

## Using Loom with the Hortonworks Sandbox

V1.2

November 11, 2013

### Introduction

#### Summary

Data science often calls for the application of a variety of tools: the Hadoop Distributed File System (HDFS) provides a place to store and process data that does not fit in memory; Hive provides a SQL-like interface for processing data in Hadoop; and R gives powerful options for munging, modeling, and visualizing “small” data. Loom provides an integrated workflow from one tool to another, capturing and storing metadata in its extensible registry.

In this tutorial, learn how to install and get started with Loom, register and transform data in HDFS through the Loom Workbench, and import transformed data into R for analysis. The tutorial is based on an analysis of the relationship between flight delays and weather. By the end of the tutorial, we will see what airports saw the most rain during the sample period. Although this tutorial shows how to use the Loom Workbench, the same steps can also be accomplished through the Loom API. For more information, see the complete Loom documentation on the Revelytix website.

If you have any questions or comments, please contact us at [hwsandbox@revelytix.com](mailto:hwsandbox@revelytix.com).

#### Prerequisites:

- Hortonworks Sandbox V1.3 (4GB RAM recommended)
- Loom 1.2.7 or higher
- RLoom 0.7.8 or higher [optional]
- R [optional]
- RStudio [optional]

#### Overview

1. Install Loom
2. Acquire Data and Login
3. Create Sources
4. Create Datasets
5. Create and Execute Transforms
6. Connect to Loom from R [optional]

## Step 1 - Install Loom

Installing Loom on the Hortonworks Sandbox is simple, but it does require using the command line interface of the Sandbox virtual machine (VM). The steps are similar to Sandbox Tutorial #12.

1. Log in to the command line of the Sandbox VM. Press CTRL + ALT and enter the following name and password. Alternatively, you can use ssh to connect on the command line.

```
login: root
password: hadoop
```

2. Download and unzip the Loom distribution from Revelytix. Go to <http://www.revelytix.com/?q=content/download-loom-trial> and register for the download. After submitting the form, you will receive an email with the Loom download URL. Download the zip file with 'wget'.

```
wget <download-URL>;
unzip loom-1.2.7-distribution.zip
```

3. Change the working directory to the distribution directory.

```
cd loom-1.2.7
```

4. Set environment variables for Hadoop and Hive.

```
export HADOOP_HOME=/usr/lib/hadoop;
export HIVE_HOME=/usr/lib/hive
```

5. Add the user 'root' to the group 'hdfs'.

```
usermod -aG hdfs root
```

6. Execute the Loom start-up script from the distribution directory as shown below. Do not change your working directory to the 'loom-1.2.7/bin' subdirectory or another subdirectory. By default, Loom starts on port 8080 of the VM. If you cannot run Loom on 8080 due to conflicts with another service, you can start Loom on a different port, such as 9090. To use another port, you may need to add a new port forwarding rule to the VM network settings.

```
bin/loom-server.sh          # Loom starts on port 8080
bin/loom-server.sh <port>  # Loom starts on <port>
```

Once you start the Loom server, leave the process running on the command line. The output will look like this:

```
[root@sandbox loom-1.2.4]# bin/loom-server.sh
Starting Database...
/usr/lib/hadoop/conf
HADOOP_CP=/usr/lib/hadoop/conf:/usr/lib/hadoop/*:/usr/lib/hadoop/lib/*
/usr/lib/hadoop/conf
HIVE_CP=bin/../plugins/hive:/usr/lib/hive/lib/*:/usr/lib/hive/conf:/usr/lib/hado
op/conf:/usr/lib/hcatalog/share/hcatalog/hcatalog-core.jar
Starting Loom Server...
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/root/loom-install/loom-1.2.4/lib/ext/slf4j-
log4j12-1.6.6.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/hadoop/lib/slf4j-log4j12-
1.4.3.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
Starting Loom Server on port 8080
Loom Server started
```

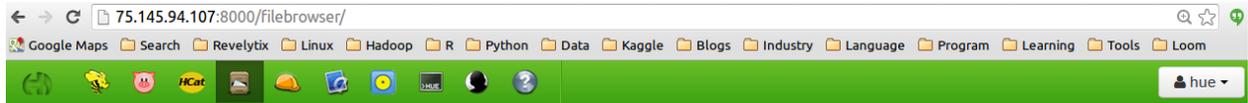
If you prefer to maintain access to the command line, you can use a utility such as screen or nohup to start the server in the background. These utilities must be downloaded separately.

While the Loom server starts up, download sample data for the tutorial.

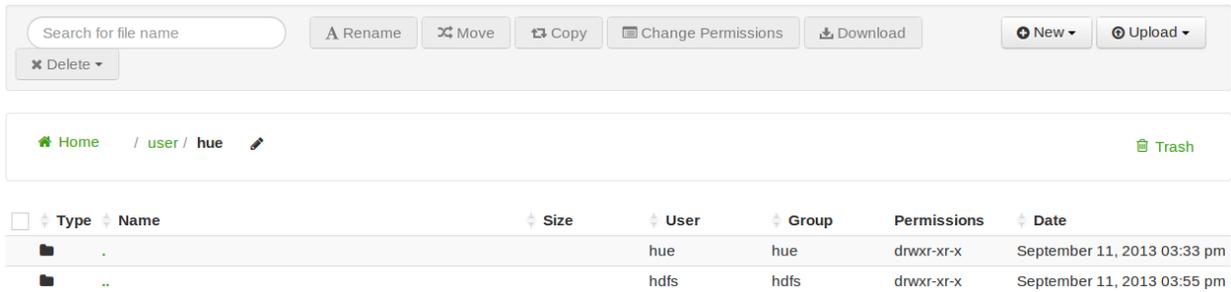
## Step 2 - Acquire Data and Log In

Sample U.S. government data on flight delays and weather, along with a table matching airports and weather stations, are available from the Revelytix Amazon S3 bucket. The airline on-time performance data comes from the Bureau of Transportation Statistics, while the weather data comes from the National Climatic Data Center's Global Historical Climatology Network.

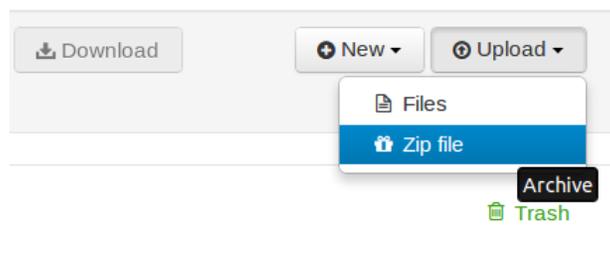
1. Download the tutorial data to your computer's local drive by clicking on this link: [https://s3.amazonaws.com/Revelytix-Public/sandbox\\_tutorial\\_data.zip](https://s3.amazonaws.com/Revelytix-Public/sandbox_tutorial_data.zip).
2. Navigate to the file browser of the Sandbox in your web browser.



