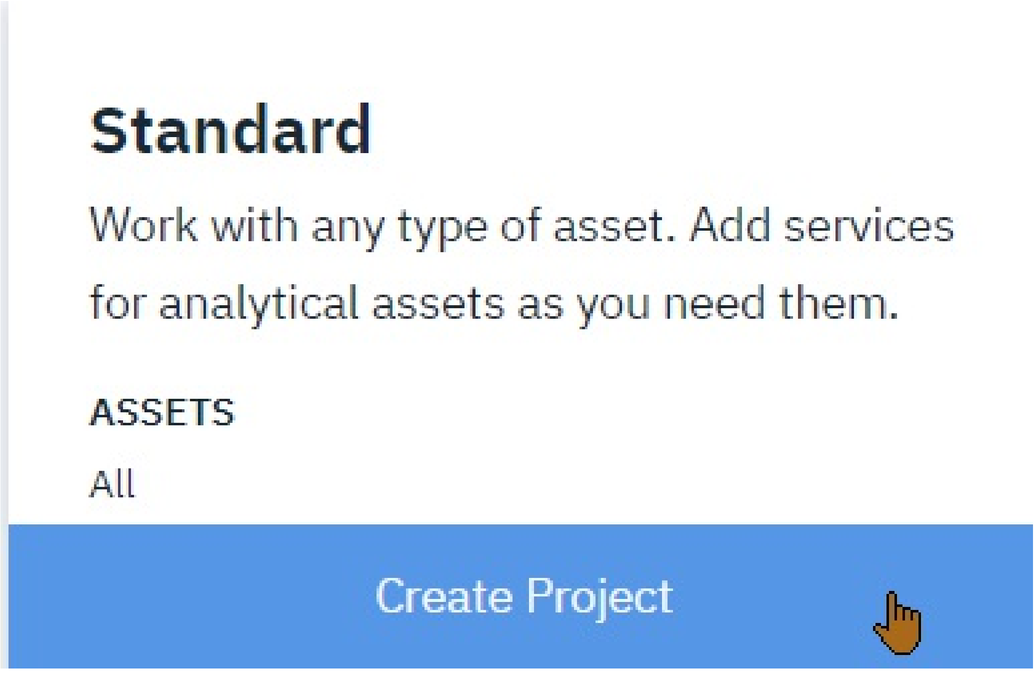
context=analytics for 'United Kingdom'
Create a new project using the `Create a Project` button tile



Create a project

Create a project, then add the tools and assets you need.

Then select a `Standard` configuration. This governs which tools are made available to the project, and can be altered later if need be



Standard

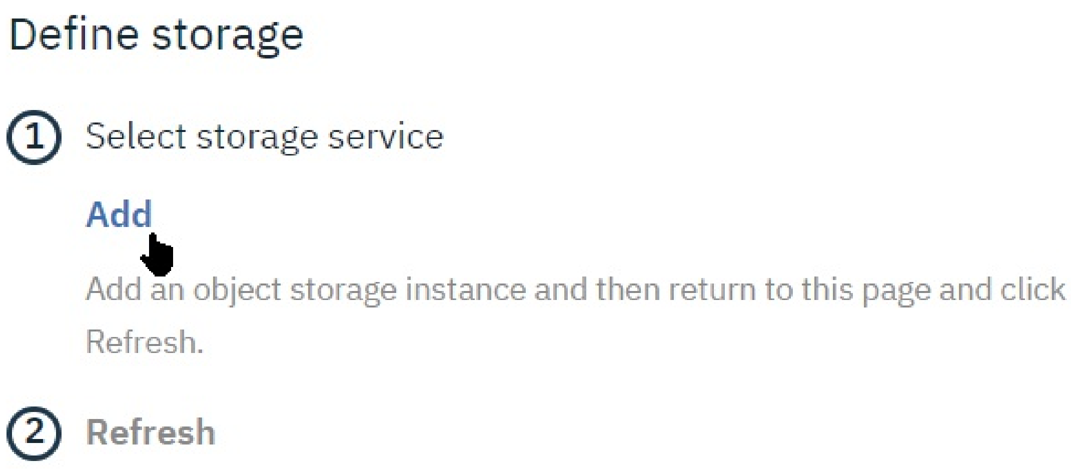
Work with any type of asset. Add services for analytical assets as you need them.

ASSETS

All

Validate with the `[OK]` button

2. Name this new project e.g. `WatStud_Workshop`.
   Note that you will want to leave the 'Restrict who can be a collaborator' unchecked, it will make sharing the project with another account more straightforward.
   Watson Studio stores its file-like artifacts into an instance of `Cloud Object Storage`, we will create a COS service instance at this stage:



Define storage

① Select storage service

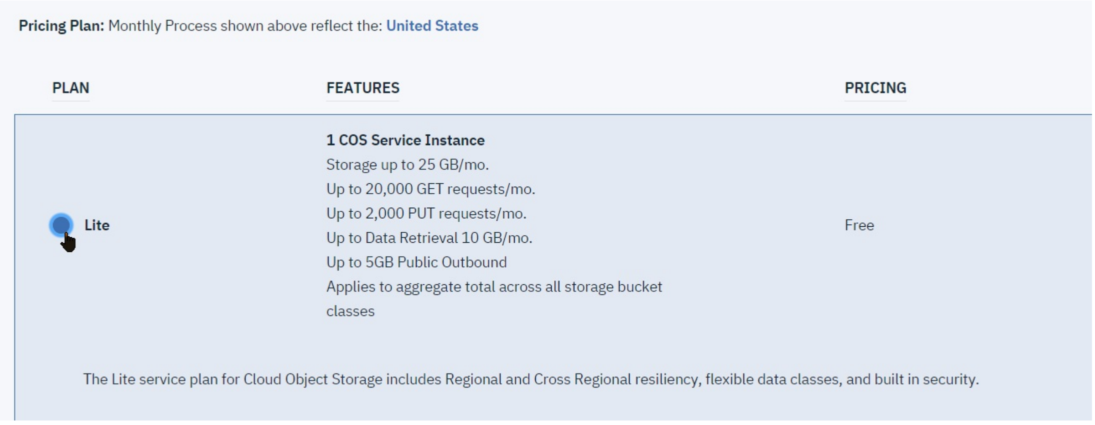
Add

Add an object storage instance and then return to this page and click Refresh.

② Refresh

Currently, your only choice is IBM Cloud Object Storage. Information stored with IBM Cloud Object Storage is encrypted, resilient and dispersed across multiple geographic locations, and accessed over HTTP using a REST API.
Each project and catalog has its own dedicated bucket.

1. Select the Lite Plan

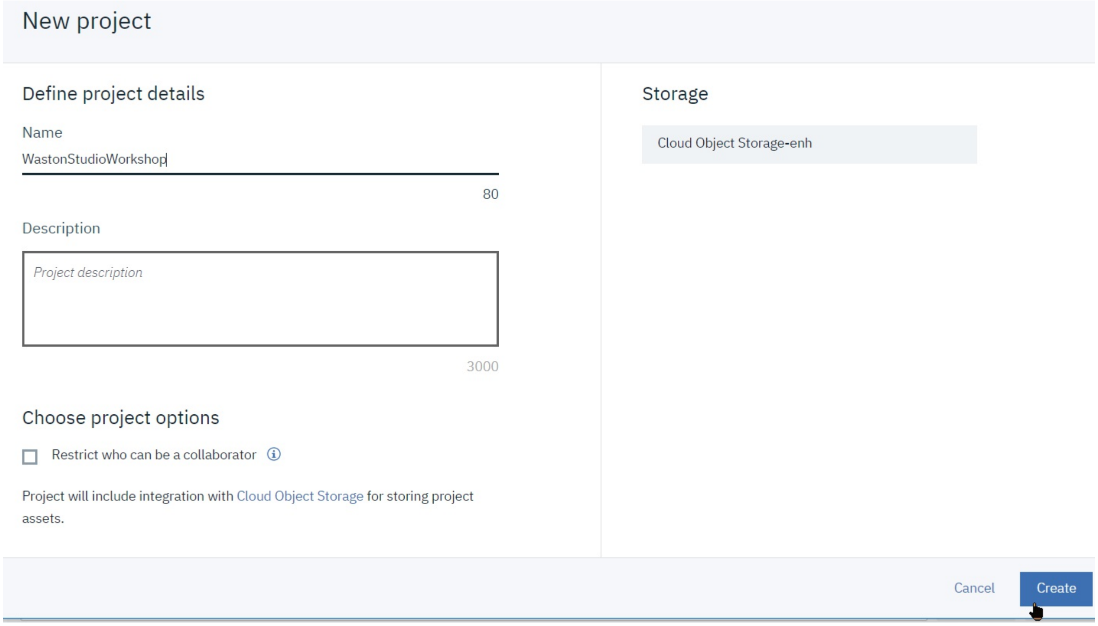


Pricing Plan: Monthly Process shown above reflect the: United States

PLAN	FEATURES	PRICING
● Lite	**1 COS Service Instance** Storage up to 25 GB/mo. Up to 20,000 GET requests/mo. Up to 2,000 PUT requests/mo. Up to Data Retrieval 10 GB/mo. Up to 5GB Public Outbound Applies to aggregate total across all storage bucket classes	Free

The Lite service plan for Cloud Object Storage includes Regional and Cross Regional resiliency, flexible data classes, and built in security.

2. Accept the default names for resource group and Service name
3. Back to the Project creation page, select Refresh then the new Object Storage service instance
4. Finally, click Create:



New project

Define project details

Name

WastonStudioWorkshop

80

Description

Project description

3000

Choose project options

□ Restrict who can be a collaborator

Project will include integration with Cloud Object Storage for storing project assets.

Storage

Cloud Object Storage-enh

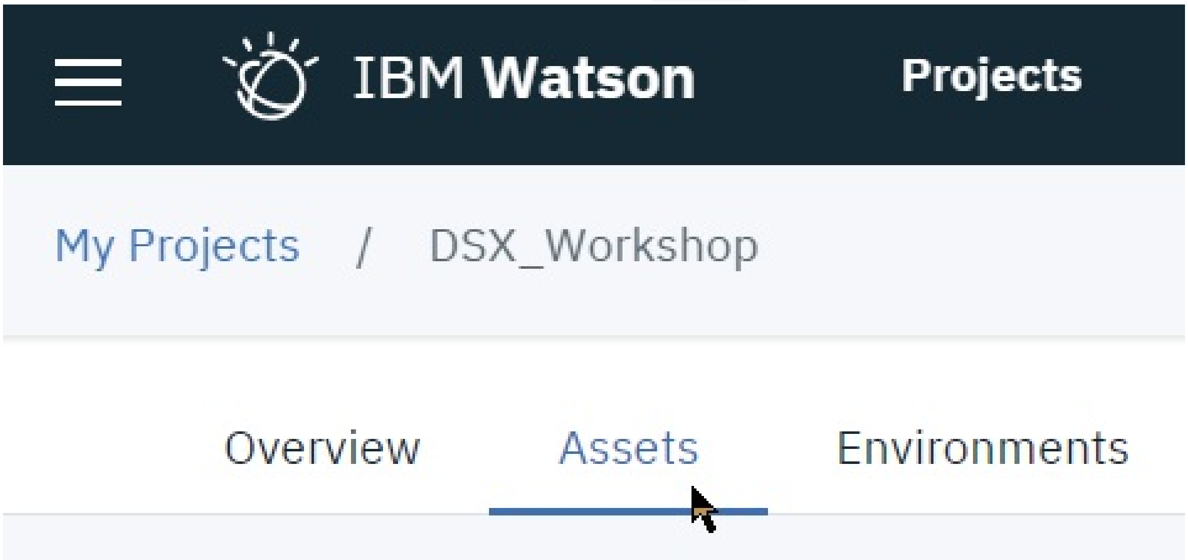
Cancel Create

> Note that COS instance needs to be created only once, it will hold projects' artifacts in separate buckets for each.

[A.2]. Loading Data Assets for the project

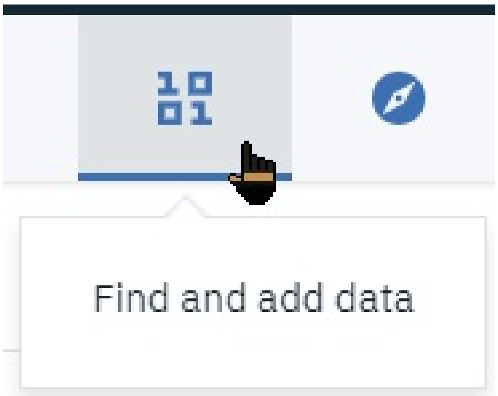
We will load some of the files used during the Hands-On lab as Data Assets available to your project.
The files are available in the Box folder.

1. In your IBM Watson Studio project, switch to the `Assets` tab:

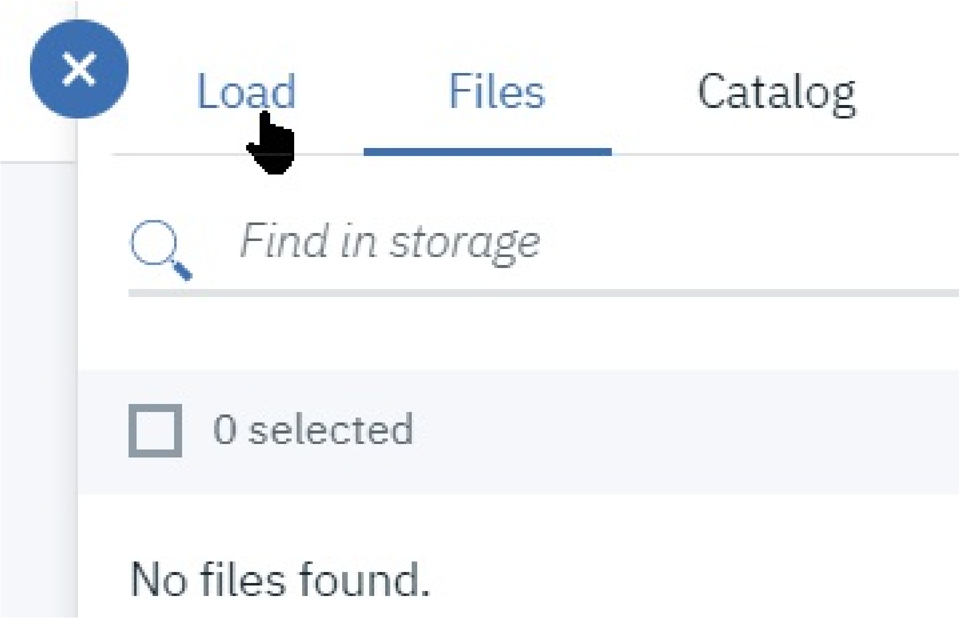


2. Initially the Data Assets list should be empty. If not opened yet, open the Data Pane by

selecting the `1001` icon at top right:



3. Select the `Load` tab



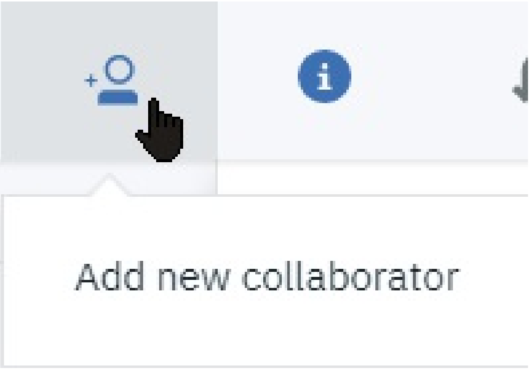
4. Click Browse to add files that you will have downloaded to your computer's disk from the Box folder.
   Among the files that we will need, you can start loading the following ones:
   The source data for these files can also be found at their original location on the web.

File name	Original location
`GoSales_Tx.csv`	https://dataplatform.cloud.ibm.com/exchange/public/entry/view/ba9a
`cars.csv`	https://dataplatform.cloud.ibm.com/api/exchange/actions/download-dataset/c81e9be8daf6941023b9dc86f303053b
`201701-citibike-tripdata.csv`	https://www.citibikenyc.com/system-data

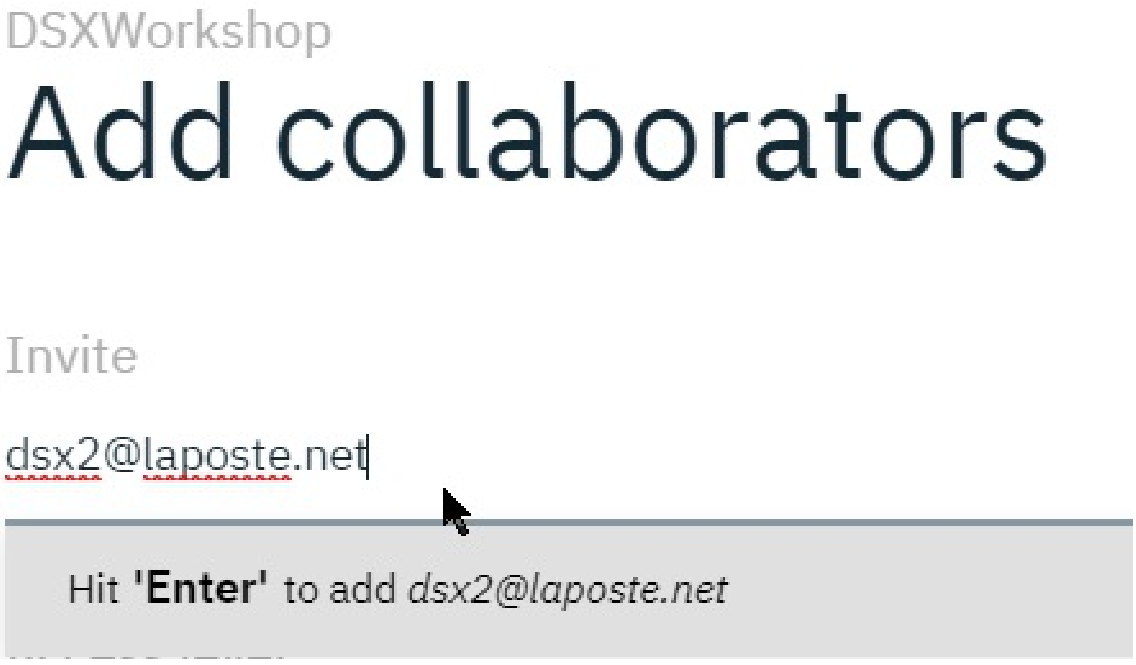
5. Once done, the files will show up in the `Data assets` list.