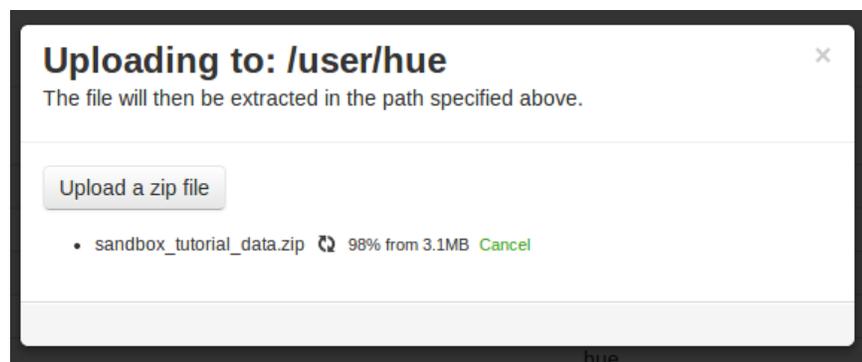
## File Browser



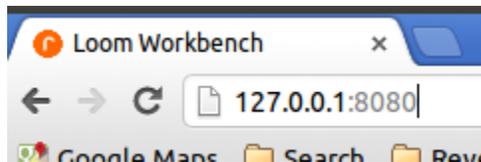
3. Click **Upload** > **Zip file** > **Upload a zip file**.



4. Select the zip file you downloaded to your computer. Click **Open**. The file will be uploaded to the VM, unzipped, and put in HDFS as a directory.



5. Open the Loom Workbench in your web browser. By default, the Workbench runs on port 8080 of the VM. Note that you will not be able to access the Workbench until the statement "Loom Server started" appears on the command line.



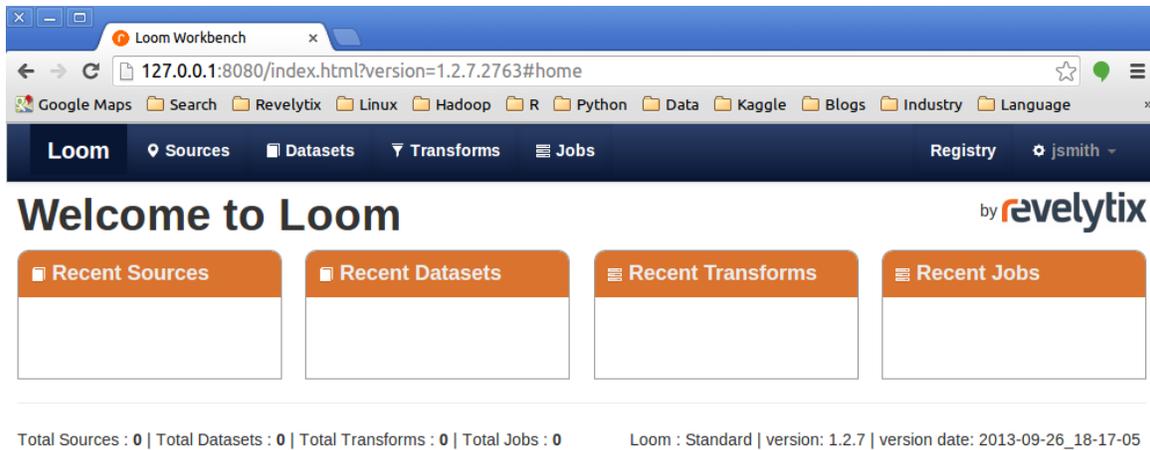
6. Click **Register**.

A screenshot of the Loom web interface. At the top, a dark blue header contains the word 'Loom' in white. Below the header, there are two tabs: 'Login' and 'Register'. The 'Register' tab is selected and highlighted. The form contains two input fields: 'Username' and 'Password'. Below the fields is a blue button labeled 'Log in'. At the bottom of the page, the text 'Dataset Management for Hadoop by revelytix' is displayed.

7. Enter a **Username** and **Password**.

A screenshot of the Loom web interface. At the top, a dark blue header contains the word 'Loom' in white. Below the header, there are two tabs: 'Login' and 'Register'. The 'Register' tab is selected and highlighted. The form contains three input fields: 'Username' (filled with 'jsmith'), 'Password' (filled with dots), and a second 'Password' field (also filled with dots). Below the fields is a blue button labeled 'Register'. At the bottom of the page, the text 'Dataset Management for Hadoop by revelytix' is displayed.

8. Click **Register**. This takes you to the Loom Home Page.

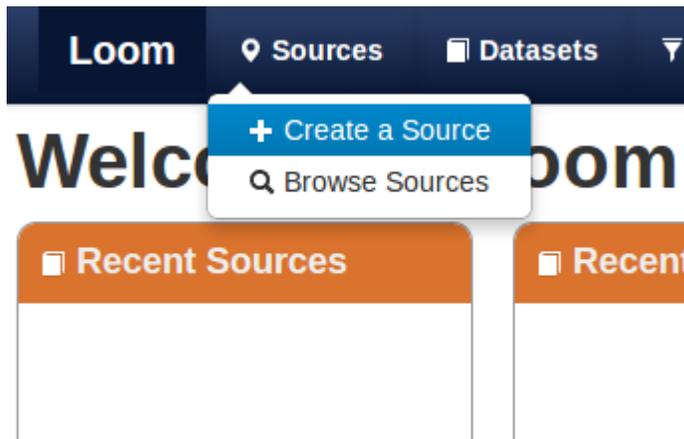


Now that the preliminaries are out of the way, we can get started with our analysis of weather data and how it relates to air travel.

### Step 3 - Create a Source

An analytic workflow in Loom typically begins with a Source. A Source is an abstraction over some data in HDFS, which can be a single file, directory, or database. The Activescan service enables Loom to identify potential Sources in HDFS automatically, but we can also create a Source manually as shown here.

1. Click **Sources > Create a Source**.



2. Click **Location** on the Source definition page. This brings up a file browser for HDFS. Select the `/user/hue/sandbox_tutorial_data/sandbox_tutorial_data` directory and click **OK**.

### Browse Filesystem

Server: hdfs://sandbox:8020

hive	2013-05-30 13:35	
hue	2013-09-24 10:31	
jobsub	2013-06-10 17:37	
oozie	2013-06-10 17:37	
sandbox_tutorial_data	2013-09-24 10:31	
sandbox_tutorial_data	2013-09-24 10:32	
Airport_and_Station.csv	2013-09-24 10:31	7.0 KB
GHCN_2013_06.csv	2013-09-24 10:32	500.4 KB
On_Time_Performance_2013_06.csv	2013-09-24 10:32	17.5 MB
oozie	2013-05-30 13:43	
sample	2013-06-10 17:38	

3. Click **Provenance Details**.

Define source Name the source and define its type and structural form.

<b>Source type</b>	<b>Format type</b>	<b>Structural form</b>
<input type="text" value="Text Files"/>	<input type="text" value="Delimited Text"/>	<input type="text" value="Table"/>
<b>Location</b>	<b>Is Directory?</b>	
<input checked="" type="checkbox"/> hdfs://localhost:8020/user/data/sandbox_tutorial	<input checked="" type="checkbox"/>	

Provenance Details : sandbox\_tutorial Edit the Name, folder, description and tags.

Default format Specify the default format for tables, which individual tables can override.

Tables Tables, or potential tables, found in source.

3 files found in source location.

Include	File	Table name	Delimiter	Quote	Skipped	Header?	
<input checked="" type="checkbox"/>	Airport_and_Station.csv	<input type="text" value="Airport_and_Station"/>	,	"	0	yes	<input type="button" value="edit"/>
<input checked="" type="checkbox"/>	GHCN_2013_06.csv	<input type="text" value="GHCN_2013_06"/>	,	"	0	yes	<input type="button" value="edit"/>
<input checked="" type="checkbox"/>	On_Time_Performance_2013_06.csv	<input type="text" value="On_Time_Performance_2013_06"/>	,	"	0	yes	<input type="button" value="edit"/>

4. Enter a new **Name** for the Source, such as 'loom\_tutorial'. Create a **Folder** called "tutorials" in the Loom registry for the Source. Add keywords "Revelytix, data science" in **Tags** and "Loom tutorial" in **Description**. Click **Provenance Details** again to hide these fields.

Define source Name the source and define its type and structural form.

---

Source type: Text Files | Format type: Delimited Text | Structural form: Table

Location:  hdfs://localhost:8020/user/data/sandbox\_tutorial | Is Directory?:

✕ Provenance Details : loom\_tutorial Edit the Name, folder, description and tags.

---

Name:  loom\_tutorial | Folder:  tutorials

Tags:  Revelytix, data science | Description:  Loom tutorial.

5. Click the **magnifying glass** to inspect each of the tables in the Source.

Default format Specify the default format for tables, which individual tables can override.

---

Tables Tables, or potential tables, found in source.

---

3 files found in source location.

Include	File	Table name	Delimiter	Quote	Skipped	Header?	
<input checked="" type="checkbox"/>	Airport_and_Station.csv	<input type="text" value="Airport_and_Station"/>	,	"	0	yes	<input type="button" value="🔍"/>
<input checked="" type="checkbox"/>	GHCN_2013_06.csv	<input type="text" value="GHCN_2013_06"/>	,	"	0	yes	<input type="button" value="🔍"/>
<input checked="" type="checkbox"/>	On_Time_Performance_2013_06.csv	<input type="text" value="On_Time_Performance_2013_06"/>	,	"	0	yes	<input type="button" value="🔍"/>

6. Inspect the **Parsed data** and **Raw data**.

Source Table: **GHCN\_2013\_06** (GHCN\_2013\_06.csv) prev next Done

**Format** Specify the format for this table

**Parsed data** Data as parsed according to format options. showing 10 rows

USW00023047	20130601	PRCP	0
USW00023047	20130601	AWND	60
USW00093862	20130601	PRCP	109
USW00093862	20130601	AWND	37
USW00093005	20130601	PRCP	0
USW00093005	20130601	AWND	26

**Raw data** An un-parsed sample from the source location

```
USW00023047,20130601,PRCP,0
USW00023047,20130601,AWND,60
USW00093862,20130601,PRCP,109
USW00093862,20130601,AWND,37
USW00093005,20130601,PRCP,0
USW00093005,20130601,AWND,26
USW00013748,20130601,PRCP,0
USW00013748,20130601,AWND,28
USW00093110,20130601,PRCP,0
USW00093110,20130601,AWND,24
```

Done

- Click **Format** and select different settings from the defaults as needed. In particular, note that the table based on 'GHCN\_2013\_06.csv' does not have a header. For this table, uncheck **Has a header row?** to keep the first row of data from being used for column names. When you are finished formatting a table, click **Done**.

Source Table: **GHCN\_2013\_06** (GHCN\_2013\_06.csv) prev next Done

**\* Format** Specify the format for this table

Field delimiter: Comma (,)

Quote character " : Double (")

Number of rows to skip: 0

Has a header row?

**Parsed data** Data as parsed according to format options. showing 10 rows

Column_1	Column_2	Column_3	Column_4
USW00023047	20130601	PRCP	0
USW00023047	20130601	AWND	60
USW00093862	20130601	PRCP	109
USW00093862	20130601	AWND	37
USW00093005	20130601	PRCP	0

**\* Raw data** An un-parsed sample from the source location

```

USW00023047,20130601,PRCP,0
USW00023047,20130601,AWND,60
USW00093862,20130601,PRCP,109
USW00093862,20130601,AWND,37
USW00093005,20130601,PRCP,0
USW00093005,20130601,AWND,26
USW00013748,20130601,PRCP,0
USW00013748,20130601,AWND,28
USW00093110,20130601,PRCP,0
    
```

- Click **Save** to create the Source.

Define source Name the source and define its type and structural form.

Source type: Text Files | Format type: Delimited Text | Structural form: Table

Location:  hdfs://localhost:8020/user/data/sandbox\_tutorial | Is Directory?:

Provenance Details : sandbox\_tutorial Edit the Name, folder, description and tags.

Default format Specify the default format for tables, which individual tables can override.

Tables Tables, or potential tables, found in source.

3 files found in source location.

Include	File	Table name	Delimiter	Quote	Skipped	Header?	
<input checked="" type="checkbox"/>	Airport_and_Station.csv	Airport_and_Station	,	"	0	yes	
<input checked="" type="checkbox"/>	GHCN_2013_06.csv	GHCN_2013_06	,	"	0	yes	
<input checked="" type="checkbox"/>	On_Time_Performance_2013_06.csv	On_Time_Performance_2013_06	,	"	0	yes	

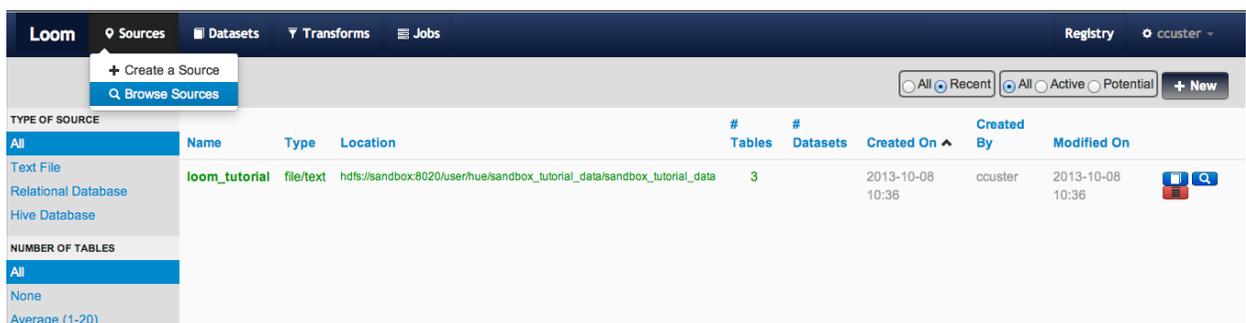
Save Save & Create Dataset Cancel

In creating the Source, you identified some data in HDFS and registered it in Loom with a particular format. Now it is time to enrich the Source with more metadata and create a Dataset.

## Step 4 - Create Datasets

The next step in the workflow is to create a Dataset from the Source. A Dataset is a Loom-managed, actionable collection of tables with complete schemas. Create two Datasets, 'ghcn' and 'matches', from the new Source. Each of these Datasets will contain a single table.

- Click **Browse Sources** from the **Sources** tab. Click the "loom\_tutorial" Source.



Registry ccuster

+ Create a Source  
Browse Sources

TYPE OF SOURCE

Name	Type	Location	# Tables	# Datasets	Created On	Created By	Modified On
loom_tutorial	file/text	hdfs://sandbox:8020/user/hue/sandbox_tutorial_data/sandbox_tutorial_data	3		2013-10-08 10:36	ccuster	2013-10-08 10:36

NUMBER OF TABLES

All  
None  
Average (1-20)

2. Start by creating a Dataset that contains the 'GHCN\_2013\_06' table. Click **Create Dataset**.



The screenshot shows the Loom interface with the source 'loom\_tutorial'. The 'Create Dataset' button is highlighted with a red circle. Below the header, there is a table listing tables in the source.