[A.2 bis]. Project collaboration (optional)

One of the strengths of IBM Watson Studio is to allow to easily collaborate on shared projects. If you have for example another IBM Cloud account, you can add that other account as a collaborator on this `WatStud_Workshop` project:
(Or you can share this with your class neighbour)



- Select the `Add new collaborator` button
- Enter the e-mail address of another account



DSXWorkshop

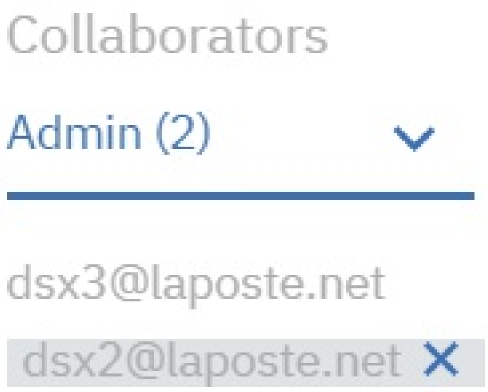
Add collaborators

Invite

dsx2@laposte.net

Hit **'Enter'** to add *dsx2@laposte.net*

- Select an access level, Admin will allow full control, the click `Add`



Collaborators

Admin (2)

dsx3@laposte.net

dsx2@laposte.net

- The new collaborator shows up in the summary
- Finally click `Invite` to validate the change
- If you login with another account to IBM Cloud and another user's instance of Watson Studio, you will be abe to access this project too.

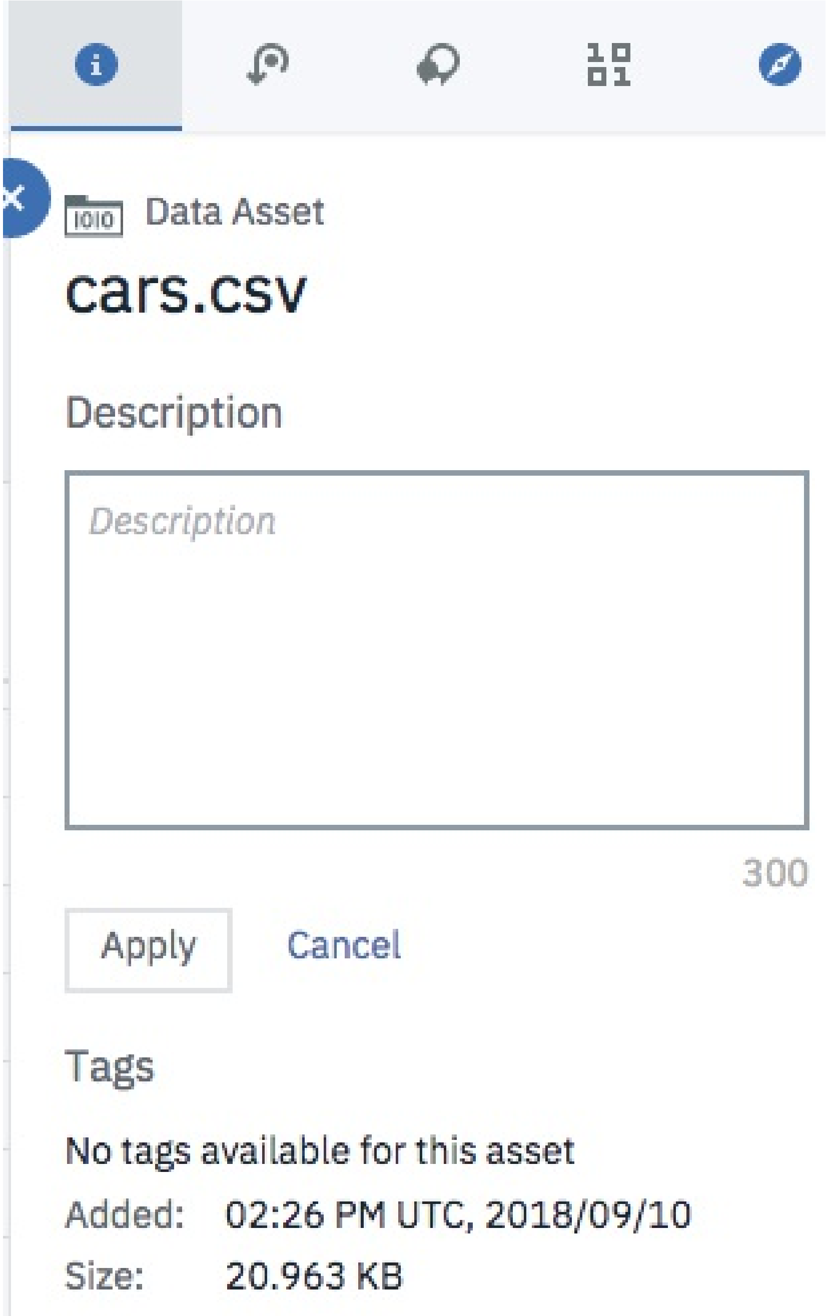
[A.3]. Quick assessment of the Data Asset

You can quickly browse through sample from one of the Data Assets, so as to get an idea of the data format.

For example:

1. From the `Assets` tab in the project page, select the `cars.csv` data asset by clicking on it
2. This opens a preview of the data in tabular format. Data set has 9 columns and 406 rows.

> Note that you can change the Data Asset metadata such as the Description and the Tags from the Information side bar and clicking on the pencil to go in edit mode.



Data Asset

cars.csv

Description

Description

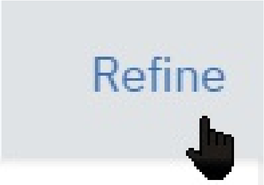
300

Tags

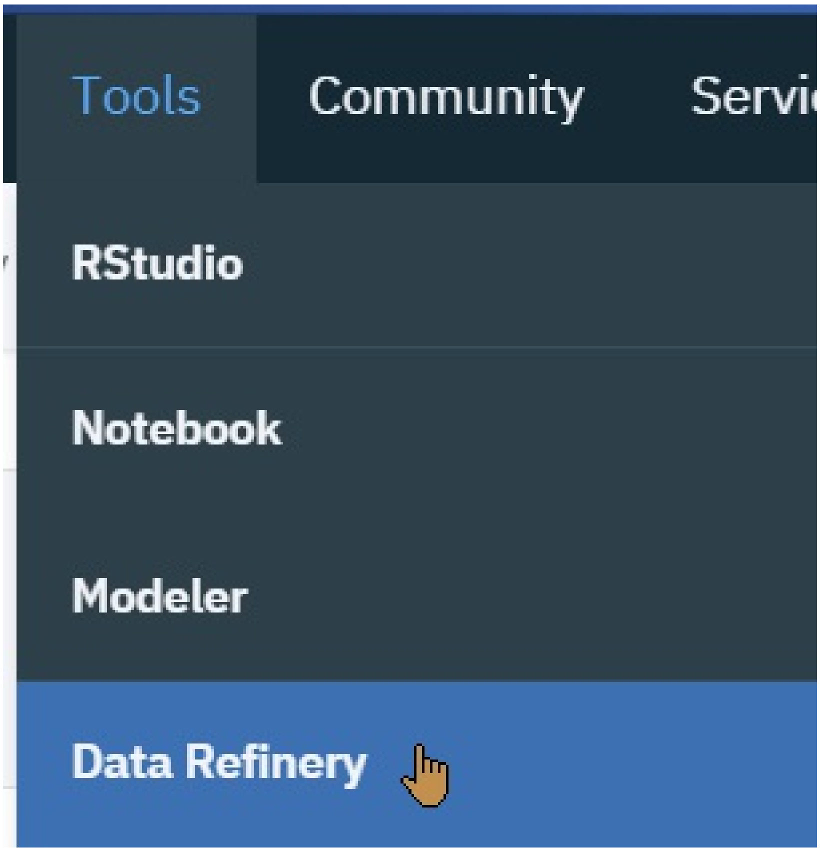
No tags available for this asset

Added: 02:26 PM UTC, 2018/09/10

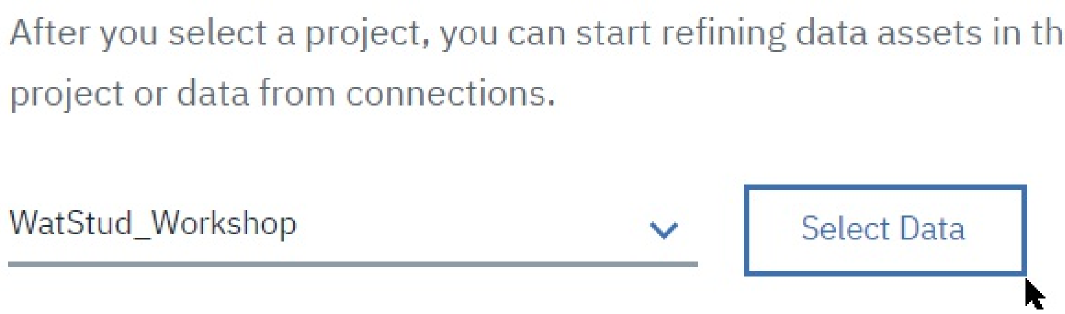
Size: 20.963 KB



3. Now select the `Refine` button
   This will open the Data Refinery tools of IBM Watson Studio which allows to cleanse and shape data, customize it by filtering, sorting, combining or removing columns, and performing operations.
   NOTE: If the `[Refine]` button is not present or grayed-out, navigate to the `Tools/Data`



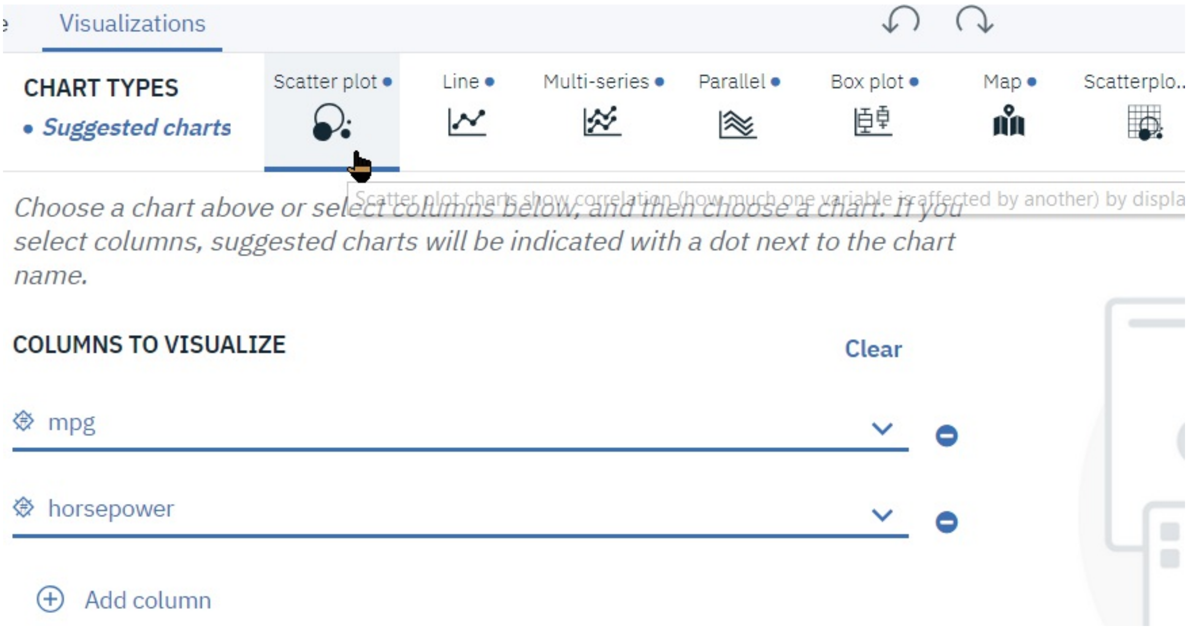
`Refinery` menu , then select your project


After you select a project, you can start refining data assets in the project or data from connections.

, and finally `[add]` the intended file.

> As you manipulate your data, you build a customized data refinery flow that you can modify in real time and save for future re-use. When you save the refined data set, you typically load it to a different location than where you read it from. In this way, your source data can remain untouched by the refinement process.

4. switch to the `Vizualisations` tab in the view that opens
5. in COLUMNS TO VISUALIZE, enter `mpg`, then `(+) Add column` and `horsepower`. Then select a graph type of `Scatterplot`



Scatter plot charts show correlation (how much one variable is affected by another) by displa

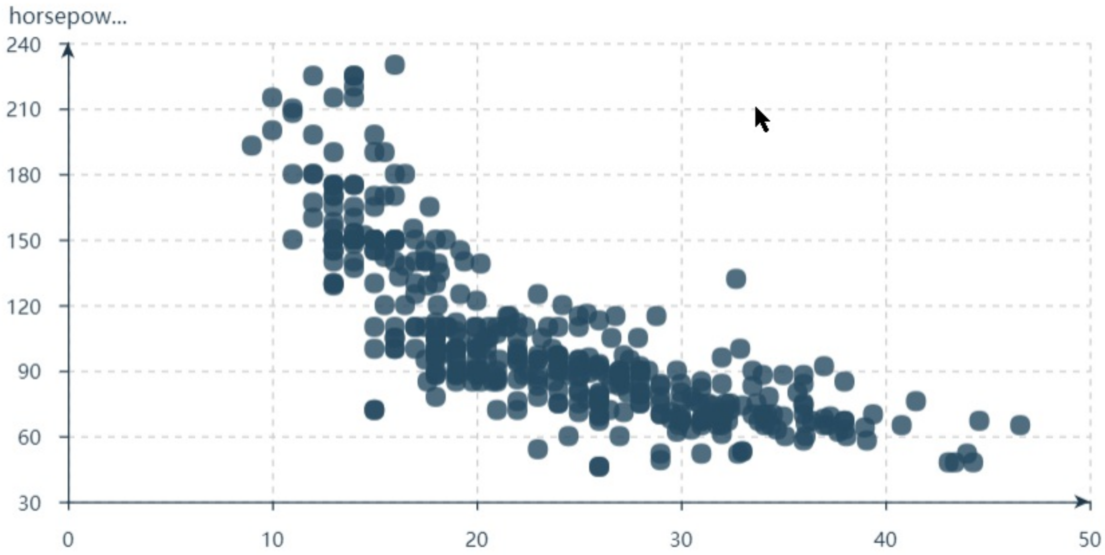
Choose a chart above or select columns below, and then choose a chart. If you select columns, suggested charts will be indicated with a dot next to the chart name.

COLUMNS TO VISUALIZE

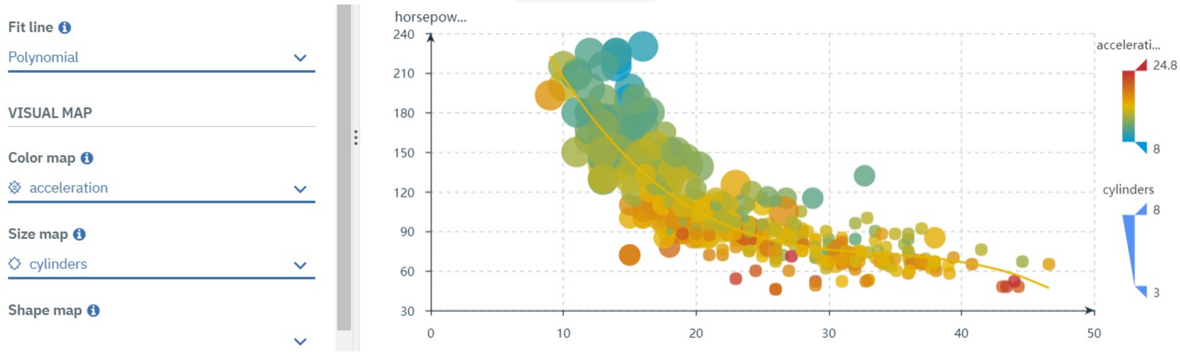
mpg

horsepower

6. the graph plots the two data columns to show their relationship. We will see in the Visualization Hands-On Lab how to programatically generate a similar graph.



7. On the left panel, you can modify the coloring of the dots according to `acceleration` , and their color according to number of `cylinders` :


Fit line: Polynomial

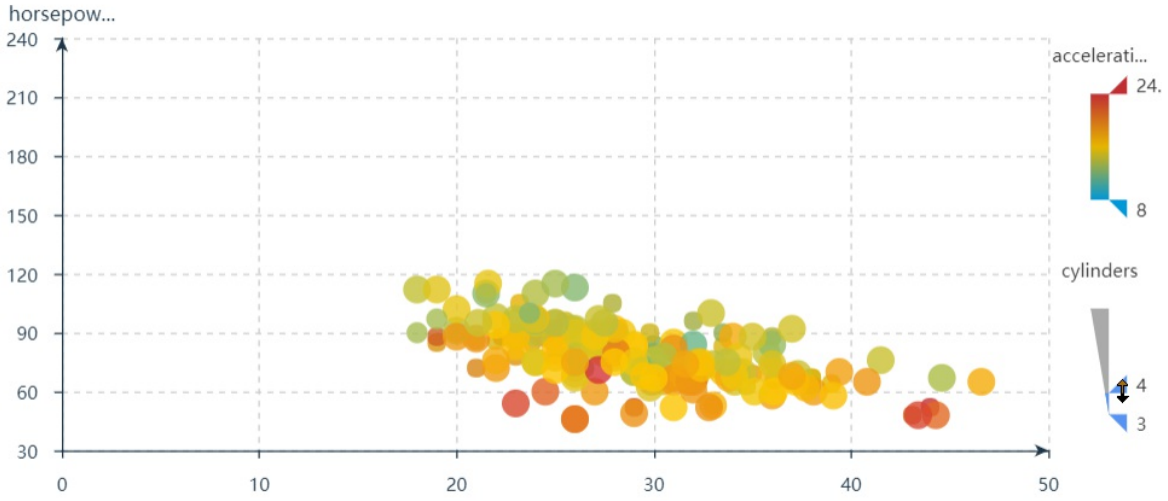
VISUAL MAP

Color map: acceleration

Size map: cylinders

Shape map

8. Finally, notice that the legend gauges on the right side are active and allow to filter the represented dataset, here we show only 2-4 cylinder cars:



Interpretation of the horsepower/mpg scatter plot

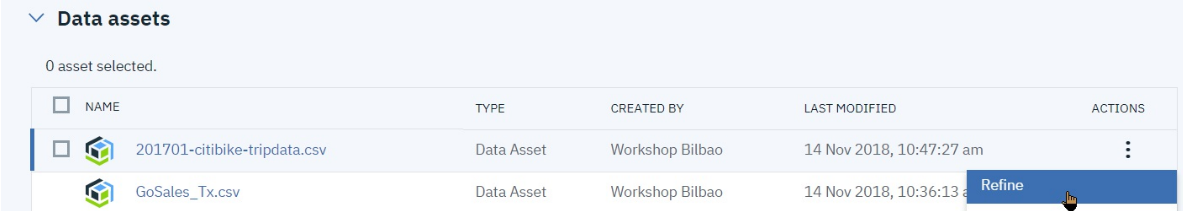
Scatter plots are very useful diagrams to quickly show if there is a relationship between two atributes.
Here we see that there is a general trend that cars with higher horsepower tend to have lower miles-per-gallon. This is kind of an expected outcome.
But we also see that the curve is not quite a straight line, it looks more hyperbolic.
Moreover, some points are clearly not on the general trend, these are called 'outliers'. You can hover at the point at `{hp: 132, mpg: 32.7}`, or `{hp: 15, mpg: 72}` for example.

[B] Data Refinery

The Data Refinery in IBM Watson Studio is an integrated ETL feature which allows to easily implement data transformation pipelines in the form of a sequence of data operations applied to a data set called data flows.

In this section, we will use Data Refinery to cleanse and filter the contents of the `201701-citibike-tripdata.csv` data file.
This file is one of the monthly reports of bike sharing usage for NYC, provided as an Open Data asset from https://www.citibikenyc.com/system-data.
We will use IBM Watson Studio to get a first understanding of the data, and apply some transformations to reduce the volume and scope of data to analyze.
Note that this file is pretty large, with over 725000 lines of data, and a raw file size of over 120MB, in CSV format, which is not the most efficient to store data (the zipped content is about one fifth of the raw data)

1. From your project's `Assets` tab, locate `201701-citibike-tripdata.csv` and select the `Refine` contextual menu option:



Data assets

0 asset selected.

□	NAME	TYPE	CREATED BY	LAST MODIFIED	ACTIONS
□	201701-citibike-tripdata.csv	Data Asset	Workshop Bilbao	14 Nov 2018, 10:47:27 am	⋮
	GoSales_Tx.csv	Data Asset	Workshop Bilbao	14 Nov 2018, 10:36:13 a	Refine

.

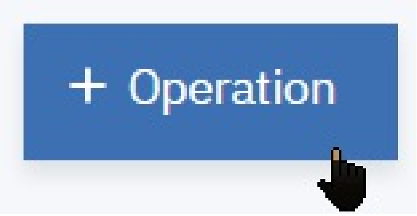
> You may also launch Data refinery from the Tools menu, in which case you will need to select the `201701-citibike-tripdata.csv` data file, and click the `[Add]` button at the bottom right, if it is not active, you will have to select `[Add]` from the main panel.

2. Data Refinery will show a table with the 1000 first rows as a sample. As part of the operations we will want to apply to the data, we will:

   i. Rename the columns so as to remove blanks that could cause handling issues later on
   ii. Specify actual data types for non-string fields. This applies to the numeric `Trip Duration`, `Birth Year` and the 4 station *latitude* and *longitude* columns.
   iii. Compute an Age column from `Birth Year`.
   iv. Extract date and time slot columns from the Start and Stop time columns.

   > Notice that as you perform data transformations, the steps of your data flow are added on the right side bar.

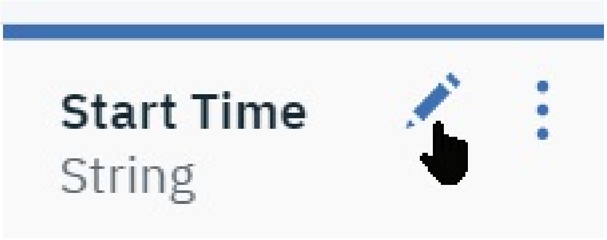
[B.1]. Columns renaming:



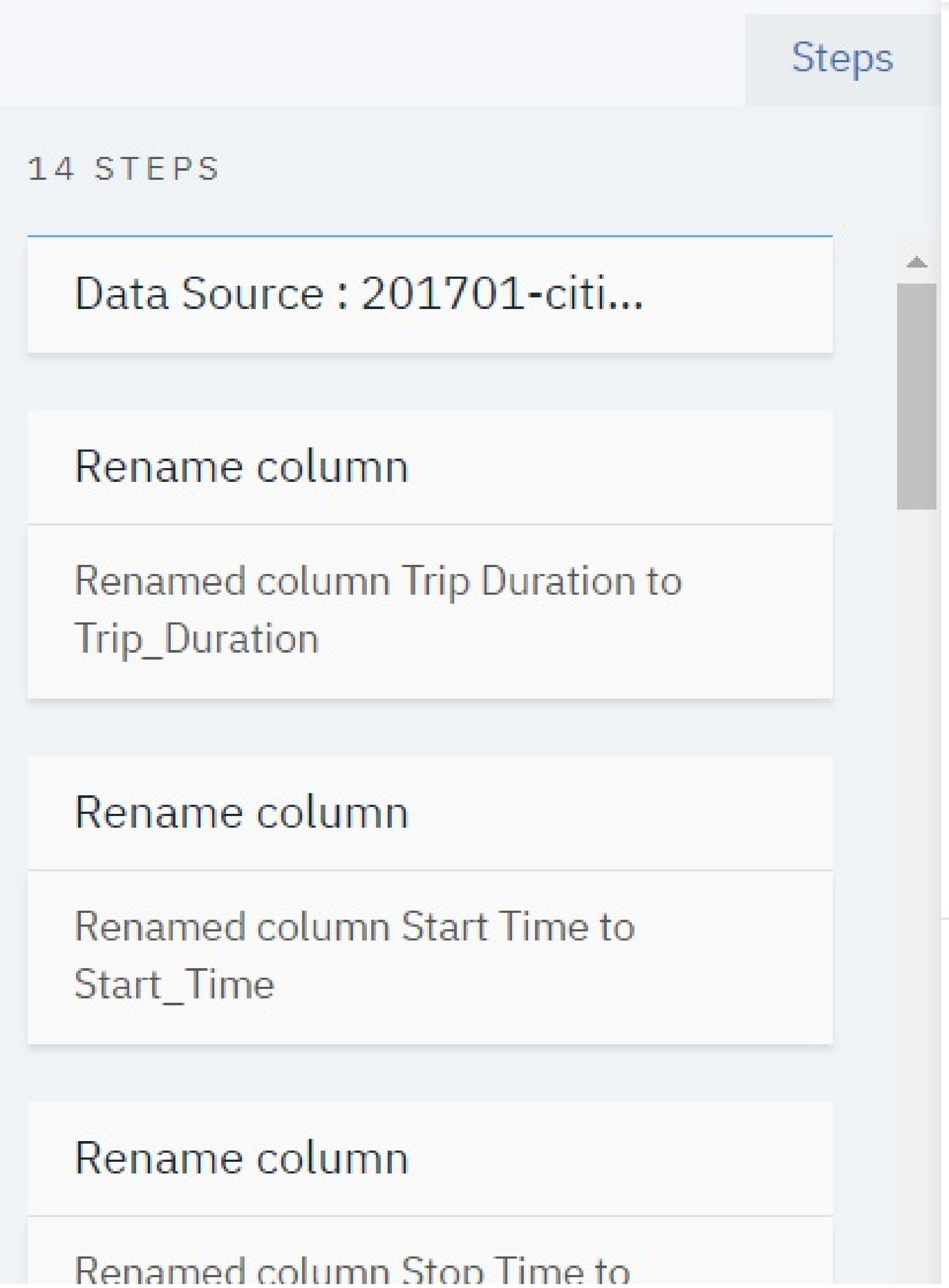
1. For the first column, select the `+ Operation` button
2. then the `Rename` operation, and replace the column name by the same with spaces replaced by underscores, e.g. `trip duration` becomes `trip_duration`.

> NOTE that it's a good idea to cut the column name before clicking the `Next` button so as to save retyping it.

2. For the other columns, there is a faster way to add a rename operation, by clicking the pencil icon in the column header and changing the name there:


Start Time
String

As you proceed through columns renaming, you will see operations being listed in the right-hand side panel. You should now have 15 operations listed in the steps list:



14 STEPS

Data Source : 201701-citi...

Rename column

Renamed column Trip Duration to Trip_Duration

Rename column

Renamed column Start Time to Start_Time

Rename column

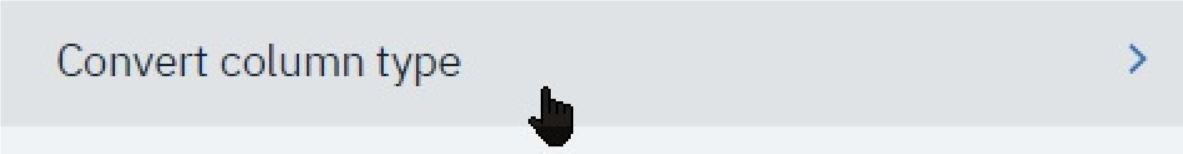
Renamed column Stop Time to

.

Save you work with the Disk icon.

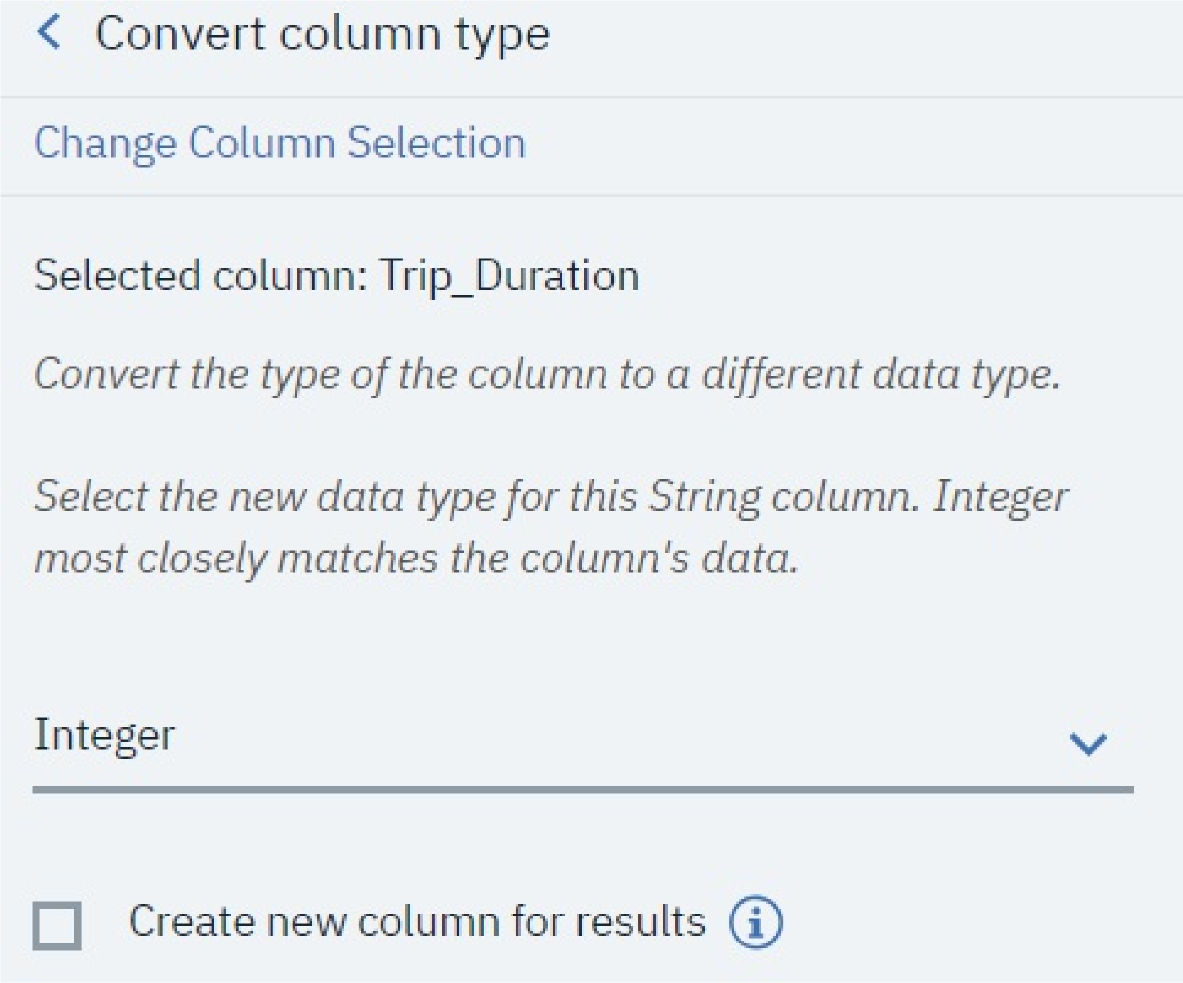
[B.2]. Data type changes

1. Now add `Convert Column type` operations for `Trip_Duration` and `Birth_Year` columns:



.

2. Select `Integer` as the target type. Leave the `Create new column` unselected:



Selected column: Trip_Duration

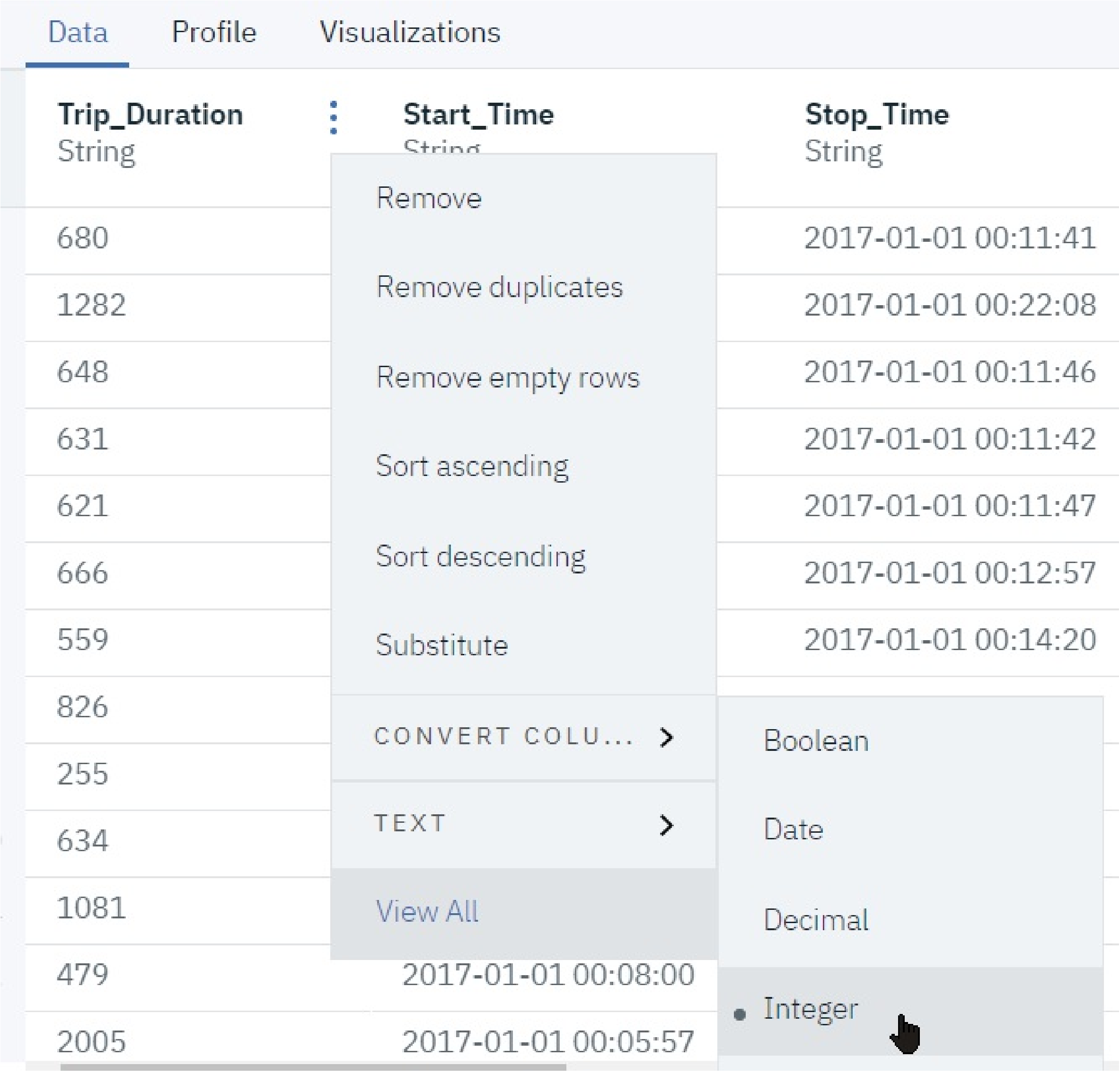
Convert the type of the column to a different data type.

Select the new data type for this String column. Integer most closely matches the column's data.