Table Name	Created	#Cols	Name in Source	Actions
Airport_and_Station	2013-09-05 17:06	3	Airport_and_Station.csv	<input type="button" value="Q"/>
GHCN_2013_06	2013-09-05 17:06	4	GHCN_2013_06.csv	<input type="button" value="Q"/>
On_Time_Performance_2013_06	2013-09-05 17:06	10	On_Time_Performance_2013_06.csv	<input type="button" value="Q"/>

3. Click **Provenance Details**. Enter "ghcn" in the **Name** field, "tutorial" in the **Folder** field, "Sample weather station data" in the **Description** field, and "weather" in the **Tags** field.

### × Provenance Details : ghcn

**Name** no spaces or special chars **Folder** no spaces / no special chars

✓ ghcn tutorial

**Description**

✓ Sample weather station data.

**Tags** optional

✓ weather

4. Uncheck the two other tables. Click **Edit** to complete the schema for the 'GHCN\_2013\_06' table.

### × Table Definitions edit column names and datatypes below

there are 1 tables in this dataset

Include	Name	Description	Created	
<input type="checkbox"/>	Airport_and_Station		2013-09-18 17:10	<input type="button" value="Edit"/>
<input checked="" type="checkbox"/>	GHCN_2013_06		2013-09-18 17:10	<input type="button" value="Edit"/>
<input type="checkbox"/>	On_Time_Performance_2013_0		2013-09-18 17:10	<input type="button" value="Edit"/>

5. Enter the field names and data types for each of the table columns as shown below. Assign a numeric **Data Type** such as 'bigint' to the column with the **Field Name**

'quantity'. The 'station' column uniquely identifies each weather station. The 'weatherdate' column provides the date of the observation. The 'stat' column marks the observation as precipitation ('PRCP') or average wind speed ('AWND'). The 'quantity' column shows the numeric value of the observation, in tenths of a millimeter or miles per hour, respectively.

Field Name	Data Type	Length	Precision	Nullable	data
station	string				USW00023047
weatherdate	string				20130601
stat	string				PRCP
quantity	bigint				0

[Done](#)

- Enter a more convenient name for the table: "ghcn". Click **Save**.

✖ **Table Definitions** edit column names and datatypes below

there are 1 tables in this dataset

Include	Name	Description	Created	
<input type="checkbox"/>	Airport_and_Station		2013-09-18 17:10	<a href="#">Edit</a>
<input checked="" type="checkbox"/>	ghcn		2013-09-18 17:10	<a href="#">Edit</a>
<input type="checkbox"/>	On_Time_Performance_2013_0		2013-09-18 17:10	<a href="#">Edit</a>

[Save](#)
[Save & Use In Transform](#)
[Save & Create Another Dataset](#)
[Cancel](#)

- Once you have saved the Dataset, you are taken to the Dataset summary tab. The 'ghcn' Dataset appears with the entity state "pending". Click **Datasets > Browse datasets** to refresh the entity state of the 'ghcn' Dataset. Once it turns to "active", click the 'ghcn' Dataset to see its details page.

Loom										
Sources Datasets Transforms Jobs Registry jsmith										
All Recent All From Source From Dataset(s)										
NUMBER OF TABLES	Dataset	# Tables	Entity State	Sourced From	Created On	Created By	# Uses	Last Used	Last User	Actions
All	ghcn	1	pending	loom_tutorial	2013-09-24 09:27	jsmith	0			<a href="#">Refresh</a> <a href="#">Delete</a>
None										
Average (1-20)										
Many (20+)										

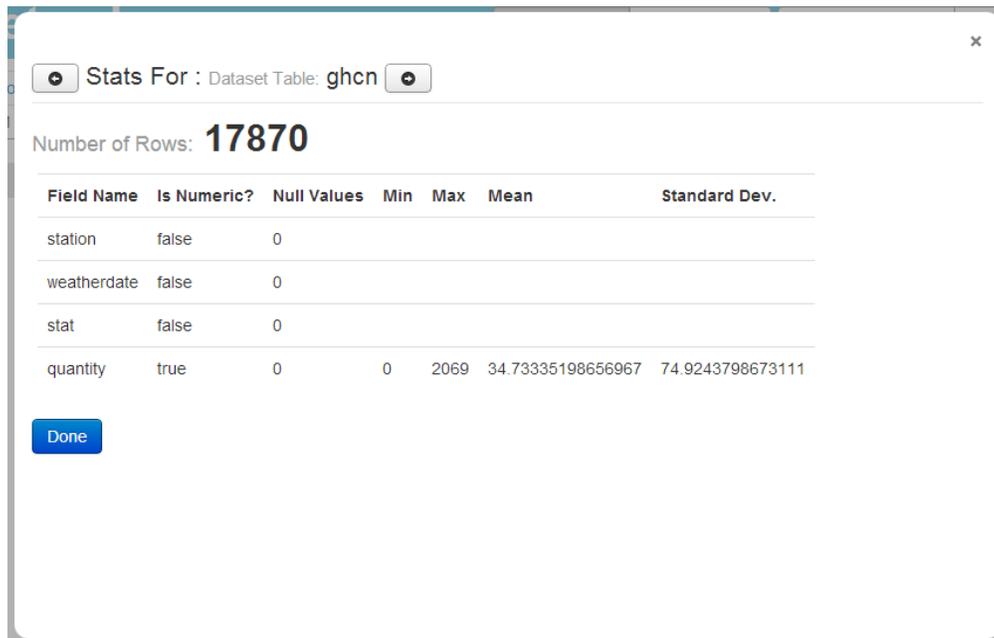
- Loom's Activescan service automatically calculates basic statistics for new tables, such as the number of rows. Click the **spreadsheet** to review column-level statistics.



The screenshot shows the Loom interface for a dataset named 'ghcn'. At the top, there are navigation tabs: Sources, Datasets, Transforms, and Jobs. Below the dataset name, there are buttons for 'Summary View', 'Advanced View', 'Use in Transform', 'Save', and 'Delete'. A small table shows 'Sourced From: loom\_tutorial' and 'Total # Tables: 1'. Below that is a main table with columns: Table Name, Description, Created, #Cols, #Rows, File Size, #Uses, and Actions. The table contains one row for 'ghcn' with 4 columns and 17870 rows. In the Actions column, there are icons for search, play, refresh, and a spreadsheet icon which is circled in red.

Table Name	Description	Created	#Cols	#Rows	File Size	#Uses	Actions
ghcn		2013-09-24 10:37	4	17870			[Search] [Play] [Refresh] [Spreadsheet]

- Activescan shows the number of null values, min, max, mean, and standard deviation for the numeric column. If the statistics are not yet available, wait a minute for the Activescan jobs to finish and try again. Click **Done**.



The screenshot shows a 'Stats For' dialog box for the dataset table 'ghcn'. It displays the total number of rows as 17870. Below this is a table with columns: Field Name, Is Numeric?, Null Values, Min, Max, Mean, and Standard Dev. The table contains four rows of data for different columns. A 'Done' button is located at the bottom left of the dialog.

Field Name	Is Numeric?	Null Values	Min	Max	Mean	Standard Dev.
station	false	0				
weatherdate	false	0				
stat	false	0				
quantity	true	0	0	2069	34.73335198656967	74.9243798673111

- Click **Sources > Browse Sources** to return to the source summary tab. Click on the 'loom\_tutorial' source.
- Create a Dataset that contains the 'Airport\_and\_Station' table. Click **Create Dataset**.

12. Click **Provenance Details**. Enter “matches” in the **Name** field, “tutorial” in the **Folder** field, “Matches airports and weather stations” in the **Description** field, and “weather” in the **Tags** field.

× Provenance Details : matches

**Name** no spaces or special chars **Folder** no spaces / no special chars

✓ matches tutorial

**Description**

✓ Matches airports and weather stations

**Tags** optional

✓ weather

13. Uncheck the two other tables. Click **Edit** to complete the schema for the ‘Airport\_and\_Station’ table.

× Table Definitions edit column names and datatypes below

there are 1 tables in this dataset

Include	Name	Description	Created	
<input checked="" type="checkbox"/>	Airport_and_Station		2013-10-03 00:22	<a href="#">Edit</a>
<input type="checkbox"/>	GHCN_2013_06		2013-10-03 00:22	<a href="#">Edit</a>
<input type="checkbox"/>	On_Time_Performance_2013_0		2013-10-03 00:22	<a href="#">Edit</a>

14. This file had a header, so the **Field Name** for each column is already given. Assign the **Data Type** ‘double’ to the column with the field name ‘distance’. The ‘airport’ column has three-letter abbreviations for U.S. domestic airports. The ‘station’ column gives the weather station nearest to the airport. The ‘distance’ column provides the distance between the weather station and airport in miles.

Dataset Table: Airport\_and\_Station prev next Done

× Description Table description

Field Name	Data Type	Length	Precision	Nullable	data	data
airport	string				MTJ	PPG
station	string				USW00093013	AQW00061705
distance	double				4.8	17.5

Done

15. Enter a more convenient name for the table: “matches”. Click **Save**.

✕ **Table Definitions** edit column names and datatypes below

there are 1 tables in this dataset

Include	Name	Description	Created	
<input checked="" type="checkbox"/>	matches		2013-10-03 00:22	
<input type="checkbox"/>	GHCN_2013_06		2013-10-03 00:22	
<input type="checkbox"/>	On_Time_Performance_2013_0		2013-10-03 00:22	

16. The ‘matches’ Dataset appears on the Dataset summary tab with the entity state “pending”. Click **Datasets > Browse datasets** to refresh the entity state until it says “active”.

Now that you have Datasets containing tables with complete schemas, you can transform those tables with Hive to learn more about your data.

### Step 5 - Create and Execute a Transform

What is the impact of precipitation on flight delays? Execute Hive queries to get started on an answer to this question. Loom automatically records the lineage of the inputs and outputs. Every execution of a Hive query creates a Job with metadata about the execution.

1. Click the ‘ghcn’ Dataset and click the **play button**.



**Loom** Sources Datasets Transforms Jobs Registry jsmith

**Dataset : ghcn** Summary View Advanced View Use in Transform Save Delete

Sourced From: loom\_tutorial  
Total # Tables: 1

Table Name	Description	Created	#Cols	#Rows	File Size	#Uses	Actions
ghcn		2013-09-24 10:37	4	17870			   

2. Enter **Provenance Details** for the Transform. Enter “join\_station\_and\_airport” in the **Name** field, “tutorial” in the **Folder** field, “weather” in the **Tags** field, and “Join airports to weather stations on station ID” in the **Description** field. This Hive query will add a column of airport abbreviations (e.g. JFK, DCA) to the ‘ghcn’ table in the ‘ghcn’ Dataset based on the matched pairs from the ‘matches’ table in the ‘matches’ Dataset.

**Default Input** Define the default input dataset and table.

**Input Dataset** optional  
ghcn(tutorial)

**Input Table** optional  
ghcn

**Table Columns**

Field Name	Data Type	
station	string	<input type="checkbox"/>
weatherdate	string	<input type="checkbox"/>
stat	string	<input type="checkbox"/>
quantity	bigint	<input type="checkbox"/>

**Transform Definition** Define the transform text.

**Transform Text**

\*

**Provenance Details : join\_station\_and\_airport** Name the transform and, optionally, define a folder, description and tags.

**Name**  join\_station\_and\_airport

**Folder**  tutorial

**Tags**  weather

**Description**  Join airports to weather stations on station ID.

3. Enter the **Transform Text** as shown below. This Hive query joins the ‘ghcn’ table with the ‘matches’ table, using weather station names as the key. This allows us to calculate weather statistics for particular airports. Make sure the transform text correctly identifies the Datasets and tables (e.g. “ghcn.ghcn”). The name before the period is the Dataset name, and the name after the period is the table name.

```
SELECT b.airport, a.station, a.weatherdate, a.stat, a.quantity FROM ghcn.ghcn a LEFT OUTER JOIN matches.matches b ON (a.station = b.station)
```

4. Click **Execution Contexts**. Enter “weather\_and\_airport” in the **Output Table** field.

**Execution Contexts** Define the inputs and outputs for execution.

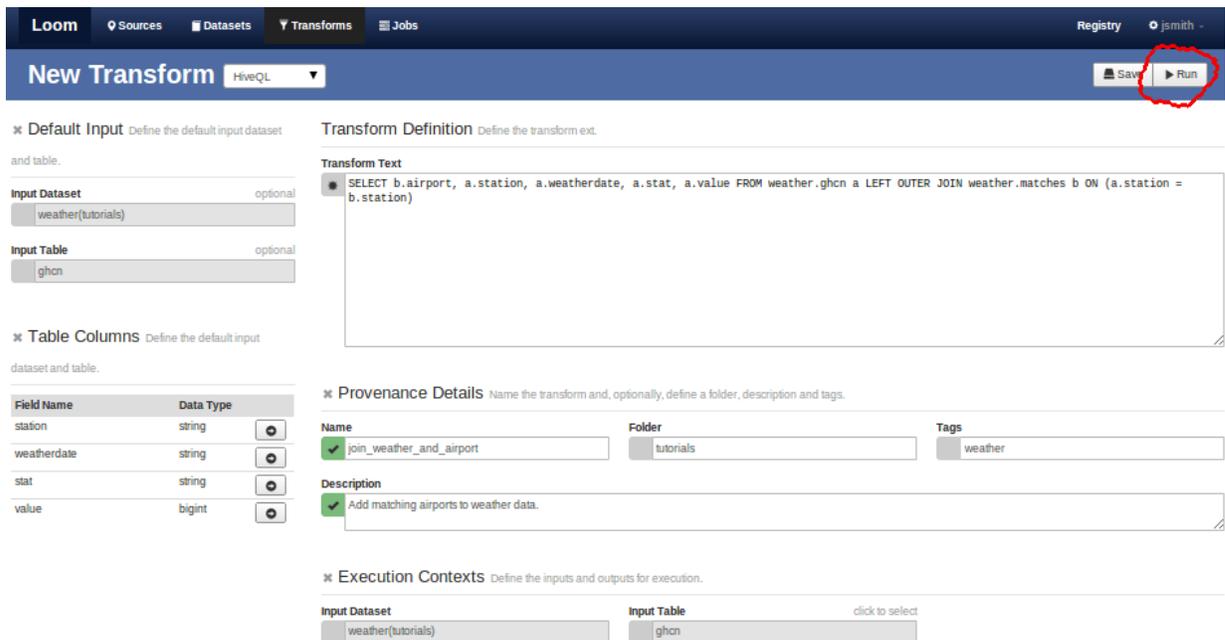
**Input Dataset**  
ghcn(tutorial)