Integer

□ Create new column for results

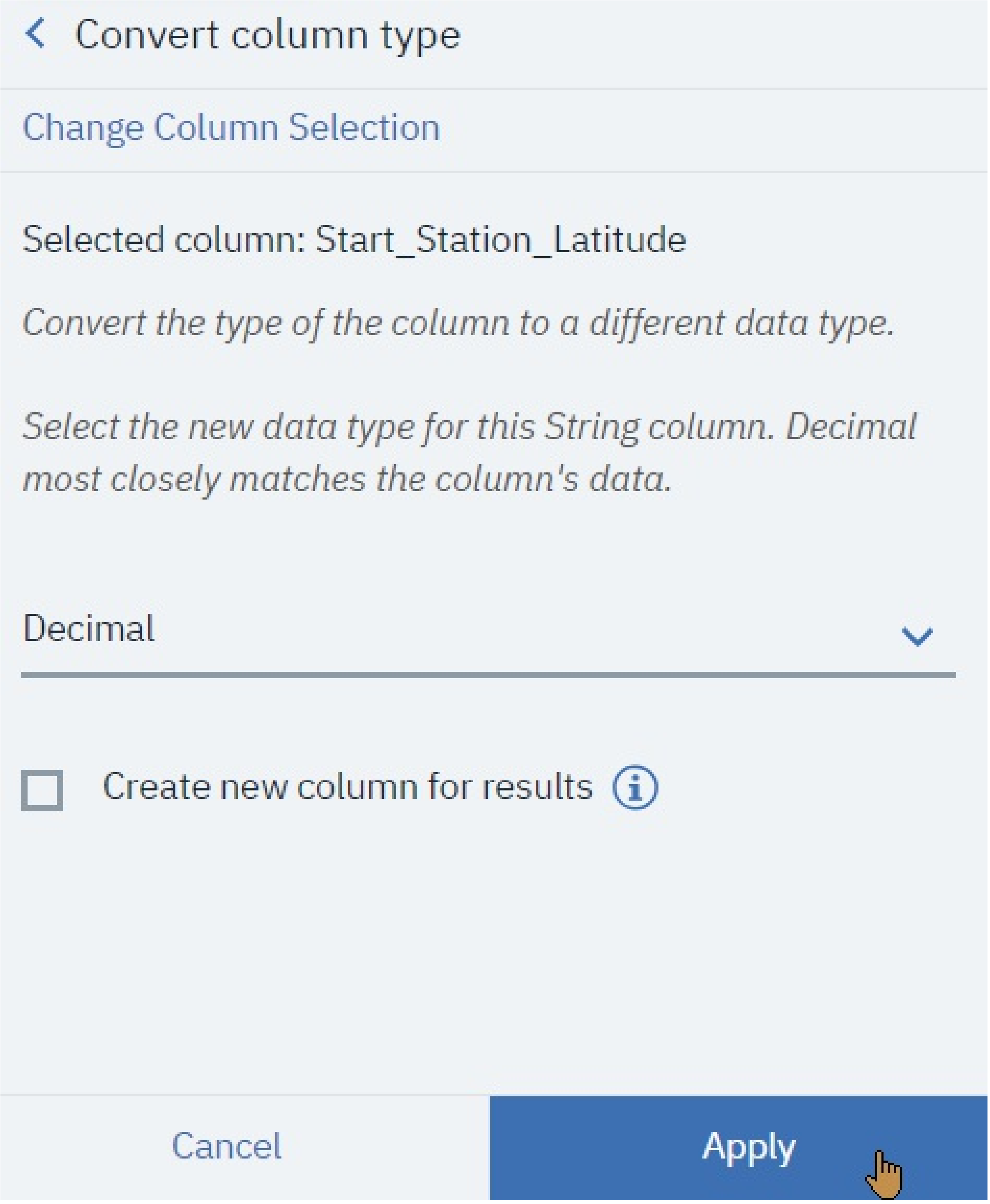
You can add the operation either from the `[+ Operation]` button at the top left, or from the column's header context menu:



Trip_Duration String	Start_Time String	Stop_Time String
680		2017-01-01 00:11:41
1282		2017-01-01 00:22:08
648		2017-01-01 00:11:46
631		2017-01-01 00:11:42
621		2017-01-01 00:11:47
666		2017-01-01 00:12:57
559		2017-01-01 00:14:20
826		
255		
634		
1081		
479	2017-01-01 00:08:00	
2005	2017-01-01 00:05:57	

. note in this case how the `integer` type is suggested with a small blue dot at its left.

3. Do the same for the 4 Start/End Latitude and Longitude columns, using `Decimal` as the type:



Convert column type

Change Column Selection

Selected column: Start_Station_Latitude

Convert the type of the column to a different data type.

Select the new data type for this String column. Decimal most closely matches the column's data.

Decimal

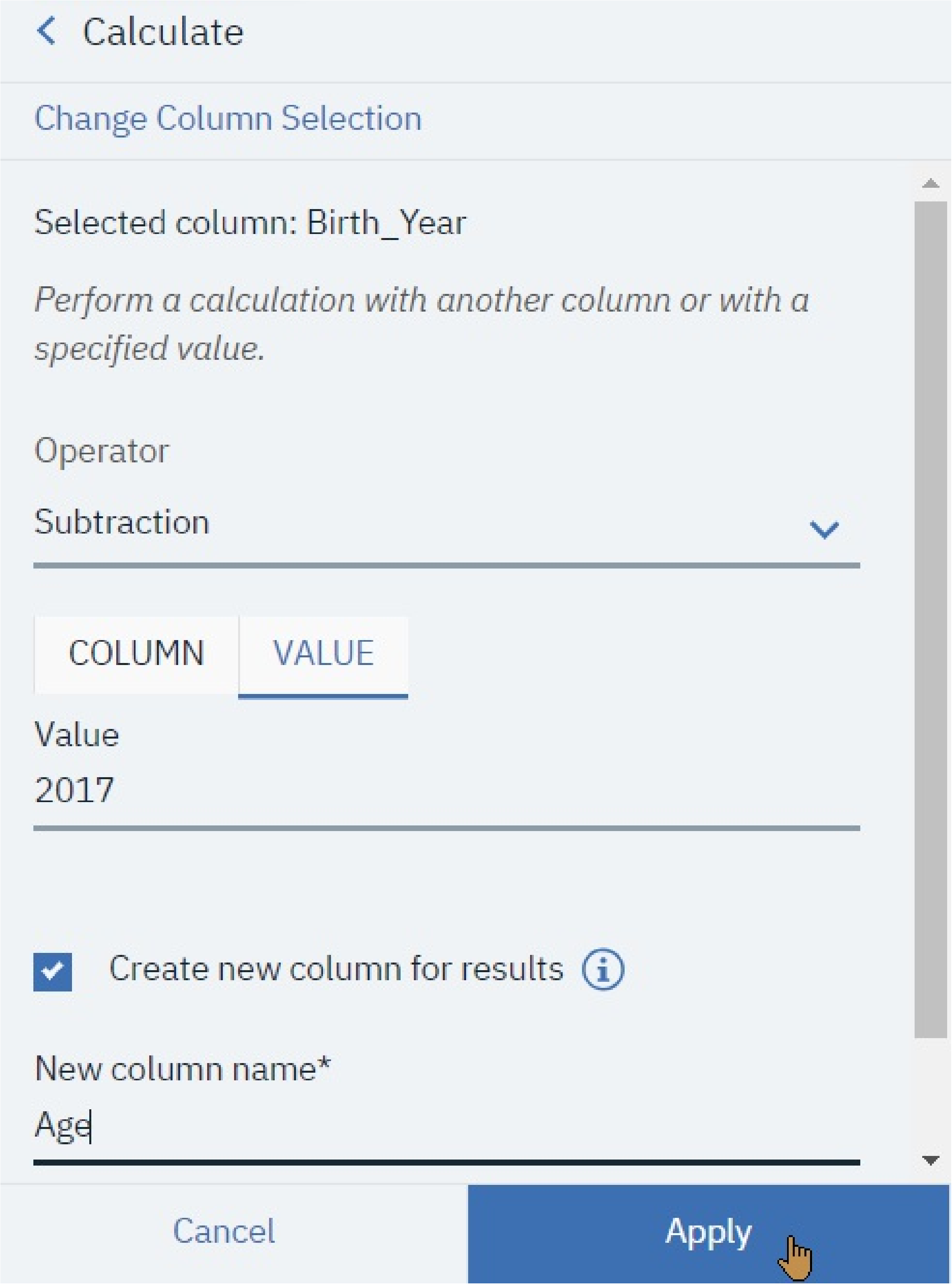
□ Create new column for results

. also note the suggested `decimal` type here.
You should now have 21 steps recorded.

[B.3]. Feature Engineering: Additional computed column

We will compute the age from the birth year. Since we have only the birth year, we will just use 2017 as the reference year from which to substract the birth and get an approximate age.
We will also remove all rows where `Age` is missing.

1. Add a `Calculate` operation, select `Birth_Year` as column, `Substraction` as operation, and `value` 2017.
2. Check the `Create new column for result` checkbox and enter `Age` as the new column name:



Calculate

Change Column Selection

Selected column: Birth_Year

Perform a calculation with another column or with a specified value.

Operator: Subtraction

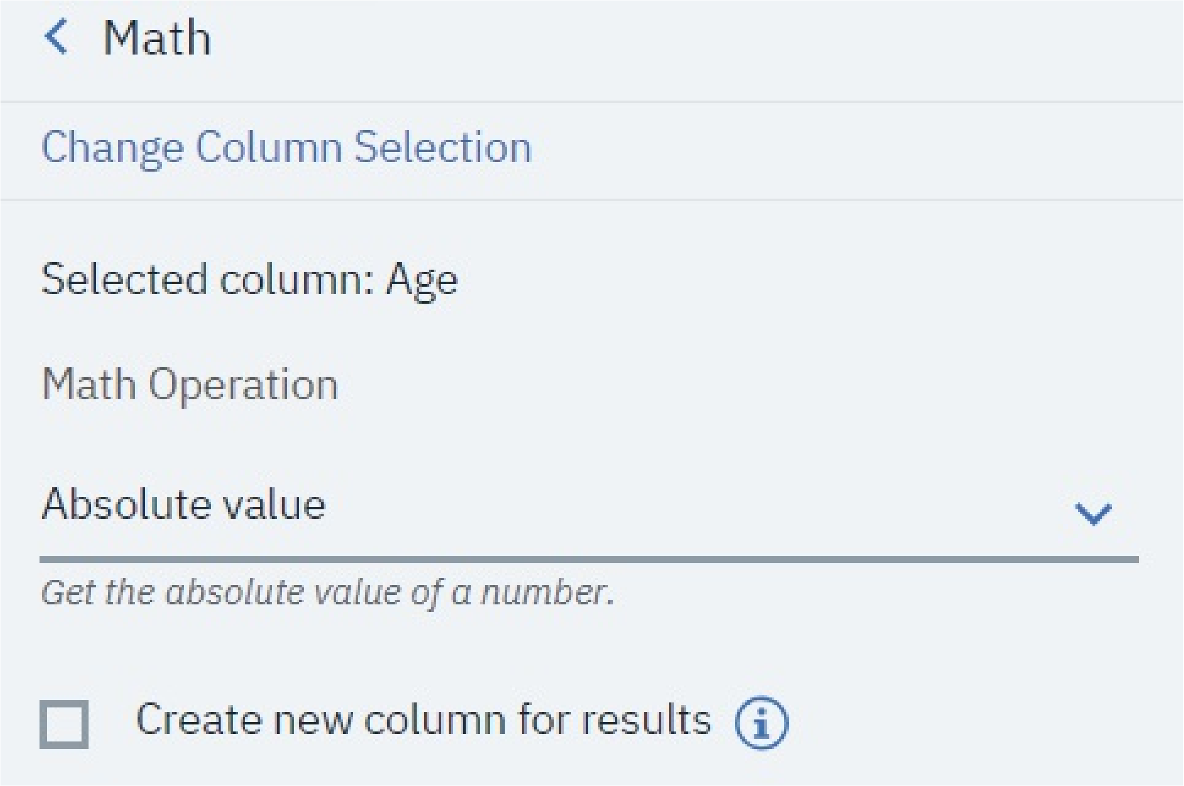
COLUMN | VALUE

Value: 2017

☑ Create new column for results

New column name*: Age

3. The compute age comes out negative, we will add a `Math` / `Absolute Value` operation to the `Age` column:



< Math

Change Column Selection

Selected column: Age

Math Operation

Absolute value

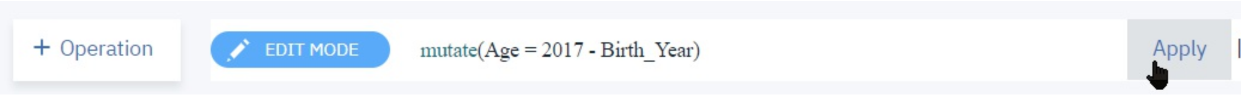
Get the absolute value of a number.

□ Create new column for results

> Note that at each step, you can see a preview of the data in the table.
> Verify that the values for `Age` column seem correct in the preview.

Also note that here we've used the UI-driven point-and-click style column ETL operations. It is also possible to add column operations using the guided formula operations entry at the top of the table preview.
For the age extraction operation, you could have entered a formula such as:

```
mutate(Age = 2017 - Birth_Year)
```

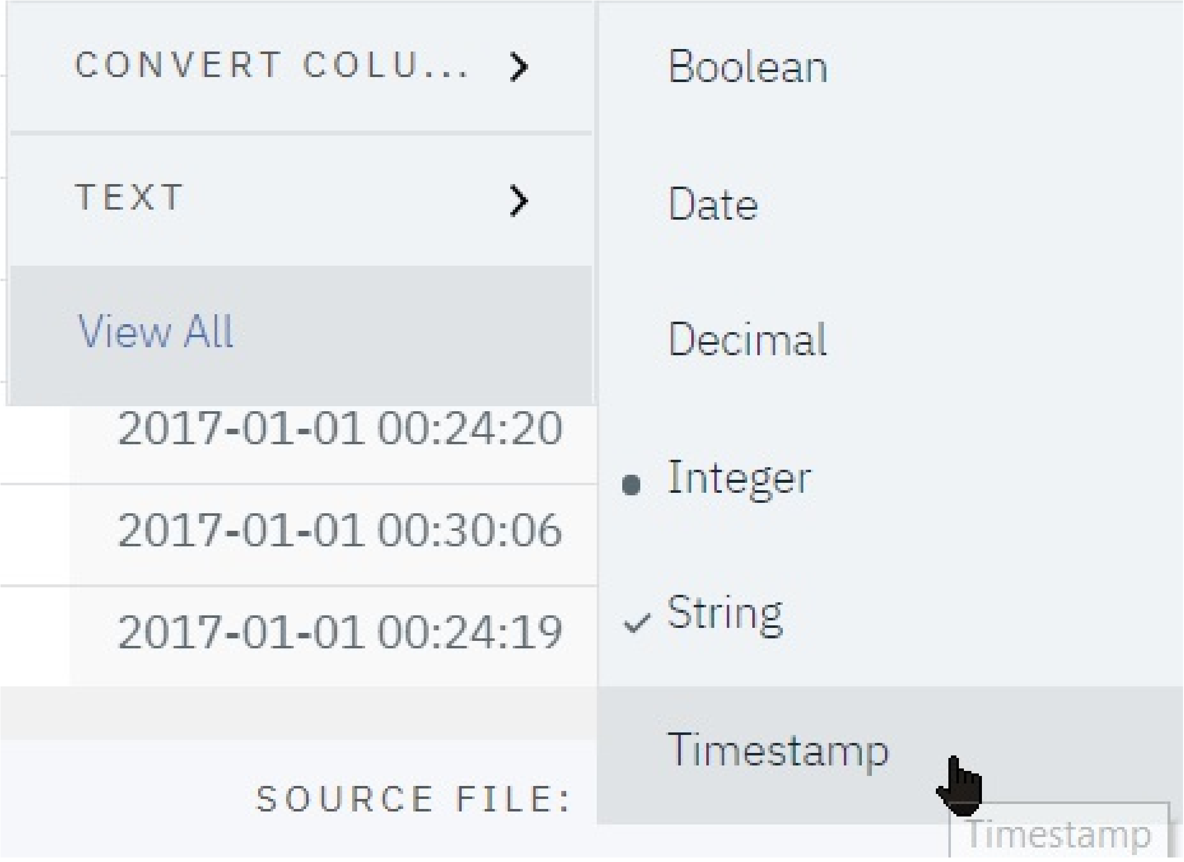


mutate(Age = 2017 - Birth_Year)

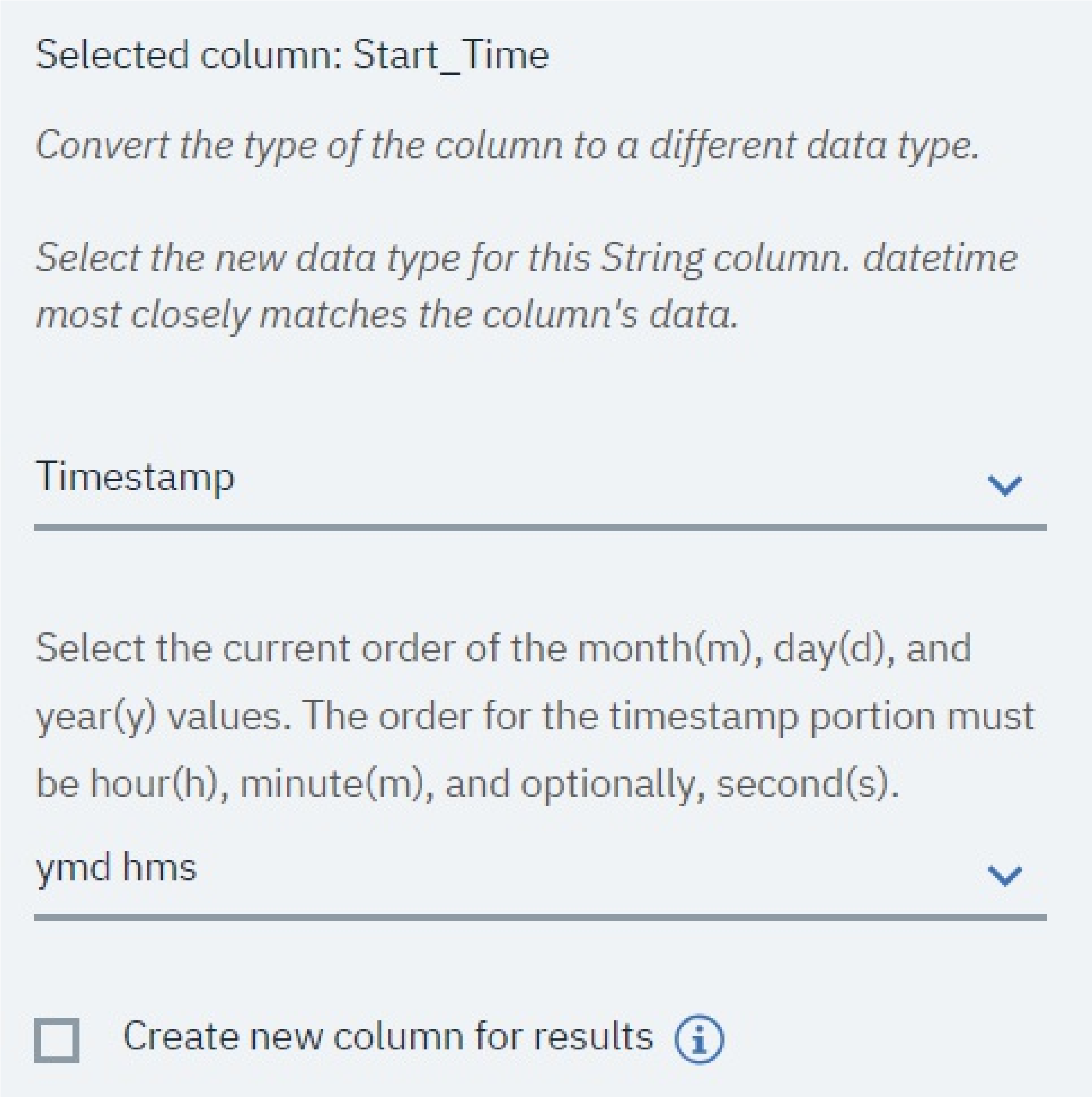
[B.4]. Feature Engineering: Process the timestamp columns

Finally, we will process the time fields. We will convert the date&time string format to a `Timestamp` type. We will then extract the `Date` into a new separate column, then the `Hour` slot from the timestamp into a new column typed `Integer`. For each of the `Start/Stop_Time` columns:

1. Select `Convert Column` to `Timestamp`:



Make sure to select `ymdhms` as the format, and do not create a new column:



Selected column: Start_Time

Convert the type of the column to a different data type.

Select the new data type for this String column. datetime most closely matches the column's data.

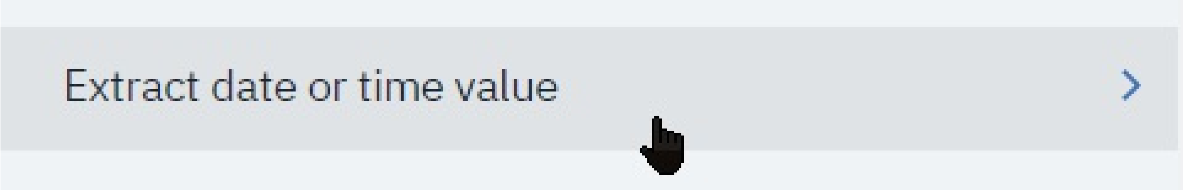
Timestamp

Select the current order of the month(m), day(d), and year(y) values. The order for the timestamp portion must be hour(h), minute(m), and optionally, second(s).

ymd hms

□ Create new column for results

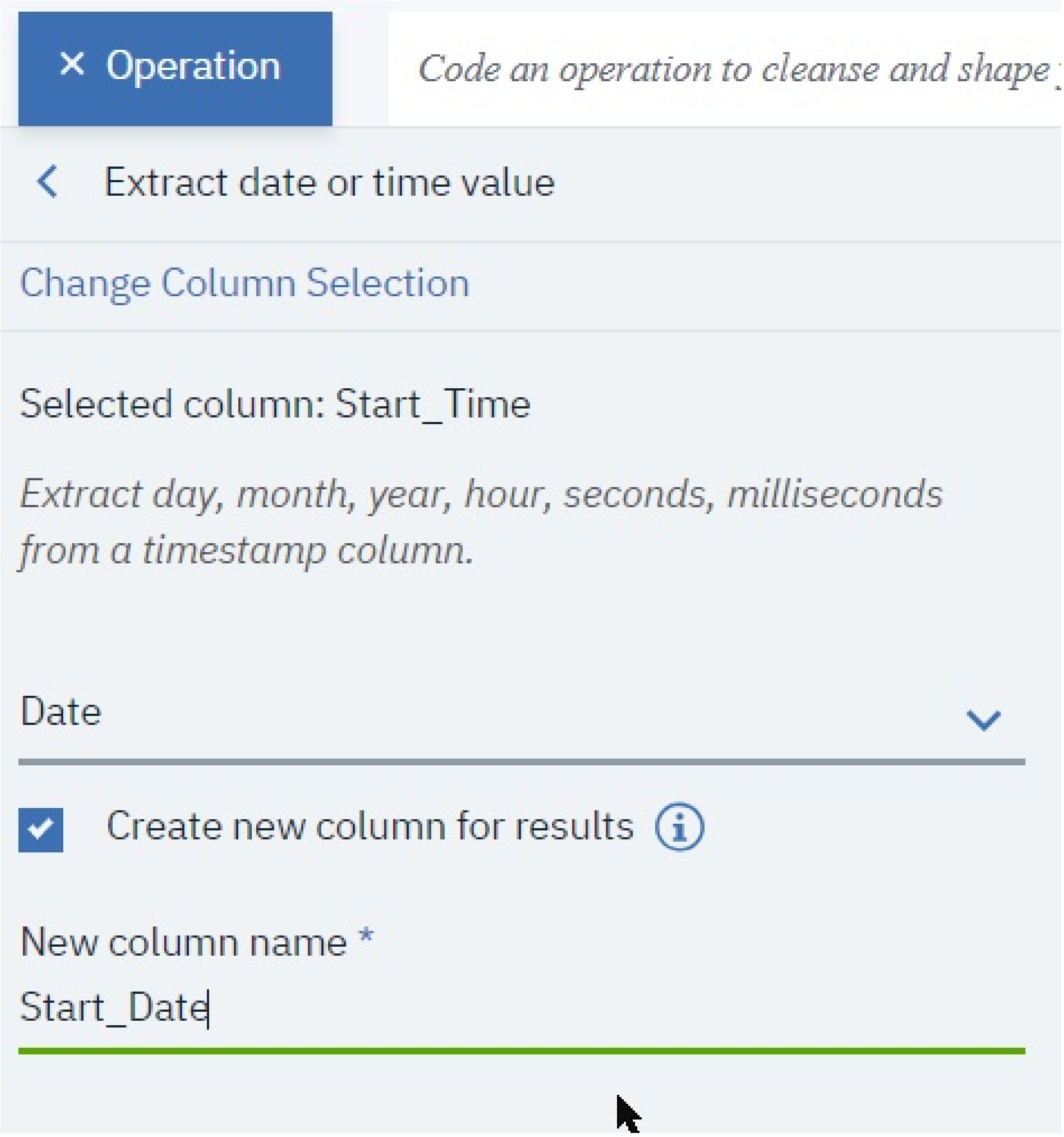
2. Extract the Date, using the `Extract date or time value` operation:



Extract date or time value

,

then `Date` and make sure to create new `Start/Stop_Date` columns:


× Operation
Code an operation to cleanse and shape
Extract date or time value
Change Column Selection

Selected column: Start_Time

Extract day, month, year, hour, seconds, milliseconds from a timestamp column.

Date

☑ Create new column for results

New column name *
Start_Date

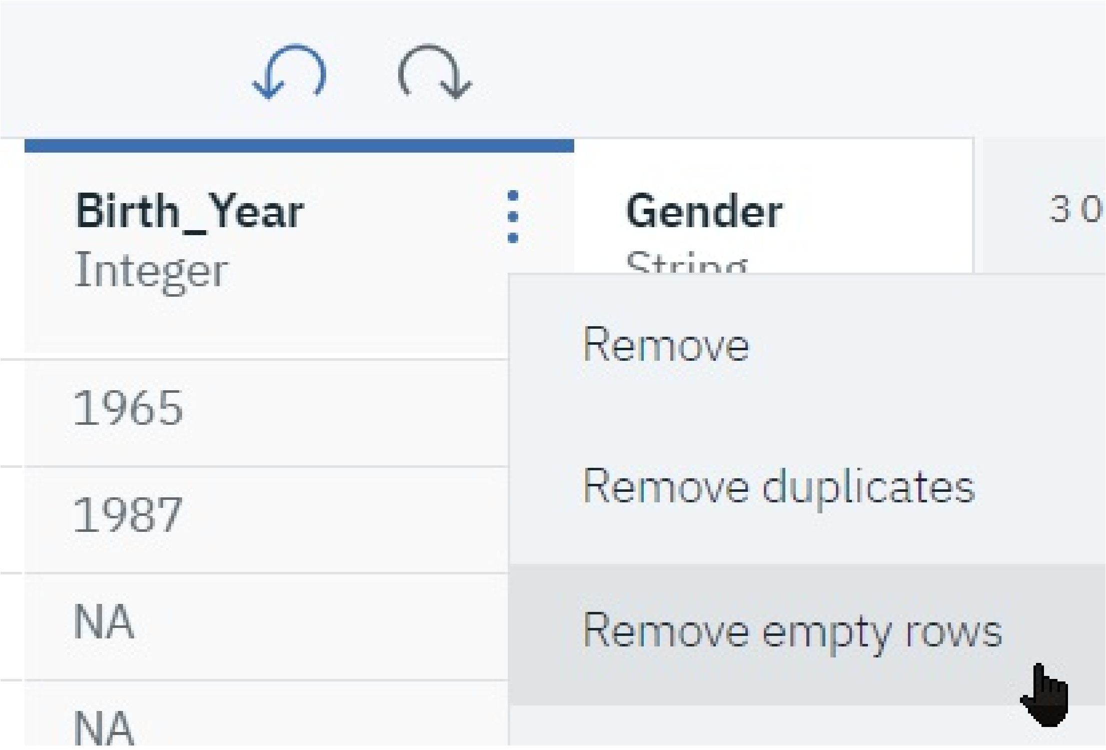
3. Repeat the operation to add new `Start/End_Hour` colunms

You should now have 29 steps defined.
Save your flow.

[B.5]. Data Cleansing: remove columns with no Birth_Date

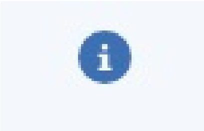
Some columns are missing the Birth_Year demographics information. We will remove the rows that have this field empty.
Note that we could (and maybe should) have added this step before computing the Age column.

1. From the `Birth_Year` column header context menu, select the `Remove empty rows` operation:

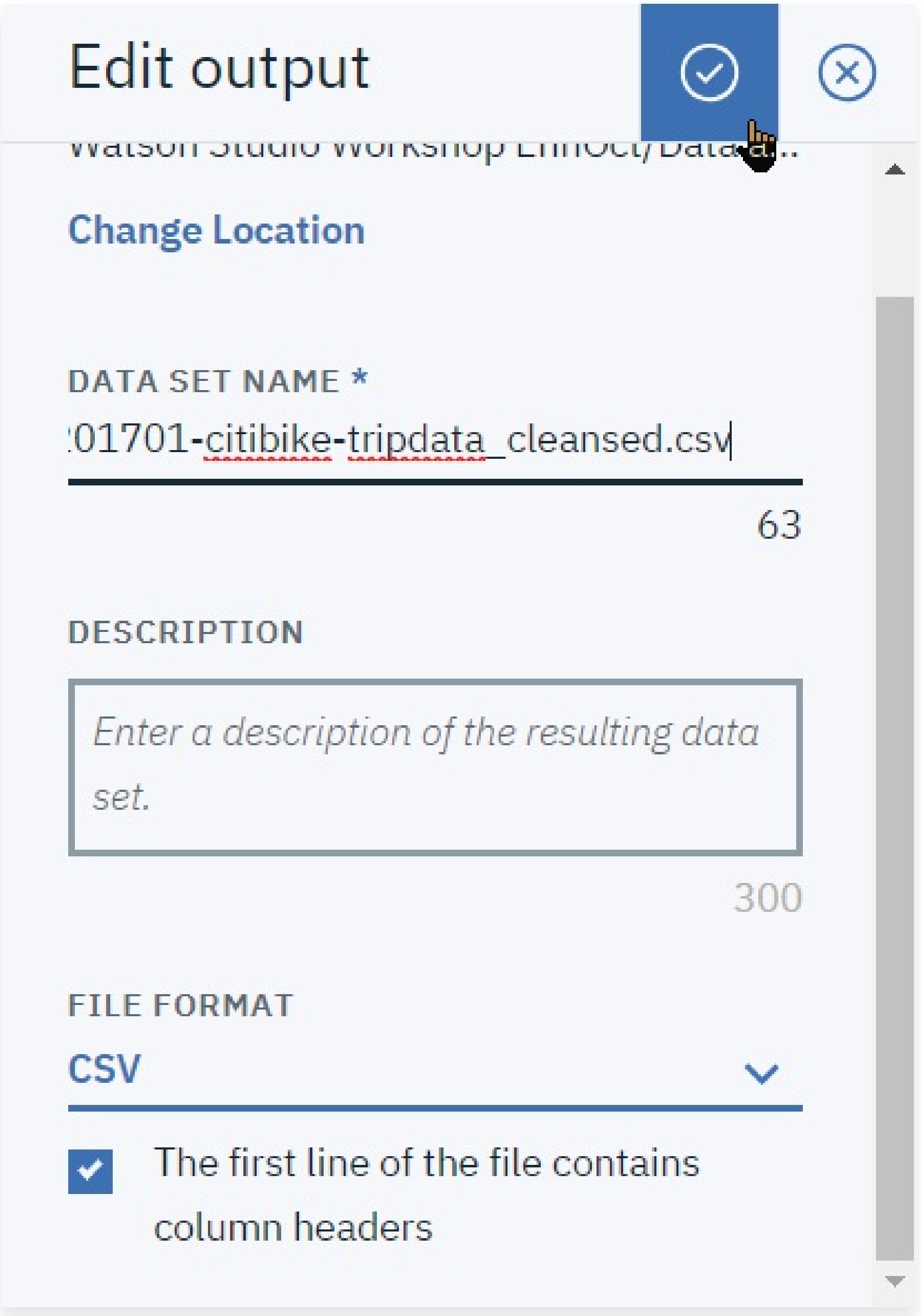


You should now have 30 steps defined.

[B.6] Change output file name

1. Select the `information`  button to reveal the details panel.
2. Click the `[Edit]` button

3. You are taken to the edit panel, change the output target file name to e.g. `201701-citibike-tripdata_cleansed.csv`, and specify format as `CSV`



Edit output

Watson Studio Workshop EnnOct/Data a...

Change Location

DATA SET NAME *

:01701-citibike-tripdata_cleansed.csv

63

DESCRIPTION

Enter a description of the resulting data set.

300

FILE FORMAT

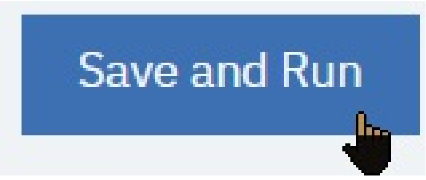
CSV

☑ The first line of the file contains column headers

> You could also select the Apache Parquet file format (PARQ) which has the advantage of retaining more type information than CSV for the columns, and can be used as input in notebooks or further data refinery operations.

> Parquet Apache Parquet is a free and open-source column-oriented data storage format of the Apache Hadoop ecosystem. It is similar to the other columnar-storage file formats available in Hadoop namely RCFile and ORC. It is compatible with most of the data processing frameworks in the Hadoop environment. It provides efficient data compression and encoding schemes with enhanced performance to handle complex data in bulk.

1. Finally click the `Save and Run` button:

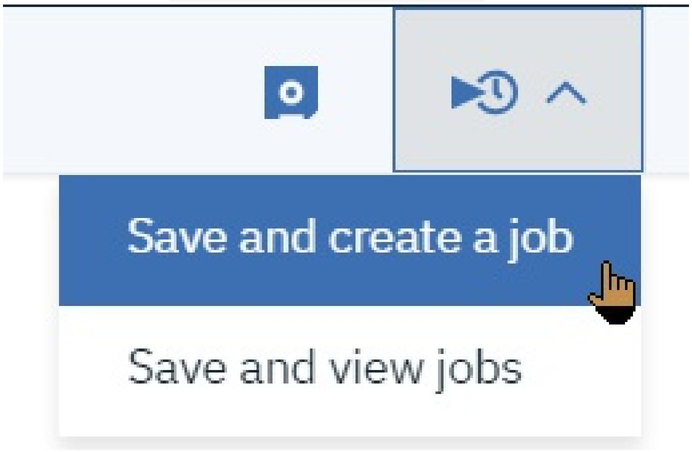


> Notice that you could have schedule your data flow to run on a defined time of the day.

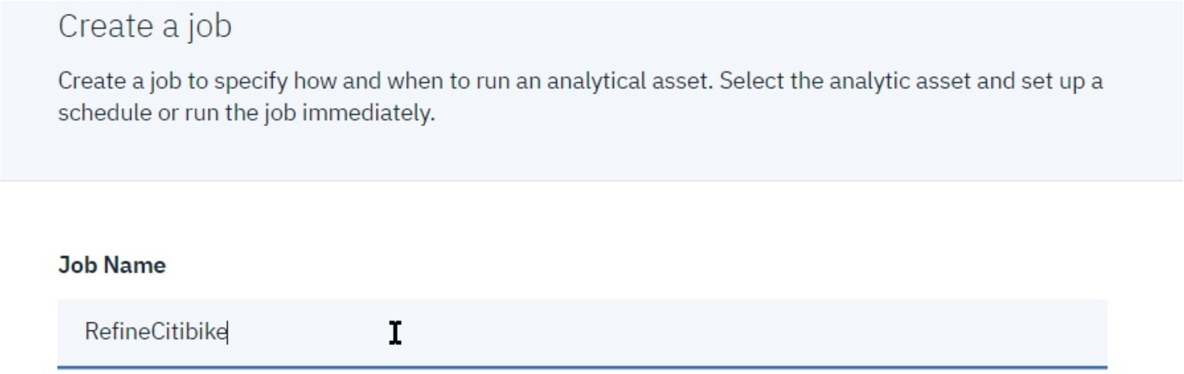
[B.7]. Apply Data Flow pipeline to the input files

We will now process the entire file with our data cleansing and feature engineering pipeline.