**Input Table** click to select  
ghcn

**Output Dataset**  
ghcn(tutorial)

**Output Table**  
 weather\_and\_airport

5. Click **Run**. Loom takes you to the Job details page.

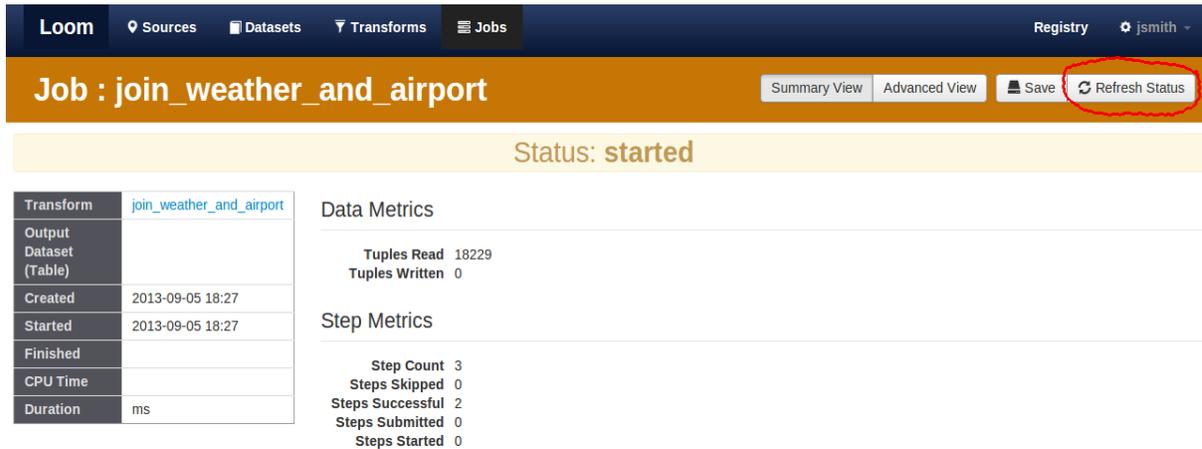


The screenshot shows the Loom 'New Transform' interface. At the top, there is a navigation bar with 'Loom', 'Sources', 'Datasets', 'Transforms', and 'Jobs'. On the right, it says 'Registry' and 'j.smith'. Below the navigation bar, there is a 'New Transform' header with a 'HiveQL' dropdown menu. On the right side of this header, there are 'Save' and 'Run' buttons. The 'Run' button is circled in red. The main content area is divided into several sections:

- Default Input:** Define the default input dataset and table. It includes an 'Input Dataset' field with 'weather(tutorials)' and an 'Input Table' field with 'ghcn'.
- Table Columns:** Define the default input dataset and table. It shows a table with columns: 'Field Name', 'Data Type', and a selection icon.
- Transform Definition:** Define the transform ext. It includes a 'Transform Text' field with the following SQL query:

```
SELECT b.airport, a.station, a.weatherdate, a.stat, a.value FROM weather.ghcn a LEFT OUTER JOIN weather.matches b ON (a.station = b.station)
```
- Provenance Details:** Name the transform and, optionally, define a folder, description and tags. It includes fields for 'Name' (join\_weather\_and\_airport), 'Folder' (tutorials), and 'Tags' (weather). There is also a 'Description' field with the text 'Add matching airports to weather data.'
- Execution Contexts:** Define the inputs and outputs for execution. It includes an 'Input Dataset' field with 'weather(tutorials)' and an 'Input Table' field with 'ghcn'.

- Click **Refresh Status** to see the latest statistics for the Job.



The screenshot shows the Loom interface for a job named 'join\_weather\_and\_airport'. The status is 'started'. The 'Refresh Status' button is circled in red. The interface includes a top navigation bar with 'Loom', 'Sources', 'Datasets', 'Transforms', and 'Jobs'. Below the navigation bar, there are buttons for 'Summary View', 'Advanced View', 'Save', and 'Refresh Status'. The main content area is divided into two columns: 'Transform' and 'Data Metrics'. The 'Transform' column shows details for the 'join\_weather\_and\_airport' transform, including 'Output Dataset (Table)', 'Created', 'Started', 'Finished', 'CPU Time', and 'Duration'. The 'Data Metrics' column shows 'Tuples Read' (18229) and 'Tuples Written' (0). Below the 'Data Metrics' section, there is a 'Step Metrics' section showing 'Step Count' (3), 'Steps Skipped' (0), 'Steps Successful' (2), 'Steps Submitted' (0), and 'Steps Started' (0).

Transform	join_weather_and_airport
Output Dataset (Table)	
Created	2013-09-05 18:27
Started	2013-09-05 18:27
Finished	
CPU Time	
Duration	ms

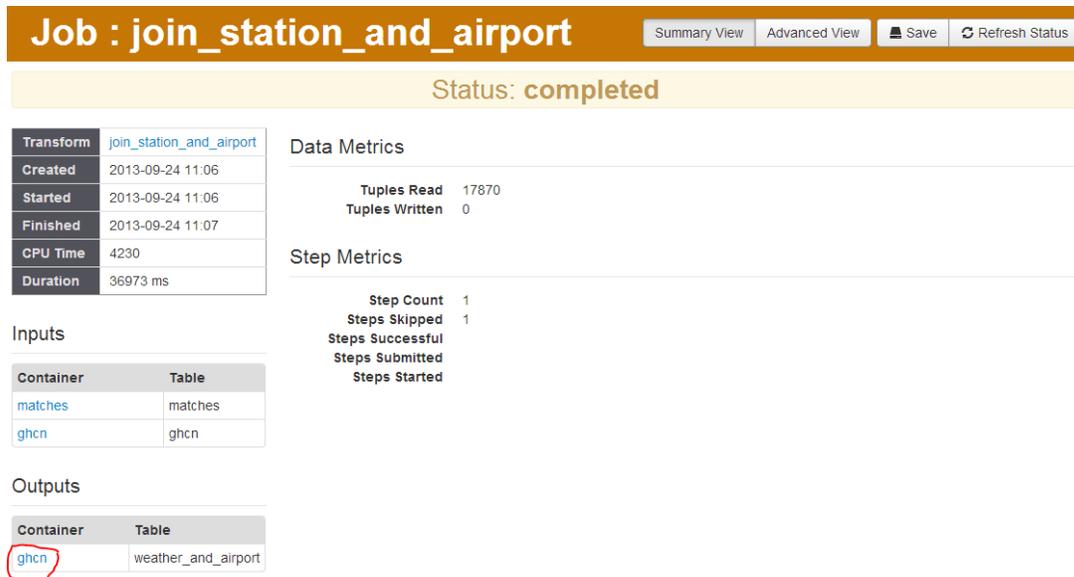
**Data Metrics**

- Tuples Read 18229
- Tuples Written 0

**Step Metrics**

- Step Count 3
- Steps Skipped 0
- Steps Successful 2
- Steps Submitted 0
- Steps Started 0

- The job may take a couple minutes to run. When the Job is 'completed', click the name of the dataset under **Outputs** to see the results.



The screenshot shows the Loom interface for a job named 'join\_station\_and\_airport'. The status is 'completed'. The 'ghcn' output dataset is circled in red. The interface includes a top navigation bar with 'Loom', 'Sources', 'Datasets', 'Transforms', and 'Jobs'. Below the navigation bar, there are buttons for 'Summary View', 'Advanced View', 'Save', and 'Refresh Status'. The main content area is divided into two columns: 'Transform' and 'Data Metrics'. The 'Transform' column shows details for the 'join\_station\_and\_airport' transform, including 'Created', 'Started', 'Finished', 'CPU Time', and 'Duration'. The 'Data Metrics' column shows 'Tuples Read' (17870) and 'Tuples Written' (0). Below the 'Data Metrics' section, there is a 'Step Metrics' section showing 'Step Count' (1), 'Steps Skipped' (1), 'Steps Successful' (0), 'Steps Submitted' (0), and 'Steps Started' (0). The 'Outputs' section shows a table with 'Container' and 'Table' columns, where 'ghcn' is circled in red and 'weather\_and\_airport' is listed as the table name.