1. Click on the `Save and Create a Job` icon at the top right:



2. Enter a name for the job, e.g. `RefineCitibike`:



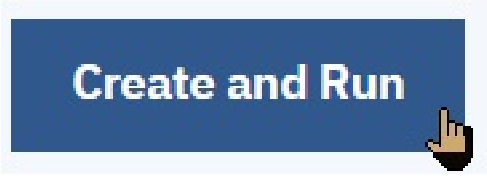
Create a job

Create a job to specify how and when to run an analytical asset. Select the analytic asset and set up a schedule or run the job immediately.

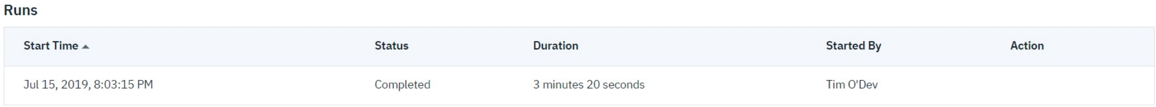
Job Name

RefineCitibike

3. Select the `[Create and run]` button



4. The job will run and report its status

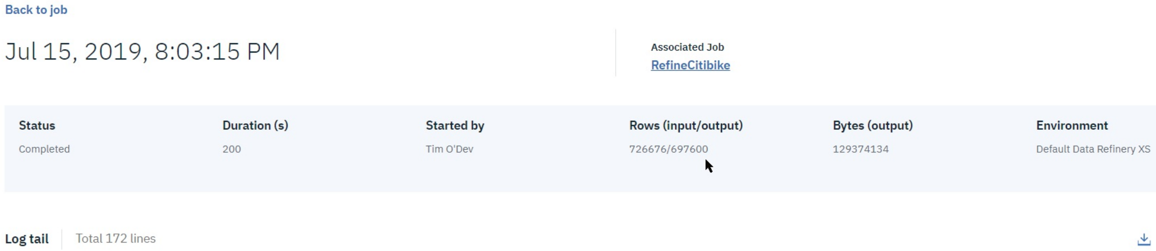


Runs

Start Time ▲	Status	Duration	Started By	Action
Jul 15, 2019, 8:03:15 PM	Completed	3 minutes 20 seconds	Tim O'Dev	

, wait for the flow to complete processing.

5. You can click on the job's line to open the log and report, which includes the read and written rows count:



Back to job

Jul 15, 2019, 8:03:15 PM

Associated Job
RefineCitibike

Status	Duration (s)	Started by	Rows (input/output)	Bytes (output)	Environment
Completed	200	Tim O'Dev	726676/697600	129374134	Default Data Refinery XS

Log tail | Total 172 lines

6. Once executed, you can go back to the project assets, and you will find the generated `201701-citibike-tripdata_cleansed.csv` or `201701-citibike-tripdata_cleansed.parq` file that you can browse by clicking on it. We will reuse this file in the following part of the Labs.
7. The same processing pipeline could be applied to another file than the one used to create the Data Flow.

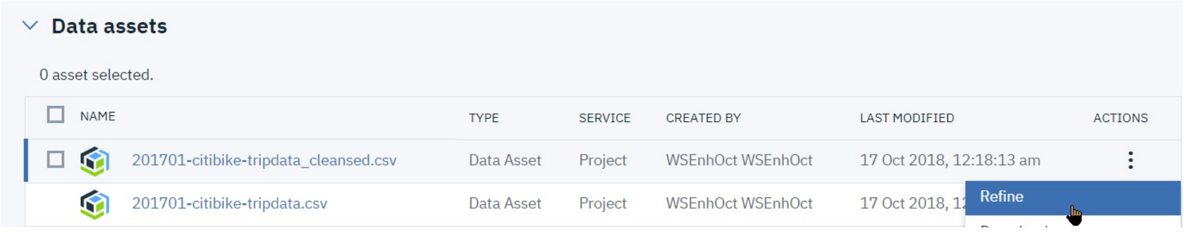
[B.8]. OPTIONAL: Data Refinery Stretch Lab

The instructions below guide you through more advanced functionality of Data Refinery. These steps are optional, you may execute them if you feel comfortable enough with the toolset.
We will show how to reduce the volums of data through aggregation functions.

Optional 1: Aggregation by day and station

From the output of the Data Refinery flow created above, we will prepare another flow which aggregates bike departures per day and starting station.
This aggregated data asset is less voluminous than the original one and could be used for efficient reporting and dashboarding.
This how the use of `group_by` functionality.

1. Go back to your project's Assets list, select your `201701-citibike-tripdata_cleansed.csv` Data Asset, and select `Refine` from its `Actions` menu:

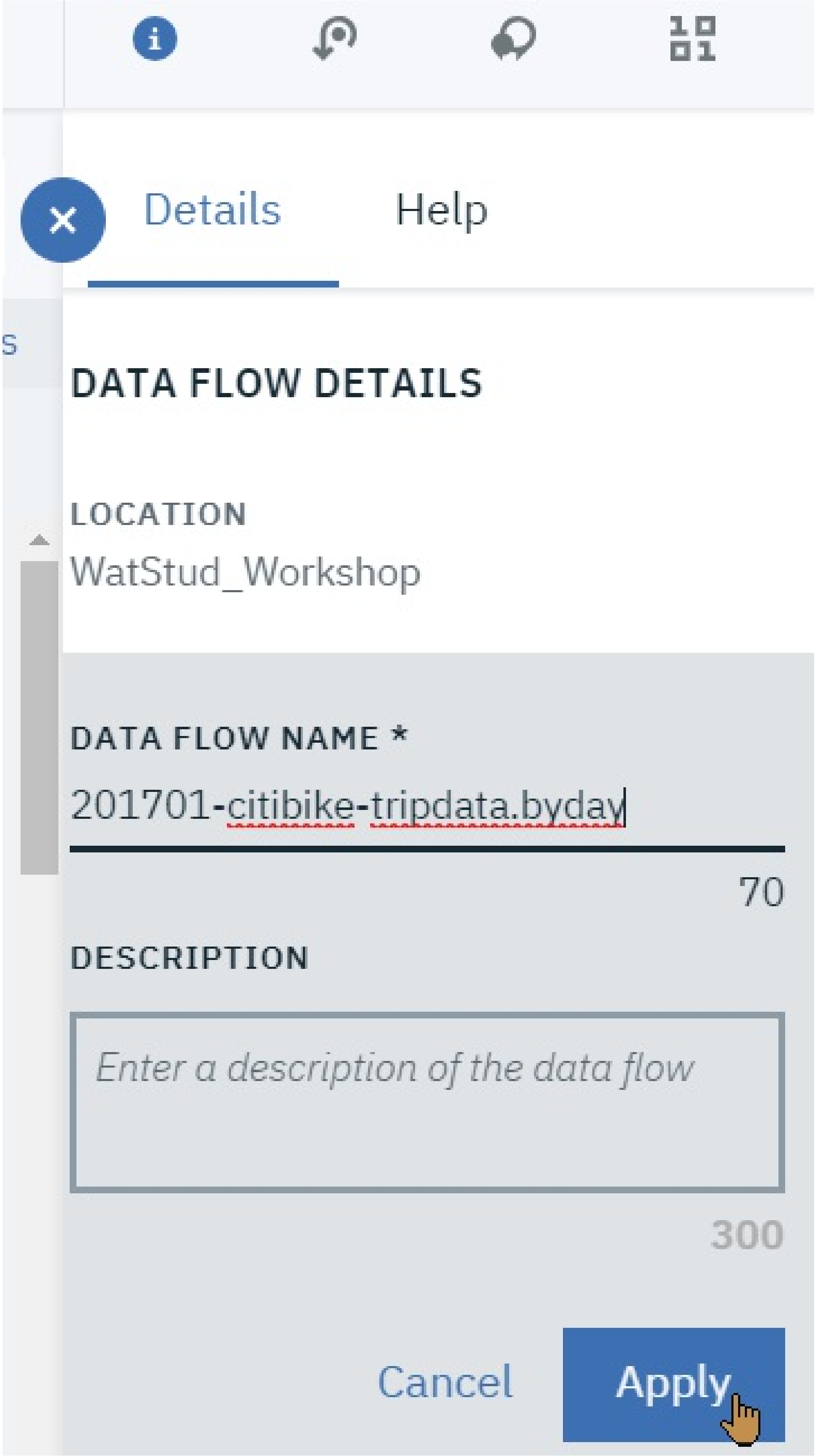


Data assets

0 asset selected.

□	NAME	TYPE	SERVICE	CREATED BY	LAST MODIFIED	ACTIONS
□	201701-citibike-tripdata_cleansed.csv	Data Asset	Project	WSEnhOct WSEnhOct	17 Oct 2018, 12:18:13 am	⋮
	201701-citibike-tripdata.csv	Data Asset	Project	WSEnhOct WSEnhOct	17 Oct 2018, 12	

2. Change the name of the flow to `201701-citibike-tripdata.byday`:



DATA FLOW DETAILS

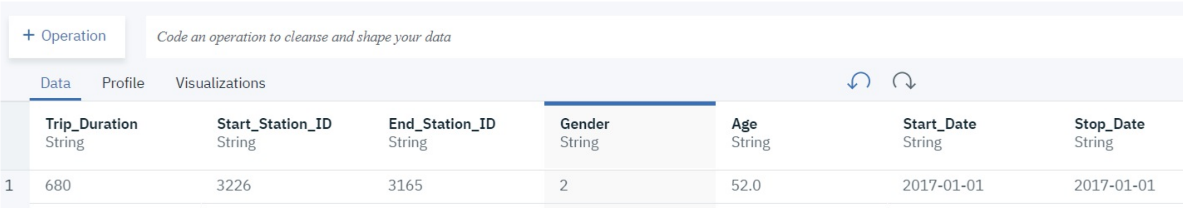
LOCATION
WatStud_Workshop

DATA FLOW NAME *
201701-citibike-tripdata.byday
70

DESCRIPTION
Enter a description of the data flow
300

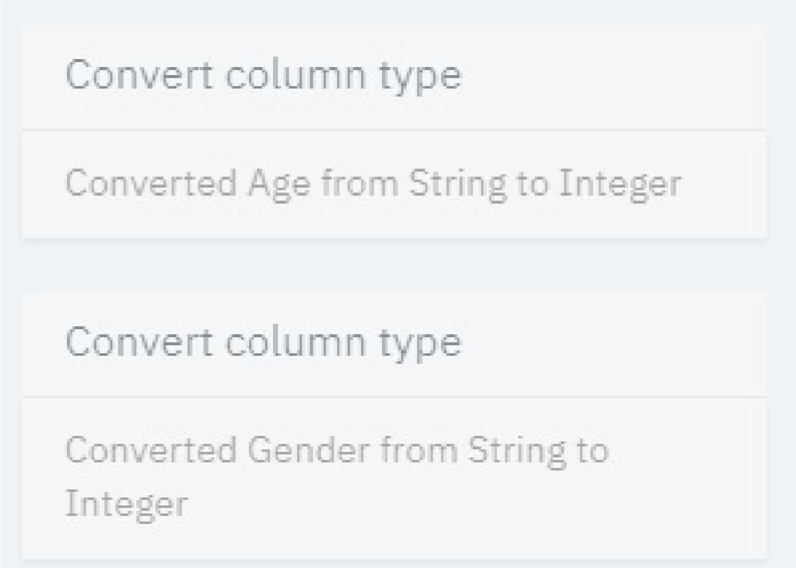
3. Add operations to Remove the 6 columns `Start/End Station Name/Latitude/Longitude`, as well as column `User_Type`, `Bike_ID` and `Birth_Year`. Also Remove `Start/Stop_Time` and `Start/Stop_Hour`.
   At the end, here should be only 7 columns remaining: `Trip_Duration`,

`Start/End_Station_ID` , `Gender` , `Age` , `Start/Stop_Date` , with 13 recorded steps:



	Trip_Duration String	Start_Station_ID String	End_Station_ID String	Gender String	Age String	Start_Date String	Stop_Date String
1	680	3226	3165	2	52.0	2017-01-01	2017-01-01

4. We need to convert the `Age` and `Trip_Duration` to `Integer` format if not already in the correct format, so that we can compute their average, add two `Convert column type` operations:



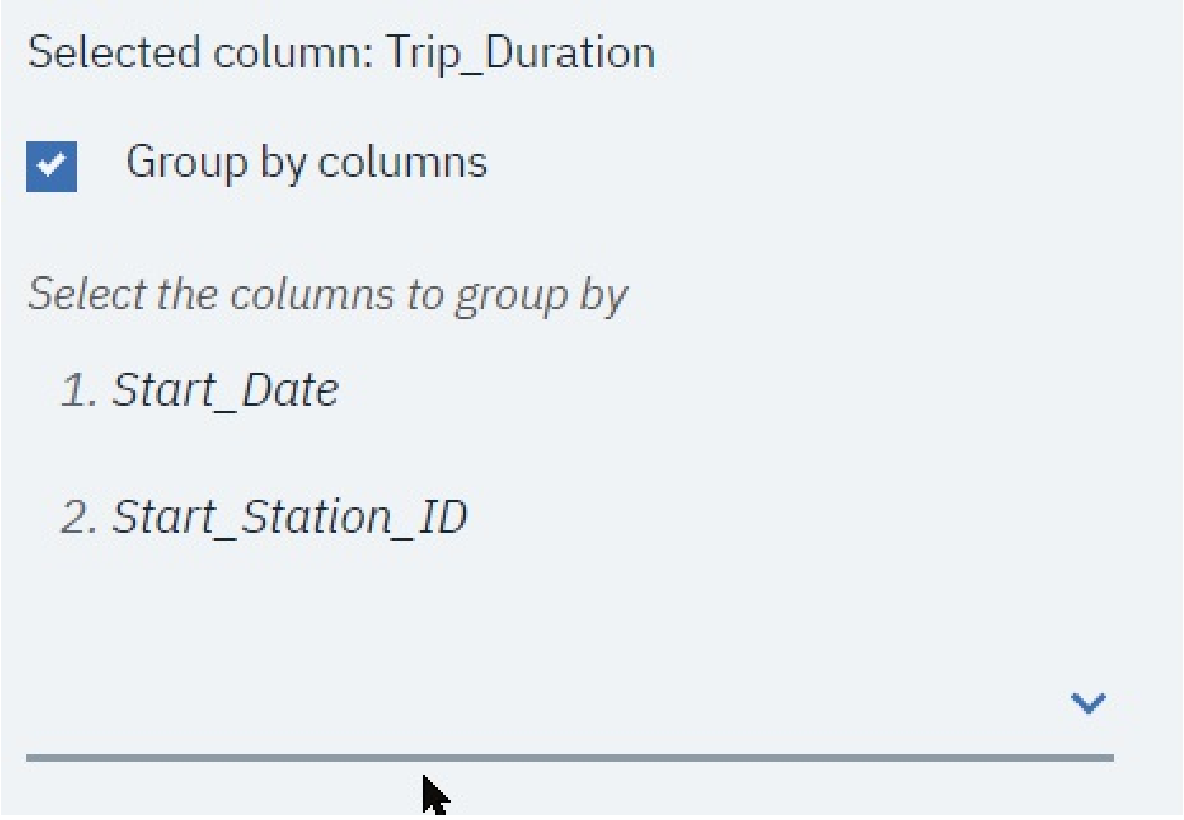
Convert column type

Converted Age from String to Integer

Convert column type

Converted Gender from String to Integer

5. You can also change the `Start_Date` and the `End_Date` data type to Date (ymd).
6. Select the `Trip_Duration` command, then the `Aggregate` operation, then check `Group by columns` , and select columns `Start_Date` and `Start_Station_ID` :



Selected column: Trip_Duration

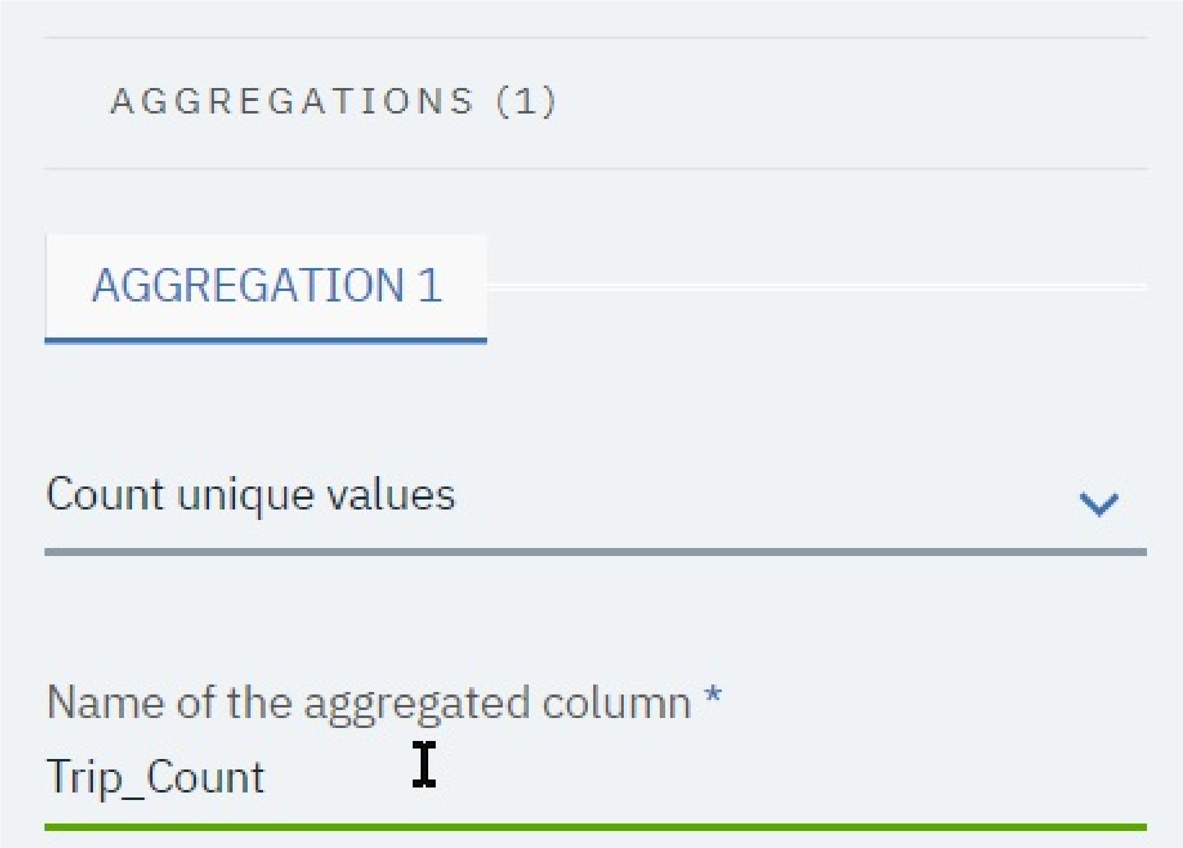
☑ Group by columns

Select the columns to group by

1. Start_Date

2. Start_Station_ID

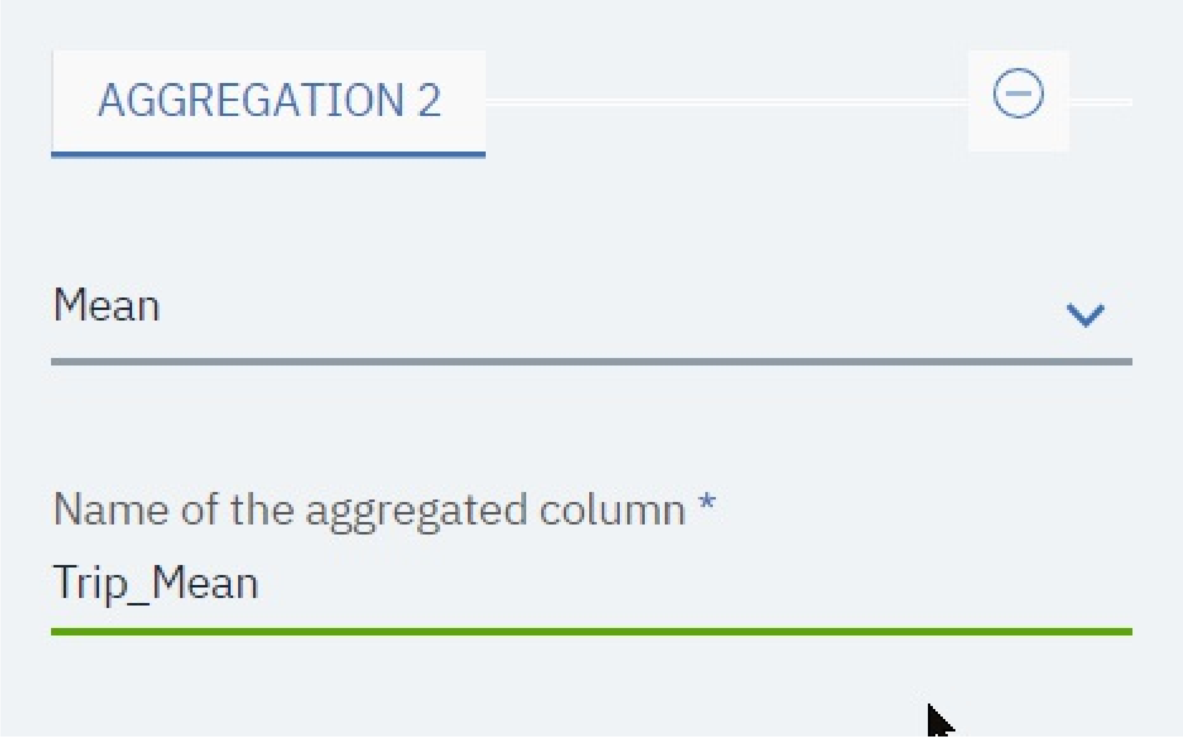
7. Select `Aggregation 1` as `Count unique values` and name it `Trip_Count`:



Count unique values

Name of the aggregated column *
Trip_Count

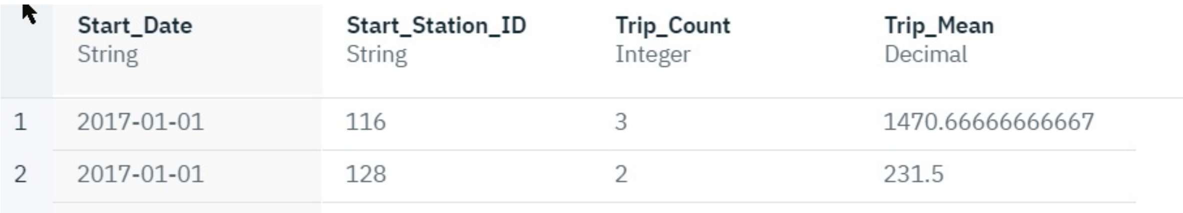
8. Add a second aggregation, of type `Mean` and name it `Trip_Mean`



Mean

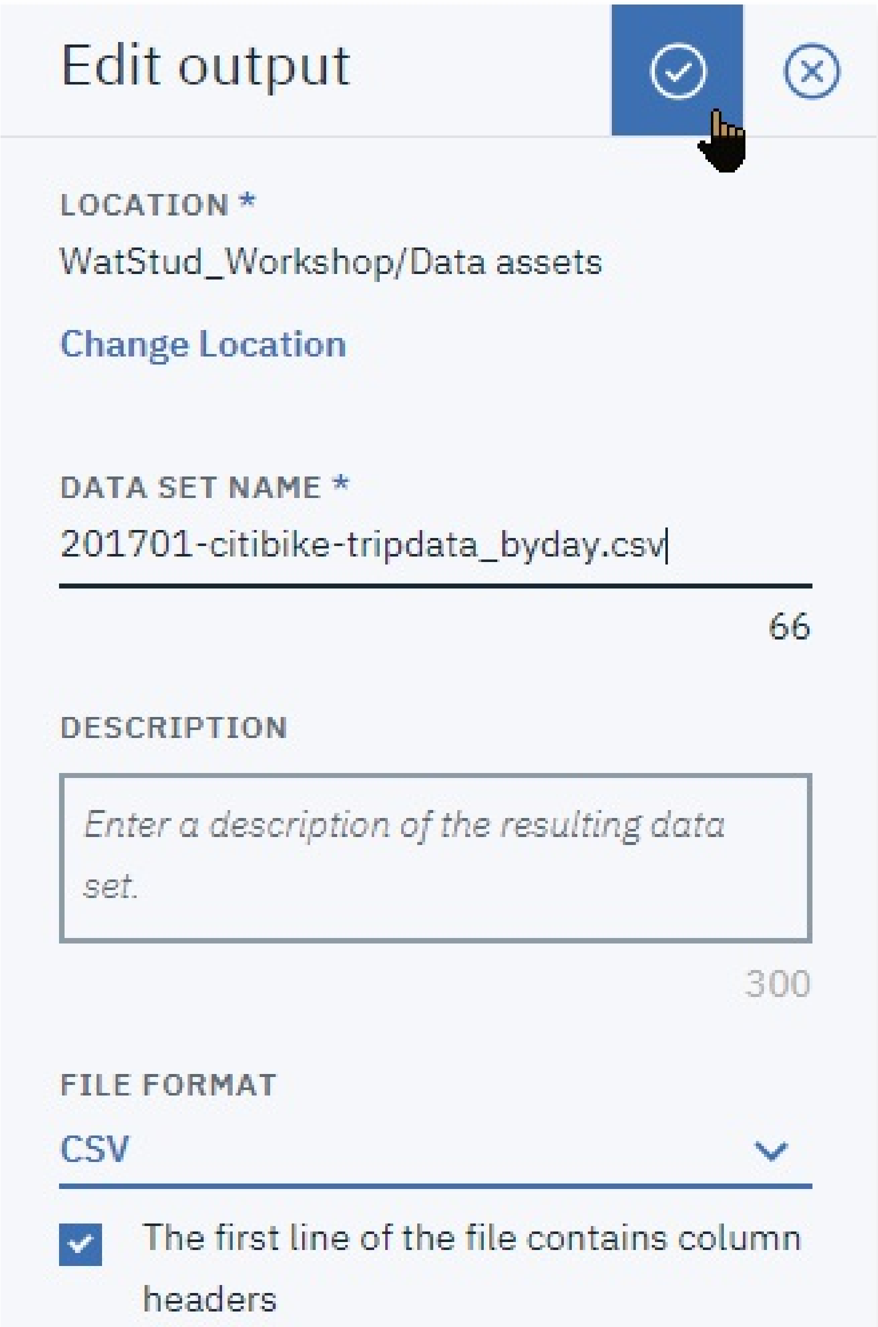
Name of the aggregated column *
Trip_Mean

9. Apply, and you will get 4 columns as result:



	Start_Date String	Start_Station_ID String	Trip_Count Integer	Trip_Mean Decimal
1	2017-01-01	116	3	1470.66666666667
2	2017-01-01	128	2	231.5

10. Save this new flow and run it. Change the output Data Set Name to `201701-citibike-tripdata_byday.csv` in CSV format:



Edit output

LOCATION *

WatStud_Workshop/Data assets

Change Location

DATA SET NAME *

201701-citibike-tripdata_byday.csv

66

DESCRIPTION

Enter a description of the resulting data set.

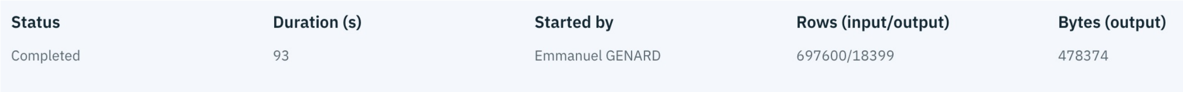
300

FILE FORMAT

CSV

☑ The first line of the file contains column headers

11. After running, the flow should yield a new Data Asset with much less rows, 18399 vs the original 697600:

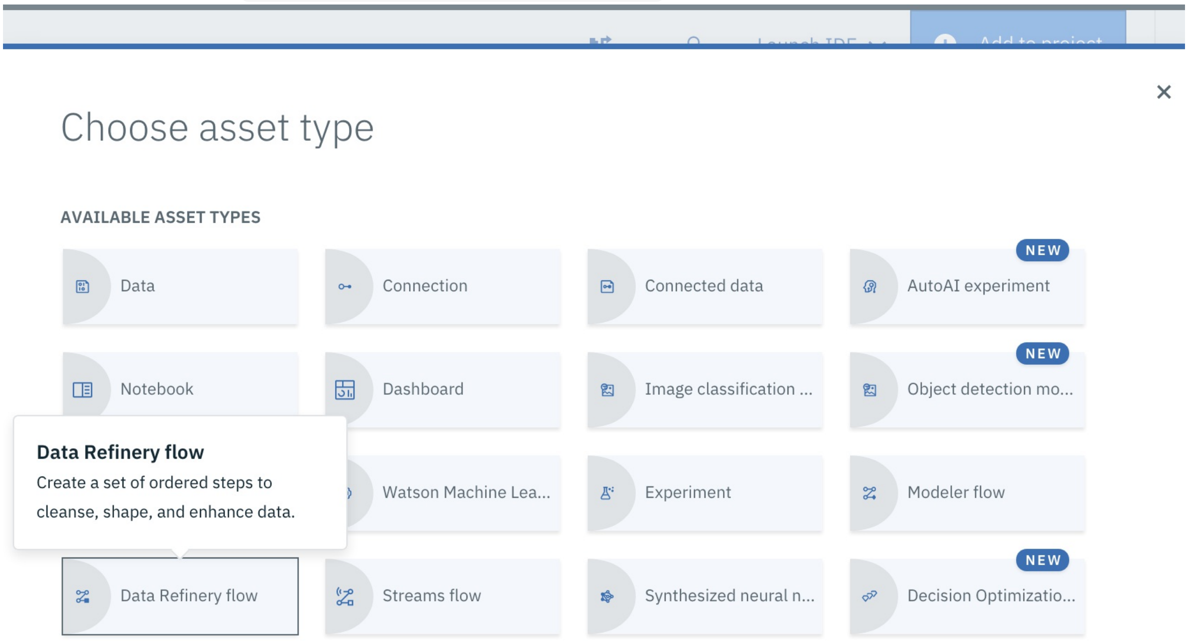


Status	Duration (s)	Started by	Rows (input/output)	Bytes (output)
Completed	93	Emmanuel GENARD	697600/18399	478374

Optional 2: Extract Station Name, Latitude, Longitude by ID

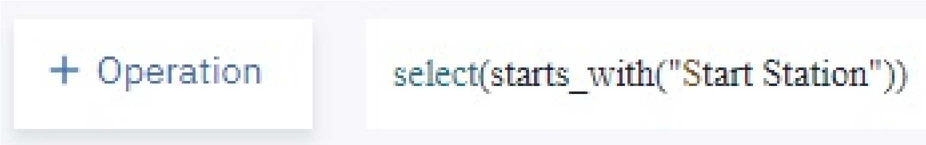
In this section, we will create a Data Flow which just extracts the stations data from the main dataset, which holds redundant information, keeping for each `StationID` its `Station_Name` and `Latitude` , `Longitude` .

1. Create a new Data Flow from the Add to Project > Data Refinery Flow button and then select the original `201701-citibike-tripdata.csv` Data Asset and click Add.



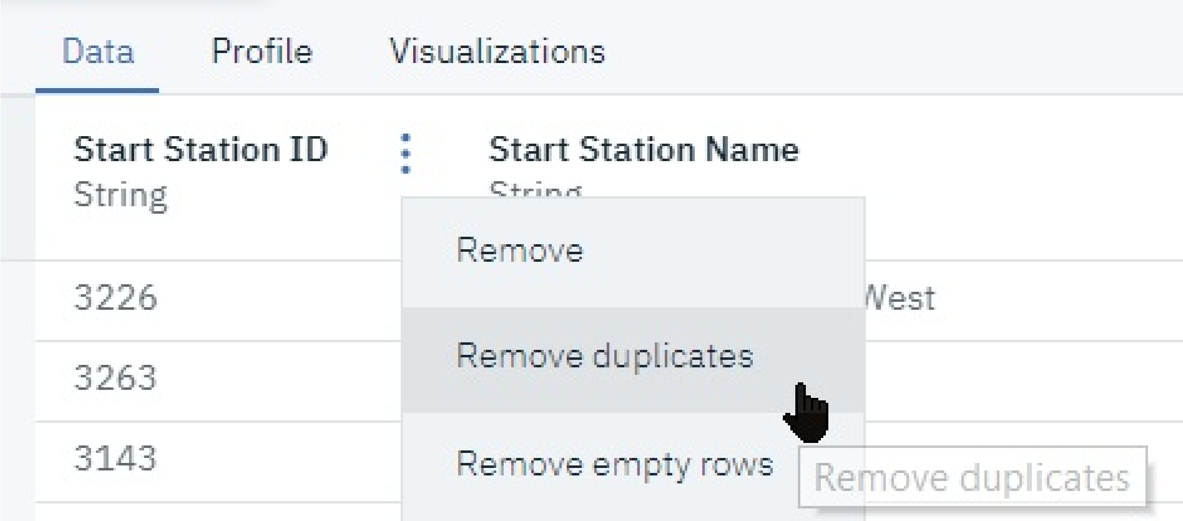
Data Refinery flow
Create a set of ordered steps to cleanse, shape, and enhance data.