Transform	join_station_and_airport
Created	2013-09-24 11:06
Started	2013-09-24 11:06
Finished	2013-09-24 11:07
CPU Time	4230
Duration	36973 ms

**Data Metrics**

- Tuples Read 17870
- Tuples Written 0

**Step Metrics**

- Step Count 1
- Steps Skipped 1
- Steps Successful 0
- Steps Submitted 0
- Steps Started 0

**Inputs**

Container	Table
matches	matches
ghcn	ghcn

**Outputs**

Container	Table
ghcn	weather_and_airport

- The Loom lineage graph provides a record of inputs and outputs for Hive queries, no matter how complicated the workflow. Click the **crossing arrows** next to one of the tables to see how the tables are related.

**Dataset : ghcn**

[Summary View](#)
[Advanced View](#)
[Use in Transform](#)
[Save](#)
[Delete](#)

<b>Sourced From</b>	loom_tutorial	<b># Uses</b>	1
<b>Total # Tables</b>	2	<b>Last Use</b>	2013-09-24 11:06
		<b>Last Used By</b>	jsmith

Table Name	Description	Created	#Cols	#Rows	File Size	#Uses	Actions
weather_and_airport		2013-09-24 11:07	5	18050			<a href="#">Q</a> <a href="#">▶</a> <span style="border: 1px solid red; border-radius: 50%; padding: 2px;">↔</span> <a href="#">☰</a>
ghcn		2013-09-24 10:37	4	17870			<a href="#">Q</a> <a href="#">▶</a> <span style="border: 1px solid red; border-radius: 50%; padding: 2px;">↔</span> <a href="#">☰</a>

- Review the lineage graph. Blue dots represent tables, and orange circles represent Jobs. Click the Job to see details on the left-hand pane.

### weather\_and\_airport - Lineage

Details Mode : [Split Screen](#) [Pop-up](#)

#### Selection Details

**Name** join\_station\_and\_a  
irport

**Type** transform

**Transform** join\_station\_and\_a  
irport

**Date** 2013-09-24 11:07

#### Arguments

- transformText** : SELECT  
b.airport, a.station,  
a.weatherdate, a.stat,  
a.quantity FROM  
ghcn.ghcn a LEFT OUTER  
JOIN matches.matches b  
ON (a.station = b.station)

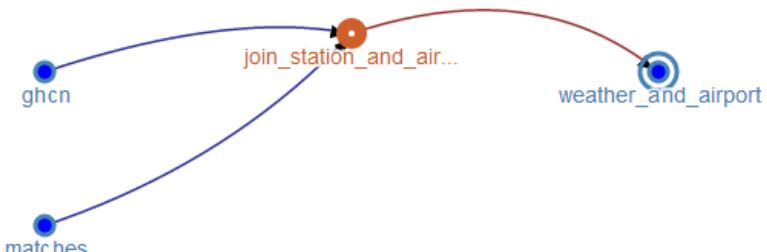
#### Contexts

**Inputs**

- matches (matches)
- ghcn (ghcn)

**Outputs**

- ghcn  
(weather\_and\_airport)



```

      graph LR
        ghcn((ghcn)) --> job((join_station_and_a irport))
        matches((matches)) --> job
        job --> weather_and_airport((weather_and_airport))
      
```

Now that you have transformed the data with Hive, you can optionally import the data into R for analysis and visualization. We encourage you to use the Loom Workbench to continue to explore the sample data. Better yet, starting using Loom with your own data and analytics!

## Step 6 - Connect to Loom from R [optional]

We can't compare precipitation and flight delays with the data registered so far, but we can see which airports got the most precipitation in the sample period. Connect to a Loom instance, import data, and create a plot.

The open-source R environment is a powerful tool for statistical analysis and visualization. R must be installed separately on your computer to complete this step. The Rloom package provides an easy way to access data and metadata in HDFS through calls to Loom's RESTful API. The same data and metadata is also available to other tools, such as Python. For more information on the API, see the complete Loom documentation on the Revelytix website. If you see errors running this script, double-check that the entity names in the script match the names in your Loom registry.

The RStudio IDE, which also must be installed separately, provides a convenient way to run through a script line-by-line. Download a script with the R code shown below by clicking this link: <https://s3.amazonaws.com/Revelytix-Public/Loom Tutorial for Hortonworks Sandbox.R>.

1. Download the Rloom package from Revelytix.

```
www.revelytix.com/transfer/Rloom-0.7.8.zip
```

2. Install Rloom and dependencies. The four dependencies are available from the main R repository. Install Rloom from the downloaded .tar.gz file.

```
> install.packages("RCurl")
> install.packages("bitops")
> install.packages("RJSONIO")
> install.packages("plyr")
> install.packages(pkgs="<your-path>/Rloom.tar.gz", repos=NULL, type="source")
```

3. Load the Rloom package and help pages.

```
> library(Rloom)
> help(package="Rloom")
```

## Access to Hadoop and the Loom Registry through Loom



### Documentation for package 'RLoom' version 0.7.8

- [DESCRIPTION file.](#)
- [Package NEWS.](#)

### Help Pages

<a href="#">add.context</a>	Add another context to a list of contexts.
<a href="#">add.process.info</a>	Add a Context or Argument struct to a local Process entity.
<a href="#">dataHead</a>	Get first rows from a dataset's data unit (table).
<a href="#">datasetCreate</a>	Create a dataset entity in Loom.
<a href="#">datasetCreateDefault</a>	Creates a 'default' dataset from an existing source entity in Loom.
<a href="#">dataseteReplace</a>	Replace an existing dataset entity in the Loom registry.
<a href="#">datasetGet</a>	Retrieves a Dataset entity from Loom.
<a href="#">datasetGetDefault</a>	Gets a 'default' dataset from an existing source entity in Loom.
<a href="#">dataStats</a>	Get the stats for a dataset's data unit (table).
<a href="#">entityCreate</a>	Create a new entity in Loom.
<a href="#">entityDelete</a>	Delete an entity in the registry.
<a href="#">entityGet</a>	Get an entity by ID.
<a href="#">entityList</a>	List all entities of a given type.
<a href="#">executeStatus</a>	Get the status of a submitted job.
<a href="#">executeTransform</a>	Execute a process (e.g., a transform such as a SQL query).
<a href="#">fileReadLines</a>	Get first lines from a file in HDFS, in record format.
<a href="#">fileReadParsed</a>	Get first records parsed from a text file containing tabular data.
<a href="#">hdfsFileInfo</a>	Get the HDFS file system details for a specified path.
<a href="#">hdfsList</a>	List files and directories in HDFS from a path.
<a href="#">loomConnect</a>	Connects to Loom.