2. Remove all the columns except those that start with `Start Station` : Add a `select` operation `select(starts_with("Start Station"))` :

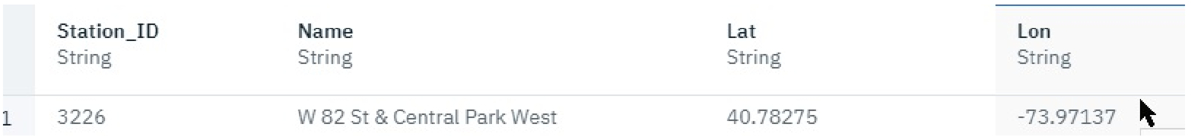


```
select(starts_with("Start Station"))
```

3. Coallesce all the rows by removing duplicate `Station ID` :

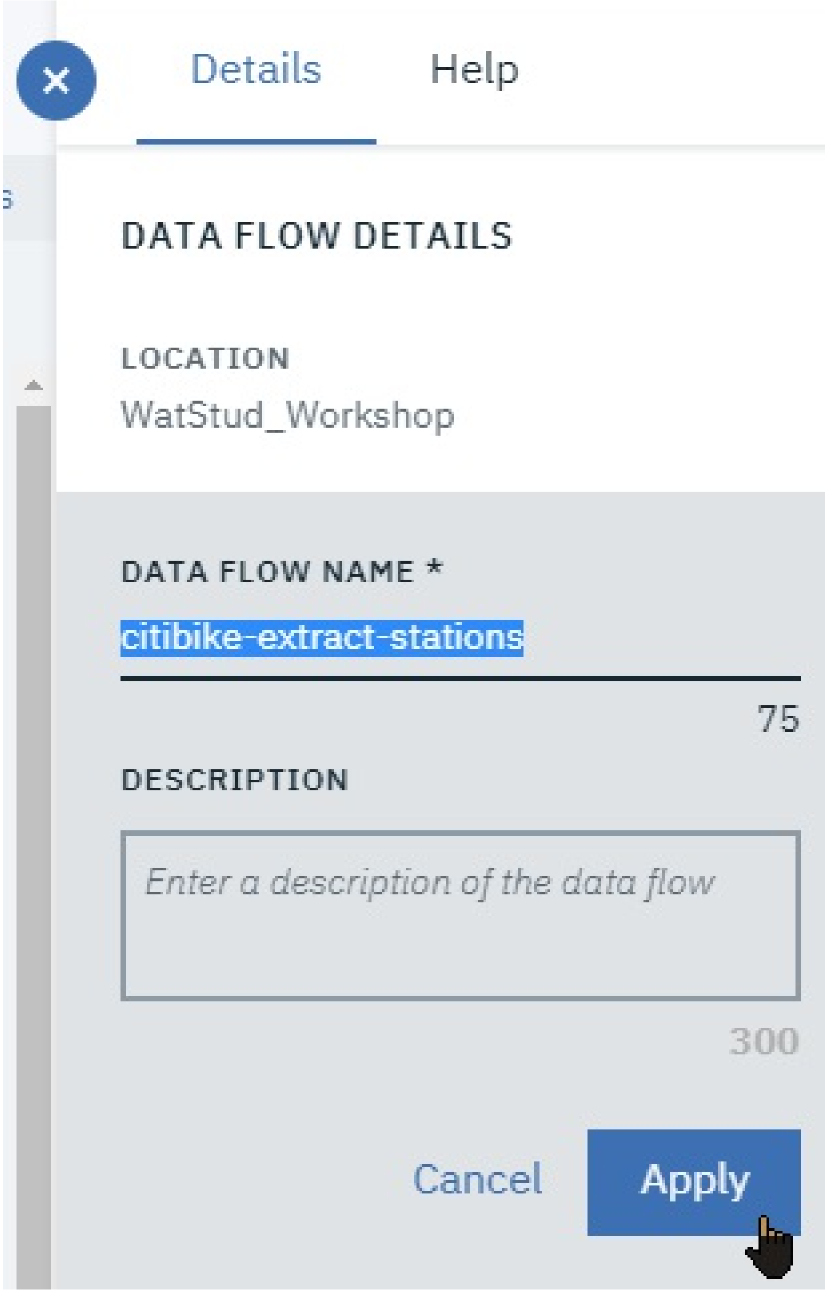


4. Change the column names to `Station_ID` , `Name` , `Latitude` , `Longitude` :



	Station_ID String	Name String	Lat String	Lon String
1	3226	W 82 St & Central Park West	40.78275	-73.97137

5. Change the flow name to `citibike-extract-stations` :



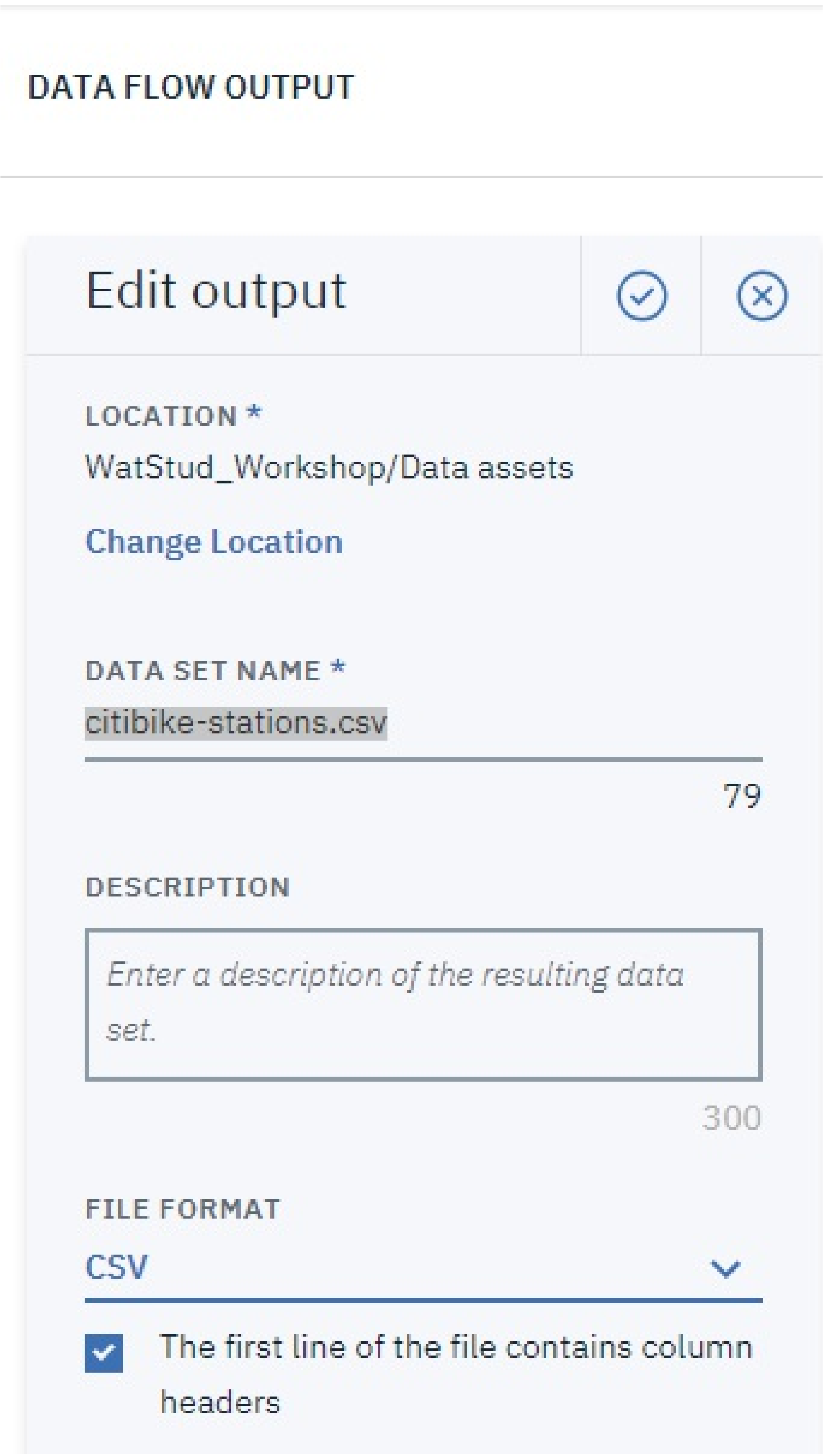
DATA FLOW DETAILS

LOCATION
WatStud_Workshop

DATA FLOW NAME *
citibike-extract-stations
75

DESCRIPTION
Enter a description of the data flow
300

6. Save and execute the flow, change the output file name to `citibike-stations.csv` :



DATA FLOW OUTPUT

Edit output

LOCATION *

WatStud_Workshop/Data assets

Change Location

DATA SET NAME *

citibike-stations.csv

79

DESCRIPTION

Enter a description of the resulting data set.

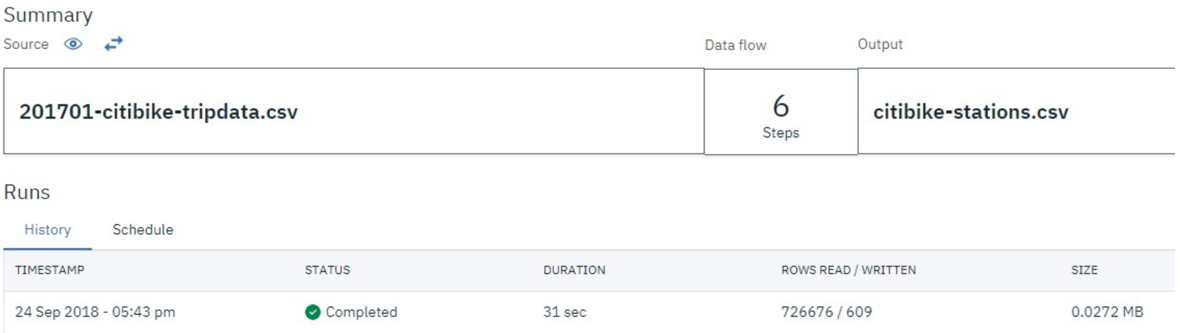
300

FILE FORMAT

CSV

☑ The first line of the file contains column headers

7. After `[Save and Run]`, then `[View flow]`, you should get an output file with 609 rows:



Summary

Source

Source	Data flow	Output
201701-citibike-tripdata.csv	6 Steps	citibike-stations.csv

Runs

TIMESTAMP	STATUS	DURATION	ROWS READ / WRITTEN	SIZE
24 Sep 2018 - 05:43 pm	✓ Completed	31 sec	726676 / 609	0.0272 MB

Conclusion of Data Refinery section

We have experienced the Data Refinery which is Watson Studio's integrated ETL (Extract, Transform and Load) tool. You have seen that the tool is designed to define ETL operations without coding, even though it can be complemented by formulas.

In a Data Science pipeline, ETL tools are almost always required as first steps in the data processing. It allows to perform Data Cleansing and Feature Engineering.

A word on file type conversion

Data Refinery also allows to generate data Asset output in `Parquet` or `Avro` file formats, which are file formats specified as part of the Apache Hadoop project, optimized for respectively column and row-oriented data storage and retrieval in a Hadoop or more generally Data Science environment.
Parquet is not as efficient as zipping a file, but can readily be used by data processing tools, and it carries meta data information such as column types.
In the case of this input file, the resulting parquet conversion would yield a file of about 42 MB, vs 116 MB for the raw CSV file and 23 MB for the zipped CSV.

[C]. Using notebooks for data exploration

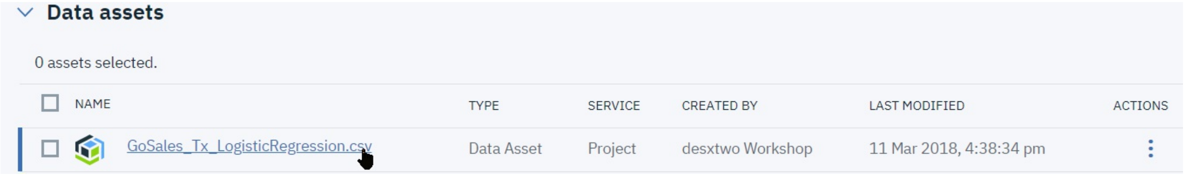
In this section, we will start exploring the data from a file which holds customer sales observations related to buying behavior of customers of an outdoor equipment company regarding tent purchases, using a Jupyter notebook.

This is a different approach to data analysis than the GUI-driven tools such as Data Refinery, here the paradigm is to perform programmatic operations on data files rather than GUI driven. Each approach has its pros and cons, and selecting one versus the other can be a matter of personal preference.

[C.1]. Explore the data set

Ensure that the `GoSales_Tx.csv` file is part of the data assets, so that we can start to have a look at the data:

1. open the corresponding asset by clicking on the file name from the list

   Data assets

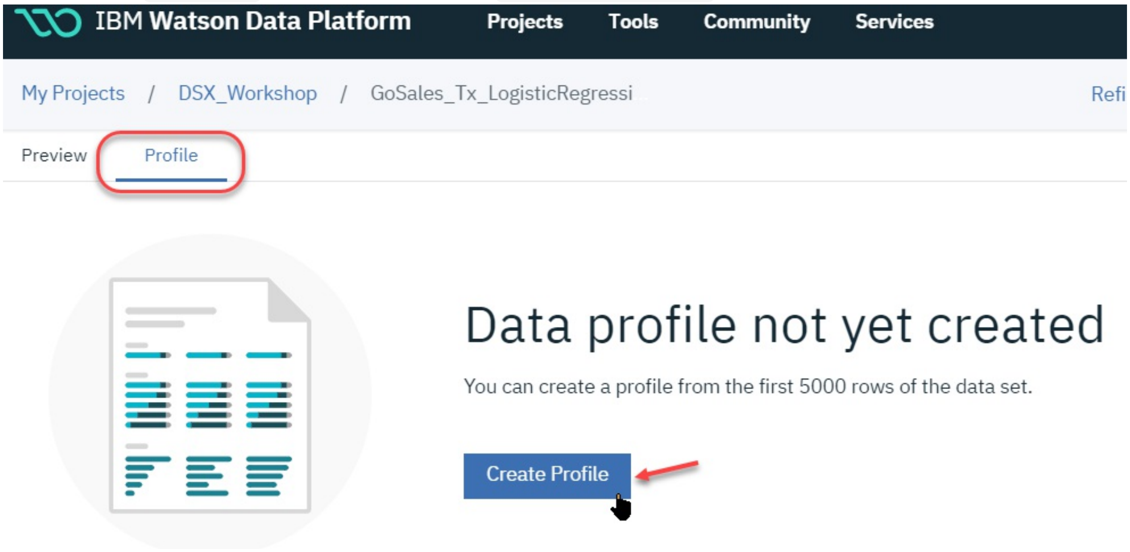
   0 assets selected.

□	NAME	TYPE	SERVICE	CREATED BY	LAST MODIFIED	ACTIONS
□	GoSales_Tx_LogisticRegression.csv	Data Asset	Project	desxtwo Workshop	11 Mar 2018, 4:38:34 pm	⋮

   This opens into the tabular preview, where we can discover the data structure:
   | `IS_TENT` | `GENDER` | `AGE` | `MARITAL_STATUS` | `PROFESSION` |
   | Type: String | Type: String | Type: String | Type: String | Type: String |
   So there are basically 4 features that can drive the buying decision held in the `IS_TENT` column.

2. To go further in the analysis, we will create the Profile for the data:

   - Select the `Profile` tab and then the `Create Profile` button.

     IBM Watson Data Platform Projects Tools Community Services

     My Projects / DSX_Workshop / GoSales_Tx_LogisticRegressi Refi

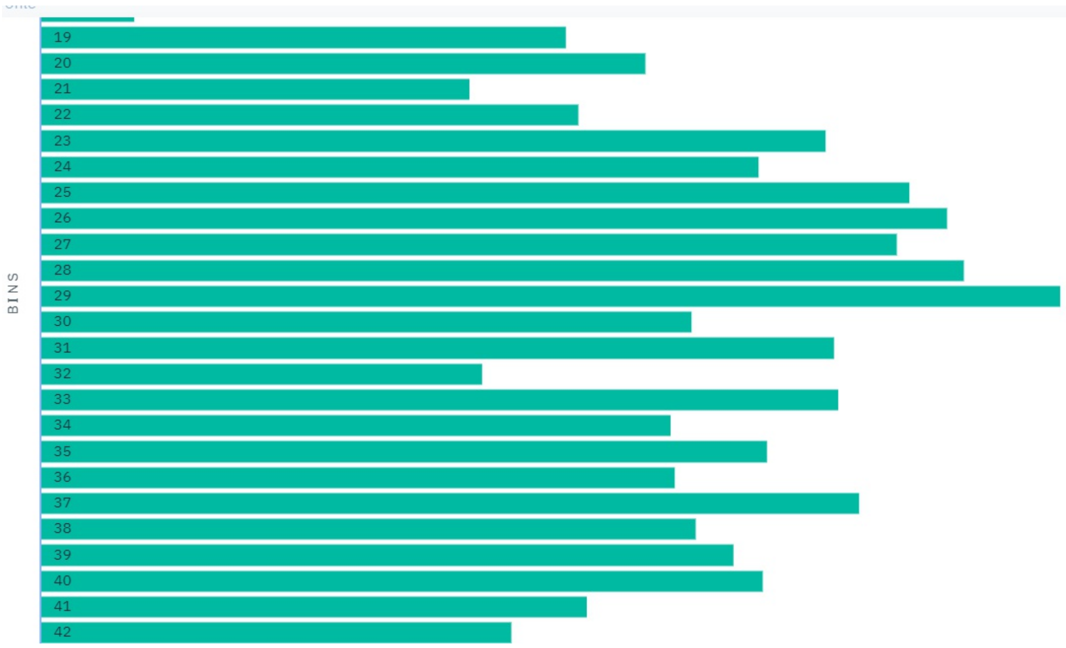
     Preview Profile

     Data profile not yet created

     You can create a profile from the first 5000 rows of the data set.

     Create Profile

   - After a while, the data profile is computed on the first 5000 lines.
   - This gives a rough idea on the structure of the data through the content of the columns in statistical terms:
     - `IS_TENT` is detected as a boolean with roughly 10% occurences of `TRUE` (509 out of the 5000 sample)
     - `GENDER` has slightly more Male than Female.
     - `AGE` distribution shows a peak in the 24-30 years, with an average of 34:

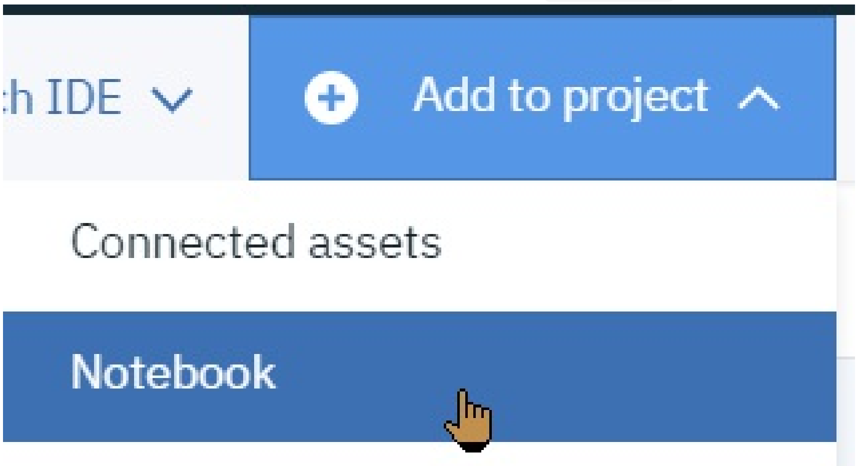


- `MARITAL_STATUS` has half of the sample as married
- `PROFESSION` shows almost half of the sample unspecified, with 8 distinct professions.

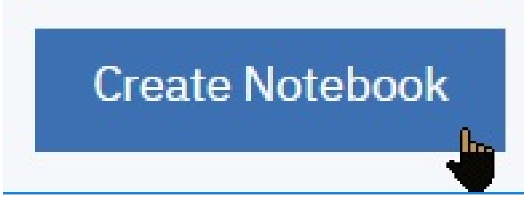
3. This gives a first-level overview of what to expect. We will now use the `GoSales_Tx_Analysis.ipynb` notebook for more data analysis:

   i. Go back to the Project page

   ii. From the (+) Add to project menu, select `Notebook` :



   iii. Select the `From file` tab, scroll down to `Choose file` and select the `GoSales_Tx_Analysis.ipynb` file



iv. For this first lab, we'll use a plain Python Jupyter notebook:
In the bottom-right section below, select the `Default Python 3.6 Free` runtime environment.

v. Open the notebook. From that point on, follow the instructions that are within the Notebook.