4. Connect to Loom. Enter your own name and password. Your host and port may also differ, depending on how the VM and Loom are set up in relation to your computer.

```
> loom <- loomConnect(
+ host="http://127.0.0.1",
+ port="8080",
+ username=<your-name>,
+ password=<your-password>
+ )
> ping(loom)
[1] TRUE
```

5. Return a data frame with the name and UUID of the Datasets registered in Loom.

```
> dataset.index <- entityList(loom=loom, type="dataset/Dataset",
+ fields=c('entity/name','entity/id'))
> colnames(dataset.index) <- c("dataset", "id")
> print(dataset.index)
  dataset                                     id
1 weather 5228faa1-92d4-427f-9d14-ccc570de6cf9
2      otp 5228fc13-29f3-4ebe-ad85-4a1a96bf53cc
```

6. Store the UUID for the 'weather' Dataset as a string.

```
> weather.id <- dataset.index[dataset.index$dataset=="weather","id"]
> print(weather.id)
[1] "5228faa1-92d4-427f-9d14-ccc570de6cf9"
```

7. Return metadata for 'weather\_and\_airport' table.

```
> weather.stats <- dataStats(loom, containerID=weather.id,
+ dataUnitName="weather_and_airport", as.frame=FALSE)
> weather.rows <- weather.stats$'scan.table/numRecords'
> print(weather.rows)
[1] 18050
```

8. Import the 'weather\_and\_airport' table into R. This may take 1-2 minutes.

```
> weather.full <- dataHead(loom=loom, containerID=weather.id,
+ dataUnitName="weather_and_airport", nrow=weather.rows)
> head(weather.full)
  airport distance      station weatherdate stat quantity
1     AMA      37.2 USW00023047   20130601 PRCP         0
2     AMA      37.2 USW00023047   20130601 AWND        60
3     TUP       0.4 USW00093862   20130601 PRCP       109
4     TUP       0.4 USW00093862   20130601 AWND        37
5     DRO       0.7 USW00093005   20130601 PRCP         0
6     DRO       0.7 USW00093005   20130601 AWND        26
```

9. Munge the data into an appropriate form. Turn the data frame of lists into a data frame of vectors; convert the value column from character to numeric.

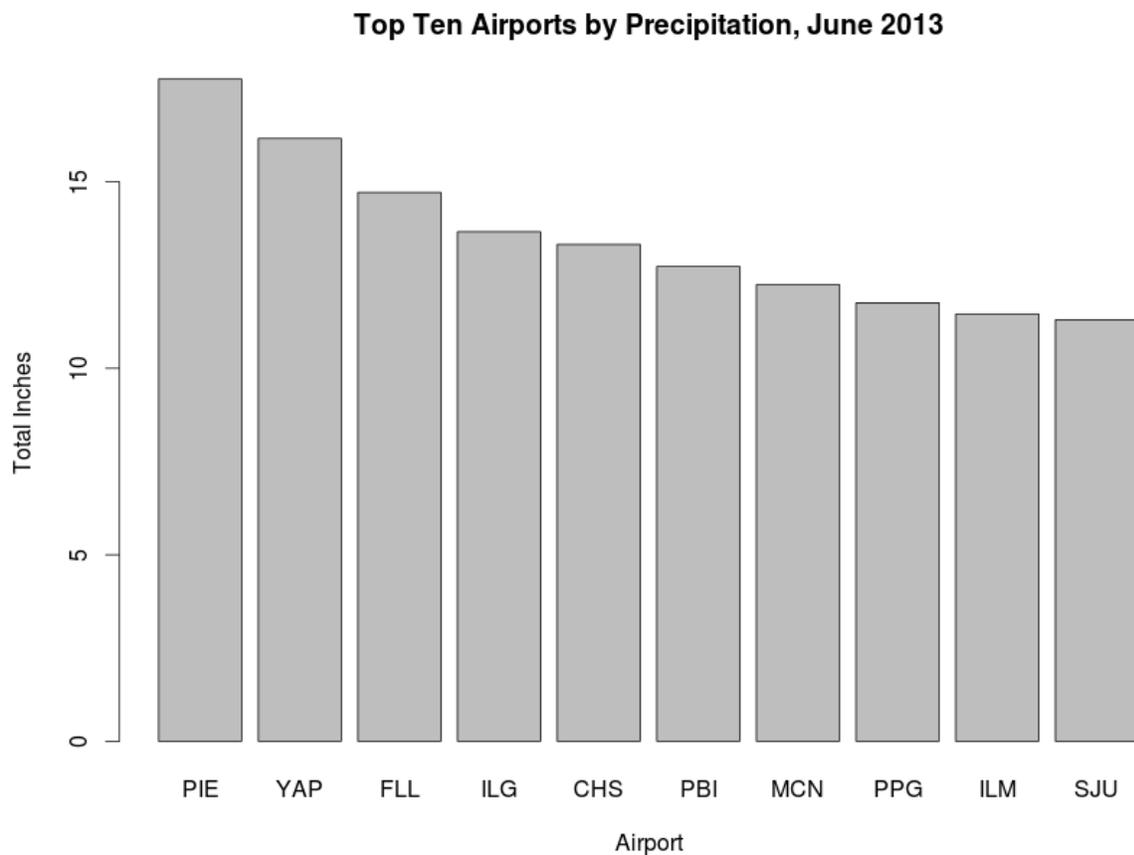
```
> weather <- as.data.frame(sapply(weather,unlist),stringsAsFactors=FALSE)
> weather$value <- as.numeric(weather$value)
```

10. Aggregate precipitation by airport. Taking the sum results in total precipitation at each airport over the sample period.

```
> airport.rain <- by(weather$value[weather$stat=="PRCP"],
+as.factor(weather$airport[weather$stat=="PRCP"]), sum)
> airport.rain.sorted <- sort(airport.rain, decreasing=TRUE)
> airport.rain.sorted.inches <- sort(airport.rain, decreasing=TRUE)/254
```

11. Plot the data with a bar plot.

```
> barplot(head(airport.rain.sorted.inches,10), main="Top Ten Airports by
Precipitation,
+ June 2013", xlab="Airport", ylab="Total Inches")
```



This tutorial is only an example of what can be done with this data using Loom, Hadoop, and R. Check out the accompanying video for an extended workflow.

## Feedback

We're interested to hear about your experience with this tutorial. Please take this [short survey](#).

## About Revelytix

Revelytix is a commercial software company providing tools for enterprise information management. The founders and engineering team have been together for 14 years, eight at Metamatrix (sold to Red Hat in 2006) and six years at Revelytix. For the first few years at Revelytix we built complex data management software for the Department of Defense.

Loom is our flagship product. Loom manages Hadoop data complexity, making data scientists and other Hadoop users more productive. Loom automatically discovers datasets, generates metadata on datasets, and tracks lineage of operations in Hadoop. Loom has a published RESTful API and is integrated with R, through the Rloom package.

For more information, please visit our website or contact us directly:

[www.revelytix.com](http://www.revelytix.com)

[info@revelytix.com](mailto:info@revelytix.com)

[hwsandbox@revelytix.com](mailto:hwsandbox@revelytix.com)

443 - 212 - 5049

## About Hortonworks

Hortonworks develops, distributes and supports the only 100-percent open source distribution of Apache Hadoop explicitly architected, built and tested for enterprise grade deployments. Developed by the original architects, builders and operators of Hadoop, Hortonworks stewards the core and delivers the critical services required by the enterprise to reliably and effectively run Hadoop at scale. Our distribution, Hortonworks Data Platform, provides an open and stable foundation for enterprises and a growing ecosystem to build and deploy big data solutions. Hortonworks also provides unmatched technical support, training and certification programs. For more information, visit [www.hortonworks.com](http://www.hortonworks.com). The Hortonworks Sandbox can be found at: [www.hortonworks.com/sandbox](http://www.hortonworks.com/sandbox).